Arithmetic circuits of the noisy-or models

Jiří Vomlel
Institute of Information Theory and Automation of the ASCR
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4,
182 08 Praha 8. Czech Republic.
e-mail: vomlel@utia.cas.cz

Petr Savicky
Institute of Computer Science
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2,
182 07 Praha 8. Czech Republic.
e-mail: savicky@cs.cas.cz

Abstract

Arithmetic circuits can be used to represent the process of probabilistic inference in Bayesian networks using methods for which the structure and complexity of the process does not depend on the evidence. For example, for the well-known junction tree methods, there is an arithmetic circuit which represents calculation with the same complexity (Darwiche, 2003). However, arithmetic circuits are more flexible and also allow representation of calculations using different types of computational savings, for example when the conditional probability tables of the Bayesian network have a certain local structure (Darwiche, 2002).

In this paper we use the size of arithmetic circuits to compare the effect of preprocessing Bayesian networks with noisy-or gates using parent divorcing and tensor rank-one decomposition. For this purpose, we use the inference methods implemented in Ace by Chavira and Darwiche for several examples of two-layered networks with noisy-or gates (BN2O type networks) in two situations. First, we use Ace on the original network directly, which means that a kind of parent divorcing is used. Second, before the application of Ace the network is preprocessed using a noisy-max decomposition, originally proposed by Díez and Galán and generalized as tensor rank-one decomposition by Savicky and Vomlel. The size of the resulting circuits depends mainly on the size of the largest clique in the triangulated graph of the network. The treewidth of the optimally triangulated graph of the transformed model is provably never larger than the treewidth of the model preprocessed using parent divorcing. Hence, one may expect that tensor rank-one decomposition produces circuits which are usually not larger than the ones from parent divorcing, even if heuristic triangulation is used. Our experiments with Ace confirm this conclusion on average. However, there are also cases where the transformed network provides a significantly larger circuit. Using a better triangulation computed by Hugin instead of the one computed by Ace we reduced the deterioration factor to at most three. This is much smaller than the best improvement factors, which exceed 100.

1 Introduction

Noisy-or models are probably the most popular examples of canonical Bayesian network models. The canonical models differ from general Bayesian network (BN) models in that they have a certain local structure within the conditional probability tables (CPTs). Canonical models were introduced by Pearl in (Pearl, 1988, Section 4.3.2). In literature they are also called causal independence models or models of independence of causal influence (ICI).

A basic task solved by BNs is the probabilistic inference, typically, the computation of all one-dimensional marginals of the joint probability distribution (represented by a BN) given evidence on a subset of network variables. The conditional independence structure of the models enables efficient probabilistic inference using the standard methods, e.g. the junction tree method (Jensen et al., 1990). It was observed by several authors that one can also benefit from the local structure of the CPTs and further improve the computational efficiency of the probabilistic inference.

In this paper we focus on the noisy-or models. A well-known family of noisy-or models are two-level noisy-or networks, abbreviated as BN2O networks. A BN2O network is a BN having the structure of a bipartite graph with all edges directed from one part (the top level) toward the other (the bottom level) and where all CPTs are noisy-or gates. See Figure 1 for an example of a BN2O network structure. An example of a real BN2O network is the decision theoretic version of the Quick Medical Reference model (QMR-DT) (Shwe et al., 1991). This model consists of approximately 600 nodes corresponding to diseases (top level) and 4000 nodes corresponding to findings (bottom level). Each finding has a subset of diseases as its parents. An algorithm tailored for the BN2O networks is the Quickscore algorithm (Heckerman, 1990), where the computations are optimized with respect to evidence on findings.

