University of Castilla-La Mancha

A publication of the

Department of Computer Science

Abductive Inference in Bayesian Networks: a review.

by

José A. Gámez

Technical Report **#DIAB-03-11-1** July 2003

To appear in *Advances in Bayesian Networks*. José A. Gámez, Serafín Moral, Antonio Salmerón (Eds.). Springer. 2004.

DEPARTAMENTO DE INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE CASTILLA-LA MANCHA
Campus Universitario s/n
Albacete - 02071 - Spain
Phone +34.967.599200, Fax +34.967.599224

Abductive Inference in Bayesian Networks: A Review

José A. Gámez
Departamento de Informática.
Universidad de Castilla-La Mancha.
Campus Universitario s/n.
Albacete, 02071. Spain.
jgamez@info-ab.uclm.es

November 19, 2003

Abstract

The goal of this paper is to serve as a survey for the problem of abductive inference (or belief revision) in Bayesian networks. Thus, the problem is introduced in its two variants: total abduction (or MPE) and partial abduction (or MAP). Also, the problem is formulated in its general case, that is, looking for the K best explanations. Then, a (non exhaustive) review of exact and approximate algorithms for dealing with both abductive inference problems is carried out. Finally, we collect the main complexity results appeared in the literature for both problems (MPE and MAP).

Keywords: Abductive inference, belief revision, MPE, MAP, probabilistic reasoning, propagation algorithms, Bayesian networks.

1 Introduction

In the last years artificial intelligence researchers have devoted increasing attention to the development of abductive reasoning methods in a wide range of applications. Probably the most clear application of abductive reasoning is in the field of diagnosis [38, 39, 42, 43], although other applications exist in natural language understanding [6, 52], vision [27], legal reasoning [53], plan recognition [4, 26], planning [40] and learning [33].

Abduction is defined as the process of generating a plausible explanation for a given set of observations or facts [41]. This kind of reasoning can be represented by the following inference rule:

$$\frac{\psi \rightarrow \omega, \omega}{\psi},$$

i.e., if we observe ω and we have the rule $\psi \rightarrow \omega$, then we can infer that ψ is a *plausible* hypothesis (or explanation) for the occurrence of ω.

In general, there are several possible abductive hypotheses and it is necessary to choose among them. In order to select the best explanations from the generated set, two kinds of criteria are used: (1) metrics based criteria (probability, weight, ...) and (2) simplicity criteria (the preferred explanation is the simplest available hypothesis).

In the context of probabilistic reasoning, abductive inference corresponds to finding the maximum *a posteriori* probability state of the system variables, given some *evidence* (observed variables). In principle, we can solve this problem by "simply" generating the joint distribution, and then taking it as our starting point, to search for the configuration with maximum probability. However, this way to proceed is intractable even for problems with a small number of variables.

The paper is organized as follows: In the second section we introduce abductive inference in the framework of probabilistic reasoning but focused on Bayesian networks (BNs). In sections three and four we review algorithms for solving both cases of abductive inference problems by using exact and approximate methods. The fifth section is devoted to collect the main complexity results about abduction in BNs. Finally, in section six we conclude.

2 Abductive Inference in Probabilistic Reasoning

In this section we will describe the problem of performing abductive reasoning in the general setting of probabilistic reasoning, although we will assume that the joint probability distribution is represented by the factorization provided by a Bayesian network [37]):

$$P(X_{\mathcal{U}}) = P(X_1, \ldots, X_n) = \prod_{X_i \in X_{\mathcal{U}}} P(X_i | pa(X_i)), \tag{1}$$

where $pa(X_i)$ is the parent set of X_i.

Before we continue, we give some notation. A lower case subscript indicates a single variable (e.g., X_i). An upper case subscript indicates a set of variables (e.g., X_I). For some particular problems, the propositional variables are denoted by capital letters without subscript $A, B, C, \ldots$. The state taken by a variable X_i will be denoted by x_i, and the configuration of states taken by a set of variables X_D will be denoted by x_D. That is, capital letters are reserved for variables and set of variables, and lower case letters are reserved for states and configurations of states.

Given a set of observations[1] x_O for a set of variables X_O, the most common query

[1]The configuration $X_O = x_O$ is known as *evidence*. For the sake of simplicity we will use only x_O in most of the cases.

in probabilistic reasoning is the computation of $P(X_i|x_O)$ for every non observed variable $(X_{\mathcal{U}} \setminus X_O)$. This kind of reasoning is called *probability propagation* or *evidence propagation* or *belief updating*, and a great variety of exact and approximate algorithms have been developed during the last years (see [5] for a description of many of them). However, this paper is not concern with evidence propagation (which can be interpreted as predictive or deductive reasoning), but with abductive or diagnostic reasoning, which consists in looking for the *best* explanation accounting for the observed evidence.

In the context of probabilistic reasoning and Bayesian networks an explanation for a given set of observations $X_O = x_O$ is a configuration of states for the network variables, $x_{\mathcal{U}}$, such that, $x_{\mathcal{U}}$ is consistent with x_O, that is, $x_{\mathcal{U}}^{\downarrow X_O} = x_O$ (by $x_{\mathcal{U}}^{\downarrow X_O}$ we are denoting the configuration obtained from $x_{\mathcal{U}}$ by removing the literal expressions not in X_O). In fact, the explanation is $x_{\mathcal{U}}^{\downarrow X_{\mathcal{U}} \setminus X_O}$, because the values taken by the variables in X_O are previously known. Given the large number of possible explanations and since we are interested in the best explanation, our goal will be to obtain the *most probable explanation*. Thus, in our setting, abductive inference or *belief revision* [36, 37] corresponds to the problem of finding the maximum a posteriori probability state of the network, given the observed variables (the evidence). In a more formal way: if X_O is the set of observed variables and X_U is the set of unobserved variables, then we aim to obtain the configuration x_U^* of X_U such that:

$$x_U^* = \arg\max_{x_U} P(x_U|x_O), \tag{2}$$

where x_O is the observed evidence. Usually, x_U^* is known as the *most probable explanation* (MPE).

