In *Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information,* pages 77-82, Stanford University, Stanford, California, March 22-24, 1999

Stochastic Sampling and Search in Belief Updating Algorithms for Very Large Bayesian Networks

Yan Lin & Marek J. Druzdzel
Decision Systems Laboratory
School of Information Sciences
and Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
{yan,marek}@sis.pitt.edu

Abstract

Bayesian networks are gaining an increasing popularity as a modeling tool for complex problems involving reasoning under uncertainty. Since belief updating in very large Bayesian networks cannot be effectively addressed by exact methods, approximate inference schemes may be often the only computationally feasible alternative. There are two basic classes of approximate schemes: stochastic sampling and search-based algorithms. We summarize the basic ideas underlying each of the classes, show how they are inter-related, discuss briefly their advantages and disadvantages, and show examples on which each of the classes fail. Finally, we study properties of a large real network from the point of view of search-based algorithms.

Introduction

Bayesian networks (Pearl 1988) are increasingly popular representations of problems involving reasoning under uncertainty. Practical models based on Bayesian networks often reach the size of hundreds of variables (e.g., (Pradhan *et al.* 1994; Conati *et al.* 1997)). Although a number of exact inference algorithms have been developed, belief updating in Bayesian networks is NP-hard (Cooper 1990) and these algorithms may be impractical in very large and complex models. It is important to focus on approximate inference schemes, as these may be the only alternative for making inference computationally feasible. Unfortunately the general problem of approximate inference with evidence is also NP-hard (Dagum & Luby 1993), even though there are restricted class of networks in which approximate inference is provably amenable to a polynomial time solution (Dagum & Chavez 1993; Dagum & Luby 1993).

There are two basic classes of approximate algorithms for Bayesian networks: stochastic sampling and search-based algorithms. The performance of both classes of algorithms depends on the properties of the underlying joint probability distribution represented by the model. While the performance of sampling algorithms is largely determined by the sampling distribution, the performance of search-based algorithms depends on the quality and quantity of found model instantiations (also called *states* or *scenarios*). Each algorithm has its advantages and disadvantages, that is, it may work well on some but poorly on other networks. It is important to study the properties of real models and subsequently to be able to tailor or combine algorithms for each model utilizing its properties.

The remainder of this paper is structured as follows. We first review briefly various sampling algorithms, discuss their advantages and disadvantages and show examples on which each of them fails. We follow with an introduction to search-based algorithms. Finally, we study the properties of a large medical model, the CPCS network (Pradhan *et al.* 1994) and use these to compare the sampling and search algorithms.

Stochastic sampling algorithms

In stochastic sampling algorithms (also called *Monte Carlo sampling*, *stochastic simulation*, or *random sampling*), the probability of an event of interest is estimated using the frequency with which that event occurs in a set of samples. Differences between various random sampling algorithms can be reduced to differences in the sampling distribution, i.e., the probability distribution from which they draw their samples. If the sampling distribution does not match the actual joint probability distribution, an algorithm may perform poorly.

We will use a simple two-node network presented in Figure 1 to illustrate the advantages and disadvantages of each algorithm. Both nodes are binary variables (denoted by upper case letters, e.g., A), the two outcomes will be represented by lower case letters (e.g., a and $\overline{a}$). We would like to stress here that our discussion aims at very large networks and the simple network of Figure 1 serves only for the purpose of illustration.

Probabilistic logic sampling

The simplest and the first proposed sampling algorithm is the probabilistic logic sampling (Henrion 1988), which works as follows. Each node is randomly instantiated to one of its possible states, according to the probability of this state given the instantiated states of its parents. This requires every instantiation to be performed in the topological order, i.e., parents are sampled before their children. Nodes with observed states

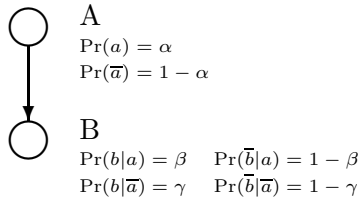

Figure 1: A simple Bayesian network and its numerical parameters (prior probability distribution over A and conditional probability distribution of B given A).

(evidence nodes) are also sampled, but if the outcome of the sampling process is inconsistent with the observed state, the entire sample is discarded.

