A Survey of Algorithms for Real-Time Bayesian Network Inference

Haipeng Guo **William Hsu**

Laboratory for Knowledge Discovery in Databases
Department of Computing and Information Sciences, Kansas State University
hpguo@cis.ksu.edu bhsu@cis.ksu.edu

Abstract

As Bayesian networks are applied to more complex and realistic real-world applications, the development of more efficient inference algorithms working under real-time constraints is becoming more and more important. This paper presents a survey of various exact and approximate Bayesian network inference algorithms. In particular, previous research on real-time inference is reviewed. It provides a framework for understanding these algorithms and the relationships between them. Some important issues in real-time Bayesian networks inference are also discussed.

1. Introduction

Over the last 20 years or so, Bayesian networks (BNs) [Pe88, Ne90, RN95, CDLS99] have become the key method for representation and reasoning under uncertainty in AI. BNs not only provide a natural and compact way to encode exponentially sized joint probability distributions, but also provide a basis for efficient probabilistic inference. Although there exists polynomial time inference algorithm for specific classes of Bayesian networks, i.e., trees and singly connected networks, in general both *exact belief update* and *belief revision* are NP-hard [Co90, Sh94]. Furthermore, approximations of them are also NP-hard [DL93b, AH98]. Given the NP-hard complexity results, one of the major challenges in applying BNs into real-world applications is the design of efficient approximate inference algorithms working under real-time constraints for very large probabilistic models. Researchers have developed various kinds of exact and approximate Bayesian network inference algorithms. Some of them are particularly designed for real-time inference. In this paper, we attempt to present a review to BN inference algorithms in general, and real-time inference algorithms in particular to provide a framework to understand the differences and relationships between these algorithms.

2. Preliminaries

2.1 Bayesian Networks

BNs (also known as Bayesian belief networks, causal networks, or probabilistic networks) are currently the dominant uncertainty knowledge representation and reasoning technique in AI [Pe88, Ne90, RN95, CDLS99]. BNs are directed acyclic graphs (DAGs) where nodes represent random variables, and edges represent conditional dependencies between random variables. These random variables can be either continuous or discrete. For simplicity, in this paper we shall only consider discrete ones.

Definition – **Bayesian network:** A Bayesian network is a graph in which the following holds [RN95]:

- A set of random variables makes up the nodes of the network.
- A set of directed links connects pairs of nodes. The intuitive meaning of an arrow from node X to node Y is that X has a *direct influence* on Y.
- Each node has a *conditional probability table* (CPT) that quantifies the effects that the parents have on the node. The parents of a node X are all those nodes that have arrows pointing to X.
- The graph has no directed cycles (hence is a *directed acyclic graph*, or DAG).

A BN represents the exponentially sized *joint probability distribution* (JPD) in a compact manner. Every entry in the JPD can be computed from the information in the BN by the chain rule:

$$P(x_1,...,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i)) \qquad (1)$$

Figure 1.1 shows a simple Bayesian network (the sprinkler network) with 4 nodes [RN95].

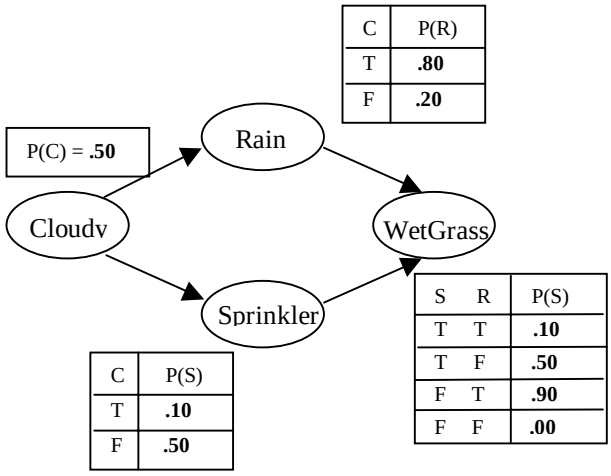

Figure 1 Sprinkler Network

2.2 Bayesian Network Inference

A BN can be considered as a probabilistic expert system in which the probabilistic knowledge base is represented by the topology of the network and the CPTs at each node. The main purpose of building a knowledge base is to use it for inference, i.e., computing the answer for particular queries about the domain. There are two main types of BN inference tasks: *belief updating* (also called *probabilistic inference*) and *belief revision* (also called *MAP explanation*) [Pe88].

The objective of *belief updating* is to calculate *P(X|E)*, the posterior probabilities of *query nodes X*, given some observed values of *evidence nodes E*. A simple form of it results when *X* is a single node, i.e., we are interested in computing the posterior marginal probabilities of a single query node. The task of *belief revision* amounts to finding the most probable instantiation of some hypothesis variables, given the observed evidence. The resulting output is an optimal list of instantiations of the hypothesis variables, a list that may change abruptly as more evidence is obtained [Pe88]. Belief revision for the case when the hypothesis variables are all non-evidence nodes is also known as computing a *most probable explanation*, or *MPE*. An explanation for the evidence is a complete assignment $\{ X_1 = x_1, \ldots, X_n = x_n\}$ that is consistent with evidence E. Computing a MPE is finding an explanation such that no other explanation has higher probability.

Although there exists exact polynomial time inference algorithm for singly connected networks, exact probabilistic inference in general has been proven to be NP-hard by Cooper [Co90]. Approximating probabilistic inference was also shown to be NP-hard by Dagum and Luby [DL93b]. In 1994 Shimony proved that finding MAPs for Bayesian networks is NP-hard [Sh94] and in 1998 Abdelbar and Hedetniemi showed that approximating MAPs is also NP-hard [AH98]. One thing we need to keep in mind is that many belief updating algorithms can be used for belief revision with just small modifications, and vice versa.

2.3 Real-time Inference

A *real-time computing system* [St92] is defined as one in which the correctness of a computation depends not only on its accuracy but also on its timeliness. In this paper, we focus on real-time Bayesian networks inference. We define hard versus soft real-time domains [St92, MHA+94]. A *hard real-time domain* admits strict deadlines where utility degrades instantly if the answer to the query is not returned and a control is not produced. Examples of hard real-time intelligent system domains include crisis monitoring [BSCC89, HGL+98] and some situated control problems. In a *soft real-time domain*, utility loss is gradual after the deadline is passed. Examples include many interactive real-time decision support systems.

