A Recurrence Local Computation Approach Towards Ordering Composite Beliefs in Bayesian Belief Networks

Bon K. Sy
Department of Computer Science,
Queens College,
City University of New York,
Flushing, New York

ABSTRACT

Finding the l Most Probable Explanations (MPE) of a given evidence, S_e, in a Bayesian belief network can be formulated as identifying and ordering a set of composite hypotheses, H_is, of which the posterior probabilities are the l largest; ie, $Pr(H_1|S_e) \geq \ldots \geq Pr(H_l|S_e)$. When an order includes all the composite hypotheses in the network in order to find all the probable explanations, it becomes a total order and the derivation of such an order has an exponential complexity. The focus of this paper is on the derivation of a partial order, with length l, for finding the l most probable composite hypotheses; where l typically is much smaller than the total number of composite hypotheses in a network. Previously, only the partial order of length two (ie, $l = 2$) in a singly connected Bayesian network could be efficiently derived without further restriction on network topologies and the increase in spatial complexity. This paper discusses an efficient algorithm for the derivation of the partial ordering of the composite hypotheses in a singly connected network with arbitrary order length. This algorithm is based on the propagation of quantitative vector streams in a feed-forward manner to a designated "root" node in a network. The time complexity of the algorithm

Address correspondence to Bon K. Sy, Department of Computer Science, Queens College, CUNY, Flushing, NY 11367.

This work is supported in part by a grant to Queens College from the General Research Branch, National Institute of Health under grant No. RR-07064 and by a grant from the City University of New York PSC-CUNY Research Award Program.

Received December 1, 1990; accepted May 21, 1992.

International Journal of Approximate Reasoning 1993; 8:17–50

655 Avenue of the Americas, New York, NY 10010 0888-613X/93/$5.00

is in the order of O(lkn); where l is the length of a partial order, k the length of the longest path in a network, and n the maximum number of node states—defined as the product of the size of the conditional probability table of a node and the number of incoming messages towards the node.

KEYWORDS: ***Bayesian network, probabilistic inference, partial order, composite hypothesis, local computation, message vectors***

1. INTRODUCTION

Finding the l Most Probable Explanations (MPE) of a given evidence, S_e, in a Bayesian belief network is an optimization problem to identify a set of composite hypotheses, Hs, which will yield the l largest $Pr(H|S_e)$s; where a composite hypothesis is an instantiation of all the nodes in the network except the evidence nodes. This optimization problem is generally NP-hard [1]. Among the methods being proposed previously for identifying the most probable composite hypotheses, two approaches were taken. One approach is to restrict the types of networks to be dealt with such as singly connected networks,[1] *BN*2 [2], or *bipartite graphs* [3]. Another approach is to shift the complexity to spatial domain in order to keep the computational complexity in a linear order. For example, Shimony and Charniak proposed a method that converts a Bayesian belief network into a Weighted Boolean Function Directed Acyclic Graph (WBFDAG), and which permits the use of the best-first search strategy in a WBFDAG. Although this method maintains a linear run time with respect to the size of a graph, the number of nodes in a WBFDAG could be in an exponential order as compared to the original network. To date, if a given network is singly connected and there is no compromise between the computational complexity and the spatial complexity, only the two most probable composite hypotheses could be determined efficiently. An efficient algorithm for finding the two most probable composite hypotheses of the MPE problem has been developed by Pearl [4]. The basic idea of Pearl's algorithm is that each node in a network is associated with a *causal* and *diagnostic* function through which the largest values of these functions will propagate to each other for obtaining the information needed to compute $Pr(H|S_e)$.

In this paper, we present an efficient computational method for obtaining the l most probable composite hypotheses in a singly connected Bayesian belief network. The mechanism of our computational method

[1] A *singly connected* network is defined as a network within which there is at most one path connecting any two nodes. For example, Fig. 1 is a singly connected network, whereas Fig. 2 is not because there are two paths of getting to node d from a via either b or c.

involves a message passing process that, in essence, is similar to Pearl's algorithm [5]. However, our method differs from Pearl's and others' algorithms in three ways. First, the message passing in our method is unidirectional as opposed to bidirectional in Pearl's algorithms. Second, each "message unit" in our method is a vector but not a value as in [5]. Third, we retain all processed information to permit its reuse in a systematically ordered fashion for the successive derivations of the most probable composite hypotheses. Regarding the spatial complexity in terms of memory size, our method is worse than Pearl's algorithm, but ours permits the derivation of more than two most probable hypotheses. Yet, it is better than that of a WBFDAG in the sense that the spatial complexity issue of our approach only occurs during the run time as opposed to the spatial complexity of a WBFDAG, which is static after compilation.

To the best of our knowledge, our attempt to identify not only the two most probable composite hypotheses, but a partial ordering of the MPE in a singly connected network without further restriction on network topology, is novel. We believe that such a consideration (ie, partial ordering of the most probable composite hypotheses) is important in several application domains such as in diagnosis, prognosis evaluation, and assessment of certain design methodologies [6, 7].

In section two we first give an overview of the formalism of a Bayesian network and the complexity of reasoning in such a network. A brief literature review is given in section three. In section four the notion of propagating "look-ahead" message streams for local computation is introduced. In section five the mechanism and characteristics of a recurrence local computation method are discussed. Then the recurrence local computation is formulated, and its correctness is proved in section six. An example to illustrate the local computation is given in section seven. In section eight the scope of this recurrence local computation approach and alternative approaches are discussed. In section nine the extension of this approach to a multiply connected network is discussed, followed by the conclusion in section ten.

2. OVERVIEW OF BAYESIAN NETWORK AND COMPLEXITY PROBLEM

A Bayesian network [8] is a directed acyclic graph within which a set of nodes are connected by a set of arcs. Each node in a graph represents a propositional variable, and an arc connecting two nodes indicates the dependency between them. For the sake of discussion, each propositional variable, represented by a lower case letter, is assumed to have only two values—true and false. An upper case letter represents the value of a

propositional variable. For example, X and $\bar{X}$ represent $x = X$ (i.e., true) and $x = \bar{X}$ (ie, false) respectively. Suppose x represents the propositional statement—*elevation* of *body temperature*. Then $X(\bar{X})$ is the hypothesis that there is (not) an elevation of body temperature.

The connections among the nodes determine the graphical structure of a network. Such a graphical structure is important in two ways. First, it indicates qualitatively the (un)conditional independencies among the propositional variables. Second, it determines the kind of efficient computation that can be applied. For example, the local computation methods described in this paper and elsewhere [5] rely on a singly connected network configuration such as the one shown in Fig. 1.

Each node in a Bayesian network is associated with a probability function. For example, in Fig. 2, node a is associated with the probability function $Pr(a)$, b with $Pr(b|a)$, c with $Pr(c|a)$, and d with $Pr(d|bc)$. The joint distribution of the variables can be computed by multiplying appropriate probability functions together. In Fig. 2, $Pr(abcd) = Pr(a)\,Pr(b|a)Pr(c|a)Pr(d|bc)$.

Each propositional variable in our discussion is assumed to be binary; ie, *true/false*, or *yes/no*. When the value of a propositional variable is known or observable, the variable is referred to as an *evidence variable*. Otherwise, it is referred to as a *non-evidence variable*. Each possible value of a non-evidence variable is considered as an assertion about a certain propo-

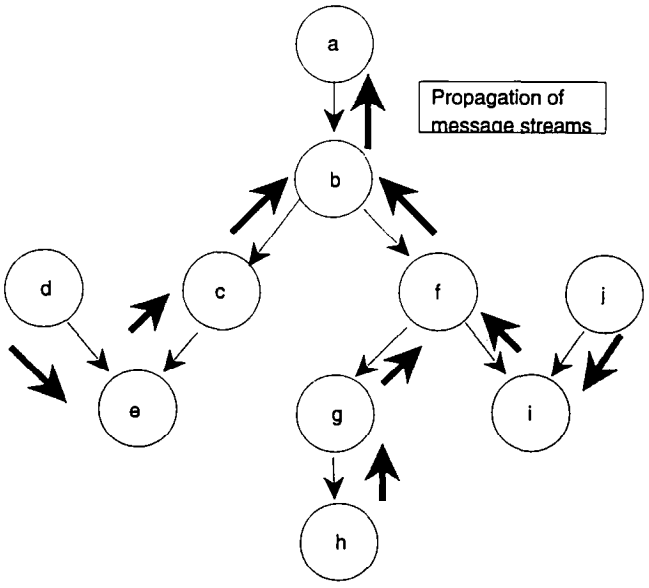


Figure 1. Ten-node singly connected Bayesian network.

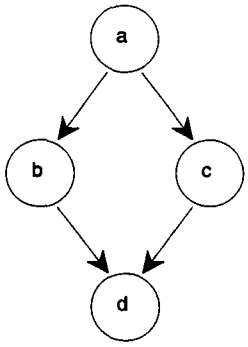


Figure 2. Multiply connected network.

sitional statement. For example, we can have an assertion: it is true that the patient has an elevation of body temperature. Such an assertion is referred to as a simple hypothesis. The combination of those assertions such as the patient has dizziness and *no* indication of heat illness and *no* elevation of body temperature is referred to as a *local composite hypothesis*. When the combination is exhaustive, meaning that all the non-evidence variables in a Bayesian network are being considered, we refer to such a combination a *composite hypothesis*.

3. REVIEW OF RELEVANT RESEARCH

Probabilistic inference in a Bayesian network has been viewed as answering queries relevant to the propositional variables in a Bayesian network [9]; in particular, the likelihoods of simple or (local) composite hypotheses in the presence of an evidence.[2] Various inference algorithms were developed elsewhere and the details were in [2–3, 5, 10–19]. To date, the most efficient computational method to deal with a simple hypothesis has a complexity order that is linear to the longest path in a network and the largest number of node states of a propositional variable [5, 11, 18]. However, this complexity order only applies to singly connected Bayesian networks and in which parallel processing is permitted. In dealing with multiply connected networks, cutset conditioning and triangulation preprocessing would be needed to *transform* the networks into polytree hypergraphs that are akin to singly connected networks. Unfortunately, finding the optimal cutsets, and obtaining the optimal hypergraph (with

[2]The set of evidence variables is not fixed and is varied over time.

respect to the number of states) using the triangulation process are both proven to be NP-hard [20, 28, 29].