Probabilistic logic sampling produces probability distributions with very small absolute errors when no evidence has been observed. If there is evidence, and it is very unlikely, most samples generated will be inconsistent with it and will be discarded. Suppose that node B (Figure 1) has been observed at an unlikely value b (both β and γ are very small). Then the percentage of useful samples consistent with the evidence will be $\alpha\,\beta + (1-\alpha)\,\gamma$, which will be very small. This means that most samples will be discarded. For example, with $\alpha = 0.5$, $\beta = \gamma = 0.01$, 98% of the generated samples will be discarded. In a large network with multiple observations, the prior probability of evidence is usually very small and, effectively, probabilistic logic sampling can perform poorly.

Likelihood weighting

Likelihood weighting (Fung & Chang 1990; Shachter & Peot 1990) enhances the logic sampling in that it never generates samples for evidence nodes but rather weights each sample by the likelihood of evidence conditional on the sample. All samples are, therefore, consistent with the evidence and none are discarded.

Also, likelihood sampling suffers from another problem. Suppose that in our example network, with b observed, $0 < \gamma << \alpha << 1$ and $\beta = 1-\gamma$. The likelihood sampling algorithm will set A to $\overline{a}$ most of time, but will assign a small weight to every sample. It will set A to a very rarely, but assign these samples a high weight. Effectively, the generated samples may not reflect the impact of evidence. For example, if $\gamma = 0.0001$, $\alpha = 0.01$, and $\beta = 0.9999$, 99.99% of all samples will be generated for a and 0.01% of the samples will be generated for $\overline{a}$. These proportions may become more extreme in very large networks and with a tractable number of samples (at the current speed of computers, usually tens of thousands of samples for networks consisting of several hundred nodes), it may happen that some states will never be sampled. It is popularly believed that the likelihood sampling suffers from unlikely evidence. This belief is inaccurate — likelihood sampling suffers mainly from a mismatch between the prior and the posterior probability distribution, as demonstrated in the above example.

Enhancements to forward sampling

There are several improvements on these two basic schemes, classified collectively as forward sampling because their order of sampling coincides with the direction of arcs in the network. Each node in the network is sampled after its parents have been sampled.

One of these improvements is *stratified simulation* (Bouckaert 1994) that divides the whole sample space evenly into many small parts, then picks one sample from each part. In other words, it allows for a systematic generation of samples without duplicates. The main problem in applying stratified sampling to large networks is that at each stage of the algorithm, we need to maintain the accumulated high and low bounds for each variable. In a network consisting of hundreds of variables, the high bound approaches the low bound as the sampling proceeds, and they will meet at some point due to the limit of number representation in computer. After this point variables will be sampled arbitrarily. Inaccurate computation will prevent the algorithm from generating desired samples, thus its performance will deteriorate.

Latin hypercube sampling follows also the idea of evenly dividing the sample space, but it focuses on the sample space of each node. It has been found to offer an improvement on any scheme, although the degree of improvement depends on the properties of the model (Cheng & Druzdzel 1999).

Importance sampling (Shachter & Peot 1990) samples from an "importance distribution" rather than the original conditional distributions, that adds flexibility in devising strategies for instantiating a network during a simulation trial. It provides a way of choosing any sampling distribution, and compensating for this by adjusting the weight of each sample. The main difficulty related to this approach is defining a good importance sampling distribution. Self-importance sampling, for example, revises conditional probability table periodically in order to make the sampling distribution gradually approach the posterior distribution.

Backward sampling

Backward sampling (Fung & del Favero 1994) allows for generating samples starting from evidence nodes based on essentially any reasonable sampling distribution. (It also, in a way, belongs to the class of importance sampling.) Backward sampling will work better than forward sampling in the example presented in the section on likelihood sampling. In some cases, however, both backward sampling and forward sampling will perform poorly. Suppose that $\alpha = \gamma << 1$ and $\beta = 1 - \alpha$. Forward sampling will tend to set A almost always to $\overline{a}$, while backward sampling will tend to set it almost always to a. For example, when $\alpha = \gamma = 0.0001$ and $\beta = 0.9999$, 99.99% of the forward samples will include $\overline{a}$ and 99.99% of the backward samples will include a.

Both schemes may fail to approach the correct posterior distribution over node A, which is uniform.