There are broadly two kinds of approaches that are being used in real-time AI research: *anytime algorithms* and *multiple methods approaches* [GL94]. Anytime algorithms are iterative refinement algorithms that are able to generate an imprecise answer quickly and refine it through some number of iterations [Bo91, HSC89, Zi93, Zi96, GL94]. The advantage of anytime algorithms is that the computation can be interrupted at any time and still produces results of a guaranteed quality. Multiple methods approaches involve a number of different algorithms that are available for a task, each of which is capable of generating solutions having different characteristics [GL94, LPD88, BH90b, LEFG+90, ET91, DL93a, GL93]. These algorithms may be more or less appropriate for different characteristics of the problems, and may make tradeoffs of solution quality versus time. Works on multiple methods include approximate processing, design-to-time scheduling, and so on. We refer interested readers to [GL93] for more detail.

3. Bayesian Network Inference Algorithms Review

In this section, we will briefly review exact and approximate Bayesian Networks inference algorithms in general.

3.1 Exact Inference

In early 1980s, Pearl published efficient message propagation inference algorithm for *polytrees* [KP83, Pe86a, Pe86b]. The algorithm is exact and has polynomial complexity in the number of nodes, but works only for singly connected networks. Pearl also presented an exact inference algorithm for multiply connected networks called *loop cutset conditioning* [Pe86b]. Loop cutset conditioning algorithm changes the connectivity of a network and renders it singly connected by instantiating a selected subset of nodes referred to as a loop cutset. The resulting single connected network is solved by the polytree algorithm, and then the results of each instantiation are combined weighted by their prior probabilities. The complexity of this algorithm results from the number of different instantiations that must be considered. This implies that the complexity grows exponentially with the size of the loop cutset being $O(dc)$, where d is the number of values that the random variables can take, and c is the size of the loop cutset. It is thus important to minimize the size of the loop cutset for a multiply connected network. Unfortunately, the loop cutset minimization problem is NP-hard.

The most popular exact BN inference algorithm is Lauritzen and Spiegelhalter's *clique-tree propagation algorithm* [LS88]. It is also called the "clustering" algorithm. It first transforms a multiply connected network into a clique tree by clustering the triangulated moral graph of the underlying undirected graph, then performs message propagation over the clique tree. The clique propagation algorithm works efficiently for sparse networks, but still can be extremely slow for dense networks. Its complexity is exponential in the size of the largest clique of the transformed undirected graph.

There exist many other classes of exact inference algorithms. *Arc reversal/node reduction* approach developed by Shachter was one of the three early exact inference algorithms (conditioning, clustering and arc reversal) [Sh86a, Sh90]. It applies a sequence of operators to the network, which reverse the links using Bayes' rule. The process continues until the network is reduced to just the query nodes with the evidence nodes as immediate predecessors [He90]. *Variable elimination (VE)* algorithm eliminates other variables one by one by summing out them [ZP94]. The complexity of VE can be measured by the number of numerical multiplications and numerical summations it performs. An optimal elimination ordering is one that results in the least complexity, but the problem of finding an optimal elimination ordering is NP-complete. *Symbolic probabilistic inference (SPI)* views probabilistic inference as a combinatorial optimization problem, the optimal factoring problem. Probabilistic inference is the problem of finding an optimal factoring given a set of probabilistic distributions [SD90, LD94]. SPI is symbolic and query-driven. *Differential approach* compiles a Bayesian network into a multivariate polynomial and then computes the partial derivatives of this polynomial with respect to each variable [Da00]. Once such derivatives are made available, one can compute answers to a very large class of probabilistic queries in constant time.

For each of these classes of BN inference algorithms as shown in Figure 2, there have been many variants, refinements, hybrids, generalizations and heuristic solutions. For example, in the class of conditioning algorithms, there are *local conditioning* [Di92], *global conditioning* [SAS94], *dynamic conditioning* [Da95], and *recursive conditioning* [Da01]; in the class of clustering algorithms, there are *Shnoey-Shafer* [SS90], *Hugin* [JLO90], and *lazy propagation* [MJ98]; in the class of elimination, there are *bucket elimination* [De96] and *general elimination* [Co00]; and so on.

Besides these general exact inference algorithms, there are some exact special case inference algorithms including *quickscore* for two-level networks with noisy-OR gates [He89], and algorithms exploiting local structures in the distributions such as *causal independency* [He93] *context-specific independencies* [BFGK96]. Figure 2 illustrates main exact BN inference algorithms.

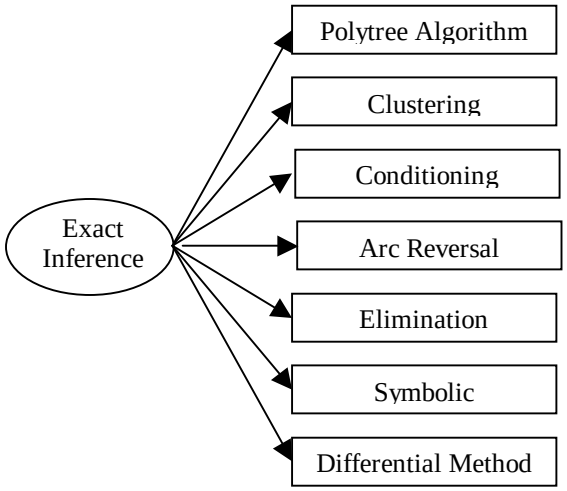

Figure 2 Categories of Exact Inference Algorithms

In general all exact Bayesian network inference algorithms share a running time exponential in the size of the largest clique of the triangulated moral graph, which is also called the induced width of the graph [LS88]. For graphs with many loops, this parameter is large and so rules out the use of exact inference algorithm. For example, Jaakkola and Jordan found that in "QMR-DT", one of the largest BNs in practice, the median size of the maximal clique of the moralized graph is 151.5 [JJ99]. Faced with the intractability of exact inference to large, complex networks, many researchers have investigated approximate inference algorithms.

3.2 Approximate Inference

Approximate BN inference algorithms include *stochastic simulation algorithms*, *model simplification methods*, *search-based methods* and *loopy belief propagation*. See Figure 3.

