Maximal Prime Subgraph Decomposition of Bayesian Networks

Kristian G. Olesen
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7C, 9220 Aalborg Ø, Denmark
kgo@cs.auc.dk

Anders L. Madsen
Hugin Expert A/S
Niels Jernes Vej 10, 9220 Aalborg Ø, Denmark
Anders.L.Madsen@hugin.com

Abstract

In this paper we present a method for decomposition of Bayesian networks into their maximal prime subgraphs. The correctness of the method is proven and results relating the maximal prime subgraph decomposition to the maximal complete subgraphs of the moral graph of the original Bayesian network are presented. The maximal prime subgraphs of a Bayesian network can be organized as a tree which can be used as the computational structure for LAZY propagation. We also identify a number of tasks performed on Bayesian networks that can benefit from maximal prime subgraph decomposition. These tasks are: divide and conquer triangulation, hybrid propagation algorithms combining exact and approximative inference techniques, and incremental construction of junction trees. We briefly compare the proposed algorithm with standard algorithms for decomposition of undirected graphs into their maximal prime subgraphs. The discussion shows that the proposed algorithm is simpler, more easy to comprehend, and it has the same complexity as the standard algorithms.

Introduction

Maximal prime subgraph decomposition is also known as decomposition by clique separators and has been used to find efficient solutions to a number of graph theoretic problems such as graph coloring and identification of maximum cliques (Tarjan 1985). The maximal prime subgraph decomposition is used in divide and conquer algorithms to speed up the solution of hard ($\mathcal{NP}$-complete) graph problems. Later, maximal prime subgraph decompositions have been used in the analysis of contingency tables (Darroch, Lauritzen, & Speed 1980) with the goal of generating a compact representation of a domain for use in decision support systems (Badsberg 1992).

Within the framework of Bayesian networks a maximal prime subgraph decomposition can be exploited as the computational structure for LAZY propagation and as the basis for a divide and conquer algorithm for triangulation, hybrid propagation algorithms combining exact and approximative inference techniques, and for incremental construction of junction trees.

(Tarjan 1985) proposed a method for identification of clique separators. This method has later been modified by (Leimer 1993) and (Badsberg 1996). The method presented by (Leimer 1993) is optimized in the sense that a graph is decomposed exactly into its maximal prime subgraphs which is a unique minimal derived system of prime subgraphs. The method presented by (Leimer 1993) is a little faster than the algorithm presented by (Tarjan 1985) even though the time complexity remains the same. In (Badsberg 1996) the method by (Leimer 1993) is extended to hypergraphs.

The method we propose is integrated into the well known procedure for construction of junction trees for Bayesian networks. We extend the construction procedure with two simple algorithms, one that makes the triangulation minimal and one that constructs a cluster tree from the junction tree by recursively aggregating cliques connected by incomplete separators (in the moral graph) to larger clusters. We state and prove that the resulting clusters are the maximal prime subgraphs of the moral graph of the Bayesian network.

Empirical results of the method are reported and the resulting algorithm is compared to standard algorithms for decomposition of undirected graphs.

Definitions and Notation

We shall assume that the reader is familiar with basic concepts and terminology of Bayesian networks and graphs. This section is therefore restricted to an introduction of central definitions and the notation used in the paper.

Bayesian Networks and Junction Trees

A *Bayesian network* $N = (G, \mathcal{P})$ consists of a directed acyclic graph $G = (V, E)$ and a set of conditional probability distributions $\mathcal{P}$. The process of adding undirected edges $e = (Y_1, Y_2)$ to E for all pairs of not already connected parents of each node and dropping the directions of all directed edges is referred to as *moralization*. The result is the *moral* graph $G^M = (V, E' \cup M)$ where E' is the set of edges produced by dropping directions of the edges of E and M is the set of edges added during mor-

alization. A moral graph G^M is *triangulated* if every cycle of length greater than 3 has a chord. The process of successively adding a set of edges T to a graph $G = (V, E)$ such that $G = (V, E \cup T)$ is triangulated is referred to as *triangulation*. The set of edges T is called the *triangulation* and the edges of T are called *fill-in edges*. Thus, the triangulated graph of a directed acyclic graph $G = (V, E)$ is $G^T = (V, E' \cup M \cup T)$.