Search-based algorithms

Joint probability distributions over practical models show large asymmetries in the probabilities of their individual instantiations. (We would like to point out that all of our examples on which sampling fails involved large asymmetries in probabilities.) A very appealing consequence of this observation is that a small fraction of most likely instantiations may cover most of the probability space. Search-based algorithms search for the most likely instantiations of a model and compute the posterior probability of an event by weighting the probability of those instantiations that are compatible with the event against the probability of those that are not. Henrion (1990) applied a variation of this idea to an approximate algorithm for belief updating in the probabilistic reformulation of the Internist–1/QMR knowledge base, in which the interactions between diseases and findings are described by two-level binary Noisy–OR gates. Later (Henrion 1991) this requirement was relaxed to gates with a negative product synergy (Druzdzel & Henrion 1993; Wellman & Henrion 1993) between pairs of diseases given each of the findings. The algorithm searches for the top n most likely hypotheses computing their relative likelihood and putting conservative upper and lower bounds on their absolute probabilities. Another search-based algorithm is due to Poole 1993b, who considered the case of models containing "normal" variables, defined as representing elements of a system working under normal conditions and only rarely deviating from this normality. This is equivalent to the assumption that the conditional distributions in the network are close to zero or close to one and implies large asymmetries in the joint probability distributions. In his work on incremental probabilistic inference, D'Ambrosio (1993) introduced "skewed distributions," which are distributions with the property of large asymmetries in probabilities.

In our research, we are probing the following questions: How are search-based algorithms applicable to general Bayesian networks? Under what circumstances can the search-based algorithms meet efficiently the accuracy requirement? Is it possible to predict the performance of a search algorithm by examining the structure and/or the distribution of the network? How to design good heuristics for search?

The two main factors determining the efficiency of the search-based algorithms are (1) how to find quickly a set of the most likely instantiations, and (2) how many such instantiations are needed to achieve the desired accuracy. The first factor is important and is related to search strategies. However, the second factor, that is largely determined by the size of the network and its probability distribution, can be critical in practical application of search-based algorithms.

Poole (1993a) gave an average-case analysis of a search algorithm for networks with extreme probabilities. This analysis, which is independent of search strategies, is based on the assumption that for each value of the parents of a variable X, one of the values of X is infinitesimally close to one and all the other values (faults) are thus close to zero. This assumption is quite strong, because it requires faults are infinitesimally close to zero as the size of the network gets very large.

Even though no practical networks may have extreme, infinitesimal probabilities, they may still have the property of large asymmetries in probability distributions. Skewness of a joint probability distribution can be in fact predicted theoretically from the properties of the prior and conditional probability distributions of individual nodes. Druzdzel (1994) has demonstrated that there are good theoretical reasons for expecting that most domains will indeed contain bulk probability mass in a relatively small fraction of scenarios. According to his analysis based on the Central Limit Theorem, p, the probability mass carried by a randomly selected instantiation follows log-normal distribution. Let $f(\ln p)$ be the density of $\ln p$. A normalized $pf(\ln p)$ expresses the expected contribution of all instantiations with probability p to the total probability mass in logarithmic scale, which belongs to the same class as $f(\ln p)$ with a shifted mean and the same variance.

The diagram in Figure 2 shows these theoretically derived and empirically verified relationships for a model consisting of 10 binary variables with probability distributions $p_1 = 0.1$ and $p_2 = 0.9$. The distribution of the

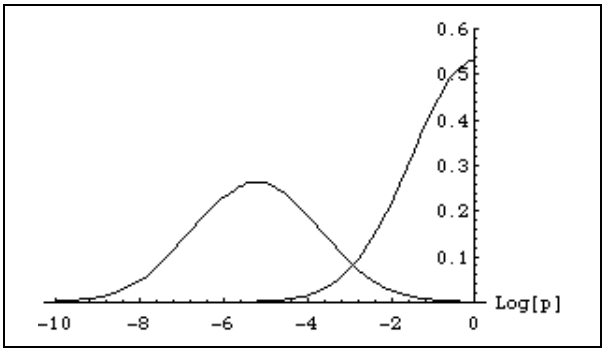

Figure 2: Theoretically derived probability distribution over probabilities of states of the joint probability distribution and the distribution of their contribution to the probability mass for identical conditional probability distributions for 10 binary variables with probabilities of outcomes equal to 0.1 and 0.9.

contributions of probabilities of states to the total probability mass is strongly shifted towards higher probabilities and cut off at point $\log p = 0$. With smaller variances in probabilities (less skewness), the shift is smaller. In such cases, most states will have low proba-

bilities and, hence, no very likely states will be observed. Given a desired error bound ϵ, this function allows to determine the probability threshold value p_0 for which all instantiations less likely than p_0 contribute collectively less than ϵ to the total probability mass. A refinement of this analysis based on the extreme value theory in (Castillo *et al.* 1995) may on theoretical grounds yield a better estimate of the threshold value p_0.