In general, this problem cannot be solved by using probability propagation. That is, x_U^* cannot be obtained as

$$x_U^* = (x_1^*, x_2^*, \ldots, x_{|U|}^*), \text{ with } x_i^* = \arg\max_{x_i \in \Omega_{X_i}} P(x_i|x_O). \tag{3}$$

As an example, let us consider the well-known Asia network [28]. If we assume that the observations are `(VisitToAsia=yes, PositiveXRay=yes)`, then figure 1 shows the result of performing different query types over this network by using the Elvira software [7]. In this software we can watch simultaneously the effect of processing different queries. In our example (see detail in figure 1) the first horizontal bar shows the *'a priori'* probability for each variable; the second bar shows the *'a posteriori'* probability for each variable given the observed evidence; and the third bar shows the state selected for each variable when looking

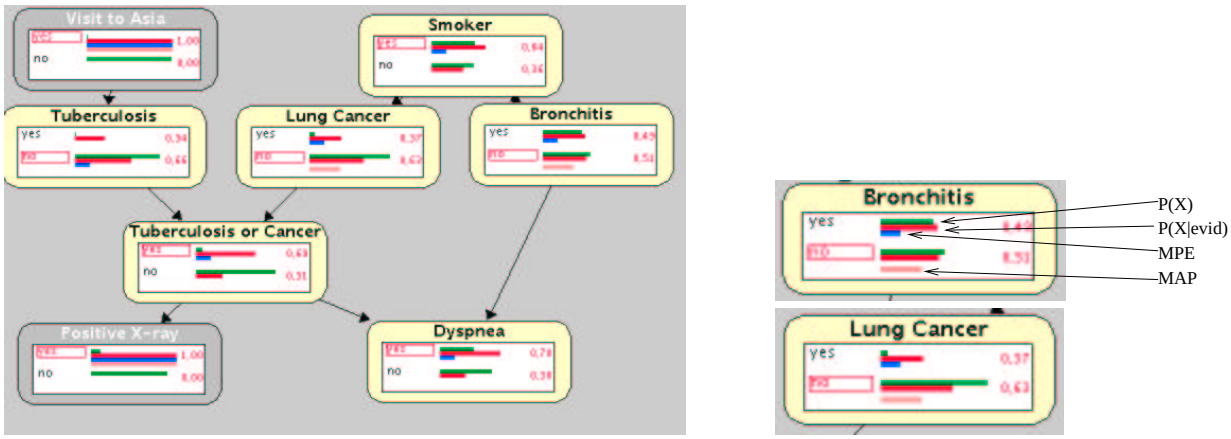
for the most probable explanation (eq. 2). As it can be seen, if we compose the MPE by using probability propagation, we obtain:

```
(Smoker=yes, Tuberculosis=no, LungCancer=no, Bronchitis=no,
TuberculosisOrCancer=yes, Dyspnea=yes),
```

with *unknown* probability. However, when total abductive inference is carried out, we obtain

```
(Smoker=yes, Tuberculosis=no, LungCancer=yes, Bronchitis=no,
TuberculosisOrCancer=yes, Dyspnea=yes),
```

with probability 0.17. As we can see the resulting configuration does not coincide with the previous one, because variable `LungCancer` takes now the state `yes`.



(a) (b)

Figure 1: (a) Different queries performed over Asia network with $X_O = (VisitToAsia = yes, PositiveXRay = yes)$. (b) Detail for nodes *Bronchitis* and *LungCancer*

The previous definition of abductive inference can be generalized by considering a subset of the unobserved variables as the interest target, instead of all of them. These variables are usually termed *explanation set* [30], and the task is known as *Partial* abductive inference (or *maximum a posteriori hypothesis*, MAP) while the previous one which is known as *Total* abductive inference (or MPE). Although this problem seems to be more useful in practical applications (because we can select the relevant variables[2] as the explanation set) than *total* abductive inference it has received much less attention.

[2]those representing diseases in a medical diagnosis problem, those representing critical components (starter, battery, alternator, ...) in a car diagnosis problem, etc ...

$$\psi(A) = \max_B \psi(A,B) = \max_B \begin{pmatrix} & b & \neg b \\ a & 0.35 & 0.2 \\ \neg a & 0.15 & 0.3 \end{pmatrix} = \begin{pmatrix} a & 0.35 \\ \neg a & 0.3 \end{pmatrix}$$

Figure 2: Marginalizing by maximum (max-marginalization)

Now, if we denote by $X_E \subset X_U$ the explanation set, then we aim to obtain the configuration x_E^* of X_E such that:

$$x_E^* = \arg\max_{x_E} P(x_E|x_O) = \arg\max_{x_E} \sum_{x_R} P(x_E, x_R|x_O), \tag{4}$$

where $X_R = X_U \setminus X_E$. In general, x_E^* is not equal to the projection of configuration x_U^* onto the variables of X_E. Therefore, we need to obtain x_E^* directly (eq. 4). As an example, let us to retake the example based on figure 1. Now, if we select variables `LungCancer` and `Bronchitis` as the explanation set, then by using partial abductive inference we get the configuration:

```
(LungCancer=no, Bronchitis=no)
```

as the most probable explanation (see detail in part (b) of the same figure). As we can see, this configuration does not coincide with the one obtained by projecting x_U^* over X_E (`LungCancer=yes, Bronchitis=no`).

To finish this section, just remark that in both queries of abductive inference, the problem is generalized to the one of looking for the K most probable explanations (K MPEs).

3 Solving total abduction (MPE) in BNs

As it has been shown in the literature, in total abduction, the MPE can be found by means of probability propagation methods but using maximum as the marginalization operator (fig. 2) instead of summation (due to the distributive property of maximum with respect to multiplication) [12].

Therefore, the process of searching for the most probable explanation (in total abduction) has the same complexity[3] as probabilities propagation. However, in order to look for the K MPEs more work has, in general, to be done. In the next subsections we review some approaches for solving this problem in an exact and approximate way.

[3]See Section 5 for details in this topic.

3.1 Exact computation

We review the two main approaches which can deal with networks of unconstrained topology.