There are some attempts to extend the query capability of a probabilistic inference to include composite hypotheses (or multiple causes) [2], and particularly in the domain of diagnosis [4, 16]. However, the attention is focused on the special cases such as the bipartite networks with a set of conditional independent disorders and a set of marginal independent manifestations. The discussion of general cases is rather limited because of the intractable computational complexity and the inconsistency problem of local computation. Nevertheless, the relationship between a probabilistic inference and a diagnostic strategy was thoroughly discussed by Peng and Reggie [16]; in particular what is meant by finding the best explanation in the domain of diagnosis. Specifically, they proposed a Parsimonious Covering Theory in which a *causation event* (CE) is defined as "d_i causes m_j" and is *true* only if both the disorder, d_i, and the manifestation, m_j, occur. Within the Parsimonious Covering Theory, the best explanation is then defined in terms of the most probable causation event—$Max[Pr(CE = d_i s \text{ cause } m_j s | S_e)]$, as opposed to Bayesian conditioning—$Max[Pr(H|S_e)]$, and the explanation must also satisfy minimality, irredundancy, and relevancy.

Further research on finding the most probable composite hypotheses should also be exemplified by the algorithms developed by Pearl [5] and Cooper [19], of which details were thoroughly discussed in Chapter 8 of [21], and the algorithm by Shimony and Charniak [17]. Pearl's algorithm [5] on finding the most probable composite hypothesis in a singly connected network is based on the propagation of the maximum probability values through a set of *causal* and *diagnostic* functions associated with the nodes in a network. The product of these probability values results in a joint probability that is proportional to the posterior likelihood of the most probable composite hypothesis. Once the most probable composite hypothesis is found, the second most probable composite hypothesis is deduced from *masking* one term at a time in the course of propagation which constitutes the most probable composite hypothesis.[3] Unfortunately, the structure of Pearl's propagation, as was pointed out by Neapolitan [21], was unable to support further derivation of the next few most probable composite hypotheses. In contrast to Pearl's approach, Cooper tackled the MPE problem by imposing the constraints that the variables in the local composite hypothesis set are: (1) binary valued, (2) mutually independent, and (3) for each variable x in the local composite hypothesis set, $Pr(X) \leq Pr(\overline{X})$. With these constraints, finding the most probable (local) composite hypotheses could be formulated as a search problem, and an incremental

[3]As a matter of fact, our algorithm is inspired by this idea.

search based on the best-first strategy with branch and bound pruning can be applied. One of the limitations of this approach is that the search complexity can grow exponentially with an extra propositional variable added in each level of the incremental search in approaching the desired (local) composite hypotheses. Another approach taken by Shimony and Charniak [17] is similar to Cooper's in that the MPE is formulated as a search problem, but no restriction is imposed on the probability distributions. The basic idea of Shimony and Charniak is to transform a Bayesian network into a Weighted Boolean Function Directed Acyclic Graph (WBFDAG) that permits the application of the best-first search strategy. Although the time complexity is showed to be linear with respect to the size of a graph, the spatial complexity in terms of number of nodes is exponentially increased in the course of converting the Bayesian networks into WBFDAGs.

Finding the total order of composite hypotheses is NP-hard due to the exponential increase in the number of composite hypotheses that depends on the number of non-evidence variables [22, 23]. In most cases, we are interested in only the most probable, or the few most probable hypotheses. In this paper, we will present a recurrence local computation method that can efficiently identify and order the few most probable composite hypotheses in a singly connected network.

4. LOCAL PROPAGATION OF LOOK-AHEAD MESSAGE STREAM

Referring to Fig. 1, the joint probability distribution of the network is:

$$\begin{aligned} &Pr(abcdefghij) \\ &\quad = Pr(a)\,Pr(b|a)\,Pr(c|b)\,Pr(d)\,Pr(e|cd)\,Pr(f|b) \\ &\quad\quad Pr(g|f)\,Pr(h|g)\,Pr(i|fj)\,Pr(j) \end{aligned}$$

In this paper each probability term on the right hand side will be referred to as a *local probability term*. It is noted that each local probability term corresponds to a node in the network. That is, $Pr(a)$ for node $a, \ldots, Pr(e|cd)$ for node $e, \ldots, Pr(i|fj)$ for node i, and $Pr(j)$ for node j. Let's suppose we are interested in finding the most probable composite hypothesis when there is no observation (ie, $S_e = \varnothing$ and the corresponding $Pr(\bullet) = Max[Pr(abcdefghij)]$). This is equivalent to finding the optimal setting for each node such that the product of the local probability terms yields the largest value.

In an extreme case when all variables are independent to each other (ie, none of the nodes are connected together), the optimal setting will be the one that corresponds to the maximum of each local probability. This is

because $Max[Pr(ab\ldots j)] = Max[Pr(a)]Max[Pr(b)]\ldots Max[Pr(j)]$. The complexity of such an extreme case is a linear combination of the complexity of finding the maximum of each local probability term. In dealing with an interconnected network such as the one in Fig. 1, we can imagine that each node in a network acts as a *local messenger* to receive and send information to its neighboring nodes. In order to determine the directional flow of information, a *root* node must be designated. A natural choice would be one of the root nodes[4] in a network. Let's suppose we choose node a as our designated root node that serves as an *absorption center*. The flow of information from each node is directed toward the designated root node. For example, the message streams that carry information about all the nodes in Fig. 1 are propagated towards node a via three different paths; namely, one through the path $d \rightarrow e \rightarrow c \rightarrow b \rightarrow a$, another one through $h \rightarrow g \rightarrow f \rightarrow b \rightarrow a$, and the last one through $j \rightarrow i \rightarrow f \rightarrow b \rightarrow a$.

Consider the simple network shown in Fig. 3 where the message stream goes from h to g to f. Note that $Pr(fgh) = Pr(f)Pr(g|f)Pr(h|g)$. Let's suppose we are going to consider the local probability terms in a bottom-up fashion (ie, consistent with the direction of message flow). In a binary case, we know that the optimal local probability term of $Pr(h|g)$ must be either $Max_h[Pr(h|G)]$ or $Max_h[Pr(h|\overline{G})]$. This is because node h receives no incoming messages and *looks ahead* to anticipate that g can only be G or $\overline{G}$. Therefore, the only important information that node g should receive from h is $Max_h[Pr(h|G)]$ and $Max_h[Pr(h|\overline{G})]$. Let's denote the message stream passed to node g from h in Fig. 3 to be $M_{h \rightarrow g}$. Then $M_{h \rightarrow g}$ will be in a form of a vector: $M_{h \rightarrow g} = [(ArgMax_{hG}[Pr(h|G)]\ Max_h[Pr(h|G)])\ (ArgMax_{h\overline{G}}[Pr(h|\overline{G})\ Max_h[Pr(h|\overline{G})])]$. To extend the consideration to mul-

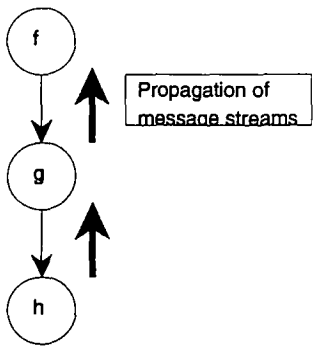


Figure 3. Simple singly connected network.

[3]As a matter of fact, our algorithm is inspired by this idea.

[4]A root node is a node without parents such as nodes a, d, and j in Fig. 1.

tivalued variables, $M_{h \to g} = [m^1_{h \to G_1}\ m^1_{h \to G_2}\ \ldots\ m^1_{h \to G_n}]$ for $\{G_1\ G_2\ \ldots\ G_n\}$ = a set of possible states of the propositional variable g; where $m^1_{h \to G_i} = (ArgMax_{hG_i}[Pr(h|G_i)]\ Max_h(Pr(h|G_i))]$, and the superscript "1" indicates the "first" round of message propagation from h to g. Note that the first part of $m^1_{h \to G_i}$ contains information about the value of the propositional variable, h, given G_i. The second part contains the information about the probability value of its argument—the propositional variables h and g. For the sake of discussion, we will abbreviate $ArgMax_{hG_i}[Pr(h|G_i)]$ as $Arg[m^1_{h \to G_i}]$, and $Max_h[Pr(h|G_i)]$ as $Val[m^1_{h \to G_i}]$.

In considering the message stream that g sends to f, it is necessary to incorporate the message stream $M_{h \to g}$ with $Arg[m^1_{g \to f}]$ and $Val[m^1_{g \to f}]$ for the composition of $M_{g \to f}$. A *convolution* operation, $\otimes$, and a *Belief* matrix are defined for this purpose.

DEFINITION 1 *Given* $M_{d \to x} = [m^1_{d \to x_1}\ m^1_{d \to x_2}\ \ldots\ m^1_{d \to x_n}]$, *and* $Pr(x|J_x) = [Pr(x|v1_x) \ldots Pr(x|vk_x)]$ (*where each* vi_x *is an instantiation of* J_x), *the convolution of* $M_{d \to x}$ *with* $Pr(x|J_x)$ *is defined as the product of every single term in* $Pr(x|J_x)$ *with a consistent* $Val[m^1_{d \to x}]$ *in* $M_{d \to x}$; *where* J_x *is the set of immediate parent nodes of* x. $Pr(x|J_x)$ *and* $Val[m^1_{d \to x}]$ *are consistent with each other if the value of* x *in* $Arg[m^1_{d \to x}]$ *is the same as the value of* x *in* $Pr(x|J_x)$.

DEFINITION 2 *A Belief Matrix of a node* x, $Bel(x)$, *is defined as the convolution of all* $M_{d_i \to x}$ *with* $Pr(x|J_x)$-$M_{d_1 \to x} \otimes M_{d_2 \to x} \otimes \ldots \otimes M_{d_w \to x} \otimes Pr(x|J_x)$; *where* d_i *is a node that propagate* $M_{d_i \to x}$ *to* x *for* $i = 1 \ldots w$.