A different approach that does not assume evidence to be known in advance and that can benefit from the local structure of CPTs is

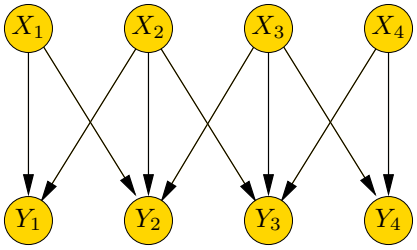

Figure 1: An example of a BN2O model structure.

the compilation of a BN into an arithmetic circuit (AC) (Darwiche, 2003). An AC is a rooted, directed acyclic graph whose leaf (input) nodes correspond to circuit inputs (variables) and whose other nodes are labeled with multiplication and addition operations. The root (output) node corresponds to circuit output. The variables are evidence indicators and parameters of the CPTs. An arithmetic circuit may be used to represent the computation determined by a junction tree (Jensen et al., 1990) in a clustering method, which consists of a fixed sequence of elementary arithmetic operations (additions and multiplications) with real numbers. However, arithmetic circuits are more flexible and may be used to represent different types of computational savings, if they are possible due to specific properties of the initial BN. We use the two methods for constructing an AC implemented in Ace, namely c2d and tabular, see (Chavira and Darwiche, 2006; Chavira and Darwiche, 2007) for more detail. The size of the circuit is used as a measure of complexity of the inference, which is more objective than the running time; the latter is influenced not only by the algorithm, but also by the efficiency of its software implementation.

In this paper we investigate transformations of CPTs representing a noisy-or model before the BN is compiled to a circuit. The standard approach to this problem is parent divorcing (Olesen et al., 1989). We propose to use a transformation based on the decomposition proposed in (Díez and Galán, 2003; Vomlel, 2002). It is a special case of tensor rank-one decomposition of CPTs and, according to the result

of (Savicky and Vomlel, 2007), it uses the minimum possible number of additive components in the case of noisy-max. We show that preprocessing of BN using the above-mentioned transformation before a compiler from BN to AC is used may significantly reduce the size of the resulting circuit obtained by Ace. We systematically tested a range of parameters of artificial networks with randomly generated edges and compared the size of the resulting circuit with parent divorcing as implemented in Ace and with tensor rank-one decomposition. In most cases, the transformed network provided a smaller model. Sometimes, the improvement ratio exceeds 100. There were also cases where the transformed network provided a larger circuit. Analysis of these cases revealed that the tabular method is quite sensitive to the choice of an elimination order of the variables, whereas Ace uses a suboptimal order. Hence, we recalculated some of the cases with the largest deterioration using the optimal order provided by Hugin. Using this more careful method, the transformation never caused an increase of the size of the circuit compared with parent divorcing by a factor larger than 3.

The paper is organized as follows. Necessary notation is introduced in Section 2. Section 3 describes the suggested transformation of a BN as a preprocessing step before an AC compiler is used. In Section 4 we present the necessary information on arithmetic circuits (ACs) and give an example of an AC for the noisy-or gate. In Section 5 we present the experiments which demonstrate the effect of the preprocessing step on the size of the resulting circuit in several randomly chosen examples. Section 6 contains conclusions and a remark on possible directions of future work.

2 Preliminaries

Let N be a set of discrete random variables. The variables will be denoted by capital letters (e.g., X_i, Y, A, etc.) and their states by lowercase letters (e.g., x_{ij_i}, y, a). For both the variables and their states boldface letters will denote vectors.

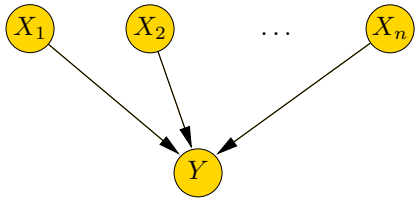

Figure 2: Graph of the noisy-or gate.

Bayesian network (BN) is defined by a pair $\mathcal{N} = (G, \mathcal{P})$, where

- $G = (N, E)$ is an acyclic directed graph (DAG), the nodes of the graph G are variables from the set N, and E is the set of directed edges.
- $\mathcal{P}$ is a system of conditional probability tables $\{P(X|pa(X)),\ X \in N\}$, where $pa(X)$ denotes the vector of parents of X in G.

The joint probability distribution satisfying just the conditional independence statements implied by the DAG and having the probabilities from $\mathcal{P}$ as its (conditional) marginals is uniquely determined by

$$P(\boldsymbol{X}) \ = \ \prod_{X \in N} P(X|pa(X)) \ .$$

A specific example of a BN that we will use in this paper is the noisy-or gate.