3.1.1 Using junction trees.

Dawid [12] developed an efficient algorithm to calculate the most probable explanation (MPE) in a junction tree. The algorithm is based on the probabilities propagation algorithm described in [3], but replacing summation (sum-marginalization) by maximum (max-marginalization) in the calculation of messages. When *max-marginalization* is used during the two stages of messages propagation (collect and distribute evidence) the process was termed *max-propagation* by Dawid.

Given an initialized junction tree $\mathcal{T}$ with cliques $\{C_0, \ldots, C_t\}$, after max-propagation the following expression holds:

$$\forall C_i \in \mathcal{T} \quad \max_{x_U \in \Omega_{\mathcal{U}}} P(x_U, x_O) = \max_{c_i \in \Omega_{C_i}} \psi(c_i) \tag{5}$$

Therefore, if there is only one configuration of maximal probability, then we can identify the MPE x_U^* by inspecting the residual set R_i of each clique C_i and picking up the state of maximal probability for the variables in R_i. However, if there are several configurations of maximal probability, then the following algorithm can be used in order to identify one of them.

FindMPE

- input: a rooted junction tree $\mathcal{T} = \{C_0, \ldots, C_t\}$ after applying max-prop.
- output: the MPE $x^* = concatenate(c_0^*, r_1^*, \ldots, r_t^*)$

1. For the root C_0: $c_0^* = \arg\max_{C_0} \psi(C_0)$

2. For $j = 1, \ldots, t$ do

$$\begin{aligned} C_i &= parent(C_j) \\ s_{ij}^* &= c_i^{*\downarrow S_{ij}} \\ r_j^* &= \arg\max_{C_j \setminus S_{ij}} \psi(C_j, s_{ij}^*) \end{aligned}$$

As an example let us consider the figure 3. On the left we have only a configuration of maximal in C_0 and so we can identify it (a, b, c) by inspection. However, in the junction tree on the right there are more than one configuration of maximal probability (highligthed in

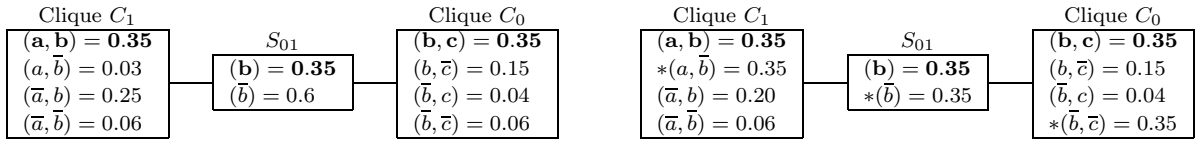


Figure 3: Identifying the MPE in two different scenarios

bold and with an (*) respectively). Thus, we have to apply the previous algorithm in order to identify one of them: (a, b, c) or $(a, \overline{b}, \overline{c})$.

Looking for the K MPEs. Looking for the K most probable explanations can be viewed as an extension of the MPE problem. Nilsson [31], has proved that using Dawid's algorithm only the three most probable explanations can directly be identified, but the fourth and subsequent explanations cannot be found directly. Therefore, more complex methods have to be used.

In [47] Seroussi and Goldmard developed a method able to find the K MPEs for every value of K. The method only requires the upward phase of max-propagation (that is, collecting evidence from the root) but modifying the messages that are sent from each clique to its parent. Now, instead of sending a single message/potential from C_i, a vector of K messages is sent, corresponding to the K most probable configurations of the subtree rooted by C_i. The main problem of this algorithm is its high computational cost.

Later, Nilsson [32] has developed a more efficient method for finding the K MPEs for every value of K. As in Seroussi and Goldmard proposal, only the upward phase of max-propagation is necessary. The algorithm for obtaining the K MPEs is based on the combination of Dawid's algorithm with a clever divide and conquer approach. The idea is as follows: after obtaining the best MPE (x_U^1), the space of possible configurations is partitioned in order to exclude it from the set of possible configurations. As the second MPE (x_U^2) must differ from the best one in at least one residual set, the search space is partitioned into t subspaces by using the value of the residual sets as evidence. The tricky point of Nilsson's algorithm is that no new propagation phases are required in order to consider these new *evidences*, but by inspecting the potentials in the junction tree. After identifying x_U^2 a new partitioning is carried out, now the clique (whose residual changes with respect to x_U^1) is considered as the root for the partition, and so only the cliques below it in the junction tree will be considered. For a complete (and formal) description of the model see [32].

3.1.2 Using variable elimination

Variable elimination [48, 57] can be directly adapted for dealing with the problem of looking for the best MPE. The main modification is to use max-marginalization instead of sum-marginalization when a variable is being eliminated. Also, we have to store the state of the variable being eliminated which takes the maximum. Finally, the configuration of maximum probability is obtained by assembling the stored states [29, 21]. In order to look for the K MPEs more work has to be done. Li and D'ambrosio [29] propose to use the standard variable elimination algorithm to look for the best MPE, and then they give a linear time algorithm which can be invoked (as many times as necessary) in order to get the next MPE.

3.2 Approximate computation

In this subsection we distinguish between search and non search based methods:

3.2.1 Search methods

As abductive inference in BNs can be viewed as a combinatorial optimization problem, several authors have used genetic algorithms to approximate a solution ([24, 44, 45, 58]). In all the cases the idea is to optimize the probability

$$P(x_U|x_O) \propto P(x_U, x_O),$$

so to evaluate the goodness of a given individual x_U the factorization provided by the network is used (eq. 1), requiring only n multiplications (that is, linear in the number of variables).

Below, we describe some relevant points of these algorithms.

- In Gelsema's algorithm [24], a chromosome is a configuration of the unobserved variables, i.e., a vector of integers. In this case, crossover is implemented as the classical one-point operator. It is worth noting that Gelsema uses the '*a priori*' probabilities of the BN and the observed evidence to generate the initial population, so that the search starts in promising regions of the search space.

- In the approach proposed by Rojas and Kramer [44, 45], a chromosome of the population is represented as a copy of the graph included in the BN, but in which each variable has been instantiated to one of its possible states. This representation makes it possible to implement the crossover operator as the interchange of a subgraph with the center in the variable X_i, X_i being randomly selected for each crossover.