3.2.1 Stochastic Simulation Algorithms

Stochastic simulation algorithms, also called stochastic sampling or Monte Carlo algorithms, are the most well known approximate BN inference algorithms. They generate a set of randomly selected samples or instantiations of the network according to the CPTs in the model, and then approximate probabilities of query variables by the frequencies of appearances in the sample. The accuracy depends on the size of samples irrespective of the structure of the network. Stochastic simulation algorithms are a big family. They can be divided into two main categories: *importance sampling algorithms* and *Markov Chain Monte Carlo (MCMC)* methods.

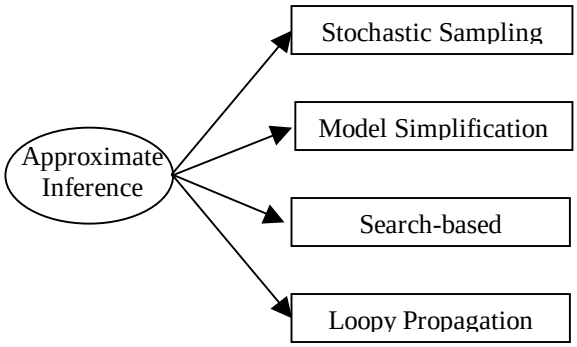

Figure 3 Categories of Approximate Inference Algorithms

Probabilistic *logic sampling* is the first and simplest forward sampling algorithm developed by Henrion in 1988 [He88]. In logic sampling, we run repeated simulations of the world described by the Bayesian network following the influence arrows (thus called forward sampling), throw away samples that are inconsistent with the evidence values, and estimate the probabilities of query nodes by counting the frequencies with which relevant events occur in the sample. When no evidence has been observed, logic sampling works very well; but if there is evidence, especially unlikely evidence, most samples generated will be inconsistent with it and be wasted. The fraction of useful samples decreased exponentially with the number of evidence variables. For a large network with several evidence nodes, the prior probability of the evidence is usually very small, and thus logic sampling performs poorly. Two other papers enhance logic sampling by examining evidential integration, which employs arc reversal to evidence nodes that are sinks to sources to avoid the computational penalty of observed nodes [FC89, CC89].

Likelihood weighting (LW) or *evidence weighting* are designed to get around the problem of logic sampling [FC89, SP90]. In likelihood weighting, every time we reach an evidence node, we don't sample and throw away inconsistent samples; instead we take the observed value of the evidence variable, and weights the sample by the likelihood of evidence conditional on the samples. Likelihood weighting usually converges much faster then logic sampling, and can handle very large networks. The main difficulty with likelihood weighting, and indeed with any stochastic sampling algorithms, is that it takes a long time to converge for unlikely events.

Both logic sampling and likelihood weighting are forward sampling methods. Backward sampling allows for generating samplings from evidence nodes in the direction that is opposite to the topological order of nodes in the network [FF94]. Backward sampling works better than forward sampling with low-likelihood evidence. However, there are still some cases where both will perform poorly because they all fail to approach the correct posterior distributions [LD99].

A well-known method to improve these sampling approaches is to use a revised "importance" distribution for sampling as an approximation to the posterior distributions. The importance distributions can be generated in many ways. Shachter and Peot introduced two variants of *importance sampling algorithms*: *self-importance sampling* (SIS) and *heuristic importance sampling* (HIS) [SP90]. SIS updates its importance function infrequently using the scores generated in the algorithm. HIS computes its importance function by performing a modified version of the singly connected evidence propagation algorithm. Other implementations of importance sampling include Cano's and Hernandez' importance sampling algorithms [CHM96, HMA98]. The experimental result reported shows that they all perform better than likelihood weighting.

Bounded-variance and *AA* algorithms are variants of likelihood weighting described by Dagum and Luby [DKLS95, DL97]. They are based on the LW algorithm and the Stopping-Rule Theorem. They work better than straight LW.

The most efficient stochastic sampling algorithm reported so far seems to be Jian Cheng and Marek Druzdzel's *adaptive importance sampling* for Bayesian networks (AIS-BN) [CD00A]. AIS-BN reduces the sampling variance by learning a sampling distribution that is as close as possible to the optimal importance sampling function. AIS-BN algorithm introduces different weights for samples generated at different learning stages. [CD01] combines AIS-BN with new stopping rules to yield two

other sampling algorithms that work well for large networks.

All stochastic sampling algorithms above can be generalized into the framework of importance sampling [CD00A] in which samples are independent to each other. The main differences between them are in how they initialize and update the important function and how they generate and weight the samples. Another group of stochastic sampling algorithms is called *Markov Chain Monte Carlo (MCMC)* methods in which the samples are dependent. MCMC methods consist of Gibbs sampling, Metropolis sampling and Hybrid Monte Carlo sampling [GG84, GRS96, Ma98, Pe87, CC90]. These algorithms work well if without extreme probabilities in CPTs. When there are extreme conditional probabilities, the convergence of MCMC sampling algorithms can be very slow.

Other sampling techniques include stratified sampling [Bo94], hypercube sampling [CD00b] and quasi-Monte Carlo methods [CD00c]. These techniques are different in how to generate random samples from the uniform distributions. Usually they can be used to any of the above sampling algorithms to improve their performance.

3.2.2 Model Simplification Methods

Model simplification methods first simplify the model until exact methods become feasible and then run an exact algorithm. Some methods reduce the model complexity by *annihilating small probabilities* [JA90]. Others involve removal of weak dependencies [Kj94] or arc removal [En97]. *Localized partial evaluation* algorithm removes selected nodes from networks [Dr95]. *Bounded conditioning* ignores some cutset instances to compute probability bounds and considers more instances to improve the accuracy [HSC89]. The s*tate space abstraction* algorithm reduces the cardinality of CPTs to simplify the model [WL94]. *Variational approach* introduces variational transformations and delinks nodes from the graph one by one until the graph is sparse enough that the exact method is feasible [JJ99, JGJS99]. *Sarkar's algorithm* approximates the Bayesian network by finding the optimal tree-decomposable representation that is the 'closest' to the actual network [Sa93]. *Context-specific approximation* algorithm takes contextual structure into account and simplifies the network by removing distinctions in the probabilities [Po97, Po98]. Other partial evaluation methods include the "*incremental SPI*" algorithm [Da93] and "*mini-buckets*" algorithm [De97].