To illustrate the definitions of convolution and belief matrix, let's suppose

$$M_{h \to g} = \left[m^1_{h \to G}\ m^1_{h \to \bar{G}}\right] = [(HG\ 0.6)\ (H\bar{G}\ 0.8)] \text{ and}$$

$$Pr(g|f) = \begin{bmatrix} (GF\ 0.3) & (G\bar{F}\ 0.45) \\ (\bar{G}F\ 0.7) & (\bar{G}\bar{F}\ 0.55) \end{bmatrix}$$

then

$$M_{h \to g} \otimes Pr(g|f) = \begin{bmatrix} (HGF\ 0.18) & (HG\bar{F}\ 0.27) \\ (H\bar{G}F\ 0.56) & (H\bar{G}\bar{F}\ 0.44) \end{bmatrix}$$

Remark: It is not critical whether $Pr(g|f)$ is listed in the form of a matrix (as shown in the example) or a vector (as shown in Definition 1) because only the relevant terms are multiplied, and the convolution operation is not a standard vector–matrix multiplication operation.

With these two definitions, $M_{g \to f}$ can be formulated as $Max_g[Bel(g|F)\ Bel(g|\bar{F})]$; ie, $M_{g \to f} = [(\bar{H}\bar{G}F\ 0.56)(H\bar{G}\bar{F}\ 0.44)]$. Lemma 1 summarizes the formulation of a message stream $M_{b \to a}$:

LEMMA 1 *A message stream that a node 'b' propagates to a node 'a' is defined as*

$$M_{b \to a} = \begin{cases} Max_b[Bel(b|A_1) \ldots Bel(b|A_n)] \\ \quad \text{if } a \text{ is an immediate parent of } b; \\ Max[Bel(B_1) \ldots Bel(B_m)] \\ \quad \text{if } b \text{ is an immediate parent of } a; \end{cases}$$

where

$$Bel(b) = M_{p_1 \to b} \otimes \ldots \otimes M_{p_k \to b} \otimes Pr(b|p_1 \ldots p_p)$$

$$\otimes M_{d_1 \to b} \otimes \ldots \otimes M_{d_k \to b}$$

$d_1 \ldots d_k$ are the immediate descendent nodes of b, and $p_1 \ldots p_l$ are the immediate ascendent nodes of b, and $Max_b[Bel(b|A_1) \ldots Bel(b|A_n)]$ is a vector in which each element is the largest value of the belief matrix $Bel(b)$ on a possible given instantiation of a. $Max[Bel(B_1) \ldots Bel(B_m)]$ is a vector in which each element is the largest value of the belief matrix $Bel(b)$ on a possible instantiation of b.

Remark: If node b is a root node, $M_{b \to a}$ is simply $[(B_1\ Pr(B_1)) \ldots (B_n\ Pr(B_n))]$.

With Lemma 1, we can realize that the most probable composite hypothesis is in $ArgMax_{R_i}[Bel(R_1) \ldots Bel(R_n)]$; where R_is are the possible states of the designated root node r. The correctness of this realization can be urged as follows: along the propagation of a message stream, the local probability terms of all the descendent nodes of x are multiplied together. The optimum setting for a node x is based on the local maximum of the product of all probability terms reflected on $Bel(x)$. Because this local maximum depends only on the nodes that a message stream traverses as is stated in Lemma 1,[5] the optimum setting with respect to the local maximum is consistent with the global maximum. In other words, once the optimum setting for a node is found along the propagation of a message stream, this setting also corresponds to the one for the most probable

[5]Note that this is only true on singly connected networks.

composite hypothesis. By the time that all message streams reach the designated root node, the setting of every propositional variable for the most probable composite hypothesis is ready. A rigorous proof will be presented under theorem 2 in Section 6.

There are two important observations about this message passing scheme. First, the amount of information propagated in a message stream—thus the complexity of a local computation—is proportional to the size of the conditional probability table of a node. Consequently, finding the most probable composite hypothesis is a linear combination of the longest path of the propagation of a message stream and the size of the conditional probability table of a node in a network. This is consistent with the previous finding reported in [10]. Second, the message streams propagated via different paths can be processed in parallel. For example, the message streams propagated via the paths $d \rightarrow e \rightarrow c \rightarrow b$, $h \rightarrow g \rightarrow f$, and $j \rightarrow i \rightarrow f$ in Fig. 1 can all be processed simultaneously.

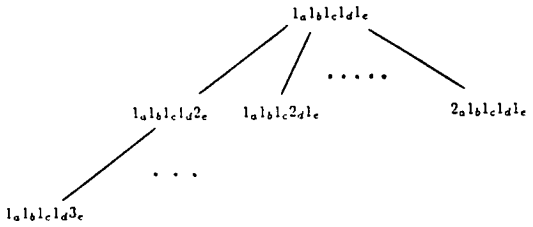
The intention of this work is not to duplicate previous effort on finding the most probable composite hypothesis; the major focus is to extend our formulation to finding the partial ordering of the few most probable composite hypotheses. Such an extension will be discussed in the next two sections.

5. PARTIAL ORDERING OF COMPOSITE HYPOTHESES

The objective of deriving the partial ordering of composite hypotheses, H_is, is to rank H_is based on $Pr(H_i|S_e)$; where S_e is an observation. For example, suppose $S_e = \bar{A}BCE$ for the network shown in Fig. 1, the four largest $Pr(H_i|S_e)$ are $Pr(DF\bar{G}\bar{H}\bar{I}\bar{J}|\bar{A}BCE) = 0.1077 \geq Pr(DF\bar{G}\bar{H}\bar{I}J|\bar{A}BCE) = 0.0984 > Pr(DFG\bar{H}\bar{I}\bar{J}|\bar{A}BCE) = 0.0561 \geq Pr(\bar{D}F\bar{G}\bar{H}\bar{I}\bar{J}|\bar{A}BCE) = 0.05386$. The partial ordering of the four most probable composite hypotheses are $DF\bar{G}\bar{H}\bar{I}\bar{J} \geq DF\bar{G}\bar{H}\bar{I}J \geq DFG\bar{H}\bar{I}\bar{J} \geq \bar{D}F\bar{G}\bar{H}\bar{I}\bar{J}$.

Let's first consider a five-node network in which all variables are independent to each other. That is, $Pr(abcde) = Pr(a)Pr(b)Pr(c)Pr(d)Pr(e)$. To obtain the most probable composite hypothesis with $S_e = \varnothing$, we need to find the maximum of each local probability term as discussed previously. To locate the second most probable composite hypothesis, we need to consider all (except one) local probability terms to be the largest, and the remaining one to be the second largest. The possible second most probable composite hypotheses are listed in the second level of the tree shown in Table 1; where, for example, 2_c refers to the instantiation of variable c such that $Pr(c)$ is the second largest. In considering the third largest, it will be either one of the settings in the

Table 1. A Tree Elaboration of the Settings of 5 Variables



second level excluding the one being the second most probable composite hypothesis, or a setting with all (except one) local probability terms being the largest and the remaining one being the third largest. To expand the discussion to include a typical case such as Fig. 1 where the variables are not all independent, there are two crucial issues to consider: (1) the dependency constraints that impose the processing sequence of the local probability terms, and (2) the consistency of the local probability settings that requires deeper consideration beyond just the largest local probability terms for the most probable composite hypothesis. For example, we can have $Pr(\overline{G}|F)$ as being the largest among $Pr(g|f)$ in Fig. 3, and yet $Pr(\overline{F})$ is the largest $Pr(f)$. In this case, there is no consistent setting from the largest of each local probability term which corresponds to the most probable composite hypothesis.

To address the first issue, it is always possible to assign a process sequence that satisfies the dependency constraints. One trivial way is to assign the process sequence according to the direction of the message streams propagated in a network. For example, in Fig. 1, we will first consider $Pr(h|g)$ for the value(s) of h given all the possible values of g, then $Pr(g|f)$. Similarly, $Pr(f|b)$ will be considered *only* after $Pr(g|f)$, thus $Pr(h|g)$, and $Pr(i|fj)$ are ready. To summarize the processes involved in Fig. 1, there are three sequential processes, listed from left to right, which can be conducted simultaneously. They are: (1) $M_{d\to e}M_{e\to c}M_{c\to b}$, (2) $M_{h\to g}M_{g\to f}$, and (3) $M_{j\to i}M_{i\to f}$. $M_{f\to b}$ is formulated when (2) and (3) are ready. Finally, $M_{b\to a}$ is formulated when (1) and $M_{f\to b}$ are ready.

To address the second issue, we shall first revisit the notion of propagating message streams for finding the most probable composite hypothesis. Referring to Lemma 1, a message stream being propagated from a node x along a certain path in a network will have anticipated the information required for its immediate parent node and also will have summarized all the incoming information. For example, in Fig. 3, $M_{h\to g}$ carries the value(s) of h such that both $Max_h[Pr(h|G)]$ and $Max_h[Pr(h|\overline{G})]$ are available to g in an anticipation that g can only be either G or $\overline{G}$. When g

prepares $M_{g \to f}$, the product of $Pr(h|g)$ and $Pr(g|f)$ is considered. Note that the largest of $Pr(h|g)Pr(g|f)$ will have included one of the $Val[m^1_{h \to g}]$s (ie, $Max_h[Pr(h|G)]$ or $Max_h[Pr(h|\overline{G})]$), and as a matter of fact, $M_{g \to f}$ carries the largest $Pr(hg|f)$ for all the possible values of f. Because the largest $Pr(fgh)$ equals $Pr(hg|f)Pr(f)$, $M_{g \to f}$ and $Pr(f)$ is the sufficient information to derive the most probable composite hypothesis. At this stage, let's call this aforementioned process as one complete iteration. A message stream involved in this initial iteration is denoted by $M^1_{x \to y}$, where the superscript indicates the first iteration.