A triangulation T is *minimal*, if removal of an edge $e \in T$ results in an untriangulated graph. There exist methods for finding triangulations with a minimal number of fill-in edges such as for example the LEX M algorithm (Rose, Tarjan, & Lueker 1976) (with time complexity $\mathcal{O}(ne)$). Heuristic methods for triangulation are in general not guaranteed to produce minimal triangulations. A non-minimal triangulation T can be made minimal by applying a recursive thinning algorithm, that removes redundant fill-in edges added during triangulation. Such an algorithm with time complexity $\mathcal{O}(|T|^2)$ is described in (Kjærulff 1993).

If the nodes V of a graph G can be partitioned into a triple (V', S, V'') of non-empty sets where S is a complete *separator* of V' and V'' in G such that every path from a node $Y' \in V'$ to a node $Y'' \in V''$ includes a node in S, then G is *decomposable* (or *reducible*) otherwise G is *prime* (or *irreducible*). Notice the terminology. In the statistical community a graph G is said to be decomposable if G and all its subgraphs can be decomposed recursively until all subgraphs are complete. We refer to a recursively decomposable graph as a triangulated graph.

A *cluster tree* representation $\mathcal{T}$ of a graph $G = (V, E)$ is a tree where the nodes of $\mathcal{T}$ are subsets (clusters) of V and the union of all clusters equals V. The edge S between two clusters C' and C'' of $\mathcal{T}$ is $C' \cap C''$, the separator of C' and C''. A *junction tree* (also referred to as a *Markov tree* or a *join tree*) is a cluster tree with the additional property that every cluster C on the path between any two clusters C' and C'' includes $C' \cap C''$.

A *clique* is a maximal complete subgraph. A *clique decomposition* of a graph G with respect to a triangulation T consists of all cliques of G^T. A clique decomposition can be represented as a junction tree $\mathcal{T}_T$ where the clusters are the cliques of G^T.

A junction tree representation of a Bayesian network $N = (G, \mathcal{P})$ can be constructed by the following steps: i) Moralize G to obtain G^M, ii) Triangulate G^M to obtain G^T, iii) Organize the clique decomposition induced by G^T as a junction tree $\mathcal{T}$.

Maximal Prime Subgraphs

A subgraph $G(U)$ of a graph $G = (V, E)$ is a *maximal prime subgraph* of G, if $G(U)$ is prime and $G(W)$ is decomposable for all W with $U \subset W \subseteq V$. The *maximal prime subgraph decomposition* (MPD) of G is the set of induced maximal prime subgraphs of G.

Notice that a clique decomposition of a graph G with respect to a triangulation T is a maximal prime subgraph decomposition of G^T.

A *maximal prime subgraph decomposition junction tree* representation $\mathcal{T}_{MPD}$ of G is a junction tree for an MPD where the nodes are the maximal prime subgraphs of G. A maximal prime subgraph decomposition junction tree representation of a graph always exists.

In the following section we give an algorithm for the construction of a maximal prime subgraph decomposition junction tree and a proof of the correctness of the algorithm.

Example 1 [Asia]
The single most famous example of a Bayesian network is the Asia network introduced by (Lauritzen & Spiegelhalter 1988). The graph of the Asia network is shown in figure 1.

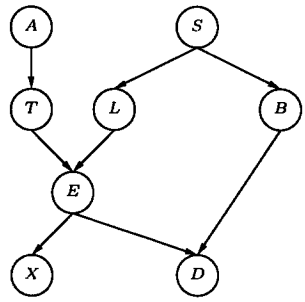

Figure 1: A Bayesian network for the Asia example.

The maximal prime subgraph decomposition of the (moralised) Asia network is shown in figure 2.

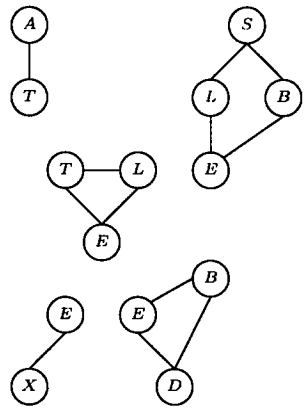

Figure 2: The maximal prime subgraph decomposition of the (moralised) Asia network.