Example 1 (Noisy-or gate)**.** Assume a BN of $n+1$ Boolean variables $X_1, \ldots, X_n$ and Y representing the noisy-or relation of Y to $X_1, \ldots, X_n$. The structure of the model is depicted in Figure 2 and the CPT of Y has a local structure defined for $(x_1, \ldots, x_n) \in \{0,1\}^n$ by

$$P(Y = 0|X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} p_i^{x_i} \ , \quad (1)$$

where $p_i \in [0,1], i = 1, \ldots, n$ represent the noise.

3 Tensor rank-one decomposition for noisy-or models

The decomposition suggested in (Díez and Galán, 2003) is applicable to any noisy-max

gate; however, let us describe it only for networks with noisy-or gates. The transformation is applied to each noisy-or gate separately. A noisy-or gate as depicted in Figure 2 is replaced by a subgraph with one additional node B and undirected edges connecting this node with the original nodes as presented in Figure 3. Then, the CPT $P(Y|pa(Y))$ is removed from the network and replaced by $|pa(Y)|+1$ two-dimensional tables for the new edges, which are as follows.

Let $pa(Y) = \{X_1, \ldots, X_n\}$. The interactions between B and X_i and between B and Y are represented by $\varphi_i(B, X_i)$ for $i = 1, \ldots, n$ and $\xi(B, Y)$, respectively, chosen so that for all $(x_1, \ldots, x_n, y) \in \{0,1\}^{n+1}$ we have

$$P(Y = y|X_1 = x_1, \ldots, X_n = x_n) \qquad (2)$$

$$= (1-2y)\prod_{i=1}^{n} p_i^{x_i} + y\prod_{i=1}^{n} 1 \qquad (3)$$

$$= \sum_{b=0}^{1} \xi(b, y) \cdot \prod_{i=1}^{n} \varphi_i(b, x_i) \ , \qquad (4)$$

where the two summands in (3) correspond to the choices $b = 0$ and $b = 1$ in (4). Equality between (2) and (3) follows from (1) and the fact that the sum of (3) for $y = 0$ and $y = 1$ is 1 independently of the values of the remaining variables. Note that the decomposition above uses tables which contain also negative numbers. The original BN may be achieved simply by marginalizing out the auxiliary variable B.

The decomposition is equivalent to the decomposition of $P(Y = y|X_1 = x_1, \ldots, X_n = x_n)$ understood as an $n+1$ dimensional tensor into a sum of tensors of rank one (De Lathauwer and De Moor, 1996). In particular, the minimum number of states of B for which such a decomposition is possible is equal to the rank of $P(Y = y|X_1 = x_1, \ldots, X_n = x_n)$. For noisy-or, this rank is 2, if $p_i < 1$ for at least one index i. Tensor rank-one decompositions are available also for some other canonical models, see (Savicky and Vomlel, 2007).

In Figure 4 we give the transformed structure of BN2O model from Figure 1.

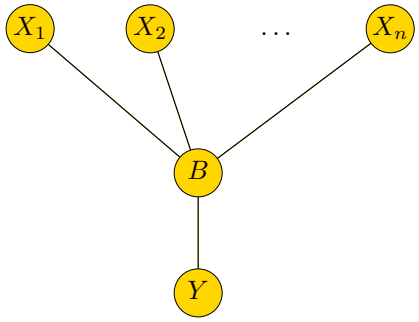

Figure 3: Noisy-or gate from Figure 2 after the transformation using tensor rank-one decomposition

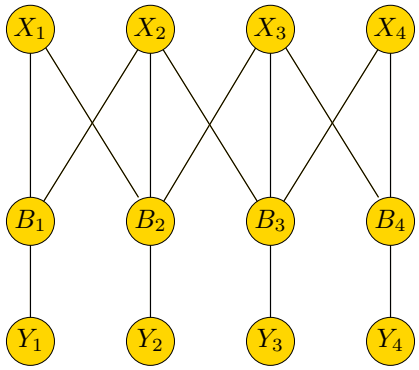

Figure 4: The BN2O model structure from Figure 1 after the transformation using tensor rank-one decomposition.