Now we are ready to extend the discussion to finding the second, and the next few most probable composite hypotheses. Let's suppose the most probable composite hypothesis in Fig. 3 is $F\overline{G}H$. The corresponding $m^1_{H \to \overline{G}}$, $m^1_{\overline{G} \to F}$, and m^1_F are *flagged* to indicate that $Arg[m^1_{H \to \overline{G}}]$, $Arg[m^1_{\overline{G} \to F}]$, and $Arg[m^1_F]$ are *consumed* in the derivation of a composite hypothesis. One way to flag those $Arg[\bullet]$ is to associate each node, x, with a stack, U^i_x; where the superscript i indicates the ith iteration. For example, after one iteration $U^2_h = \{m^1_{H \to \overline{G}}\}$, $U^2_g = \{m^1_{\overline{G} \to F}\}$, and $U^2_f = \{m^1_F\}$. In order to consider the second most probable composite hypothesis in the second iteration, we start from node h again. There are three possibilities for the value of g from h's point of view. One is that the second most probable composite hypothesis has an identical value of g as the most probable one, and $Pr(h|g)$ is a contributing term to the second most probable composite hypothesis. In this case, the second largest of $Pr(h|\overline{G})$, $Max^2_h[Pr(h|\overline{G})]$[6] must be included in $M_{h \to g}$. Second is that the second most probable composite hypothesis has an identical value of g as the most probable one, but $Pr(h|g)$ is not a contributing term. In this case, $m^1_{h \to \overline{G}}$ in $M^1_{h \to g}$ is needed. Third is that the second most probable composite hypothesis is due to a new value of g. In this case, $m^1_{h \to G}$ in $M^1_{h \to g}$ is needed. We can easily see in this simple case that $M^2_{h \to g}$ is simply $M^1_{h \to g}$ (ie, $[m^1_{h \to G}\, m^1_{h \to \overline{G}}]$), plus one additional piece of information—$Max^2_h[Bel(h|\overline{G})]$. In a general case, this additional peice of information can be represented by $Max^2_h[Bel(h|g_{U^2_h})]$. $g_{U^2_h}$ is the value of g in $Arg[m^1_{h \to g_{U^2h}}]$ such that the exclusion of $m^1_{h \to g_{U^2h}}$ from $M^1_{h \to g}$ will leave no information about $g_{U^2_h}$. Mathematically, we can formally define g_{U^ih} as follows:

DEFINITION 3 *The value, $g_{U^i_h}$, of a variable 'g' concerning the propagation of a message stream, $M^i_{h \to g}$, from 'h' to 'g' in the 'i' iteration is defined as*:

$$g_{U^i_h}: M^{i-1}_{h \to Arg_g[U^i_h]} - U^i_h = \varnothing$$

[6] We will use Max^i to denote the ith largest.

For example, suppose $U_h^4 = \{m^1_{H \to \overline{G}}\, m^2_{\overline{H} \to \overline{G}}\, m^1_{H \to G}\}$, then

$$Arg_g[U_h^4] = \{G\ \overline{G}\}.$$

If

$$M^3_{h \to g} = \left\{m^1_{H \to \overline{G}}\, m^2_{\overline{H} \to \overline{G}}\, m^1_{H \to G}\, m^2_{\overline{H} \to G}\right\},$$

then

$$\begin{aligned} M^3_{h \to G} - U_h^4 &= \left\{m^1_{H \to G}\, m^2_{\overline{H} \to G}\right\} - U_h^4 \\ &= \left\{m^2_{\overline{H} \to G}\right\}, \text{ and} \\ M^3_{h \to \overline{G}} - U_h^4 &= \varnothing. \end{aligned}$$

In this case, $g_{U_h^4} = \{g\colon M^3_{h \to Arg_g[U_h^4]} - U_h^4 = \varnothing\} = \overline{G}$.

Notice that the information in the message streams of successive iterations differs by only one piece of information. We can summarize this important observation in the following lemma:

LEMMA 2 *A message stream $M^i_{b \to a}$ for the ith iteration requires at most one more piece of information than the previous iteration. The relationship between the message stream of successive iterations can be defined recursively as follow*:

For $i = 1$

$$M^1_{b \to a} = \begin{cases} Max_b[Bel_1(b|A_1) \ldots Bel_1(b|A_n)] \\ \text{if } a \text{ is an immediate parent of } b; \\ Max[Bel_1(B_1) \ldots Bel_1(B_m)] \\ \text{if } b \text{ is an immediate parent of } a; \end{cases}$$

where $U_b^1 = \varnothing$

$$Bel_1(b) = \begin{cases} M^1_{d_1 \to b} \otimes \ldots \otimes M^1_{d_k \to b} \otimes Pr(b|a, p_1 \ldots p_p) \\ \text{if } a \text{ is an immediate parent of } b; \\ M^1_{p_1 \to b} \otimes \ldots \otimes M^1_{p_k \to b} \otimes Pr(b|p_1 \ldots p_p) \\ \text{if } b \text{ is an immediate parent of } a; \end{cases}$$

For $i > 1$

$$M^i_{b \to a} = \begin{cases} M^{i-1}_{b \to a} \cup ArgMax^i_b\left[Bel_i(b|a_{U_b^i})\right] \\ \text{if } a \text{ is an immediate parent of } b; \\ M^{i-1}_{b \to a} \cup ArgMax^i_b\left[Bel_i(b_{U_b^i})\right] \\ \text{if } b \text{ is an immediate parent of } a; \end{cases}$$

Remark: A subscript is introduced for *Bel* whose purpose is the same as the superscript of a message stream $M^i_{b \to a}$ to indicate the ith iteration.

THEOREM 1 *The size of a message stream grows at most incrementally with the number of iterations.*

Proof A direct consequence of Lemma 2. Q.E.D.

In the next section we will formulate the algorithm that derives the partial ordering of the few most probable composite hypotheses.

6. RECURRENCE LOCAL COMPUTATION APPROACH

Before the discussion of the recurrence local computation approach for the derivation of composite hypotheses, there is an important characteristic about the completeness of both $M_{b \to a}$ and $Bel(b)$ in Lemma 1 to be noted. It is summarized in the following theorem:

THEOREM 2 *The message stream that 'b' propagates to 'a' in the ith iteration,*
$M^i_{b \to a}$, carries sufficient and complete information for the derivation of the first i largest $Pr(p1|J_{p1}) \ldots Pr(pn|J_{pn})Pr(b|J_b)Pr(d1|bJ_{d1}) \ldots Pr(dm|bJ_{dm})$; where $p1, p2, \ldots, pn$ are the parent nodes of 'b', and $d1, d2, \ldots, dm$ are the daughter nodes of 'b'.

Proof Without the loss of generality, let's suppose $p1, p2, \ldots, pn$ and $d1, d2, \ldots, dm$ form two paths to propagate message streams to b[7] ie, $p1 \to p2 \to \ldots \to pn \to b$ (ie, $J_{PT} = P_T$ for $T > 1$) and $d1 \to \ldots \to dm \to b$ (ie, $J_{dT} = d(i + 1)$ and $Pr(dT/bJ_{dT}) = Pr(dT/J_{dT})$.

When $i = 1$, $M^1_{p1 \to p2}$ carries $Pr(p1)$ for $p1$ = all possible values of $p1$, $M^1_{p2 \to p3}$ carries $Max_{p2}[Pr(p1)Pr(p2|J_{p2})]$ for every possible value of $p2$, and so forth according to Lemma 2. When the message stream carrying all pi reaches b, we have $Max[Pr(p1)Pr(p2|J_{p2}) \ldots Pr(pn|J_{pn})Pr(b|J_b)]$. Similarly, $M^1_{d1 \to d2}$ carries $Max_{d1}[Pr(d1|J_{d1})]$ for every possible value of $d2$, $M^1_{d2 \to d3}$ carries $Max_{d2}[Pr(d1|J_{d1})Pr(d2|J_{d2})]$ for every possible value of $d3$, and so forth according to Lemma 2. When the message streams carrying all di reach b and combined with all pi, we get, in a general form, $Max[Pr(p1|J_{p1}) \ldots Pr(pn|J_{pn})Pr(b|J_b)Pr(d1|J_{d1}) \ldots Pr(dm|J_{dm})]$ for every possible value of b. From here we can see that one of the $Max_b[\bullet]$ must be the largest of $Max[Pr(pn|J_{pn}) \ldots Pr(d1|J_{d1}) \ldots Pr(b|J_b)]$.

[7] It is always possible to do so by recursively considering a subnetwork in which all pis and dis are conditionally independent with respect to "b," and treat the subnetwork as a special node in the original network.

When $i = 2$, $M^2_{p1 \to p2}$ carries the information of $M^1_{p1 \to p2}$, plus one additional term $Bel(p1_{U^2_{p1}})$. We note that one of the terms in $M^1_{p1 \to p2}$, or the new term being introduced according to Lemma 2 must be the second largest, because any other settings not included will definitely be less than the largest one, and at least one term in $M^2_{p1 \to p2}$, thus it can at most be the third largest, which can never be the second largest. Applying a similar argument through the propagation of the message streams about *pi* and *di*, and inductively following the same line of reasoning over i, we will find that $M^i_{b \to a}$ carries sufficient and complete information for the derivation of the first i largest $Pr(p1|J_{p1}) \ldots Pr(pn|J_{pn})Pr(b|J_b)Pr(d1|bJ_{d1}) \ldots Pr(dm|bJ_{dm})$. Q.E.D.

Using theorem 2, a straightforward manner of applying the idea of propagating message streams for the derivation of composite hypotheses will be to identify (1) a designated root node and (2) the propagation path for message passing. In Lemma 2 and Theorem 1, we see that the amount of information carried in $M_{b \to a}$, thus the local computational load, is linearly proportional to the number of iterations, thus the length of the partial ordering. In addition, the size of a message propagated from a node n_i to a node n_j, is at most the number of states of n_j—if n_j is an immediate parent node of n_i—and is at most the number of states of n_i otherwise. Although the complexity of the convolution operation seems to grow exponentially with the number of messages merged in a node, it turns out to be a linear order—with a scaling proportionality equal to the maximum number of states among the nodes that *communicate* via message streams. This is so because the convolution operation imposes the constraint that only consistent terms are combined through multiplication. This observation can be summarized in the following lemma:

LEMMA 3 *The time complexity of a convolution operation in a node is in a linear order with respect to the number of messages, e, to the node and the size of the conditional probability table of the node, p.*

Proof Because the size of each message to a node $n_i - m$, is at most equal to the number of states of n_i, m is always less than or equal to p. The number of multiplications in the convolution operation is at most ep because only the "like" terms are multiplied. Q.E.D.

The formulation discussed so far, however, is not an optimal solution for the derivation of partial ordering yet. The complexity can be further reduced from a second order reasoning of the relative magnitudes of the probability terms.