□

Identifying Maximal Prime Subgraphs

The method we propose for identification of maximal prime subgraphs of a Bayesian network $N = (G, \mathcal{P})$ is based on a junction tree representation $\mathcal{T}$ of G. A precondition of the method is that the triangulation T from which $\mathcal{T}$ is constructed is minimal. This can be obtained directly through an algorithm such as LEX M that guarantees a minimal triangulation or, alter-

natively, recursive thinning can be applied to remove redundant fill-in edges.

The main contributions of this paper is the algorithm for constructing a maximal prime subgraph decomposition junction tree $\mathcal{T}_{MPD}$ by aggregation of cliques connected by separators which are incomplete in the moral graph and the proof of correctness of the algorithm which follows below. A Bayesian network $N = (G, \mathcal{P})$ is decomposed into its maximal prime subgraphs as follows. Let G^M be the moral graph of G, let $G^{\mathcal{T}_{min}}$ be the graph corresponding to a minimal triangulation T_{min} of G^M, and let $\mathcal{T}_{min}$ be a junction tree corresponding to the clique decomposition induced by $G^{\mathcal{T}_{min}}$. The maximal prime subgraphs of G^M are formed by aggregating adjacent cliques connected by a separator which is incomplete in G^M. The algorithm is:

Algorithm 1 [Construct MPD Junction Tree]
Let $N = (G, \mathcal{P})$ be a Bayesian network. Let $\mathcal{T}_{min}$ be a junction tree representation of N constructed from a minimal triangulation T_{min} of G^M. If *Construct MPD junction tree* is invoked on $\mathcal{T}_{min}$, then:

1. Set $\mathcal{T}' = \mathcal{T}_{min}$.
2. **Repeat**
   (a) Let S be a separator of $\mathcal{T}'$ connecting C' and C''.
   (b) **If** $G^M(S)$ is incomplete, **then** aggregate C' and C'' in $\mathcal{T}'$.
   Until no separator S of $\mathcal{T}'$ such that $G^M(S)$ is incomplete exists.
3. Return $\mathcal{T}_{MPD} = \mathcal{T}'$.

□

In order to prove that $\mathcal{T}_{MPD}$ is an MPD junction tree for G^M the following lemma is needed:

Lemma 1. *Each clique C of $\mathcal{T}_{min}$ is a prime subgraph of G^M.*

Proof. By contradiction. Let C be a clique in $\mathcal{T}_{min}$. Assume C is reducible in G^M. Then there exist a complete maximal set $S \subset C$ such that (C', S, C'') is a decomposition of C for some $C', C'' \subset C$, see figure 3.

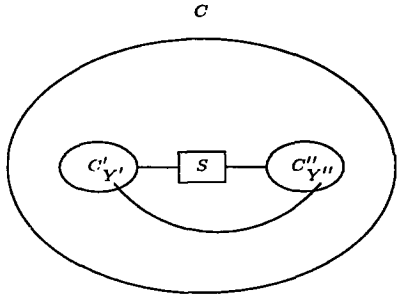

Figure 3: A decomposition of a cluster C into (C', S, C'').

But this means that the fill-in edge between $Y' \in C'$ and $Y'' \in C''$ is redundant, contradicting the fact that the triangulation is minimal. Thus C is a prime subgraph of G^M. □

Next, we prove the main theorem of the paper.

Theorem 2
The clusters of $\mathcal{T}_{MPD}$ constructed with algorithm 1 are exactly the maximal prime subgraphs of G^M.

Proof. First we prove that every separator of $\mathcal{T}_{MPD}$ is a complete separator in G^M. Let S be a separator of $\mathcal{T}_{MPD}$ and let C' and C'' denote the union of all clusters on each side of S, see figure 4.

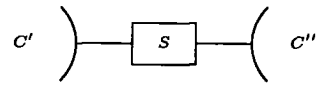

Figure 4: C' and C'' denote the union of all clusters on each side of S.