- Finally, Zhong and Santos [58] propose to group the variables in the network into several clusters attending to the probabilistic correlations among the variables, and to use cluster-based mutation and crossover. The cluster are identified by using reinforcement learning, and the output of the genetic algorithm is used to refine the clustering process in order to get a better grouping.

Recently, Kask and Dechter [25] have studied the applicability of local search to this problem. They compared several stochastic local search algorithms for solving the MPE problem. In their analysis, they found that the greedy approach combined with stochastic simulation outperforms the other analyzed techniques: stochastic simulation (alone), greedy local search (alone) and simulated annealing.

The problem of looking for the K MPEs is solved by these algorithms by returning the K best individuals/chromosomes visited during the search.

3.2.2 Non search methods

Non search methods for dealing with the MPE problem are based on the methods discussed in section 3.1:

- The Elvira software [7] allows to solve the MPE problem by using junction tree based approximate propagation. The idea is to use Nilsson's algorithm [32] but dealing with an approximate representation of the potentials associated to cliques and messages. Concretely, approximate probability trees are used instead of probability tables/trees (see [46] for a description on (approximate) probability trees).

- A method based on variable elimination is the so-called mini-bucket approach [22]. The idea of the mini-bucket approach is to move some marginalizations outside of the product, in order to deal with smaller potentials. Of course, the result of this transformation will be, in general, an approximation of the true value. For example, in variable (bucket) elimination we could have the following scenario when eliminating variable X:

$$\max_X \left(f(X, Y_1, Y_2) \times f(X, Y_2, Y_3) \times f(X, Y_4, Y_5) \times f(X, Y_5, Y_6)\right)$$

  If each variable has ten different states, then this operation has to build a potential of size 10^7. By using the mini-bucket approach, the previous computation could be

organized in the following way:

$$\left\{\max_X \left(f(X,Y_1,Y_2)\times f(X,Y_2,Y_3)\right), \max_X \left(f(X,Y_4,Y_5)\times f(X,Y_5,Y_6)\right)\right\}$$

which builds two potentials of size 10^4. An extra advantage of the mini-buckets approach is that it gives bounds on the quality of the approximate solution.

4 Solving partial abduction (MAP) in BNs

As was stated in Sec. 2 when we are interested in the MPE for a given subset (X_E) of the network's variables, then equation 4 has to be used. Therefore, two types of marginalization operators have to be applied: max-marginalization over the variables in the X_E and sum-marginalization over the rest of variables.

In the next two subsections we describe exact and approximate methods proposed to cope with the problem of partial abductive inference (MAP). In the case of exact computation we focus on junction tree based algorithms, although some remarks (and references) to variable elimination will be provided.

4.1 Exact computation

Given the algorithms described for total abduction in BNs and the need of using two types of marginalization, it seems easy to solve the problem of partial abductive inference, we can use the following two-stage based algorithm:

1. Marginalize out (by summation) over the variables not in the explanation set. This process will yield a junction tree containing only the variables in the explanation set (X_E).

2. Apply an algorithm of *total* abductive inference.

However, the problem is more complex than it looks like, because due to the non-commutative behaviour of summation and maximum the following constraint has to be considered: *no summation can be carried out over a potential obtained by maximum.* This fact motivates that not all junction trees obtained from the original network are valid. Thus, the previous two-stage algorithm can be only directly applied when X_E is included in a node of the junction tree, or when the variables of X_E constitute a subtree of the complete junction tree. As an example, let us consider the junction tree in figure 4.(a) and the explanation

set $X_E = \{A, T, B\}$. Clearly this tree is not valid for the given explanation set, because in clique C_2 we have to sum-marginalize variables $\{E, L\}$ from the potential $\psi(L, E, T)$ whose computation has involved max-marginalizing over B.

Therefore, we have modified the problem to the following one: what happen when the variables of the explanation set are associated with several disconnected subtrees of the junction tree?. Below, we describe several approaches which have been proposed in the literature in the last years.

4.1.1 Adapting a given junction tree.

With the proposal of computing marginal values over a set of variables, Xu [55] gives a method for transforming the initial junction tree into another one containing a node/cluster in which the variables of X_E are included. The problem of this approach is that if X_E contains many variables, then the size of the potential associated with that cluster will be to large. Later, Nilsson [32] briefly outlines how to slightly modify Xu's algorithm in order to allow (when possible) that the variables in X_E constitute a sub-tree and not a single cluster. More recently, de Campos et al. [18] have detailed this process, studying heuristics and introducing and intermediate step. In the rest of this section we briefly describe this work.

The goal of obtaining a junction tree $\mathcal{T}_E$ containing only the variables in X_E, from a given junction tree $\mathcal{T}$ can be performed as follows:

1. Identify the smallest subtree $\mathcal{T}'$ of $\mathcal{T}$ that contains the variables of X_E.

2. Pass sum-flows from the rest of the cliques to $\mathcal{T}'$. In this way $\mathcal{T}'$ factorizes over the variables contained in its cliques.

3. While $\mathcal{T}'$ contains variables not in X_E do: Select two neighbors C_i and C_j in $\mathcal{T}'$, and replace them by their fusion into a new node C_{ij} obtained from $C^*_{ij} = C_i \cup C_j$ by deleting the variables that are not necessary to maintain the running intersection property and do not belong to X_E. The potential assigned to C_{ij} is[4]:

$$\psi(C_{ij}) = \sum_{C^*_{ij} \setminus C_{ij}} \frac{\psi(C_i) \cdot \psi(C_j)}{\psi(S_{ij})} \tag{6}$$

Notice that C^*_{ij} is relevant because though we produce a clique C_{ij}, during the process we have to deal with the potential defined on the whole set C^*_{ij}. Figure 4.(b) shows $\mathcal{T}'$ (wrt

[4]The division can be omitted if we know that no propagation has been previously carried out over the junction tree, because in such a case S_{ij} contains an unitary potential

$X_E = \{A, T, B\}$) obtained from $\mathcal{T}$ in part (a) of the same figure. Finally part (c) of the same figure shows an example for the Bayesian network Asia taking $X_E = \{A, T, B\}$. Notice that the potential over $\{B, T, E, L\}$ is built during the fusion process.