Let's consider $M^1_D = [(D_1\ 0.7)(D_2\ 0.3)(D_3\ 0.5)]$; where D_is are the composite hypotheses derived in a designated root node. We can easily see that D_1 is the most probable composite hypothesis. In addition, D_3 is a potential second most probable composite hypothesis. It is clear that 0.7

(from $Pr(D_1)$) is the upper bound of the probability value of the second most probable composite hypothesis. But from $Pr(D_3) = 0.5$, we also know that the probability value of the second most probable composite hypothesis must be at least equal to 0.5. To extend this to a general case, we can define a *rejection threshold* as in definition 4 to determine whether the introduction of the additional term, $ArgMax_b^i[Bel_i(b|a_{U_b^i})]$ or $ArgMax_b^i[Bel_i(b_{U_b^i})]$, in Lemma 2 is necessary at each level of local computation. This result is summarized in Lemma 4.

DEFINITION 4 *A rejection threshold, R_i for the ith iteration is defined as $Max_d^i[Bel_{i-1}(d)]/Max_d^{i-1}[Bel_{i-1}(d)]$; where $i > 1$, and d is the combination of the variables for a composite hypothesis.*

LEMMA 4 *A second order reasoning using R_i can be realized as a condition test to determine, at any level of local computation about Bel, whether the introduction of $ArgMax_b^i[Bel_i(b|a_{U_b^i})]$ or $ArgMax_b^i[Bel_i(b_{U_b^i})]$ in Lemma 2 is necessary. The additional term, $ArgMax_b^i[Bel_i(b|a_{U_b^i})]$ or $ArgMax_b^i[Bel_i(b_{U_b^i})]$ will be introduced* **ONLY IF** *one of the following cases is true*:

CASE 1: a is an immediate parent of b and

$$\frac{Max_b^i\left[Bel_i\left(b|a_{U_b^i}\right)\right]}{Max_b^{i-1}\left[Bel_i\left(b|a_{U_b^i}\right)\right]} \geq R_i; \text{ or}$$

CASE 2: b is an immediate parent of a and

$$\frac{Max_b^i\left[Bel_i(b_{U_b^i})\right]}{Max_b^{i-1}\left[Bel_i(b_{U_b^i})\right]} \geq R_i;$$

By combining the results obtained from Lemma 2, Lemma 4, and Theorem 1, the recurrence local computation algorithm for the derivation of a partial ordering of the most probable composite hypotheses can be described as follows:

STEP 1:

Define $l \leftarrow$ length of partial ordering (ie, number of most probable composite hypotheses to be sought).

STEP 2:

Designate a "root" node as an absorption center, identify the settings for the evidence variables in S_e, and identify the paths for the propagation of message streams.

STEP 3:

Initialize the iteration count, $i = 1$, rejection ratio $R_i = 0$, and $U_x^1 = \varnothing$ for all the non-evidence nodes x in a network.

STEP 4: (Loop starts here—composing message streams)

Compose $M^i_{b \to a|S_e}$ using Lemma 4; ie,

$$M^i_{b \to a|S_e} = \begin{cases} M^{i-1}_{b \to a|S_e} \cup ArgMax^i_b\left[Bel_i\left(b|S_e, a_{U_b^i}\right)\right] \\ \text{if case 1 or 2 is true;} \\ M^{i-1}_{b \to a|S_e} \qquad \text{otherwise;} \end{cases}$$

where

CASE 1: a is an immediate parent of b and

$$Max^i_b\left[Bel_i\left(b|S_e, a_{U_b^i}\right)\right] / Max^{i-1}_b\left[Bel_i\left(b|S_e, a_{U_b^i}\right)\right] \geq R_i$$

CASE 2: b is an immediate parent of a and

$$Max^i_b\left[Bel_i\left(b_{U_b^i}|S_e\right)\right] / Max^{i-1}_b\left[Bel_i\left(b_{U_b^i}|S_e\right)\right] \geq R_i$$

STEP 5: (Deriving composite hypothesis)

Identify the setting of the composite hypothesis with the largest $Pr(H_i|S_e)$ in the designated root node.

STEP 6: (Updating parameters)

Update all U_x^is. Re-estimate a better bound of R if a lower bound is still available, otherwise arbitrary choose a safe lower bound and recompute R. Increment the iteration count i.

STEP 7:

Repeat steps 4 to 7 until i reaches l.

The pseudocode of a sequential version of the algorithm is shown in the Appendix, and the time complexity of the algorithm is discussed in theorem 3.

THEOREM 3 *The time complexity of the recurrence local computation method shown above is in an order $O(lkn)$ when parallel processing is permitted; where l is the length of a partial order, k the length of the longest path in a network, and n the maximum number of node states—defined as the product of the size of the conditional probability table of a node and the number of incoming messages towards the node.*

Proof First, we noted that the algorithm is based on the propagation of quantitative vector streams in a feed-forward manner to a designated "root" node in a network. In one complete iteration of propagating the vector streams to the "root" node, one composite hypothesis of the ordering can be identified. To obtain the l most probable composite hypotheses, l iterations will be needed.

When parallel processing is permitted, the amount of time required for each iteration will be at most the amount of time required for the convolution operations in the longest path (ie, length k stated in the theorem). Because the node states is the worst case of the time complexity of one convolution operation (see Lemma 3), the time complexity for one iteration is $O(kn)$, and for l iterations, the time complexity is $O(lkn)$. Q.E.D.

7. EXAMPLE ILLUSTRATION

To illustrate the recurrence local computation algorithm, we will make use of the Bayesian network shown in Fig. 1. Let's assume the observation $S_e = F\bar{H}$, and the query to the system is the first three composite hypotheses that are most probable with respect to $Pr(abcdegij|F\bar{H})$. Note that $Pr(abcdegij|F\bar{H}) = [1/Pr(F\bar{H})]Pr(abcdeFg\bar{H}ij)$; where $1/Pr(F\bar{H})$ is a constant. The partial ordering of $Pr(abcdegij|F\bar{H})$s is identical to the ordering of $Pr(abcdeFg\bar{H}ij)$s. Indeed $Pr(abcdeFg\bar{H}ij)$ is a scaled version of $Pr(abcdegij|F\bar{H})$ with scaling factor $1/Pr(F\bar{H})$. Therefore, we can derive the three most probable composite hypotheses (given $S_e = F\bar{H}$) from $Pr(abcdeFg\bar{H}ij)$ without actually knowing the value of $1/Pr(F\bar{H})$. Of course, the value of $1/Pr(F\bar{H})$ must be known in order to obtain the quantitative values of $Pr(abcdegij|F\bar{H})$s. In this example, we assume the quantitative values of $Pr(abcdegij|F\bar{H})$s are of no interest to us.

Referring to the algorithm in the previous section, we first initialize the appropriate parameters. They are: $S_e = F\bar{H}$, length of partial ordering $l = 3$, iteration count $i = 1$, rejection ratio $R_1 = 0$, and $U_x^1 = \varnothing$ for $x \in \{a, b, c, d, e, g, i, j\}$. We also select node a as the absorption center (ie, the designated root node), and follow the *thick* arrows in Fig. 1 to define the direction of propagation.

Table 2. Probabilistic Knowledge of Fig. 1

$Pr(A) = 0.2$	$Pr(D) = 0.6$	$Pr(J) = 0.54$	
$Pr(B\|A) = 0.7$	$Pr(B\|\bar{A}) = 0.1$	$Pr(C\|B) = 0.1$	$Pr(C\|\bar{B}) = 0.7$
$Pr(E\|CD) = 0.4$	$Pr(E\|\bar{C}D) = 0.65$	$Pr(E\|C\bar{D}) = 0.3$	$Pr(E\|\bar{C}\bar{D}) = 0.45$
$Pr(F\|B) = 0.82$	$Pr(F\|\bar{B}) = 0.6$	$Pr(G\|F) = 0.32$	$Pr(G\|\bar{F}) = 0.6$
$Pr(H\|G) = 0.22$	$Pr(H\|\bar{G}) = 0.3$		
$Pr(I\|FJ) = 0.3$	$Pr(I\|F\bar{J}) = 0.1$	$Pr(I\|\bar{F}J) = 0.6$	$Pr(I\|\bar{F}\bar{J}) = 0.2$

$Pr(abcdefghij)$
$= Pr(a)Pr(b|a)Pr(c|b)Pr(d)Pr(e|cd)Pr(f|b)Pr(g|f)Pr(h|g)Pr(i|fj)Pr(j)$

Starting from nodes d, h, and j, the belief matrices and initial message streams can be derived directly from the probabilistic information in Table 2:

$$U_d^1 = \varnothing$$

$$Bel_1(d|S_e) = Pr(d|S_e) = [(D\ 0.6)\ (\bar{D}\ 0.4)]$$

$$M^1_{d \to e|S_e} = [(D\ 0.6)\ (\bar{D}\ 0.4)]$$

$$U_h^1 = \varnothing$$

$$Bel_1(h|S_e) = Pr(h|g, S_e) = \left[(G\bar{H}\ 0.78)\ (\bar{G}\bar{H}\ 0.7)\right]$$

$$M^1_{h \to g|S_e} = Max^1_h\left[Bel_1(h|S_e, G)Bel_1\left(h|S_e, \bar{G}\right)\right]$$

$$= \left[(G\bar{H}\ 0.78)\ (\bar{G}\bar{H}\ 0.7)\right]$$

$$U_j^1 = \varnothing$$

$$Bel_1(j|S_e) = Pr(j|S_e) = [(J\ 0.54)\ (\bar{J}\ 0.46)]$$

$$M^1_{j \to i|S_e} = [(J\ 0.54)\ (\bar{J}\ 0.46)]$$

When node d propagates the message streams to e, h to g, and j to i, we obtain:

$$U_e^1 = \varnothing$$

$$Bel_1(e|S_e)$$

$$= M^1_{d \to e|S_e} \otimes Pr(e|dc, S_e)$$

$$= [(D\ 0.6)\ (\bar{D}\ 0.4)]$$

$$\otimes \begin{bmatrix} (ECD\ 0.4) & (E\bar{C}D\ 0.65) & (\bar{E}CD\ 0.6) & (\bar{E}\bar{C}D\ 0.35) \\ (EC\bar{D}\ 0.3) & (E\bar{C}\bar{D}\ 0.45) & (\bar{E}C\bar{D}\ 0.7) & (\bar{E}\bar{C}\bar{D}\ 0.55) \end{bmatrix}$$

$$= \begin{bmatrix} (ECD\ 0.24) & (E\bar{C}D\ 0.39) & (\bar{E}CD\ 0.36) & (\bar{E}\bar{C}D\ 0.21) \\ (EC\bar{D}\ 0.12) & (E\bar{C}\bar{D}\ 0.18) & (\bar{E}C\bar{D}\ 0.28) & (\bar{E}\bar{C}\bar{D}\ 0.22) \end{bmatrix}$$

$$M^1_{e \to c|S_e} = [(E\bar{C}D\ 0.39)\ (\bar{E}CD\ 0.36)]$$

$$U_g^1 = \varnothing$$

$$Bel_1(g|S_e) = M^1_{h \to g|S_e} \otimes Pr(g|f, S_e)$$

$$= \left[(G\bar{H}\ 0.78)\ (\bar{G}\bar{H}\ 0.7)\right] \begin{bmatrix} (GF\ 0.32) \\ (\bar{G}F\ 0.68) \end{bmatrix}$$

$$= \begin{bmatrix} (G\bar{H}F\ 0.2496) \\ (\bar{G}\bar{H}F\ 0.476) \end{bmatrix}$$