The separator S is complete in G^M by construction and due to the junction tree property $S = C' \cap C''$. Hence, $(C' \setminus S, S, C'' \setminus S)$ is a decomposition of G^M.

Next, we prove that all clusters of $\mathcal{T}_{MPD}$ are prime subgraphs of G^M. Let C be a cluster in $\mathcal{T}_{MPD}$ which is a clique in $\mathcal{T}_{min}$. By lemma 1, C is a prime subgraph of G^M. Now, let C be a cluster of $\mathcal{T}_{MPD}$ constructed by aggregation of cliques $C_1, \ldots, C_n$ connected by incomplete separators $S_1, \ldots, S_{n-1}$ in $\mathcal{T}_{min}$. Using lemma 1, a decomposition of C_i (for $1 \leq i \leq n$) cannot exists. Next, assume there exists a complete separator $S = C' \cap C''$ of two subsets $C', C'' \subset C$ such that $C = C' \cup C''$. There are three different ways in which S can be related to $C_1, \ldots, C_n$ and $S_1, \ldots, S_{n-1}$:

1. S could be equal to S_i for some $1 \leq i \leq n-1$, but $\nexists S_i : S = S_i$ as S_i is incomplete for all $1 \leq i \leq n-1$.
2. S could be a subset of C_i for some $1 \leq i \leq n$, but $\nexists C_i : S \subset C_i$ as C_i is a prime subgraph for all $1 \leq i \leq n$.
3. Due to steps 1 and 2, it must be the case that $\exists X, Y \in S : X \in C_i, X \notin C_j, Y \notin C_i, Y \in C_j$ for some i and j where $1 \leq i \leq n$ and $1 \leq j \leq n$, but $X \notin adj(Y)$ implies that S is incomplete.

Thus, the complete separator S cannot exists. Thus, C must be a prime subgraph of G^M.

Finally, C is maximal as all separators of $\mathcal{T}_{MPD}$ are complete separators of G^M. □

Corollary 3. *The maximal prime subgraph decomposition of G^M is unique.*

This result has also been proved by (Leimer 1993), but here it follows directly from the proof of theorem 2 as all separators of $\mathcal{T}_{MPD}$ are complete separators in G^M and no other complete separators exist. Although of minor importance, we mention that $\mathcal{T}_{MPD}$ is not necessarily unique. If two separators S_i and S_j are identical there exists at least three clusters sharing the same set of nodes and they can be singly connected arbitrarily.

We can establish some further results relating the clusters of $\mathcal{T}_{MPD}$ to the cliques of G^M.

Proposition 4. *The clusters of $\mathcal{T}_{MPD}$ formed by aggregation of cliques of $\mathcal{T}_{min}$ are not cliques in G^M.*

Proof. If C is a cluster in $\mathcal{T}_{MPD}$ and not a clique in $\mathcal{T}_{min}$ then there exists a separator $S \subset C$ that is not complete in G^M. Therefore C cannot be complete in G^M. □

Theorem 5
The clusters of $\mathcal{T}_{MPD}$ that are cliques of $\mathcal{T}_{min}$ are also cliques of G^M.

Corollary 6. *If C is a clique of $\mathcal{T}_{min}$ and all separators connected to C are complete in G^M, then C is a clique of G^M.*

This follows directly from theorem 5 as the cliques of corollary 6 are exactly the clusters of theorem 5.

The condition in corollary 6 that all separators have to be complete in G^M is necessary, which follows from the proof of the theorem. The proof can be found in (Olesen & Madsen 1999).

The MPD Identification Algorithm

The purpose of the previous sections was to introduce concepts, establish theoretical results including a proof of the correctness of the algorithm, and describe an algorithm simplifying the description of the main algorithm of the paper:

Algorithm 2 [Construct Maximal Prime Subgraph Decomposition Junction Tree]
Let $N = (G, \mathcal{P})$ be a Bayesian network. If *Construct maximal prime subgraph decomposition junction tree* is invoked on G, then:

1. Moralize G to obtain G^M.
2. Triangulate G^M to obtain G^T.
3. Thin out redundant fill-in edges (only performed if necessary) to obtain $G^{T'}$.
4. Organize the clique decomposition induced by $G^{T'}$ as a junction tree $\mathcal{T}$.
5. Construct the MPD junction tree $\mathcal{T}_{MPD}$ (algorithm 1).
6. Return $\mathcal{T}_{MPD}$.