3.2.3 Search-based Methods

Search based methods assume that a relative small fraction of the joint probability space contains a majority of the probability mass. These algorithms search for the high probability instantiations and then use them to obtain a reasonable approximation [Co85, PR87, He91, Po93a, Po93b, Dr94]. They include Henrion's "*Top-N*" search-based method [He91], Poole's search approach using "*conflicts*" [Po93a, Po93b, Po96], Santos' "*Deterministic Approximation*" and "*Sample-and-Accumulate*" methods, and so on [SS96, SSSW96, SS98]. Druzdzel [Dr94] has theoretically demonstrated that the skewness of the joint probability distributions can be predicted by the asymmetries of the CPTs. However, Lin and Druzdzel's empirical results on a subset of CPCS network (179 nodes) show 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 [LD99].

3.2.4 Loopy Belief Propagation

In the past few years, *loopy belief propagation* (BP) - the use of Pearl's polytree propagation algorithm in a Bayesian network with loops - has become a hot topic [MWJ99, WF99]. Researchers have empirically demonstrated that loopy belief can perform well in the context of error-correcting codes and computer vision [MMC98]. But for some other graphs with loops, BP may give poor results or even fail to converge. More recently, it has been shown that there is a close connection between the belief propagation algorithm and certain approximations to the *variational free energy* in statistical physics [YFW01, Ye01, PA02]. Specifically, the fixed points of BP are shown to coincide with the stationary points of the *Bethe approximate free energy* subject to consistency constraints. Bethe free energy approximation is known to be a special case of a general class of approximations called *Kikuchi free energy* approximations. A general class of BP algorithms are also introduced which attempts to find the stationary points of Kikuchi free energy [YFW01].

4. Algorithms for Real-time Bayesian Network Inference

In this section we focus on discussing real-time Bayesian networks inference algorithms.

4.1 Anytime BN Inference Algorithms

Theoretically, any Bayesian networks inference algorithm that temporarily ignores partial information contained in a BN, and recovers those ignored information whenever the allocated computational time allowed, is an anytime inference algorithm [WL94]. This partial information could be partial instantiations [Po96, SS96, SS98], partial nodes [Dr95], partial edges [Kj94, En97], partial probabilities in CPTs [JA90], partial node states [WL94], and partial cutset or other computational items [HSC89]. Therefore, most approximate inference algorithms can be

easily used as an anytime algorithms by applying them iteratively.

Stochastic sampling algorithms are anytime algorithms. The precision obtained by stochastic sampling algorithms generally increases with the number of samples generated so far and is not affected by the network topology and size. Computation can be interrupted by anytime to yield an approximation. The main problem of stochastic sampling algorithms is that the convergence becomes worse if there are extreme probabilities in the CPTs and the probability of evidence is low. It is also difficult to judge how close of the simulation results to the exact results, especially when the probability of evidence is very low [Ch01].

Many model simplification methods and search-based approaches are also anytime algorithms. In late 1980s Eric Horvitz first investigated the problem of uncertain reasoning under limited computational resources under the name of flexible computation [Ho87, Ho88, Ho90]. His *bounded conditioning* algorithm was the first anytime Bayesian network inference algorithm (under the name of flexible computation) [HSC89]. Bounded conditioning uses conditioning method, but conditions only on a small, high probability cutset instances. As more resources (time) are allocated, more cutset instances are included and the accuracy is improved incrementally. Horvitz also suggested a list of useful strategies for bounded-resource computation including *bound calculation and propagation*, *stochastic simulation*, *completeness modulation*, *abstraction modulation*, *local reformulation* and *default reasoning and compilation* [Ho87].

D'Ambrosio redefined probabilistic inference as term computation in *Incremental Probabilistic Inference* [Da93]. They assumed that most distributions in a belief net are "skewed" and considering only a small number of the largest terms can lead to a good approximation. The process is made incremental by computing larger terms first, and constructing an error bound on the possible remaining probability mass. It can also be made incremental with respect to queries, evidence and model revisions. D'Ambrosio also reported some experiments with real-time decision algorithms in the domain of On-line Maintenance Agent (OLMA) [DB96].

Poole's *search-based algorithm* works very similarly [Po93a, Po93b, Po96]. It uses conflicts to generate a set of possible worlds for computing posterior probabilities. When the distributions are skewed, it produces small error bounds. All search-based algorithms exploit the skewness of the joint distribution probability. In 1994 Druzdezl proved the skewness assumption by applying Central limit theorem [Dr94].

Localized partial evaluation (LPE) considers parts of the network incrementally [DS94, Dr95]. LPE is based on standard message-passing method but uses interval-valued messages instead of point-valued messages, and perform its calculation only on a subset of the nodes in the network. It ignores nodes that are too "far away" from the queried nodes to have much impact on its values. LPE is anytime because it is able to produce better solutions given more time to consider more of the network.

State space abstraction algorithm [LW94] reduces the complexity of the model by changing the state space cardinalities. It produces progressively improving estimates of a probabilistic query via a series of incremental refinements of the state spaces of random variables.

Santos' *deterministic approximate* algorithm enumerates high-probability Independence-Based (IB) assignments to approximate marginal probabilities [SS96, SSSW96, SS98]. IB assignments are partial assignments that take advantage of local independencies not represented by the topology of the network to reduce the number of assigned variables, and hence the probability mass in each assignment. This method is a search-based method and will also benefit from the probability skewness assumption.

Welch's *real-time estimation* method combines genetic algorithm (GA) with stochastic simulation [We96]. It first utilizes a Monte Carlo sampling method (forward or backward sampling) to generate a sufficient number of trails for an initial breeding set. Then it runs a GA search until either the time allocated for a real-time responses runs out, the accuracy reaches a tolerate level, or the graph of the archive probability mass from trails that conform to the evidence is flat. Finally, it uses an estimation based on partial sums of the joint probabilities of trails in the archive rather than the weighted frequency of the values encountered during sampling.

Mini-Clustering is a parameterized approximation scheme that extends the partition-based approximation of mini-bucket elimination to tree decompositions [MKD01]. The scheme is governed by a controlling parameter that allows adjustable levels of accuracy and efficiency in an anytime style.

Variational probabilistic inference transforms the network to a simpler network by variationally delinking nodes and executes an exact algorithm when the resulted graph is sparse enough. Its accuracy generally improves as more nodes are treated exactly and hence has an anytime characteristic of time-accuracy tradeoff [JJ99].

The *TreeNets* framework also allows incremental evaluation of a BN, but the main limitation is that it is only applicable to polytree structure [JN97].