The importance of the theoretical analysis is that it allows for predictions concerning the expected convergence of the algorithms and the error bounds on the probabilities during the search process, something that was impossible based purely on the assumption of asymmetry. We can use p_0 to improve search efficiency by pruning all branches of the search tree that give probability of an instantiation lower than p_0. Also we can calculate the fraction q of instantiations that are less likely than p_0 from the density function $f(\ln p)$. The number of instantiations needed for covering a given percentage $1-\epsilon$ of the total probability mass will be $n\,(1-q)$, where n is the total number of instantiations. This number provides a good estimate of the feasibility of the search.

Search and sampling in practice

We studied the properties of a subset of the CPCS network (Pradhan *et al.* 1994), a multiply-connected multi-layer network consisting of 422 multi-valued nodes and covering a subset of the domain of internal medicine. Among the 422 nodes, 14 nodes describe diseases, 33 nodes describe history and risk factors, and the remaining 375 nodes describe various findings related to the diseases. The CPCS network is among the largest real networks available to the research community at present time. Our analysis is based on a subset of 179 nodes of the CPCS network, created by Max Henrion and Malcolm Pradhan. We used this smaller version in order to be able to compute the exact solution for the purpose of measuring approximation error in the sampling and search-based algorithms. We used both absolute and relative error. Absolute error measures the difference between the approximated probability and the true posterior probability. Both mean squared error and relative entropy or Kullback-Leibler distance between two probability distributions (Cover & Thomas 1991), of which we used the first, are consistent with absolute error. Relative error equals to the absolute error divided by the true posterior probability and it is more stringent than absolute error.

We performed experiments to test how many instantiations cover how much probability space. We applied the stratified sampling algorithm to generate distinct most likely instantiations, and accumulated their weights (probabilities). The experimental results presented in Figure 3 show that a prohibitive number of instantiations would be needed in a moderately sized practical network for search algorithms to generate reasonably accurate results. For example, 100,000,000 (one hundred million) most probable instantiations covered

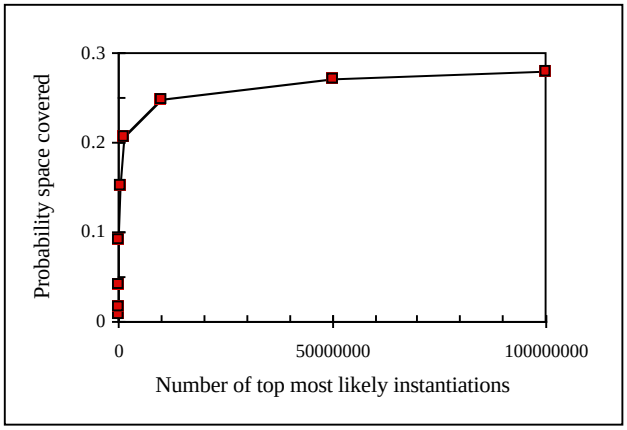

Figure 3: Coverage of the probability space as a function of the number of samples in the CPCS network with 179 nodes.

less than 28% of the total probability space. The total number of possible instantiations of the CPCS network is almost 10^{62}, which is about 10^{54} times more than the number of instantiations found in our experiment. This indicates that even though a small fraction of the total number of instantiations may indeed cover most of the probability mass, it is still intractably huge by all standards.

Figure 4 shows mean squared error in the network as a function of the fraction of the probability space covered. The observed relationship seems to indicate that accuracy increases very slowly with further search for the most likely instantiations.

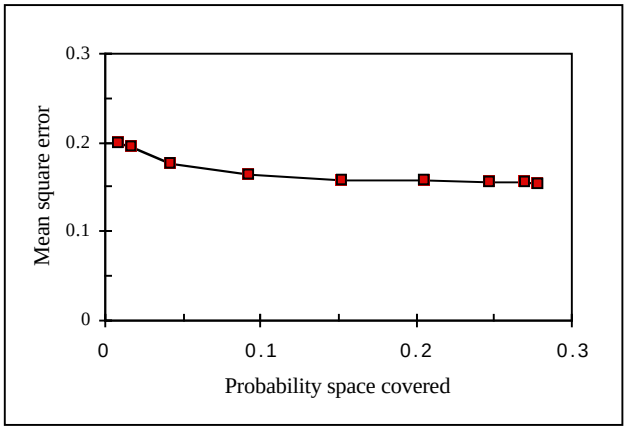

Figure 4: Mean squared error as a function of the probability space covered by the most likely instantiations.