□

Using the above algorithms the organization of the junction tree $\mathcal{T}$ also determines the organization of the maximal prime subgraph decomposition junction tree $\mathcal{T}_{MPD}$. It is, however, possible to organize $\mathcal{T}_{MPD}$ independently of the structure of $\mathcal{T}$.

Example 2 [Asia]
A junction tree for Asia constructed with moral links (T, L) and (E, B), and fill-in edge (B, L) is shown in figure 5. Cliques BLS and BEL are aggregated to form the MPD junction tree since the separator BL is incomplete in G^M.

□

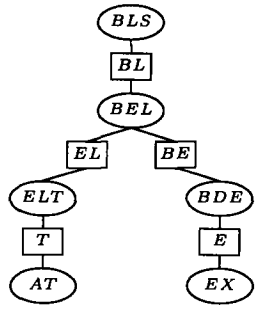

Figure 5: A junction tree for the Bayesian network shown in figure 1.

Applications

In this section we discuss some tasks commonly performed on Bayesian networks that might benefit from maximal prime subgraph decomposition.

Lazy Propagation in MPD Junction Trees

In (Madsen & Jensen 1999) the LAZY propagation architecture was described as a junction tree message passing algorithm. Any computational tree structure maintaining the independence relations of the corresponding Bayesian network can be used to control the flow of messages in the LAZY propagation architecture. An MPD junction tree could, for instance, be used instead of a junction tree.

Message passing and absorption of evidence in an MPD junction tree proceeds in exactly the same way as in junction trees. Thus, changing the underlying structure of the LAZY propagation architecture from junction trees to MPD junction trees does not impose any major adjustments to the inference algorithm. LAZY propagation often produces smaller potentials when performed in MPD junction trees than when performed in general junction trees. It can be demonstrated that an MPD junction tree does not eliminate the possibility of introducing unnecessary fill-in edges during message passing with LAZY propagation. The number of unnecessary fill-in edges introduced will, however, be reduced.

Triangulation by Divide and Conquer

In order to obtain a computationally efficient triangulation of the graph G of a Bayesian network we can exploit the cluster tree representation $\mathcal{T}_{MPD}$ of the maximal prime subgraph decomposition of G. By proposition 4 and theorem 5 we can identify a partial triangulation of G^M. Although a complete triangulation was identified during the construction of $\mathcal{T}_{MPD}$, this need not be a good one. Hence we can re-triangulate the clusters of $\mathcal{T}_{MPD}$ that are not triangulated already and obtain a new total triangulation. The non-triangulated clusters of G^M are smaller than the original graph and they can be triangulated independently. Therefore, the original triangulation problem is split into a number of independent subproblems of smaller sizes. It follows that a divide and conquer strategy for triangulation consists

of a maximal prime subgraph decomposition of G^M followed by a triangulation of all clusters of $\mathcal{T}_{MPD}$. Different methods for triangulation can be used for different clusters, or the clusters can be triangulated in parallel. It is clear that the efficiency of the approach increases with a decreasing size of the maximal prime subgraphs.

Hybrid Propagation

Even though efficient triangulations can be obtained by the divide and conquer approach just described, the cliques of the junction tree may be too large for inference by exact methods. When Bayesian networks become so large that probabilistic inference cannot be performed on the available hardware approximative methods can be applied. In such cases inference can be performed by stochastic simulation or through other approximation methods. Such methods are also relevant if the Bayesian network includes arbitrarily distributed continuous variables for which exact methods do not exist. Approximative and stochastic simulation methods are most often applied to the entire computational structure. In (Dawid, Kjærulff, & Lauritzen 1995) a methodology for performing approximate computations in large junction trees where some cliques are handled exactly and others by approximation or stochastic simulation is described. The emphasis of (Dawid, Kjærulff, & Lauritzen 1995) is on the methodology for combining the inference algorithms whereas the construction of the computational structure for mixed propagations is not treated. MPD junction trees can be used as the basis for construction of computational structures for hybrid propagation.