4.2 Metalevel Reasoning and Multiple Methods

4.2.1 Ideal Reformulation

The intelligent reformulation of a Bayesian network can greatly increase the efficiency of inference. But there is a tradeoff between the time dedicated to reformulating and the time applied to executing inference. This problem is called the *metareasoning-partition problem*. Breese and Horvitz studied this problem of ideally apportioning resources between a meta-analysis and the solution of a base problem [BH90a]. They studied different prototypical models of cost. Their analysis showed that the use of metareasoning to dynamically optimize the amount of time expended on reformulation frequently is more valuable than the static policy of halting reformulation after the first valid clique-tree is discovered.

4.2.2 Performance Prediction and Algorithm Selection

So far we have reviewed various exact and approximate Bayesian networks inference algorithms. Each of them has different properties and works better for different class of inference problems. Given a Bayesian network inference problem instance, it is usually important but hard to decide in advance which algorithm among a set of choices is most appropriate. This problem is known as the *algorithm selection problem* [Ri76]. This problem becomes more crucial when there are limited resources. Metareasoning from the characteristics of the problem instances about what algorithm should be selected and executed can significantly increase the inference efficiency.

Jitnah and Nicholson proposed a system for network characterization as the basis for inference algorithms comparison and selection [NJ96]. The domain characterization they proposed includes whole network characteristics (size and connectivity of the network), node characteristics (skewness of the CPT), and characteristics of an evaluation (map of the evidence and query nodes). They presented performance results on logic sampling, likelihood weighting, the polytree and the Jensen algorithm. Their results indicated that domain characterization could be useful for predicting inference algorithm performance. The ability of predicting algorithm performance could provide the basis for more optimal metareasoner that guide the selection and execution of BN inference algorithms.

Borghetti developed an empirical method for using performance data gathered offline to predict future performance of algorithms on new problems [Bo96]. A knowledge base is analyzed into a set of characteristics. Each algorithm is executed individually on a single knowledge base and provides a set of solutions. These solutions are analyzed and the performance profiles for each algorithm are generated. A mapping relation is created between the characteristics and the performance profile of each algorithm. The mapping relation is used to predict the performance of algorithms based on the domain characteristics of the knowledge base. The relation mapping is made by examining the critical points where two or more algorithms' utility curves intersect. Whenever a decision needs to be made, we simply determine whether or not the domain characteristics for a network are above or below the critical points. Then an algorithm is chosen accordingly.

Williams also studied the inference algorithm selection problem but he addressed it in a very different way. He developed and implemented a selection process based on algorithm analysis instead of empirical methods [Wi97].

4.2.3 Algorithms Cooperative Inference

Santos developed a distributed architecture, OVERMIND, for unifying various probabilistic reasoning algorithms that have both anytime and anywhere properties [SSSW95, Wi97]. Anywhere algorithms can exploit intermediate results produced by other algorithms. When different algorithms have both anytime and anywhere properties, they can be harnessed together into a cooperative system. The resultant system can exploit the best characteristics of each algorithm. OVERMIND architecture consists of an intelligent resource allocator, an overseer task manager, and a library of tasks. The library of tasks contains a Genetic algorithm (GA), an integer linear programming (ILP) algorithm, a best first search algorithm (A*) and a hybrid stochastic sampling algorithm (HySS). When performing inference, the A* tended to produce reasonable solutions immediately, Gas took these solutions near some maximas, HySS fine-tuned those maximas, and the ILP finished the optimization and generated the optimal solution [SSSW95].

5. Discussions

5.1 Summary

As just reviewed, there exists a number of exact and approximate Bayesian networks inference algorithms, different algorithms exploit different characteristics of the problem instances and work better for different classes of algorithms. For example, Pearl's poly-tree propagation algorithm has polynomial complexity in the number of nodes, but only works for singly connected networks; the clique-tree propagation is efficient only on sparse networks; the accuracy of stochastic sampling algorithms depends on the size of samples irrespective of the structure of the network, but they converge very slowly if there are extreme probabilities in the CPTs; search-based algorithms are only efficient on networks that have highly-skewed distributions; variational approaches work better on dense networks than sparse ones.

The No Free Lunch (NFL) theorem [WM95, WM97] states that the performance of any algorithm averaged over all possible cost functions will be the same as that of any

other algorithm. Put another way, NFL suggests that there are classes of problems and that for each class there exists an algorithm that solves the problems of that class most efficiently. Applying NFL to the problem of Bayesian network inference, we can draw the conclusion that there does not exist a single champion Bayesian network inference algorithm, and we need to run different algorithms for different problems to maximize run time performance. Therefore, it is necessary to systematically study the matching relationships between the spaces of problems and the spaces of algorithms. This line of research has been explored in [NJ96, Bo96, Wi97], but we believe that there are much more work could be done in this direction.

In real-time BN inference research, we can see that the majority of algorithms for real-time Bayesian networks inference are anytime algorithms. Less work has been done on the multiple methods approach, probably because of the empirical difficulty and its less obviously appealing theoretical properties. We believe that in the future it is still necessary to develop new efficient inference algorithms for particular class of problem instances. However, to gain more efficient inference under real-time constraints, the more important and more promising goal should be integrating various kinds of exact and approximate inference algorithms into one unified inference system such that the most appropriate algorithm or a group of most appropriate algorithms are selected and executed depending on the characteristics of the input inference problem instance [He90, LD99, JJ99].

Other metalevel strategies that could be very helpful for real-time Bayesian networks inference include pre-computation techniques such as pruning the network to delete the irrelevant portions of the network for the given inference problem [BB90, DS94], the problem-solving process monitoring and running time algorithms cooperation [FM01, SSSW95], hybrid inference algorithms that combine the advantages of different algorithms [JJ99], and continual computation [Ho97, Ho01].

Acknowledgements

Support for this research was provided in part by the Army Research Lab under grant ARL-PET-IMT-KSU-07 and by the Office of Naval Research under grant N00014-01-1-0519.

References

[AH98] A. M. Abdelbar and S. M. Hedetniemi. Approximating MAPs for belief networks in NP-hard and other theorems. *Artificial Intelligence* vol. 102, pp. 21-38, 1998.

[BB90] M. Baker and T. E. Boult. Pruning Bayesian networks for efficient computation. In Proc. Sixth Conf. on Uncertainty in Artificial Intelligence, pages 257--264, Cambridge, Mass., 1990.