Remark: $Pr(g|\bar{F}, S_e)$ and $Bel_1(g|\bar{F}, S_e)$ are don't care terms because f is instantiated to be F in S_e.

$$M^1_{g \to f|S_e} = Max^1_g[Bel_1(g|f, S_e)]$$

$$= \left[(\bar{G}\bar{H}F\ 0.476)\right]$$

$$U_i^1 = \varnothing$$

$$Bel_1(i|S_e) = M^1_{j \to i|S_e} \otimes Pr(i|fj, S_e)$$

$$= [(J\ 0.54)\ (\bar{J}\ 0.46)]$$

$$\otimes \begin{bmatrix} (IFJ\ 0.3) & (\bar{I}FJ\ 0.7) \\ (IF\bar{J}\ 0.1) & (\bar{I}F\bar{J}\ 0.9) \end{bmatrix}$$

$$= \begin{bmatrix} (IFJ\ 0.162) & (\bar{I}FJ\ 0.378) \\ (IF\bar{J}\ 0.046) & (\bar{I}F\bar{J}\ 0.414) \end{bmatrix}$$

$$M^1_{i \to f|S_e} = \left[(F\bar{I}\bar{J}\ 0.414)\right]$$

Note that at this stage node b is idle and has to wait the message stream from f. When the message streams from node g and i reach f, we are ready to construct the message stream $M^1_{f\to b|S_e}$.

$$U^1_f = \varnothing$$

$$Bel_1(f|S_e) = M^1_{g\to f|S_e} \otimes M^1_{i\to f|S_e} \otimes Pr(f|b, S_e)$$

$$= \left[(\bar{G}\bar{H}F\ 0.476)\right]\left[(F\bar{I}\bar{J}\ 0.414)\right]\left[(FB\ 0.82)\ (F\bar{B}\ 0.6)\right]$$

$$= \left[\left(BF\bar{H}\bar{G}\bar{I}\bar{J}\ 0.16159\right)\left(\bar{B}\bar{G}\bar{H}\bar{I}\bar{J}F\ 0.11823\right)\right]$$

$$M^1_{f\to b|S_e} = \left[\left(BF\bar{H}\bar{G}\bar{I}\bar{J}\ 0.16159\right)\left(\bar{B}\bar{G}\bar{H}\bar{I}\bar{J}F\ 0.11823\right)\right]$$

Following similar procedure, we can compute $M^1_{c\to b|S_e}$, and thus $M^1_{b\to a|S_e}$ as below:

$$U^1_b = \varnothing$$

$$Bel_1(b|S_e) = M^1_{c\to b|S_e} \otimes M^1_{f\to b|S_e} \otimes Pr(b|a, S_e)$$

$$M^1_{b\to a|S_e} = \left[\left(AB\bar{C}DEF\bar{G}\bar{H}\bar{I}\bar{J}\ 0.0397\right)\left(\bar{A}\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.0268\right)\right]$$

At node a,

$$U^1_a = \varnothing$$

$$Bel_1(a|S_e) = M^1_{b\to a|S_e} \otimes Pr(a|S_e)$$

$$= \left[\left(AB\bar{C}DEF\bar{G}\bar{H}\bar{I}\bar{J}\ 0.00794\right)\left(\bar{A}\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.02144\right)\right]$$

$$= M^1_{a|S_e}$$

The most probable composite hypothesis can now easily be derived from $M^1_{a|S_e}$. That is, given the observation $S_e = F\bar{H}$, the most probable composite hypothesis is $\bar{A}\bar{B}CD\bar{E}\bar{G}\bar{I}\bar{J}$ and $Pr(\bar{A}\bar{B}CD\bar{E}\bar{G}\bar{I}\bar{J}|F\bar{H}) = 0.02144/Pr(F\bar{H})$.

Before we start the second iteration to derive the second most probable composite hypothesis, we first update the following parameters accordingly:

$$\text{Iteration count } i = 2,\ R_2 = \frac{0.00794}{0.02144} = 0.37 \text{ (definition 4)}$$

Given $H^* = \overline{A}BCD\overline{E}\overline{G}\overline{I}\overline{J}$

$U_d^2 = \{(D\ 0.6)\}$ $U_h^2 = \{(\overline{HG}\ 0.7)\}$ $U_i^2 = \{(\overline{I}F\overline{J}\ 0.414)\}$

$U_e^2 = \{(\overline{E}CD\ 0.36)\}$ $U_g^2 = \{(\overline{G}\overline{H}F\ 0.476)\}$

$U_c^2 = \{(\overline{B}CD\overline{E}\ 0.252)\}$ $U_f^2 = \{(F\overline{B}\overline{G}\overline{H}\overline{I}\overline{J}\ 0.11823)\}$

$U_b^2 = \{(\overline{A}\overline{B}CD\overline{E}F\overline{G}\overline{H}\overline{I}\overline{J}\ 0.0268)\}$

Starting from nodes d, h, and j again, because there is only one possible $Pr(d)$ for $d = D$ and $Pr(j)$ for $j = J$, $M^2_{d\to e|S_e} = M^1_{d\to e|S_e}$, and $M^2_{j\to i|S_e} = M^1_{j\to i|S_e}$. In addition, because S_e indicates that $h = \overline{H}$, there is no second largest term of $Pr(h|\overline{G})$ other than the only one that has already been included in $M^1_{h\to g|S_e}$. Because of this, the examination of rejection ratio can be avoided and $M^2_{h\to g|S_e} = M^1_{h\to g|S_e}$. However, we generally have to use rejection ratio to examine the necessity of introducing a new term into a message stream. In considering $M^2_{e\to c|S_e}$, note that:

$$M^2_{e\to c|S_e} = \begin{cases} M^1_{e\to c|S_e} \cup ArgMax^2_e\left[Bel_2\left(e|S_e, c_{U_e^2}\right)\right] \\ \text{if } \dfrac{Max^2_e\left[Bel_2\left(e|S_e, c_{U_e^2}\right)\right]}{Max^1_e\left[Bel_2\left(e|S_e, c_{U_e^2}\right)\right]} \geq R_2 = 0.37 \\ M^1_{e\to c|S_e} \qquad \text{otherwise;} \end{cases}$$

Because $U_e^2 = \{(\overline{E}CD\ 0.36)\}$, we get $c_{U_e^2} = C$ (definition 3), and the second largest of $Bel_2(e|S_e, C)$, $Max^2_e[\bullet] = 0.28$ (from $Pr(\overline{E}|C\overline{D}) = 0.28$). As $Max^1_e[\bullet] = 0.36$ (from $Pr(\overline{E}|CD) = 0.36$), $0.28/0.36 \geq R_2 = 0.37$. Therefore,

$$M^2_{e\to c|S_e} = \left[(CD\overline{E}\ 0.36)\ (C\overline{D}\overline{E}\ 0.28)\ (\overline{C}DE\ 0.39)\right]$$

Because now $M^2_{e\to c|S_e}$ differs from $M^1_{e\to c|S_e}$, $Bel_2(c|S_e)$ has to be computed due to the inclusion of the new term $(C\overline{D}\overline{E}\ 0.28)$. This can be easily done by incorporating $Bel_1(c|S_e)$ with $[(C\overline{D}\overline{E}\ 0.28)] \otimes [(CB\ 0.1)\ (C\overline{B}\ 0.7)]$. By adding the results of the convolution,

$$Bel_2(c|S_e) = \begin{bmatrix} (\overline{E}CBD\ 0.036) & (\overline{E}CD\overline{B}\ 0.252) \\ (\overline{E}CB\overline{D}\ 0.028) & (\overline{E}C\overline{D}\overline{B}\ 0.196) \\ (E\overline{C}BD\ 0.351) & (E\overline{C}D\overline{B}\ 0.117) \end{bmatrix}$$

Following similar procedures for $g \rightarrow f$, $i \rightarrow f$, $f \rightarrow b$, $c \rightarrow b$, and $b \rightarrow a$, we will get

$$Bel_2(a|S_e) = \begin{bmatrix} (AB\bar{C}DEF\bar{G}\bar{H}\bar{I}\bar{J}\ 0.00794) \\ (\bar{A}\,\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.02144) \\ (\bar{A}\,\bar{B}CD\bar{E}F\bar{G}\bar{H}IJ\ 0.019587) \end{bmatrix}$$

Now we obtain the second most probable composite hypotheses as $\bar{A}\bar{B}CD\bar{E}\bar{G}IJ$. By updating the appropriate parameters accordingly, we get:

Iteration count $i = 3$, $R_3 = 0.00794/0.019587 = 0.4054$ (Note that $R_3 \geq R_2$)

Given $\bar{A}\bar{B}CD\bar{E}\bar{G}\bar{I}\bar{J}$ and $\bar{A}\bar{B}CD\bar{E}\bar{G}IJ$ as the two most probable composite hypotheses,

$$U_d^3 = \{(D\ 0.6)\} \qquad U_h^3 = \{(\overline{HG}\ 0.7)\} \qquad U_i^3 = \{(\bar{I}F\bar{J}\ 0.414)\}$$

$$U_e^3 = \{(\bar{E}CD\ 0.36)\} \qquad U_g^3 = \{(\bar{G}\bar{H}F\ 0.476)\}$$

$$U_c^3 = \{(\bar{B}CD\bar{E}\ 0.252)\} \qquad U_f^3 = \{(F\bar{B}\bar{G}\bar{H}\bar{I}\bar{J}\ 0.1823)\ (F\bar{B}\bar{G}\bar{H}\bar{I}J\ 0.108)\}$$

$$U_b^3 = \{(\bar{A}\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.0268)\ (\bar{A}\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}J\ 0.024484)\}$$