Incremental Compilation

During the model design and construction phases, the structure of the Bayesian network will often be changed frequently. The model is changed both to improve the conceptual representation of the domain being modeled, but model constructs are also changed to improve the efficiency of the model. Changing the structure of the Bayesian network has implications to the structure of corresponding junction trees and the efficiency of a Bayesian network is almost always measured as the efficiency of the corresponding junction tree. Therefore, it is important to consider how structural changes to the Bayesian network effects the structure of corresponding junction trees. Once a junction tree representation of a Bayesian network has been generated incremental changes made to the Bayesian network should produce incremental changes to the initial junction tree representation. Methods for dynamically adjusting the computational structure has been developed, see for instance (Draper 1995) and (Darwiche 1998).

(Draper 1995) argues that if the Bayesian network is large, then it will often be considerably less expensive to modify an existing junction tree than to perform a new compilation and that incremental modification of a junction tree will often produce more stable results. This is especially important if the designer is trying to optimize the junction tree representation. With the MPD junction tree as an intermediate representation it is possible to modify the junction tree dynamically as changes in the Bayesian network are introduced. These changes will typically influence only one or a few clusters in the MPD junction tree and only the cliques of the junction tree constructed from the triangulation of these clusters have to be modified, while the rest of the junction tree remains intact. Methods for incremental compilation based on this idea is under current investigation.

Sample Empirical Results

The utilization of MPD junction trees to some degree depends on the size of the maximal prime subgraphs of the decomposition. A series of empirical evaluations investigating the sizes of maximal prime subgraphs of a number of large real-world Bayesian networks have been performed.

The Bayesian networks are decomposed into maximal prime subgraphs. For each size of the maximal prime subgraphs, the accumulated number of variables in subgraphs of this size is computed. These numbers are related to the size of the maximal prime subgraphs as illustrated in figure 6.

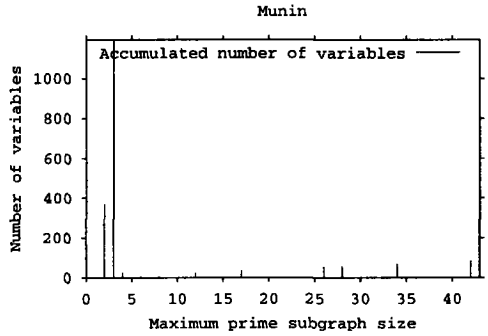

Figure 6: The accumulated number of variables in maximal prime subgraphs of equal size as a function of the size of the maximal prime subgraphs for the Munin network.

A typical result is that a large fraction of the variables lies in small maximal prime subgraphs and a single subgraph is quite large. In some networks up to half of the variables are contained in a single subgraph. Figure 6 shows the results for a fraction of the Munin network (Andreassen, Falck, & Olesen 1992) containing 1066 nodes. This network contains 12 maximal prime subgraphs with 12 or more variables in each. Thus e.g. an exhaustive search for an optimal triangulation is still without reach, but the resources can be directed towards the crucial parts of the network.

Comparison with Existing Methods

In this section we briefly compare the proposed algorithm for maximal prime subgraph decomposition (algorithm 2) to existing algorithms proposed by (Tarjan 1985) and (Leimer 1993). Space limitations prohibits an in depth analysis, instead we refer the interested reader to (Olesen & Madsen 1999).

The time complexity of the algorithms for finding maximal prime subgraph decompositions presented by (Tarjan 1985) and (Leimer 1993) is $\mathcal{O}(ne)$ where n is the number of nodes and e is the number of edges of the decomposed graph.

The algorithm proposed by (Tarjan 1985) finds a prime decomposition of the undirected graph G, but the prime subgraphs of the decomposition are not necessarily the maximal prime subgraphs nor does the algorithm produce a unique decomposition.