[BFGK96] C. Boutillier, N. Friedman, M. Goldszmidt, and D. Koller. Context-Specific independence in Bayesian networks. In Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 115-123. San Francisco, Calif.: Morgan Kaufmann, 1996.

[BH90a] J. S. Breese and E. J. Horvitz. Ideal reformulation of belief networks. In Proceedings of UAI-90, Cambridge, MA, pages 64-72. Assoc. for Uncertainty in AI, Mountain View, CA, July 1990.

[BH90b] P. P. Bonissone and P. C. Halverson. Time constrained reasoning under uncertainty. *The Journal of Real-Time Systems*, 2(1/2):25--45, 1990.

[Bo91] M. Boddy. Anytime problem solving using dynamic programming. In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), p. 738-743. AAAI Press, San Mateo, CA, 1991.

[Bo94] R. R. Bouckaert. A stratified simulation scheme for inference in Bayesian belief networks. In Uncertainty in AI, Proceedings of the Tenth Conference, pages 110--117, July 1994.

[Bo96] B. J. Borghetti, Inference algorithm performance and selection under contrained resources, MS Thesis, AFIT/GCS/ENG/96D-05, 1996.

[BSCC89] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The *ALARM* monitoring system: a case study with two probabilistic inference techniques for Belief networks. In Proceedings of the European Conference on AI in Medicine (ECAIM '89), p. 247-256, 1989.

[CC89] H. L. Chin and G. F. Cooper. Bayesian belief network inference using simulation. In L. N. Kanal, J. F. Lemmer, and T. S. Levitt, editors, Uncertainty in Artificial Intelligence 3, pages 129--148. North Holland, New York, 1989.

[CC90] R. M. Chavez and G. F. Cooper. A randomized approximation algorithm for probabilistic inference on Bayesian belief networks. *Networks* 20:661-685, 1990.

[CD00a] J. Cheng and M. J. Druzdzel. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155--188, 2000.

[CD00b] J. Cheng and M. J. Druzdzel. Latin hypercube sampling in Bayesian networks. In Proceedings of the Thirteenth International Florida Artificial Intelligence Research Symposium Conference (FLAIRS-2000), pp. 287-292, Orlando, Florida, AAAI Publishers, 2000.

[CD00c] J. Cheng and M. J. Druzdzel. Computational investigations of low--discrepancy sequences in simulation algorithms for Bayesian networks. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI2000), pp. 72-81, Stanford, CA, Morgan Kaufmann Publishers, 2000.

[CD01] J. Cheng and M. J. Druzdzel. Confidence inference in Bayesian networks, In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI--2001), pp. 75-82, Seattle, WA, Morgan Kaufmann Publishers, 2001.

[CDLS99] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[Ch01] J. Cheng. Efficient stochastic sampling algorithms for Bayesian networks, Ph.D. Dissertation, School of Information Sciences, University of Pittsburgh, 2001.

[CHM96] J. E. Cano, L. D. Hernandez and S. Moral. Importance sampling algorithms for the propagation of probabilities in belief networks. *International Journal of Approximate Reasoning*, 15, 77-92, 1996.

[Co00] F. G. Cozman. Generalizing variable elimination in Bayesian networks, Proceedings of the BERAMIA/SBIA Workshops, 2000.

[Co85] G. F. Cooper. NESTOR: A computer-based medical diagnostic aid that integrates causal and probabilistic knowledge. Ph.D. Dissertation, Medical Informatics Sciences, Stanford University, Stanford, CA, 1985.

[Co90] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence*, vol. 42, pp. 393-405, 1990.

[Da00] A. Darwiche. A differential approach to inference in Bayesian networks. In UAI'00, 2000.

[Da01] A. Darwiche, Recursive conditioning. In *Artificial Intelligence Journal*. Vol 125, No 1-2, pages 5-41, 2001.

[Da93] B. D'Ambrosio. Incremental probabilistic inference. In Proc. Ninth Conf. on Uncertainty in Artificial Intelligence, pages 301--308, Washington, D.C., July 1993.

[Da95] A. Darwiche. Conditioning methods for exact and approximate inference in causal networks. In Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, Montreal, Morgan Kauffman, August 1995.

[DB96] B. D'Ambrosio and S. Burgess. Some experiments with real-time decision algorithms. AAAI Fall Symposium on Flexible Computation in Intelligent Systems, pp. 25-32, Boston, Massachusetts, 1996.

[De96] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In E. Horvits and F. Jensen, editor, Proc. Twelthth Conf. on Uncertainty in Artificial Intelligence, pages 211-219, Portland, Oregon, 1996.

[De97] R. Dechter. Mini-buckets: A general scheme for generating approximations in automated reasoning in probabilistic inference. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pages 1297--1302, Nagoya, 1997.

[DH94] D. Draper and S. Hanks. Localized partial evaluation of belief networks. In Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI '94) , pp. 170—177, 1994.

[Di92] F. J. Diez. Local conditioning in Bayesian networks. Technical Report R-181, Cognitive Systems Lab., Dept. of Computer Science, UCLA, July 1992.

[DKLS95] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for Monte Carlo estimation (extended abstract). In Proceedings of the 36th IEEE Symposium on Foundations of Computer Science, pages 142-149, Portland, Oregon, 1995.

[DL93a] K. Decker and V. Lesser. Quantitative modeling of complex computational task environments. Proceedings of the Eleventh National Conference on Artificial Intelligence, 1993.

[DL93b] P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, vol. 60, pp. 141-153, 1993.

[DL97] Paul Dagum and Michael Luby. An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence*, 93:1-27, 1997.

[Dr94] M. J. Druzdzel. Some properties of joint probability distributions. In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 187--194, 1994.

[Dr95] D. Draper. Localized partial evaluation of Bayesian belief networks. PhD thesis, Computer Science Dept., Univ. of Washington, Seattle, November 1995.

[DS94] M. J. Druzdzel and H. J. Suermondt. Relevance in probabilistic models: backyards in a small world, In Working notes of the AAAI-1994 Fall Symposium Series: Relevance, pages 60-63, New Orleans, LA, November 4-6, 1994.

[En97] R. A. van Engelen. Approximating Bayesian belief networks by arc removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):916-920, 1997.

[Et91] O. Etzioni. Embedding decision-analytic control in a learning architecture. *Artificial Intelligence*, 49:129-159, 1991.