4 Arithmetic circuits

Let $\boldsymbol{e}$ be evidence $(\boldsymbol{X}_A = \boldsymbol{x}_A^*) = (X_i = x_i^*)_{i \in A}$. Arithmetic circuits as described in the introduction are used to efficiently calculate the probability $P(\boldsymbol{e})$, whose value is given by the multilinear polynomial (Darwiche, 2003)

$$P(\boldsymbol{e}) \quad = \quad \sum_{\boldsymbol{x}} \prod_{X} \lambda_x \ \theta_{x|\boldsymbol{u}} \ ,$$

where the polynomial variables are:

- *BN parameters*: for each variable X and its parents $\boldsymbol{U} = pa(X)$ we have for all values x of X and all values $\boldsymbol{u}$ of $\boldsymbol{U}$ a variable $\theta_{x|\boldsymbol{u}}$, which represents the value

$$\theta_{x|\boldsymbol{u}} \quad = \quad P(X = x|\boldsymbol{U} = \boldsymbol{u})$$

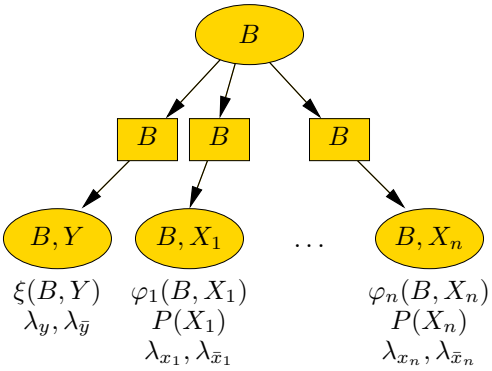

Figure 5: A join tree for the transformed noisy-or gate from Figure 3.

- *evidence indicators*: for each variable X and for each value x of X we have the variable λ_x. The values of the indicator variables encode the given evidence $\boldsymbol{e}$ as follows. The variable λ_x equals one if state x is consistent[1] with the evidence $\boldsymbol{e}$ and zero otherwise. In particular, if there is no evidence for variable X, then $\lambda_x = 1$ for all states x of X.

The structure of the circuit is usually very different from the expression above in order to achieve efficiency.

Example 2 (AC for the noisy-or gate)**.** Let the states of Boolean variables X_i $(i = 1, \ldots, n)$ be denoted x_i (if X_i is true) and $\bar{x}_i$ (if X_i is false) and similarly for the Boolean variable Y.

An AC of a noisy-or gate can be constructed directly from a join tree of the transformed noisy-or gate from Figure 3 (which is a decomposable model) using the construction described in (Darwiche, 2003, Definition 5). We present a join tree[2] of the transformed noisy-or model in Figure 5. Note that each model parameter and each evidence indicator is attached to a node of the join tree.

First, we create one output addition node for the root cluster $\{B\}$ of the join tree. It has as its children two multiplication nodes – one for each state of the variable B. Both multiplication nodes have as their children one addition node from every child separator $\{B\}$ of the root cluster with the compatible state of B. Every addition node from every separator $\{B\}$ has as its children two multiplication nodes from its child cluster ($\{B, Y\}$ or $\{B, X_i\}$ for $i \in \{1, \ldots, n\}$) with the compatible state of B, one for each state of the other variable of the cluster (Y or X_i for $i \in \{1, \ldots, n\}$). Finally, each multiplication node of a leaf cluster has as its children all model parameters and evidence indicators with the compatible states attached to that cluster.

The AC shown in Figures 6 and 7 is the result of the following construction:

1. construct the join tree in Figure 5 for the transformed noisy-or model from Figure 3,
2. use the construction described above to get an AC from the join tree,
3. substitute network parameters from tables $\xi, \varphi_1, \ldots, \varphi_n$ by their values $+1, -1, 0$ and $p_i, i = 1, \ldots, n$, set λ_b and $\lambda_{\bar{b}}$ to 1,
4. simplify the AC by omitting multiplications by one and zero additions, and
5. coalesce parts of the AC that compute the same value so that they are present only once.