The main conclusion from our experiment is that a search-based algorithm cannot work efficiently in such a network, even though its probability distribution is fairly skewed. Generation of 100,000,000 samples, which fall short of coming even close to covering a reasonable fraction of the probability space, takes hours at

the current speed of computers.

Combining search and sampling

Both search-based schemes and stochastic sampling schemes aim at most probable samples. The key difference between them is how to weight the samples. Search-based schemes simply accumulate probabilities of all different instantiations that have been found so far. We call this weighting method "scenario weighting." Stochastic sampling schemes may generate duplicated samples. To avoid over-counting duplicated samples, each sample is discounted by its probability in the sampling distribution. We call this weighting method "discounted weighting." Our preliminary empirical study shows that different weighting methods result in different precision in the approximated posterior distributions.

We used the stratified sampling algorithm to generate unique samples from the CPCS network. We applied "scenario weighting" and "discounted weighting" separately to approximate the posterior distributions, and computed the maximum absolute error and maximum relative error respectively for these two different weighting methods as shown in Figure 5. The "discounted weighting" gave quite small absolute errors, and the "scenario weighting" produced small relative errors.

Since the probability distribution of network instantiations follows roughly a log-normal distribution, the difference in probability between even highly probable scenarios can be large. Therefore, "scenario weighting" can be strongly biased by a few very large scenarios that only cover a small percentage of the probability space, and it may produce large absolute errors. The "discounted weighting" smoothes the differences in probability among scenarios, but it may produce large relative error for the nodes whose posterior distribution is skewed. Depending on the requirements of the system, different weighting schemes can be chosen for different purposes.

To this point, we know that the quality of the approximation does not only depend on the samples, but also on the weighting methods. As search-based schemes and sampling schemes may find different instantiations, it is interesting to explore how they can help each other with an appropriate weighting method. Search algorithms systematically go through the whole sample space, and can get easily stuck in local maxima. It may be a good idea to start with samples generated from sampling algorithms, as these will be usually very likely, and then search for very likely neighbors.

Discussion

Stochastic sampling and search are two classes of belief updating algorithms for Bayesian networks that are conceptually quite close together. They both rely on instantiations of the network and their performance depends strongly on the joint probability distribution over the network's variables.

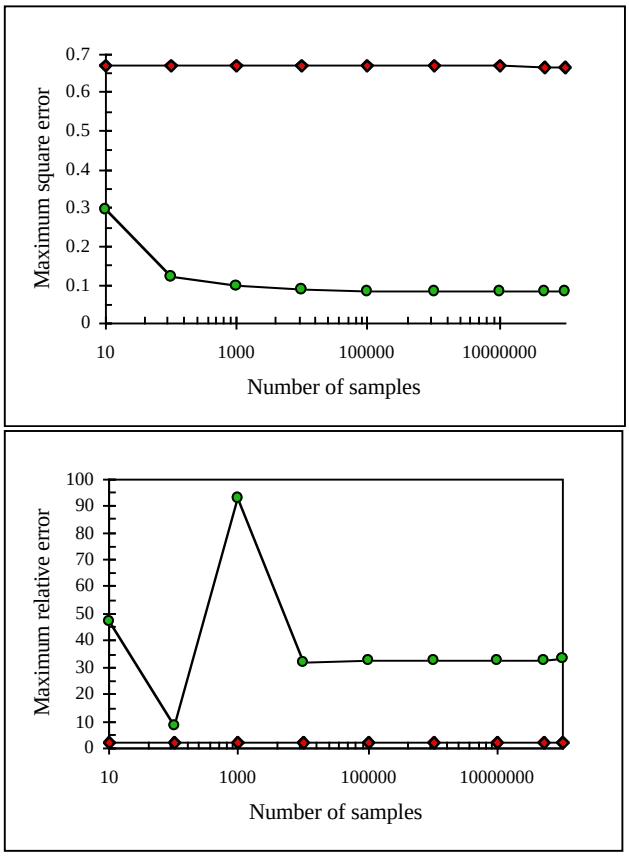

Figure 5: Mean squared error (upper diagram) and relative error (lower diagram) as a function of the number of the most likely instantiations for the "scenario weighting" (diamonds) and the "discounted weighting" (circles), shown in logarithmic scale.