[FC89] R. Fung and K. C. Chang. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In Uncertainty in Artificial Intelligence 5, pages 209-219, New York, N. Y., 1989. Elsevier Science Publishing Company, Inc., 1989.

[FF94] R. Fung and B. D. Favero. Backward simulation in Bayesian networks. In R. Lopez de Mantaras and D. Poole, editor, Proc. Tenth Conf. on Uncertainty in Artificial Intelligence, pages 227--234, Seattle, July 1994.

[FM01] L. Finkelstein and S. Markovitch. Optimal schedules for monitoring anytime algorithms. *Artificial Intelligence*, Vol. 126, Pages 63-108, February 2001.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721--741, November 1984.

[GL93] A. J. Garvey and V. R. Lesser. Design-to-time real-time scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6):1491--1502, 1993.

[GL94] A. J. Garvey and V. R. Lesser. A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, 6(3):317--47, 1994.

[GRS96] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.

[He88] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In Uncertainty in Artificial Intelligence 2, eds. J. Lemmer and L. Kanal, 149-163. New York: Elsevier Science, 1988.

[He89] D. Heckerman. A tractable inference algorithm for diagnosing multiple diseases. Fifth Workshop on Uncertainty in Artificial Intelligence, Windsor, Ontario, 174-181. 1989.

[He90] M. Henrion. An introduction to algorithms for inference in belief nets. In M. Henrion, et al., editor, Uncertainty in Artificial Intelligence 5, pages 129--138. North Holland, 1990.

[He91] M. Henrion. Search-based methods to bound diagnostic probabilities in very large belief nets. In Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence, 1991.

[He93] D. Heckerman. Causal independence for knowledge acquisition and inference. In UAI-93, pp. 122--137, 1993.

[HGL+98a] W. H. Hsu, N. D. Gettings, V. E. Lease, Y. Pan, and D. C. Wilkins. Heterogeneous time series learning for crisis monitoring. In Proceedings of the International Workshop on Multistrategy Learning (MSL-98), Milan, Italy. June, 1998.

[HMA98] L. D. Hernandez, S. Moral, and S. Antonio. A Monte Carlo algorithm for probabilistic propagation in belief networks based on importance sampling and stratified simulation techniques. *International Journal of Approximate Reasoning*, 18:53-91, 1998.

[Ho87] E. Horvitz. Reasoning about Beliefs and Actions under Computational Resource Constraints, Third Workshop on Uncertainty in Artificial Intelligence, Seattle, Washington. July 1987.

[Ho88] E. Horvitz. Reasoning under varying and uncertain resource constraints. Proceedings of the Seventh National Conference on Artificial Intelligence, Minneapolis, MN. August 1988. Morgan Kaufmann, San Mateo, CA. pp. 111-116, 1988.

[Ho90] E. Horvitz. Computation and action under bounded resources, PhD dissertation. Stanford University, 1990.

[Ho97] E. Horvitz. Models of continual computation. Proceedings of the Fourteenth National Conference on Artificial Intelligence, Providence, RI, July 1997.

[HSC89] E. Horvitz, H. J. Suermondt and G. F. Cooper. Bounded conditioning: Flexible inference for decisions under scarce resources, Proc. 5th conference on Uncertainty in Artificial Intelligence, Windsor, Ontario, 182--193, 1989.

[JA90] F. Jensen and S. Andersen. Approximations in Bayesian belief universes for knowledge-based systems. In UAI-90, pp. 162--169, 1990.

[JGJS99] M. I. Jordan, Z. Ghahramani, T. S. Jaakola, and L. K. Saul. An introduction to variational methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105--161, Cambridge, MA, MIT Press, 1999.

[JJ99] T. S Jaakkola and M. Jordan. Variational probabilistic inference and the QMR--DT network. *Journal of Artificial Intelligence Research*, 10:291--322, 1999.

[JLO90] F. V. Jensen, S. Lauritzen and K. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly* 4:269-282, 1990.

[JN97] N. Jitnah and A. Nicholson. Treenets: A framework for anytime evaluation of belief networks. In First International Joint Conference on Qualitative and Quantitative Practical Reasoning, ECSQARU-FAPR'97, 1997.

[Kj94] U. Kjaerulff. Reduction of computation complexity in bayesian networks through removal of weak dependencies. In Proc. of UAI-94, pages 374--382, 1994.

[KP83] J. H. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference engines. In

Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, pages 190--193. International Joint Conference on Artificial Intelligence, August 1983.

[LD94] Z. Li and B. D'Ambrosio. Efficient Inference in Bayes Networks as a Combinatorial Optimization Problem. *International Journal of Approximate Reasoning*, 11, 55--81.

[LD99] Y. Lin and M. Druzdzel. Stochastic sampling and search in belief updating algorithms for very large Bayesian networks. In Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information, pages 77--82, Stanford, California, March 1999.

[LEFG+90] J. S. Lark, L. D. Erman, S. Forrest, K. P. Gostelow, F. Hayes-Roth and D. M. Smith. Concepts, Methods, and Languages for Building Timely Intelligent Systems, Real-Time Systems 2:127-148, 1990.

[LPD88] V. R. Lesser, J. Pavlin and E. H. Durfee. Approximate processing in real time problem solving, *AI Magazine*, Vol. 9, No. 1, pp. 49-61, 1988.

[LS88] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their applications to expert systems, Proceedings of the Royal Statistical Society, Series B., 50, 154-227, 1988.

[Ma98] D. MacKay. Introduction to Monte Carlo methods. In M. Jordan, editor, *Learning in graphical models*. MIT Press, 1998.

[MHA+94] D. J. Musliner, J. A. Hendler, A. K. Agrawala, E. H. Durfee, J. K. Strosnider, and C. J. Paul. The challenges of real-time AI. Technical Report CS-TR-3290, UMIACS-TR-94-69. University of Maryland Institute for Advanced Computer Studies, 1994.

[MJ98] A. L. Madsen and F. V. Jensen. Lazy propagation in junction trees. In Gregory F. Cooper and Serafin Moral, editors, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pages 362--369. Morgan Kaufmann Publishers, 1998.

[MKD01] R. Mateescu, R. Dechter and K. Kask. Partition-based anytime approximation for belief updating. 2001.

[MMC98] R. J. McEliece, D. J. MacKay and J. F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm, *IEEE Journal of Selected Areas of Communication*, pp. 140--152, Feb, 1998.