The structure of the obtained AC may also be represented by the following formula.

$$\begin{aligned} P(\boldsymbol{e}) \;\; = \;\; & \lambda_y \prod_{i=1}^{n} (\lambda_{\bar{x}_i} \theta_{\bar{x}_i} + \lambda_{x_i} \theta_{x_i}) \\ & + (\lambda_{\bar{y}} - \lambda_y) \prod_{i=1}^{n} (\lambda_{\bar{x}_i} \theta_{\bar{x}_i} + \lambda_{x_i} \theta_{x_i} p_i) \;\; . \end{aligned}$$

Note that the size of this formula is linear in n, while the number of monomials in the expanded polynomial represented by the formula is 2^{n+1}.

If we want to compute all one-dimensional marginals using an AC we can use methods for computing partial derivatives of functions of several variables, e.g., Sawyer (1984). Similarly

[1] Value x of X is consistent with evidence either if $x = x^*$ or if variable X is not present in evidence $\boldsymbol{e}$.

[2] Note that Darwiche's definition (Darwiche, 2003, Section 5.1) of the join tree is more general than the usual one in that it does not require nodes of the join tree to correspond to a maximal clique of the chordal graph.

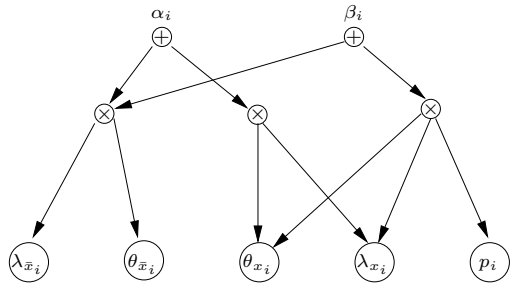

Figure 6: The part of the AC for a noisy-or model corresponding to the variable X_i.

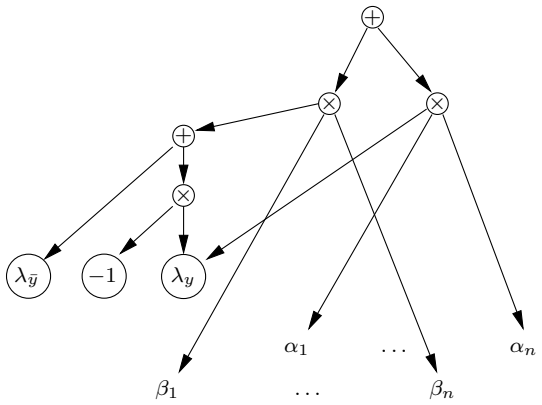

Figure 7: The part of the AC for a noisy-or model that joins the parts of each variable X_i from Figure 6.

to the junction tree methods after two passes through the AC (one upward and one downward pass) we get marginal probabilities for all variables given the evidence $\boldsymbol{e}$. Therefore, in order to get efficient inference it is crucial to have the arithmetic circuit as small as possible. The size of an AC is typically measured by the number of its edges since it is approximately proportional to the number of binary operations.

5 Experimental comparisons

For the experiments we used a development release of Ace (2008). Ace is a package that compiles a BN into an AC using one of two available methods – c2d (Chavira and Darwiche, 2006) or the tabular compilation (Chavira and Darwiche, 2007) – see the Ace manual for details. We carried out experiments with BN2O models of various sizes. The name of the BN2O model in the form of `bn2o-x-y-e-i` contains information about the BN2O structure:

- `x` is the number of nodes in the top level,
- `y` is the number of nodes in the bottom level,
- `e` is the total number of edges in the BN2O model, and `e/y` defines the number of parents for each node from the bottom level.

For each `x-y-e` type ($\texttt{x}, \texttt{y} = 10, 20, 30, 40, 50$ and $\texttt{e/y} = 2, 5, 10, 20$, excluding those with $\texttt{e/y} > \texttt{x}$) we generated randomly ten models (indexed by $\texttt{i} = 0, \ldots, 9$). For every node from the bottom level we randomly selected `e/y` nodes from the top level as its parents. All models and results are available at: `http://www.utia.cz/vomlel/ac/`

