ELSEVIER International Journal of Approximate Reasoning 30 (2002) 1–39

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Evaluation of Bayesian networks with flexible state-space abstraction methods

Chao-Lin Liu [a,*], Michael P. Wellman [b]

[a] *National Chengchi University, 64 Sec. 2 Chih-Nan Road, Wen-Shan, Taipei 11605, Taiwan*
[b] *University of Michigan, 1101 Beal Avenue, Ann Arbor, MI 48109, USA*

Received 1 March 2000; accepted 1 November 2001

Abstract

We investigate state-space abstraction methods for computing approximate probabilities with Bayesian networks. These methods approximate Bayesian networks by aggregating the states of variables. We implement an iterative approximation procedure based on this idea, and the procedure demonstrates the desirable anytime property in experiments. Further theoretical analysis reveals special properties of the approximations, and we exploit these properties to design heuristics for improving performance profiles of the iterative procedure.

1. Introduction

Bayesian networks (also called *Bayes networks*, *probabilistic networks*, and *belief networks*, among others) have become a popular form for capturing uncertainty in problem solving [18,19]. The networks themselves are directed acyclic graphs with nodes augmented by conditional probability tables (CPTs) [26,42]. Nodes in the networks represent variables in the problem being modeled, thus we use "node" and "variable" interchangeably herein. The possible values that a variable can take on are called *states* of the variable.

* Corresponding author.
E-mail addresses: chaolin@nccu.edu.tw (C.-L. Liu), wellman@umich.edu (M.P. Wellman).

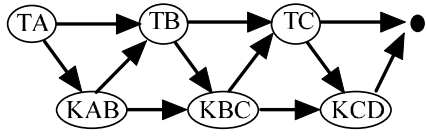
0888-613X/02/$ - see front matter
PII: S0888-613X(01)00067-6

Links qualitatively denote dependence relationships among the variables. A directed link in a Bayesian network connects its *tail* node to its *head*. The link that connects two variables indicates direct dependence between these variables. In other words, information about the state of the tail node generally affects the probability distribution of the head, and vice versa. Variables not linked directly are conditionally independent. The probability distribution of a variable may or may not change with the other indirectly connected variable under specific conditions.

The dependence relationships among variables are described quantitatively by conditional probability tables associated with each node. A CPT describes the conditional probability distribution of its corresponding variable given the possible combinations of states of all its parents.

Fig. 1 shows the graphical structure of a Bayesian network and two of its CPTs for a simplified traffic model. Node *TA* represents the time a traveler leaves location A. Its arrival times at locations B, C, D, ... are represented by nodes *TB*, *TC*, *TD*, ..., respectively. This model assumes that the traveler leaves the intermediate locations upon arrival, so we do not distinguish arrival and departure times for intermediate locations. The average driving speeds from location X to location Y are represented by nodes named *KXY*. Since traffic conditions vary during a typical day, the probability distribution for *KXY* depends on when the traveler leaves location X. For instance, the probability of traveling from A to B between 50 and 60 miles per hour given $TA = ta_2$ is 0.3. In addition, direct links from *TA* and *KAB* to *TB* indicate that the arrival time at location B directly depends on the departure time from A and the average driving speed from A to B.

Given a Bayesian network, we compute probability distributions of interest by *evaluating* the network. For instance, given the network shown in Fig. 1, we can compute the probability of a traveler arriving at C at time tc_1 given a departure from A at ta_2. Conversely, we can compute the probability of the traveler having departed from A at ta_2 when we observe that the traveler arrives at C at time tc_1.



TA	
ta_1	0.5
ta_2	0.4
ta_3	0.1

KAB	$TA = ta_1$	$TA = ta_2$	$TA = ta_3$
50-60	0.3	0.6	0.2
60-70	0.4	0.3	0.1
70-80	0.3	0.1	0.7

Fig. 1. A Bayesian network for a simple traffic model.

The calculation of the desired probability distributions can be carried out with a variety of evaluation algorithms for Bayesian networks. These algorithms include, but not limited to, graph reduction [49], junction tree [27], symbolic probabilistic inference [9,11,55], cutset algorithms [15,52], and the Shenoy–Shafer architecture [50].

The computational cost of the evaluation task varies from problem instance to problem instance, depending on factors such as network structure [12,33,40]. In a few special cases, there exist efficient algorithms for the task. For instance, the computational complexity of evaluating singly connected networks is linear in the number of nodes in the networks [42], and the complexity can be further reduced to sublinear if we compile the network in a preprocessing phase [14]. Compilation of Bayesian networks may help to reduce evaluation time for other more general network structures [10]. Despite these special techniques, the task of exact evaluation of general Bayesian networks, that is, computing exact values of the probability distributions, is NP-hard [4].

In light of this complexity result, many research projects seek to find good approximation methods for the evaluation task. Unfortunately, approximate evaluation of Bayesian networks is not easier than exact evaluation in terms of computational complexity. The task of approximating marginal probabilities to a fixed degree in Bayesian networks is also NP-hard [6,47].