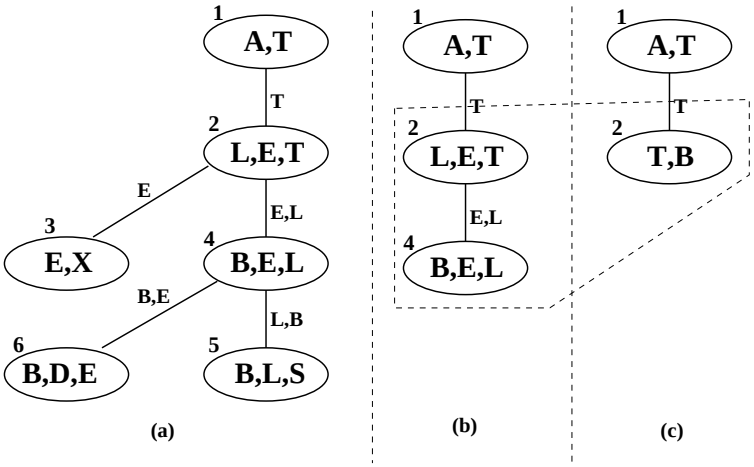


Figure 4: (a) The initial join tree. (b) $\mathcal{T}'$ for $X_E = \{A, T, B\}$. (c) The junction tree obtained by the fusion process.

The following improvements were proposed for this algorithm in [18]:

- Introducing an intermediate step. An optimization can be added to the previous algorithm between steps 2 and 3. The idea is to look for the variables not in X_E that are only included in one clique of $\mathcal{T}'$, then it is clear that these variables can be marginalized out directly. This idea, although it is very simple can improve the efficiency of the fusion process carried out in step 3 and also the quality of the final tree (see [18] for details and examples).

- Defining heuristics for step 3. In [18] the fusion process performed in step 3 is formally defined as a *link* deletion process. It is important to emphasize that unlike the case of triangulation (deletion of nodes) in this case not all the links have to be removed. Therefore the first think to do is to identify which are the links candidates for deletion.

  Definition 1 *[18] Given a junction tree $\mathcal{T}'$ and an explanation set X_E, a link (C_i, C_j) with separator $S_{i,j}$ has to be removed if:*

  i) $S_{i,j} \not\subseteq X_E$, *or*
  ii) $S_{i,j} \subseteq X_E$, *but* $S_{i,j} = C_i$ *or* $S_{i,j} = C_j$.

The second condition of the previous definition is necessary to deal with non-maximal nodes, which can be introduced by the intermediate step presented in the previous subsection. To finish with this section, we reproduce here the heuristics which obtained better results in [18] when applied to select the next link to be removed:

– Select the link (C_i, C_j) with yields smallest $s(C_{ij})$.

The goal of this heuristics is the creation of nodes with smallest state space, with the expectation of an earlier deletion of the variables not in X_E.

4.1.2 Looking for an explanation set oriented junction tree.

A disadvantage of the previous method is that the quality of the final junction tree, depends on the topology of the initial one, which was obtained for general inference purposes and not thinking about partial abductive inference. An alternative approach could be to search for a specific junction tree given an explanation set X_E [23, 18]. This task can be achieved by taking advantage of the available degrees of freedom in the compilation/triangulation process. Concretely, during the triangulation we can constrain the deletion sequence, in such a way that we start to delete the variables of X_E only when all the variables not in X_E have been deleted.

Using this kind of deletion sequences, and adding all the clusters (and not only those which are maximal) to the tree, we can build a junction tree in which a subtree for X_E can be directly identified. If only cliques[5] are added to the tree, it is necessary to apply *maximum cardinality search* (see [28]) beginning with a variable of X_E and breaking ties in favor of the variables in X_E during the numbering of the graph's nodes. Figure 5.a shows a junction tree valid for the Asia network and $X_E = \{A, B, T\}$.

Notice that now $\mathcal{T}'$ practically coincides with $\mathcal{T}_E$. In fact the fusion process (step 3) is not necessary because the variables in $\mathcal{T}'$ and not in X_E are in the leaves and will be deleted (by summation) by the intermediate step described above (this is the case in the tree depicted on part (b) of figure 5, where L and E can be directly removed.

The experiments carried out in [18] show that, as expected, the size of the join tree over which partial abductive inference will be performed is smaller when using constrained

[5]Sometimes it is no possible to obtain a valid clique tree [23], but the structure appears valid after applying the intermediate step proposed above.

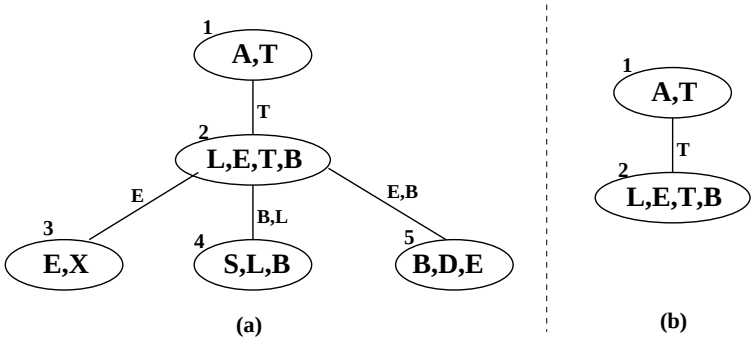


Figure 5: (a) Specific junction tree for Asia and $X_E = \{A, B, T\}$.(b) Identified $\mathcal{T}'$

deletion sequences than when using adapted junction trees. However, the authors also get an unexpected result: in general, the abductive process runs faster over adapted junction trees than over specific ones. De Campos et al. analyze this problem in [20] and conclude that this is due to the fact that when using specific junction trees the abductive inference method deals with very large potentials since the beginning of the process. However, as the initial information, that is the conditional probability tables in the network, is the same in both cases, then there should be lots of regularities in those large potentials. In fact, when using probability trees instead of tables to represent potentials, as regularities and contextual independences are exploited, the efficiency of the method is increased no matter which kind of junction tree (adapted or specific) is used. However, the greater benefit is obtained when using specific ones, to the point that, in general, (as expected initially) the algorithm runs faster over specific junction trees than over adapted ones.