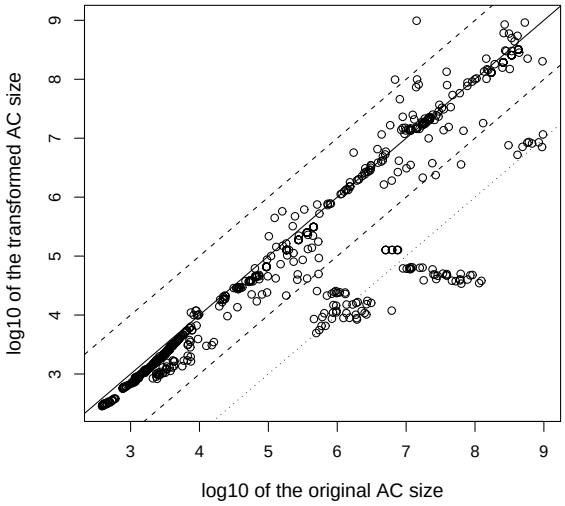

Figure 8: Logarithm of the AC size for the original (o) and transformed (t) models. The straight lines correspond to t/o ratios equal to $10, 1, 1/10, 1/100$.

In Figure 8 we plot the pairs of values of $\log_{10}$ of the ACs' size of the original model[3] (horizontal axis) and the transformed model[4] (vertical

[3]The original model is the model constructed by Ace using parent divorcing.

[4]The transformed model is model obtained using tensor rank-one decomposition.

axis). Since randomness is used during the Ace compilation process, Ace often provides different circuits for the same input model. Therefore every value presented in the plot is the minimum over ten runs – five runs of c2d plus five runs of the tabular method – for both the original and the transformed models.

The AC of the transformed model was smaller in 88% of the BN2O models solved by Ace for both - the original and the transformed models. In several cases we got significant reductions in the AC size (in a few cases multiple order of magnitude), in most of the remaining cases we got smaller reductions. There are also a few cases where the AC of the transformed model is significantly larger. We comment on these cases below. For some of larger models Ace ran out of memory[5] – for 26% of the original models and 21% of the transformed models. For 85% of the tested BN2O models the tabular method led to smaller ACs than c2d.

It is well-known that the efficiency of the inference in BN depends on the size of the largest clique in the triangulated graph of the network or, more exactly, on the total size of the tables in the resulting network. We observed this also in our experiments with Ace; the size of the AC is significantly influenced by the total table size corresponding to the triangulated graph of the models.

The triangulated graph of the transformed model may be obtained from the triangulated graph of the original model by contracting edges between the nodes in the groups of nodes, which are added by parent divorcing for each node in the bottom level. It can be shown that contracting edges does not increase the treewidth of the graph. Hence, the treewidth of the optimally triangulated graph of the transformed model is never larger than the treewidth of the original model. However, if a (non-optimal) heuristic method is used for triangulation then it may happen that we get larger treewidth for the triangulated graph of the transformed model. We believe this is the main reason behind the few large losses of the transformed model.

We conducted additional experiments with all models where Ace provided ACs of the transformed model at least three times larger. In all of these eleven cases we were able to reduce the deterioration factor to less than three using a better triangulation method. Let us have a closer look at a BN2O model having the largest loss for the transformed model – `bn2o-20-30-300-6`. The AC size for the original model was 21 539 918 edges, while the AC of the transformed model generated by Ace when using the minfill heuristics for triangulation had 999 809 342 edges, i.e. about 46 times larger. But, when we constructed the AC of the transformed model using an elimination ordering minimizing the total table size found by Hugin (2008), the AC size dropped to 25 117 892 edges.

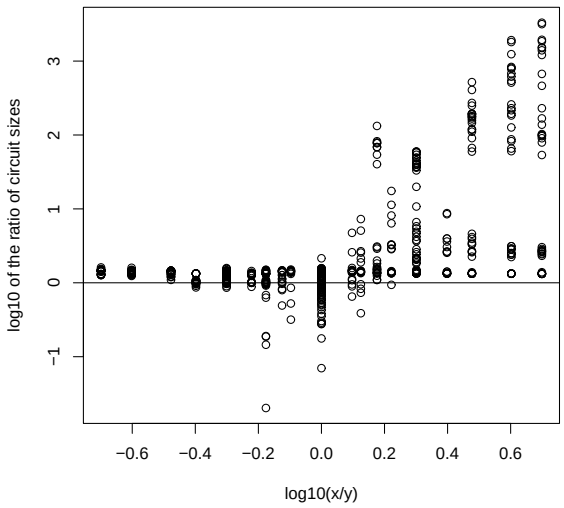

Figure 9: $\log_{10}(o/t)$ with respect to $\log_{10}(x/y)$.