Note that U_f^3 and U_b^3 are different from U_f^2 and U_b^2. By going through similar procedures as discussed previously,

$$Bel_3(a|S_e) = \begin{bmatrix} (AB\bar{C}DEF\bar{G}\bar{H}\bar{I}\bar{J}\ 0.00794) \\ (\bar{A}\,\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.02144) \\ (\bar{A}\,\bar{B}CD\bar{E}F\bar{G}\bar{H}\bar{I}J\ 0.019587) \\ (\bar{A}\,\bar{B}C\bar{D}\bar{E}F\bar{G}\bar{H}\bar{I}\bar{J}\ 0.016684) \end{bmatrix}$$

Now the system is ready to supply an answer to the query about the three most probable composite hypotheses as below:

Rank	Composite hypothesis	Quantitative value
Most probable	$\bar{A}\bar{B}CD\bar{E}\bar{G}\bar{I}J$	$0.02144/Pr(F\bar{H})$
Second most probable	$\bar{A}\bar{B}CD\bar{E}\bar{G}IJ$	$0.019587/Pr(F\bar{H})$
Third most probable	$\bar{A}\bar{B}C\bar{D}\bar{E}\bar{G}\bar{I}J$	$0.016684/Pr(F\bar{H})$

It is interesting to note that only 29 multiplications are involved in deriving the most probable composite hypothesis, 12 extra multiplications to locate the second most probable composite hypothesis, and 6 additional to identify the third most probable one, or a total of 47 multiplications for the three most probable composite hypotheses.

8. SCOPE OF RECURRENCE LOCAL COMPUTATION AND ALTERNATIVE APPROACHES

From our previous discussion and the example illustrated in section 7, there is only feed-forward propagation of message streams towards a designated root node. Only one designated root node acts as an absorption center and no back propagation is involved. This differs from other local computation approaches [5, 18]. It is noted that if more than one absorption center is allowed, back propagation of local computation is necessary to ensure an equilibrium state in which the settings of all variables are consistent. The major reason that this recurrence local computation can avoid back propagation is that the propagation of message streams is forced towards one and only one absorption center in a unique path. In addition, the information being propagated in a message stream is exhaustive and complete as discussed in theorem 2. This guarantees consistency per iteration, thus convergence, because no inconsistency or back propagation is involved.

This approach, however, renders a major limitation as other local computational approaches do. It works only on singly connected networks. In terms of the classes of problems to which this approach can be applied, any Bayesian network whose topological structure is a subclass of singly connected network falls into the category. One obvious class will be tree-structured Bayesian networks. If we consider the classes of Bayesian networks in a hierarchy that tree-structured networks $\subseteq$ singly connected networks $\subseteq$ multiply connected networks, the scope of the classes of problems being covered by this recurrence local computation approach is rather limited. Two research avenues can be taken to deal with this

limitation. One is to focus on the possible transformation of a multiply connected network into a singly connected network. Second is to focus on the development of alternative heuristic and local computational approaches to deal with other classes of networks that are not covered under singly connected networks. If this second avenue is taken, not only has the development of alternative approach to be focused, but the categorization of the networks to which the alternative approach can be applied is equally important. The development of an alternative local computation approach is underway, and an attempt of classifying Bayesian networks on which this alternative approach can be applied is also initiated.

Regarding the expressiveness of the conclusion derived from this recurrence local computation approach, we found that the conclusion can be considered as the lower bound of any query made to a Bayesian network. Consider a query about nodes d, c, and i in Fig. 1 is made given the observation $S_e = \bar{A}BH$ (ie, $Pr(dci|S_e = \bar{A}BH)$), the local computational approaches elsewhere [5, 11, 24] can be used to compute $Pr(d|S_e)$, $Pr(c|S_e)$, $Pr(i|S_e)$, and the upper bound of $Pr(S_e)$ (in this case, the upper bound is $Min[Pr(\bar{A})\ Pr(B)\ Pr(H)] \geq Pr(S_e)$). In addition, this recurrence local computation approach can be used to compute $Pr(dci, efgij|S_e)$s. One can easily see that $Min[Pr(d|S_e)\ Pr(c|S_e)\ Pr(i|S_e)]$ and $Max[Pr(dci, efgij|S_e)]$ serve as the upper and lower bound of $Pr(dci|S_e)$ respectively; ie, $Min[Pr(d|S_e)\ Pr(c|S_e)\ Pr(i|S_e)] \geq Pr(dci|S_e) \geq Max[Pr(dci, efgij|S_e)]$. Although the posterior likelihood of a local composite hypothesis, X (ie, a nonexhaustive instantiation of non-evidence variables), $Pr(X|S_e)$, can be efficiently computed in time that is linear in the size of the belief network, the upper and lower bounds of $Pr(X|S_e)$ are available at no added cost when the likelihoods of the corresponding simple hypothesis (S) and composite hypothesis (H), $Pr(S|S_e)$ and $Pr(H|S_e)$, are available. This provides an "opportunistic" derivation of the partial ordering of local composite hypotheses whose details are beyond the scope of this paper and readers are referred to [25].

9. EXTENSION TO MULTIPLY CONNECTED NETWORK

As we pointed out in the previous section, the recurrence local computation approach fails to produce the correct answer in a multiply connected network. One of the main reasons is that a common parent node or a common daughter node (such as nodes a and d in Fig. 2, respectively) may receive conflict message streams along the path(s) of propagation. For example, suppose the propagation is bottom-up (ie, message streams propagating from node d to a) and the values of b, c, and d are identified along the propagation of message streams. Because $Pr(abcd) =$

$Pr(a)Pr(b|a)Pr(c|a)Pr(d|bc)$, node a may receive different values of b and c when the maximum of $Pr(b|a)Pr(d|bc)$ and $Pr(c|a)Pr(d|bc)$ are considered via different paths.

One possible way to avoid conflict information is to use the technique of *clustering* discussed in [4]. The idea of clustering is to lump variables together to form compound variables in such a way that the resultant clustering produces a singly connected network. For example, nodes b and c in Fig. 2 can be grouped together to form a compound variable bc such that $Pr(bc|a) = Pr(b|a)Pr(c|a)$, and $Pr(d|bc)$ remains the same. Once we lump nodes b and c together, the network becomes a singly connected network as shown in Fig. 4. There are two important points to note about clustering. First, there could be more than one way to cluster variables into compound variables in order to get a singly connected network. For example, Fig. 6a and 6b are two possible transformations of Fig. 5. Second, the computational complexity remains the same even though an arbitrary network can always be transformed into a seemingly simpler network. For example, we can lump as many nodes together as possible to reduce the number of nodes in a network, thus the length of the longest path. However, by clustering the variables into a compound variable, the computational load of processing the compound variable is exponentially increased. Because the computational complexity is proportional to the length of the longest path in a network and the maximum number of node states, the overall computational load in terms of the number of multiplications will still be the same. However, it is possible that certain configurations of a network with compound variables are *more efficient* than the others with respect to a fixed directional flow of message streams. For example, if we compare Fig. 6a and 6b, and assuming the propagation of messages are from the root nodes to the leaf node, the processing time

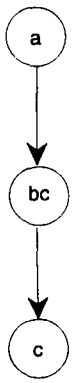


Figure 4. Clustering of fig. 2.

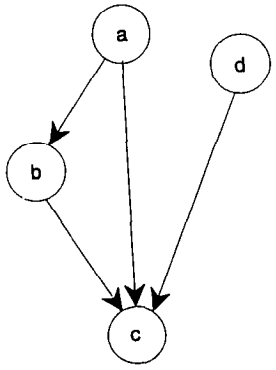


Figure 5. Simple multiply connected network.

required for Fig. 6b will be less because of the symmetrical configuration. Because both *a* and *b* have two states, the awaiting time for the synchronization of nodes *a* and *d* in Fig. 6b will be less than the awaiting time of nodes *ab* and *d* in Fig. 6a. We expect that further research is necessary to better understand how to optimize node clustering in a multiply connected network.

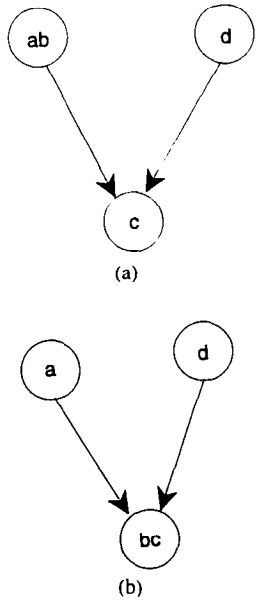


Figure 6. (a) Clustering of fig. 5. (b) Second clustering of fig. 5.

10. CONCLUSION

A recurrence local computation approach is discussed for the derivation of the partial ordering of the few most probable composite hypotheses. This approach is based on the propagation of message streams that carry local probability terms towards a designated root node from which the most probable composite hypotheses can be deduced. A simple rejection ratio based on second order reasoning is derived to reduce the amount of information being propagated in a message stream, thus reducing the computational load. The complexity involved in each local computation is linearly proportional to the size of the conditional probability table of a node and the amount of information in the incoming message streams.

As is illustrated in Section 7, the recurrence local computation approach relies only on the values of local probability terms and their relative magnitude in the derivation of a partial ordering. This allows us to avoid the evaluation of $Pr(S_e)$.[8] Whenever the quantitative value of $Pr(S_e)$ is known, the probability values, $Pr(H_i|S_e)$s, of the most probable composite hypotheses can also be derived.

There are, however, two major limitations of this recurrence local computation approach. First, this approach is limited to singly connected networks. A transformation algorithm to handle any arbitrary Bayesian network is required, or alternative approaches to compliment this approach are needed. Second, in order to make use of the result of this approach to estimate a lower bound of any query as is discussed in section 8, the quantitative values of $Pr(H_i|S_e)$, thus $Pr(S_e)$, must be known. In a non-trivial case (eg, $|Se| > 1$), we will need an efficient algorithm to compute $Pr(S_e)$. These two limitations lead to two open questions for future research:

1. In view of the current research efforts on reasoning in Bayesian networks, how do we gather the meta (common sense) knowledge about the boundary or limitation of various inference algorithms with respect to the queries that one can handle?
2. What is the best way of categorizing Bayesian networks so that we can tell which inference algorithms will be most efficient for a certain class of Bayesian networks? Should the classification be qualitative in terms of the topological structure of a network, quantitative in terms of the probability distribution and the total entropy [26] of a network, a combination of two, or other criteria?[9]

[8] Computing $Pr(S_e)$ can be NP-hard in a multiply connected network as was pointed out by a reviewer [22] and a reviewer of this paper.