A somewhat depressing finding of our empirical studies is that both sampling and search may not work too well in very large networks. Even if the probability distribution in question is very skewed and a small fraction of all instantiations covers a large area of the probability space, this fraction can still consist of billions of instantiations, a number that is infeasible to search for and use in any practical algorithm.

Generation of an instantiation in stochastic sampling algorithms is usually simpler and computationally less intensive than search for the most likely instantiations. As sampling will usually identify very likely instantiations, it can aid search-based algorithms. Another promising direction of our pursuits of approximate belief updating algorithms for Bayesian networks is combining various algorithms in such a way that an appropriate algorithm is executed depending on the properties of the network.

Acknowledgments

This research was supported by the National Science Foundation under Faculty Early Career Development (CAREER) Program, grant IRI–9624629, and by the Air Force Office of Scientific Research under grant number F49620–97–1–0225. Malcolm Pradhan and Max Henrion of the Institute for Decision Systems Research shared with us the CPCS network with a kind permission from the developers of the Internist system at the University of Pittsburgh. All experimental data have been obtained using SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory and available at `http://www2.sis.pitt.edu/~genie`.

References

Bouckaert, R. R. 1994. A stratified simulation scheme for inference in Bayesian belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, 102–109.

Castillo, E. F.; Bouckaert, R. R.; Sarabia, J.; and Solares, C. 1995. Error estimation in approximate Bayesian belief networks inference. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, 55–62.

Cheng, J., and Druzdzel, M. J. 1999. Latin hypoercube sampling in Bayesian networks. In preparation.

Conati, C.; Gertner, A. S.; VanLehn, K.; and Druzdzel, M. J. 1997. On-line student modeling for coached problem solving using Bayesian networks. In *Proceedings of the Sixth International Conference on User Modeling (UM–96)*, 231–242. Vienna, New York: Springer Verlag.

Cooper, G. F. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* 42(2–3):393–405.

Cover, T., and Thomas, J. 1991. *Elements of Information Theory*. John Wiley & Sons, Chichester, UK.

Dagum, P., and Chavez, R. M. 1993. Approximating probabilistic inference in Bayesian belief networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):246–255.

Dagum, P., and Luby, M. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence* 60(1):141–153.

D'Ambrosio, B. 1993. Incremental probabilistic inference. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, 301–308.

Druzdzel, M. J., and Henrion, M. 1993. Intercausal reasoning with uninstantiated ancestor nodes. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, 317–325. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Druzdzel, M. J. 1994. Some properties of joint probability distributions. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, 187–194. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Fung, R., and Chang, K.-C. 1990. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In Henrion, M.; Shachter, R.; Kanal, L.; and Lemmer, J., eds., *Uncertainty in Artificial Intelligence 5*. North Holland: Elsevier Science Publishers B.V. 209–219.

Fung, R., and del Favero, B. 1994. Backward simulation in Bayesian networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*. San Mateo, CA: Morgan Kaufmann Publishers, Inc. 102–109.

Henrion, M. 1988. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In Lemmer, J., and Kanal, L., eds., *Uncertainty in Artificial Intelligence 2*. Elsevier Science Publishers B.V. (North Holland). 149–163.

Henrion, M. 1990. Towards efficient probabilistic diagnosis in multiply connected belief networks. In Smith, J., and Oliver, R., eds., *Influence Diagrams for Decision Analysis, Inference, and Prediction*. New York: John Wiley & Sons Ltd.

Henrion, M. 1991. Search-based methods to bound diagnostic probabilities in very large belief nets. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI–91)*, 142–150. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

Poole, D. 1993a. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI–93)*, 606–612.

Poole, D. 1993b. The use of conflicts in searching Bayesian networks. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, 359–367.

Pradhan, M.; Provan, G.; Middleton, B.; and Henrion, M. 1994. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, 484–490.

Shachter, R. D., and Peot, M. A. 1990. Simulation approaches to general probabilistic inference on belief networks. In Henrion, M.; Shachter, R.; Kanal, L.; and Lemmer, J., eds., *Uncertainty in Artificial Intelligence 5*. North Holland: Elsevier Science Publishers B.V. 221–231.

Wellman, M. P., and Henrion, M. 1993. Explaining "explaining away". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):287–292.