From the plot in Figure 9 we can see that there is a higher probability of larger gains when using the transformed model if there are more nodes in the first level than in the second level of the BN2O network since the log-ratio of the AC sizes of the original and transformed model $\log_{10}(o/t)$ has more often higher values with a positive log-ratio of the number of nodes in the first and the second level $\log_{10}(x/y)$.

[5] All experiments were done with the maximum possible memory for 32 bit Ace, which is 3.6 GB RAM.

We should mention that in (Chavira et al., 2005) the authors perform experiments with BN2O networks, but unlike them we do not assume evidence to be known before the AC is constructed.

6 Conclusions and future work

The performed experiments suggest that tensor rank-one decomposition can help to find smaller ACs for BN2O type networks. In some cases it may even find a solution that could not be found without the transformation. Future work should determine whether similar savings are possible for other canonical models for which a compact tensor rank-one decomposition is known.

Acknowledgments

We are grateful to Mark Chavira for his flexible support of Ace and to Frank Jensen and Anders Madsen for providing us with the optimal triangulation method used in Hugin.

The authors were supported by the Ministry of Education of the Czech Republic under the projects 1M0545 (P. Savicky), 1M0572, and 2C06019 (J. Vomlel). J. Vomlel was also supported by the Eurocores LogICCCC Project FP005 (LcpR) and by the project 201/08/0539 of the Grant Agency of the Czech Republic.

References

Ace. 2008. A Bayesian network compiler. `http://reasoning.cs.ucla.edu/ace/`.

M. Chavira and A. Darwiche. 2006. Encoding CNFs to empower component analysis. In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, Lecture Notes in Computer Science, Volume 4121, pages 61–74. Springer Berlin /Heidelberg.

M. Chavira and A. Darwiche. 2007. Compiling bayesian networks using variable elimination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2443–2449.

M. Chavira, D. Allen, and A. Darwiche. 2005. Exploiting evidence in probabilistic inference. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), Edinburgh, Scotland*.

A. Darwiche. 2003. A differential approach to inference in Bayesian networks. *Journal of the ACM*, 50:280–305.

L. De Lathauwer and B. De Moor. 1996. From matrix to tensor: multilinear algebra and signal processing. In *4th Int. Conf. on Mathematics in Signal Processing, Part I*, IMA Conf. Series, pages 1–11, Warwick. Keynote paper.

F. J. Díez and S. F. Galán. 2003. An efficient factorization for the noisy MAX. *International Journal of Intelligent Systems*, 18:165–177.

D. Heckerman. 1990. A tractable inference algorithm for diagnosing multiple diseases. In *Proceedings of Fifth Conference on Uncertainty in Artificial Intelligence, Windsor, Ontario*, pages 163–171. Elsevier.

Hugin. 2008. Decision engine, version 6.7. `http://www.hugin.com/`.

F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. 1990. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, (4):269–282.

K. G. Olesen, U. Kjærulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, and S. K. Andersen. 1989. A MUNIN network for the median nerve — a case study on loops. *Applied Artificial Intelligence*, 3:384–403. Special issue: Towards Causal AI Models in Practice.

J. Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

P. Savicky and J. Vomlel. 2007. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764.

J. W. Sawyer. 1984. First partial differentiation by computer with an application to categorical data analysis. *The American Statistician*, 38(4):300–308.

M. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, and G. Cooper. 1991. Probabilistic diagnosis using a reformulation of the INTERNIST–1/QMR knowledge base I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30:241–255.

J. Vomlel. 2002. Exploiting functional dependence in Bayesian network inference. In *Proceedings of the 18th Conference on Uncertainty in AI (UAI)*, pages 528–535. Morgan Kaufmann Publishers.