[MWJ99] K. P. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In Proceedings of Uncertainty in AI, pages 467-475, 1999.

[Ne90] R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems*. John Wiley & Sons, NY, 1990.

[NJ96] A. E. Nicholson and N. Jitnah. Belief network algorithms: a study of performance using domain characterisation. Technical Report 96/249, Department of Computer Science, Monash University, 1996.

[PA02] P. Pakzad and V. Anantharam. Belief propagation and statistical physics, conference on Information Science and Systems, Princeton Univ., 2002.

[Pe86a] J. Pearl. A constraint-propagation approach to probabilistic reasoning. In: L. N. Kanal and J. F. Lemmer (eds), Uncertainty in Artificial Intelligence, Amsterdam, NorthHolland, 3718-382, 1986.

[Pe86b] J. Pearl. Fusion, propagation and structuring in belief networks. UCLA Computer Science Department Technical Report 850022 (R-42); *Artificial Intelligence*, Vol. 29, No. 3, 241-288, September 1986.

[Pe87] J. Pearl: Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence* 32(2): 245-257, 1987.

[Pe88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.

[Po93a] D. Poole. The Use of conflicts in searching Bayesian networks, Proceedings of the Ninth Conference on Uncertainty in AI, Washington D.C., pages 359-367, July 1993.

[Po93b] D. Poole. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities, Proc. Thirteenth International Joint Conference on Artificial Intelligence, pages 606-612, France, August 1993.

[Po96] D. Poole. Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks, *Artificial Intelligence*, 88, 69-100, 1996.

[Po97] D. Poole. Exploiting contextual independence and approximation in belief network inference. Technical report, University of British Columbia, 1997.

[Po98] D. Poole. Context-specific approximation in probabilistic inference. Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.

[PR87] Y. Peng and J. Reggia. A probabilistic causal model for diagnostic problem solving - Part 2: Diagnostic strategy. IEEE Trans. on Systems, Man, and Cybernetics: Special Issue for Diagnosis, 17, 395-406, 1987.

[Ri76] J. R. Rice. The algorithm selection problem. *Advances in computers* 15:65-118, 1976.

[RN95] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence, 1995.

[Sa93] S. Sarkar. Using tree-decomposable structures to approximate belief networks. Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence , Washington, DC, Morgan Kaufmann, 1993.

[SAS94] R. D. Shachter, S. K. Anderson, and P. Szolovits. Global conditioning for prababilistic inference in belief networks. Proceedings of the Uncertainty in AI Conference 1994, San Francisco, CA: Morgan Kaufmann, 514—522, 1994.

[SD90] R. D. Shachter, B. D'Ambrosio, and B. D. Del Favero. Symbolic probabilistic inference in belief networks, Proc. 8th National Conference on Artificial Intelligence, MIT Press, Boston, pp. 126—131, 1990.

[Sh86a] R. Shachter. Intelligent probabilistic inference. In J. F. Lemmer and L. N. Kanal, editors, Uncertainty in Artificial Intelligence, pages 371--382. North Holland, Amsterdam, 1986.

[Sh90] R. Shachter. Evidence Absorption and Propagation through Evidence Reversals. In M. Henrion,R. D. Shachter,J. F. Lemmer, & L. N. Kanal (Eds.), Uncertainty in Artificial Intelligence 5 (pp. 173-190), 1990.

[Sh94] S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, vol. 68, pp. 399--410, 1994.

[SP90] R. D. Shachter and M. A. Peot. Simulation approaches to general probabilistic inference on belief networks. In Proc. of the Conf. on Uncertainty in AI, volume 5, 1990.

[SS90] P. Shenoy and G. Shafer. Axioms for Probability and Belief-Function Propagation. In Shachter, R. et al., eds., Uncertainty in AI, volume 4, 169-198, 1990.

[SS96] E. J. Santos and S. E. Shimony. Exploiting case-based independence for approximating marginal probabilities, *International Journal of Approximate Reasoning*, 14, pp. 25-54, 1996.

[SS98] E. J. Santos and S. E. Shimony. Deterministic approximation of marginal probabilities in bayes nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(4):377-393, 1998.

[SSSW95] E. J. Santos, S. E. Shimony, E. Solomon, and E. Williams. On a distributed anytime architecture for probabilistic reasoning, AFIT/EN/TR95-02, Department of Electrical and Computer Engineering, Air Force Institute of Technology, 1995.

[SSSW96] E. J. Santos, S. E. Shimony, E. Solomon, and E. Williams. Sample-and-accumulate algorithms for belief updating in Bayes networks, Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence, 477-484, Portland, OR, 1996.

[St92] J. A. Stankovic. *Real-Time Computing*. Technical Report. University of Massachusetts, Amherst, 1992.

[We96] R. L. Welch. Real-time estimation of bayesian networks. In Uncertainty in Artificial Intelligence, Proceedings of the 12th Conference, pages 533--544. Morgan Kaufmann, August 1996.

[WF99] Y. Weiss and W. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. In NIPS, volume 12, 1999.

[Wi97] E. M. Williams. Modeling intelligent control of distributed cooperative inferencing, PhD Dissertation, Department of Electrical and Computer Engineering, Air Force Institute of Technology, 1997.

[WL94] M. P. Wellman and C. L. Liu. State-space abstraction for anytime evaluation of probabilistic networks. In UAI '94, pp. 567 -- 574, 1994.

[WM95] D. H. Wolpert and W. Macready. No free lunch theorems for search, The Santa Fe Institute, 1995.

[WM97] D. H. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67-82, 1997.

[Ye01] J. S. Yedidia. An idiosyncratic journey beyond mean field theory, in *Advanced Mean Field Methods, Theory and Practice*, eds. Manfred Opper andDavid Saad, MIT Press, June 2001.

[YFW01] J. S. Yedidia, W. T. Freeman and Y. Weiss. Bethe free energies, Kikuchi approximations, and belief propagation algorithms. TR 2001-16, 2001.

[Zi93] S. Zilberstein. Operational rationality through compilation of anytime algorithms Ph.D. dissertation, Computer Science Division, University of California at Berkeley, 1993.

[Zi96] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, 17(3):73-83, 1996.

[ZP94] N. L. Zhang and D. Poole. A simple approach to Bayesian network computations. In Proc. of the Tenth Canadian Conference on Artificial Intelligence, pages 171--178, 1994.