The algorithm proposed by (Leimer 1993) is a modication of the one presented by (Tarjan 1985) that determines the unique maximal prime subgraph decomposition of G. The algorithm iterates through all separators in a fixed order resulting in a skewed binary decomposition tree.

The method for maximal prime subgraph decomposition we propose, is based on arranging the cliques of G^T in a arbitrary junction tree and then inspecting the separators of the junction tree. Hence, the order in which the separators are inspected is unimportant. This enables parallel instead of sequential inspection of separators.

The methods proposed by (Tarjan 1985) and (Leimer 1993) are both based on decomposition of undirected graphs. The method we propose is based on decomposition of the moral graph of a directed acyclic graph. For instance, the graph of a Bayesian network. For the sake of comparison with the existing methods it is therefore fair to disregard the moralization step of the method we propose. The time complexity of the three methods is $\mathcal{O}(ne)$, see (Olesen & Madsen 1999).

Conclusion

In this paper we have presented a method for maximal prime subgraph decomposition of Bayesian network. The method is similar to the method of (Tarjan 1985) as modified by (Leimer 1993). The advantage of the proposed method is that it integrates nicely into the construction of junction trees for Bayesian networks. Furthermore, the requirement that the cliques of the triangulated graph are arranged in a specific unique order is relaxed and the separators of the triangulated graph can be inspected in any order.

The maximal prime subgraphs are organized into an MPD junction tree. The MPD junction tree can serve as the computational structure for LAZY propagation and for hybrid propagation schemes. Furthermore, by maintaining the MPD junction tree structure as an intermediate representation between the Bayesian network and a corresponding junction tree methods for divide and conquer triangulation, and incremental construction of junction trees can be employed efficiently.

The results of empirical experiments indicate that a maximal prime subgraph decomposition typically produces a few relative large maximal prime subgraphs and a number of small maximal prime subgraphs. Further research is needed in order to draw precise conclusions on the importance of exploiting maximal prime subgraph decompositions.

References

Andreassen, S.; Falck, B.; and Olesen, K. G. 1992. Diagnostic function of the microhuman prototype of the expert system MUNIN. *Electroencephalography and Clinical Neurophysiology* 85:143-157.

Badsberg, J. H. 1992. Model search in contingency table by CoCo. In Dodge, Y., and Whittaker, J., eds., *Computational Statistics, COMPSTAT*, 251–256.

Badsberg, J. H. 1996. Decomposition of graphs and hypergraphs with identification of conformal hypergraphs. Technical Report R-96-2032, Department of Computer Science, Aalborg University, Denmark.

Darroch, J.; Lauritzen, S.; and Speed, T. 1980. Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics* 8(3):522–539.

Darwiche, A. 1998. Dynamic jointrees. In *Proceedings of the 14th Conference on UAI*, 97–104.

Dawid, A.; Kjærulff, U.; and Lauritzen, S. 1995. Hybrid propagation in junction trees. In *Advances in Intelligent Computing*, volume 945 of *Lecture Notes in Computer Science*. B. Bouchon-Meunier et al. 87–97.

Draper, D. 1995. Clustering Without (Thinking About) Triangulation. In *Proceedings of the 11th Conference on UAI*, 125–133.

Kjærulff, U. 1993. *Aspects of efficiency improvement in Bayesian networks.* Ph.D. Dissertation, Department of Computer Science, Aalborg University, Denmark.

Lauritzen, S. L., and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, B.* 50(2):157–224.

Leimer, H.-G. 1993. Optimal decomposition by clique separators. *Discrete Mathematics* 113:99–123.

Madsen, A. L., and Jensen, F. V. 1999. Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence* 113(1-2):203–245.

Olesen, K. G., and Madsen, A. L. 1999. Maximal prime subgraph decomposition of Bayesian networks. Technical report, Aalborg University, Denmark.

Rose, D.; Tarjan, R.; and Lueker, G. 1976. Algorithmic aspects of vertex elimination on graphs. *Siam Journal of Computing* 5(2):266–283.

Tarjan, R. E. 1985. Decomposition by clique separators. *Discrete Mathematics* 55:221–232.