With respect to variable/bucket elimination, Dechter [21] proposed the algorithm *elim-map* which basically uses a constrained deletion sequence. Firstly, the variables not in the explanation set are eliminated by summation, and then the variables in X_E are eliminated by maximum, storing in each step the state of maximum probability of the variable being eliminated (as in *elim-mpe*). As well as in total abduction, the algorithm of Li and D'Ambrosio [29] could be used to look for the remaining $K - 1$ explanations.

4.2 Approximate computation

As in the total case, in this subsection we distinguish between search and non search based methods:

4.2.1 Search methods

Of course, partial abductive inference can also be defined as a combinatorial optimization problem, being now Ω_{X_E} instead of Ω_{X_U} the search space. Thus, several combinatorial optimization techniques have been used:

- Evolutionary algorithms. As in the total case, genetic algorithms have been used to approach this problem [14, 19, 15]. Now, eq. 1 cannot be used to evaluate an individual or candidate configuration because we have to remove (by addition) the variables in $X_U \setminus X_E$. That is, if we use eq. 1 we have to invoke it $|\Omega_{X_U \setminus X_E}|$ times, which is impracticable in most of the cases.

  In [14] the fitness $P(x_E, x_O)$ of a configuration x_E is computed by the process described below, where $\mathcal{T}$ is a rooted clique tree, being C_0 the root.

  1. Enter the evidence x_O in $\mathcal{T}$,
  2. Enter (as evidence) the configuration x_E in $\mathcal{T}$,
  3. Perform *CollectEvidence* from the root (C_0) (i.e., an upward propagation), and
  4. $P(x_E, x_O)$ is equal to the sum of the potential stored in the root (C_0).

  Therefore, to evaluate a configuration an exact sum-propagation is carried out, or more correctly half propagation, because only the *upward* phase is performed and not the *downward* one. Furthermore, for this propagation we can use a clique tree obtained without constraints and so its size is much smaller than the clique tree used for exact partial abductive inference. In addition, in [14] it is shown how the tree can be pruned in order to avoid the repetition of unnecessary computations when a new chromosome is being evaluated.

  An improvement to this algorithm can be found in [19], consisting in the use of specific genetic operators that allow us to take advantage of the calculations previously carried out when a new individual is being evaluated. In this way the need to perform a whole upward propagation is avoided, although more memory is needed.

  In [13] *estimation of distribution algorithms* (EDAs) are used to compare with this two genetic approaches. EDAs are evolutionary algorithms in which genetic has been

replaced by estimation and sampling of probability distribution. The same evaluation function is used and the results are really competitive, depending on the type of EDA.

In order to reduce the complexity of evaluating a configuration, in [15] the junction tree is pre-processed by means of explanation set absorption. That is, variables in X_E are considered as evidence, but all its possible states are taken into account. In this way, some cliques store vector of potentials instead of single potentials, and the correct one, which depends on the chromosome being evaluated, is selected in execution time. The advantage of this approach is to deal with smaller junction trees, and the authors show a considerable speed-up in the efficiency of the algorithm. However, the main disadvantage is the need of (much) more memory when many variables are included in the explanation set.

- Simulated annealing. In De Campos et al. [16] a simulated annealing based algorithm is proposed. As in evolutionary algorithms, the evaluation function is based on clique tree propagation, but now neighbor configurations can be evaluated by means of local computations, improving the efficiency with respect to global evaluation. The algorithm maintain a state composed of the current configuration and the current clique C_i, then the neighborhood of that configuration is defined as the set of configurations in which only the variables included in the residual sets of adjacent cliques to C_i can change their value. This assumption allows to organize computations in such a way that to evaluate a neighbor configuration only the current clique and its neighbors in $\mathcal{T}$ are involved. After selecting the new configuration, the algorithm also change its current clique to the neighbor containing (in its residual set) the variable(s) who has change its value (wrt the previous configuration).

- Local search. Park and Darwiche [35] propose to use local search to solve partial abductive inference (or MAP). The algorithm is based in a hill climbing (with restart) strategy in which we move from a configuration x_E to one of its neighbors only if $P(x'_E, x_O) > P(x_E, x_O)$. A neighbor of x_E is defined as the result of changing the value of a single variable, so there are $|X_E|$ possible neighbors. The novelty of this method resides in the way in which configurations are evaluated. The authors propose to use a method based on *differential inference* [11], which allow them to compute the value of all the neighbors in $O(n \exp(w))$, where n is the number of variables and w the width

of an (unconstrained) elimination order.

Later, Park [34] extend the approach to larger networks by replacing the evaluation method. In this work, the score for a given configuration is computed by using a method for approximating the probability of evidence based on *loopy belief propagation* [54, 56].

4.2.2 Non search methods

As wells as in the MPE problem, the mini-bucket approach [22] and the use of approximate probability trees [20] have been used as tools for obtaining approximate solutions to the partial abductive inference problem.

5 Complexity results

It is well known that exact probabilistic inference can be performed efficiently in networks of restricted topology (trees and polytrees), but in the general case (networks with cycles) propagation is NP-hard [8]. Also, approximate inference in BNs is NP-hard too [9], although the class of solvable problems is wider. Therefore, as abductive inference in BNs is solved by propagation algorithms it is expected to be NP-hard as well. Below we revised some complexity results about abductive inference in BNs.

Looking for the MPE in BNs can be solved efficiently in the same cases as probability propagation, but Shimony [50] proves that in the general case the MPE problem is NP-hard. Besides, with minor modifications to this result, Shimony shows that the problem remains NP-hard even when the topology of the network is restricted to an in-degree (number of parents) of 2 or to an out-degree (number of children) of 2.