[9] Some suggestions based on the topological structures and probability distributions of a network have been discussed [2, 27].

Our future research will focus on these two open questions and on continuing our exploration of this uncertain research.

ACKNOWLEDGMENT

I am indebted to the anonymous referees for their constructive comments and excellent guidelines that have been greatly helpful in improving the manuscript. I also thank Peter Szolovits of M.I.T. for his valuable comments on an early draft of this paper and for his reference to some current relevant works.

APPENDIX

```
  /Pseudo code of the RLCM/
function max(N, BELMATRIX, EVIDENCE, FOCUSNODE): valuelist;
  {return the Nth largest valuelist of all the possible states of
    FOCUSNODE from the given BELMATRIX conditioned on EVI-
DENCE set, or return the largest unused valuelist at the Nth iteration if
FOCUSNODE = nil}
  /***Example:
max(1,[(IFJ 0.3)(IFJ' 0.1)(I'FJ 0.7)(I'FJ 0.9)], {H}, F)
  → ((IFJ 0.3)(I'FJ 0.9))
max(3,[(IFJ 0.3)(IFJ' 0.1)(I'FJ 0.7)(I'FJ 0.9)], {H}, nil)
  → (IFJ 0.3) if (I'FJ 0.9) and (I'FJ 0.7) are used in Max^1 and Max^2
  → (I'FJ 0.7) if only (I'FJ 0.9) is used so far                    ***/
function send-msg-process(SENDER, RECEIVER, DEPTH): void;
{
    set Upperlist = max((DEPTH-1), node[SENDER].belief-matrix,
                        union(EVIDENCE, get-setting(SENDER,
                                RECEIVER, SENDER-OF-
                                SENDER-LIST), nil);
    set Lowerlist = max(DEPTH, node[SENDER].belief-matrix,
                        union(EVIDENCE, node[RECEIVER].setting-
                                used), nil);
    if value(Lowerlist)/value(Upperlist) > rejection-ratio then
      {if (iter = = 1) then
          set send-msg[SENDER, RECEIVER]
            = (parent[SENDER, RECEIVER = = true)?
              max(1, node[SENDER].belief-matrix, EVIDENCE,
                    SENDER)
              /***if SENDER is a parent of RECEIVER***/
```

```
            :max(1, node[SENDER].belief-matrix, EVIDENCE,
                    RECEIVER)
              /***if RECEIVER is a parent of SENDER***/
    else
      set send-msg[SENDER, RECEIVER] =
          append(Lowerlist, send-msg[SENDER, RECEIVER])
        /***include additional information***/
    }
  set msg-for[RECEIVER] =
      union(msg-for[RECEIVER], send-msg[SENDER, RECEIVER]);
}
function update-belief-matrix(LIST-OF-MSG, PROCESS-NODE): void;
{
    set node[PROCESS-NODE].belief-matrix =
      union(node[PROCESS-NODE].belief-matrix,
          convolve(node[PROCESS-NODE].prob-table,  LIST-OF-
                  MSG));
}
                    /Sequential version of the RLCM/
{
 input L, evidence                  /length of partial ordering and evi-
                                      dence/
 set root, prog-path                /define propagation path and root
                                      node/
 initialize {
          rejection-ratio = 0;
          node[x].belief-matrix = nil     ∀x ∈ N   /belief-matrix of
                                                   every node = nil/
          send-msg[from, to] = nil     ∀ from, to: succ(from) = to ∈
                                         Prog-path
            /initially no message sent by any node/
          node[x].depth = 1 Level of the most recently instantiated X in
            the Lattice tree
          msg-for[x] = nil     ∀x ∈ prog-path
          node[x].setting-used = nil/record the most recently instantiated
            x − X_{U^{ii}_{prec(s)}}/
          }
loop (iter: 1 to l; + + iter){
  loop (process-node: first (prog-path); succ(prog-path)){
    loop (sender-node: first(node[process-node].msgsender);
         succ(node[process-node].msgsender)){
         if sender-node ≠ nil then
```

```
            send-msg-process(sender-node, process-node,
              node[sender-node].depth)
          else
            node[process-node].belief-matrix = node[process-
              node].prob-table
  }
  update-belief-matrix(msg-for[process-node], process-node);
}
output the most recently found hypotheses;
update node[x].depth      ∀x ∈ prog-path;
  /**backtrack (from a queue) to an available one which is most recently
    used**/
update node[x].setting-used      ∀x ∈ prog-path;
  /**backtrack (from the queue) to an available one which is most
    recently used**/
  update rejection-ratio;
  clear msg-for[x]      ∀x ∈ prog-path                    /re-initialization/
  }
}
```

References

1. Cooper, G., The computational complexity of probabilistic inference using Bayesian belief networks, *Artif. Intell.* 42:393–405, 1990.
2. Henrion, M., An introduction to algorithm for inference in belief nets, *Uncertainty in Artificial Intelligence* 5 (MIPR 10), (M. Henrion, R. Shachter, L. Kanal, and J. Lemmer, Eds.), Elsevier, North-Holland, 129–138, 1990.
3. Wu, T., A problem decomposition method for efficient diagnosis and interpretation of multiple disorders, *Proceedings of 14th Symp. on Computer Appl. in Medical Care*, 86–92, 1990.
4. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, 1988.
5. Pearl, J., Distributed revision of composite beliefs, *Artif. Intell.* 33:173–215, 1987.
6. Sy, B. K., An AI-based CAD/CAM approach to assess design methodology of a user-specific nonvocal communication device, *Proceedings of 15th Northeast Bioengineering Conference*, Boston, Mass., 185, 1989.
7. Keramaris, V., Sy, B. K., A syllable based approach towards the intelligibility study of impaired speech, *Proceedings of 12th Annual International Conference of IEEE–EMBS*, Philadelphia, Penn., 1990.

8. Pearl, J., Fusion, propagation, and structuring in Bayesian networks, *Artif. Intelligence* 28:241–288, 1986.

9. Shachter, R. D., D'Ambrosio, B., Del Favero, B. A., Symbolic probabilistic inference in belief networks, *Proceedings of the 8th National Conference on AI*, Boston, Mass., 126–131, 1990.

10. Jensen, F. V., and Lauritzen, S. L., Olesen, K. G., Bayesian updating in recursive graphical models by local computation, Internal Technical Report, R-*89-15*, Institute of Electronic System, University of Aalborg, Denmark, 1989.

11. Jensen, F. V., Olesen, K. G., Anderson, S. K., An algebra of Bayesian belief universes for knowledge based systems, Internal Technical Report, R-88-25, Institute of Electronic System, University of Aalborg, Denmark, 1988.

12. Lauritzen, S. L., Spiegelhalter, D. J., Local computations with probabilities on graphical structures and their application to expert system, *J. Royal Statistics Society—British*. 50(2):157–224, 1988.

13. Rege, A., and Agogino, A. M., Topological framework for representing and solving probabilistic inference problems in expert systems, *IEEE Transactions on Systems, Man, and Cybernetics*. 18:402–414, 1988.

14. Wellman, M. P., Qualitative probabilistic networks for planning under uncertainty (revised), *Readings in Uncertain Reasoning*, (G. Shafer and J. Pearl, Eds.), Morgan Kaufmann, San Mateo, CA, 711–722, 1990.

15. Wellman, M. P., Fundamental concepts of qualitative probabilistic network, *Artif. Intell.* 44:257–304, 1990.

16a. Peng, Y., and Reggia, J. A., A probabilistic causal model for diagnostic problem solving—part I: integrating symbolic causal inference with numeric probabilistic inference, *IEEE Transactions on Systems, Man, and Cybernetics*. (2):146–162, 1987.

16b. Peng, Y., and Reggia, J. A., A probabilistic causal model for diagnostic problem solving—part II: diagnostic strategy, *IEEE Transactions on Systems, Man, and Cybernetics*. (3):395–406, 1987.

17. Shimony, S. E., and Charniak, E., A new algorithm for finding MAP assignments to belief networks, *Proceedings of the Conference on Uncertainty in AI*, Cambridge, MA 98–103, 1990.

18. Peot, M., Shachter, R. D., Fusion and propagation with multiple observations in belief networks, *Artif. Intell.* 1991.

19. Cooper, G., NESTOR: A computer-based medical diagnosis that integrates causal and probabilistic knowledge, *Technical Report HPP-84-48*, Stanford University, Stanford, CA.

20. Kjaerulff, U., Triangulation of graphs—algorithms giving small total state space, *Internal Technical Report, R-90-09*, Institute of Electronic System, University of Aalborg, Denmark, 1990.

21. Neapolitan, R., *Probabilistic Reasoning in Expert Systems*, John Wiley & Sons, New York, 317–370, 1990.

22. Sy, B. K., Qualitative reasoning of Bayesian belief using meta-knowledge, *Proceedings of the 11th International Joint Conference of AI*, Detroit, Michigan, 1128–113, 1989.

23. Sy, B. K., Reasoning composite hypotheses using qualitative approach. *Ann. Math. Artif. Intell.* (4):1–24, 1991.

24. Cooper, G. F., A diagnostic method that uses causal knowledge and linear programming in the application of Bayes' formula. *Comput. Methods Programs Biomed.* 22:223–237, 1986.

25. Sy, B. K., An adaptive reasoning approach towards efficient ordering of composite hypotheses, *Technical Report #91-001*, Queens College, CUNY, 1991.

26. Yaglom, A. M., and Yaglom, I. M., *Probability and Information*, D. Reidel Publishing Co., 1983.

27. Horvitz, E. J., Breese, J. S., Henrion, M., Decision theory in expert system and artificial intelligence. *Int. J. Approx. Reasoning.* 2:247–302, 1988.

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

29. Suermondt, H. J., Cooper G. F., A combination of exact algorithms for inference on Bayesian belief networks. *Int. J. Approx. Reasoning.* 5(6):521–542, 1991.