With respect to the use of approximations, Abdelbar and Hedetniemi [1] have shown that approximating the MPE problem with a constant ratio bound is also NP-hard. In the same paper the authors also prove that given an explanation, the process of approximating the next one with a constant ratio bound is NP-hard too. Abdelbar and Hedetniemi [1] also gave an additional and interesting result with respect to what they termed *dynamic* abduction: given the MPE (or K MPEs) for a BN and an initial evidence x_O, the problem of finding or approximating the MPE (K MPEs) for a modified evidence $x_O{}'$ which differs form x_O by the addition or removal of even a single pair (variable,value) is also NP-hard. Finally, as in [50] they extend their results to the case of networks with restricted topology (in-degree=2 or out-degree=2). Later, Abdelbar et al. [2] show that on the contrary to probability propagation

[10], finding or approximating the MPE in BNs remains NP-hard even for networks in which probabilities are bounded within the range $[l, u]$ for any $0 \leq l < 0.5 < u \leq 1$.

With respect to partial abduction or MAP, it seems that solving this problem is harder than solving related inference problems as probability propagation or total abductive inference [23, 16, 34], being really complex even for networks in which the other tasks can be solved efficiently. With respect to the *hardness* of MAP, Park [34] shows that MAP is $\mathrm{NP}^{\mathrm{PP}}$-complete (MPE is NP-complete and probability propagation is PP-complete in its decision version). Besides, Park shows that elimination algorithms require exponential resources to perform MAP, even on some polytrees; and that MAP is NP-complete when restricted to polytrees.

Related with MAP, Gámez [23] gives a result which can be used to obtain a lower bound for clique size (or elimination order width) for any given problem. In fact, given a network G and an explanation set X_E, Gámez shows that for all constrained elimination sequence (variables of X_E are eliminated last), and for all variable X_i not in X_E, there is at least one clique containing $X_i \cup acc_E(X_i)$, where

$$\begin{aligned} acc_E(X_i) \;\; = \;\; & \{X_j \in X_E \mid \exists \text{ an undirected path } (X_i, X_{ij}^1, \ldots, X_{ij}^p, X_j) \text{ in } G, \\ & \text{such that}, \{X_{ij}^1, \ldots, X_{ij}^p\} \subset X_{\mathcal{U}} \setminus X_E\}. \end{aligned}$$

Therefore, the lower bound for any constrained elimination order for that problem will be:

$$\max_{X_i \text{ not in } X_E} |\{X_i\} \cup acc_E(X_i)|$$

As an example of this result, let us consider the classical Alarm network and all the root nodes (which is very usual) as the explanation set. In this case, there are several variables (not in X_E) such that $acc_E(\cdot) = X_E$, and so there will be at least one clique containing (among others) all the variables in the explanation set.

6 Conclusions

In this paper we have revised the problems of performing total and partial abductive inference in Bayesian networks. A review of exact and approximate algorithm for dealing with both problems and their extension to looking for the K best explanations has been carried out. Of course, we are aware of the fact that not all algorithms have been discussed by our revision, but due to the lack of space we have selected the approaches with wider application (networks of unconstrained topology) and of more recent publication.

There is a very interesting problem related with partial abduction or MAP which has not been covered in this paper. We refer to the problem of selecting the explanation set. In the literature, it is mostly assumed that the explanation set is provided *a priori* by an expert, another algorithm, etc, ... It is an extended practice to select root nodes (which usually represents diseases or disorders) or the evidence ancestors as the explanation set [29]. Shimony [49, 51] proposed a method based on irrelevance criteria that does not require a given explanation set, on the contrary the algorithm tries to identify the relevant nodes directly. De Campos et al. [17] propose to start with an explanation set, but to simplify the obtained explanations by using independence-based criteria. However, in our opinion this problem should receive more attention from the BNs research community.

Acknowledgments

This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnología (MCyT), under project TIC2001-2973-C05-05. The author is grateful to Julia Flores for her comments and suggestions on previous versions of this work.

References

[1] A.M. Abdelbar and S.M. Hedetniemi. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38, 1998.

[2] A.M. Abdelbar, S.T. Hedetniemi, and S.M. Hedetniemi. The complexity of approximating MAPs for belief networks with bounded probabilities. *Artificial Intelligence*, 124:283–288, 2000.

[3] S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. Hugin: a shell for buiding belief universes for expert systems. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, 1989.

[4] D.E. Appelt and M. Pollack. Weighted abduction for plan ascription. Technical report, Artificial Intelligence Center and Center for the Study of Language and Information, SRI International, Menlo Park, California, 1990.

[5] E. Castillo, J.M. Gutiérrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models*. Monographs in Computer Science. Springer-Verlag, New York, 1997.

[6] E. Charniak and E. McDermott. *Introduction to Artificial Intelligence*. Addison-Wesley, 1985.

[7] Elvira Consortium. Elvira: An environment for creating and using probabilistic graphical models. In *Proceedings of the First European Workshop on Probabilistic Grpahical Models*, pages 220–230, 2002.

[8] G.F. Cooper. Probabilistic inference using belief networks is NP-hard. *Artificial Intelligence*, pages 393–405, 1990.

[9] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[10] P. Dagum and M. Luby. An optimal approximation algorithm for bayesian inference. *Artificial Intelligence*, 93:1–27, 1997.

[11] A. Darwiche. A differential approach to inference in bayesian networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 123–132, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[12] A.P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36, 1992.

[13] L.M. de Campos, J.A. Gámez, P. Larrañaga, S. Moral, and T. Romero. *Estimation of Distribution Algorithms. A new tool for evolutionary computation*, chapter Partial Abductive Inference in Bayesian Networks: An empirical comparison between Gas and EDAs. Kluwer Academic Publishers, 2001.

[14] L.M. de Campos, J.A. Gámez, and S. Moral. Partial Abductive Inference in Bayesian Belief Networks using a Genetic Algorithm. *Pattern Recognition Letters*, 20(11-13):1211–1217, 1999.

[15] L.M. de Campos, J.A. Gámez, and S. Moral. Accelerating chromosome evaluation for partial abductive inference in bayesian networks by means of explanation set absorption. *International Journal of Approximate Reasoning*, (27):121–142, 2001.

[16] L.M. de Campos, J.A. Gámez, and S. Moral. Partial Abductive Inference in Bayesian Belief Networks by using Simulated Annealing. *International Journal of Approximate Reasoning*, (27):263–283, 2001.

[17] L.M. de Campos, J.A. Gámez, and S. Moral. Simplifying explanations in bayesian belief networks. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, (9):461–489, 2001.

[18] L.M. de Campos, J.A. Gámez, and S. Moral. On the problem of performing exact partial abductive inference in Bayesian belief networks using junction trees. In B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, and R.R. Yager, editors, *Technologies for Constructing Intelligent Systems 2: Tools*, pages 289–302, 2002.

[19] L.M. de Campos, J.A. Gámez, and S. Moral. Partial Abductive Inference in Bayesian Belief Networks: An Evolutionary Computation Approach by Using Problem Specific Genetic Operators. *IEEE Transaction on Evolutionary Computation*, (6):105–131, 2002.

[20] L.M. de Campos, J.A. Gámez, and S. Moral. Partial abductive inference in bayesian networks by using probability trees. In *Proceedings of the 5th International Conference On Enterprise Information Systems (ICEIS'03). Vol 2.*, pages 83–91, Angers, 2003.

[21] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 211–219, Portland, Oregon, 1996.

[22] R. Decther and I. Rish. A scheme for approximating probabilistic inference. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 132–141, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[23] J.A. Gámez. *Abductive inference in Bayesian networks (*in Spanish*)*. PhD thesis, Departamento de Ciencias de la Computación e I.A. Escuela Técnica Superior de Ingeniería Informática. Universidad de Granada, 1998.

[24] E.S. Gelsema. Abductive reasoning in Bayesian belief networks using a genetic algorithm. *Pattern Recognition Letters*, 16:865–871, 1995.

[25] K. Kask and R. Dechter. Stochastic local search for Bayesian networks. In *Seventh International Workshop on Artificial Intelligence*, pages 469–480. Morgan Kaufmann Publishers, 1999.

[26] H. Kautz and J. Allen. Generalized plan recognition. In *Proc. of National Conference on Artificial Intelligence*, pages 32–37. AAAI, 1986.

[27] U.P. Kumar and U.B. Desai. Image interpretation using Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):74–78, 1996.

[28] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J.R. Statistics Society. Serie B*, 50(2):157–224, 1988.

[29] Z. Li and D'Ambrosio B. An efficient approach for finding the MPE in belief networks. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 342–349. Morgan and Kaufmann, San Mateo, 1993.

[30] R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems. Theory and Algorithms.* Wiley Interscience, New York, 1990.

[31] D. Nilsson. An algorithm for finding the M most probable configurations of discrete variables that are specified in probabilistic expert systems. *MSc. Thesis*, University of Copenhagen, 1994.

[32] D. Nilsson. An efficient algorithm for finding the M most probable configurations in Bayesian networks. *Statistics and Computing*, 9:159–173, 1998.

[33] P. O'Rorke, S. Morris, and D. Schulenberg. Theory formation by abduction: initial results of a case study based on the chemical revolution. Technical Report ICS-TR-89-25, University of California, Irvine, Department of Information and Computer Science, 1989.

[34] J.D. Park. MAP complexity results and approximation methods. In *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference (UAI-2002)*, pages 388–396, San Francisco, CA, 2002. Morgan Kaufmann Publishers.

[35] J.D. Park and A. Darwiche. Approximating MAP using local search. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 403–410, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[36] J. Pearl. Distributed revision of composite beliefs. *Artificial Intelligence*, 33:173–215, 1987.

[37] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, 1988.

[38] Y. Peng and J.A. Reggia. A probabilistic causal model for diagnostic problem solving. Part One. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(2):146–162, 1987.

[39] Y. Peng and J.A. Reggia. A probabilistic causal model for diagnostic problem solving. Part Two. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):395–406, 1987.

[40] D. Poole and K. Kanazawa. A decision-theoretic abductive basis for planning. In *Proc. of AAAI Spring Symposium on Decision-Theoretic Planning*, pages 232–239. Stanford University, March 1994.

[41] H.E. Pople. On the mechanization of abductive logic. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 147–152, 1973.

[42] J.A. Reggia. Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, 19:437–460, 1983.

[43] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32, 1987.

[44] C. Rojas-Guzman and M.A. Kramer. Galgo: A genetic algorithm decision support tool for complex uncertain systems modeled with Bayesian belief networks. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 368–375. Morgan and Kauffman, San Mateo, 1993.

[45] C. Rojas-Guzman and M.A. Kramer. An evolutionary computing approach to probabilistic reasoning in Bayesian networks. *Evolutionary Computation*, 4:57–85, 1996.

[46] A. Salmerón, A. Cano and S. Moral. Importance sampling in bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.

[47] B. Seroussi and J.L. Goldmard. An algorithm directly finding the K most probable configurations in Bayesian networks. *International Journal of Approximate Reasoning*, 11:205–233, 1994.

[48] R.D. Shachter, B.D. D'Ambrosio, and B.D. Del Favero. Symbolic probabilistic inference in belief networks. In *8th National Conference on Artificial Intelligence*, pages 126–131, Boston, 1990. MIT Press.

[49] S.E. Shimony. The role of relevance in explanation I: Irrelevance as statistical independence. *International Journal of Approximate Reasoning*, 8:281–324, 1993.

[50] S.E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68:399–410, 1994.

[51] S.E. Shimony. The role of relevance in explanation II: Disjunctive assignments and approximate independence. *International Journal of Approximate Reasoning*, 13:27–60, 1995.

[52] M.E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural language interpretation. Technical Report 451, AI Center, SRI International, 1988.

[53] P. Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12:435–467, 1989.

[54] Y. Weiss. *Approximate inference using belief propagation.* UAI tutorial, 2001.

[55] H. Xu. Computing marginals for arbitrary subsets from marginal representation in markov trees. *Artificial Intelligence*, 74:177–189, 1995.

[56] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS, volume 13*, 2000.

[57] N.L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.

[58] X. Zhong and E. Santos Jr. Directing genetic algorithms for probabilistic reasoning through reinforcement learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8:167–186, 2000.