Nevertheless, we can still benefit from the study of approximation methods. Even without guarantees of fixed degrees of accuracy, approximation methods offer reasonable prospects of significant accuracy, which is a lot better than many alternatives. Also, approximation methods offer the opportunity to consider problems much larger than we could otherwise, which could compensate substantially for a loss of accuracy. Moreover, studying approximation methods provides a foundation for the design of algorithms that support robust performance over a range of real-time applications.

Approximation algorithms for Bayesian networks may compute two types of solutions: bounds of the desired probabilities or point-valued approximations. The literature has seen a wide range of approaches for computing bounds of the desired probabilities. For instance, *bounded conditioning* ignores some cutset instances to compute lower and upper bounds of the desired probabilities [22]. *Localized partial evaluation* computes bounds of the desired probabilities by temporarily ignoring selected nodes from the given networks [16]. Search-based algorithms consider more probable assignments of all variables and use these instances to compute probability bounds [8,44]. There are also algorithms that employ max and min operations in computing probability bounds [13,36,45], while some others exploit special features of the conditional probability distributions in the networks in computing bounds [23,36].

The literature has also seen a variety of approaches for computing point-valued approximations of the desired probabilities. Stochastic simulation

algorithms apply Monte Carlo sampling to compute approximations of the desired probabilities [7,20,38,41]. Given sufficient time, results computed by these simulation-based algorithms converge to the exact solutions. Jensen and Andersen propose an algorithm that sets to zero very small values in the potential functions of cliques in junction trees [25]. Roughly speaking, cases with very small values in the potential functions correspond to relatively less probable assignments of all variables, and therefore may be ignored in computing approximate probabilities. It is also possible to approximately evaluate Bayesian networks by ignoring weak dependencies among variables, and this can be achieved either in the junction tree of the given Bayesian network [29] or directly in the given Bayesian network [53].

Conceivably, Bayesian networks are useful for applications that require solutions other than probability distributions [11]. For instance, some applications need to find the most probable assignment to all the variables that are consistent with the states of observed variables. These so-called maximum a posteriori (MAP) explanations can be computed with exact [11] or approximation algorithms [13,48]. The tasks for computing MAPs are also intractable in general, both exactly [51] and approximately [1].

In this paper, we report on a particular family of algorithms for approximating probability distributions by aggregating states of variables [54]. We introduce the notation used in this paper in the next section. Section 3 presents motivation and methods for approximating Bayesian networks by state-space abstraction. Section 4 delineates the alternatives for measuring the quality of approximate probability distributions. In Section 5, we present an algorithm for anytime evaluation of Bayesian networks that employ the state-space abstraction methods, and study its performance in experiments. We then discuss theorems regarding the quality of approximations of the probability distributions in Section 6. Applying the theorems, we devise heuristics for controlling the anytime algorithm to achieve better performance. We study the resulting performance of the anytime algorithm in experiments in Section 7. Finally, we summarize and discuss our main contributions.

2. Notation

We denote variables by capital letters and their states by corresponding small letters. Sets of variables and their states are denoted by bold-faced letters. When necessary, we use superscripts to distinguish variables that belong to a set of variables, and subscripts to distinguish states of a variable. For instance, $\boldsymbol{X}$ represents a set of variables, and this set may contain three nodes X^1, X^2, and X^3. The state space of the variable X^1 may contain three possible states x_1^1, x_2^1, and x_3^1. The *cardinality* of a variable is the number of states in its state space. Hence, the cardinality of X^1 is three.

The conditional probability tables associated with nodes specify the probability distribution of the node given the states of parent nodes of the node. Variables at the tail of the incoming links of X are *parents* of X, denoted $\boldsymbol{P}(X)$, and variables at the head of the outgoing links of X are *children* of X, denoted $\boldsymbol{C}(X)$. We use the shorthand $\Pr(x|\boldsymbol{p}(X))$ to represent an entry, $\Pr(X = x|\boldsymbol{P}(X) = \boldsymbol{p}(X))$, in the conditional probability table associated with node X. Take the Bayesian network shown in Fig. 1, for example. The set of parents of KBC, $\boldsymbol{P}(KBC)$, consists of TB and KAB. The state space of TA contains three states: ta_1, ta_2, and ta_3.

The joint probability distribution of all variables $\boldsymbol{V} = \{V^1, V^2, \ldots, V^n\}$ in the network is implicitly encoded in the conditional probability distributions stored with the nodes. Specifically,

$$\Pr(\boldsymbol{v}) = \prod_{i=1}^{n} \Pr(v^i|\boldsymbol{p}(V^i)).$$

3. State-space abstraction

3.1. Motivation

The cardinality of variables has a substantial influence on the computational complexity of the evaluation of Bayesian networks. Although the precise quantitative influence depends on the evaluation method [12], in general the relationship is exponential for known algorithms. For instance, the computational complexity of the junction-tree method is $O(pr^m)$, where r, p, and m are, respectively, the maximum number of states of an individual node in the Bayesian network, the number of clique nodes, and the maximum number of nodes in a clique [39]. When we increase the cardinality of all the variables in a Bayesian network at the same time, the sizes of the conditional probability tables of all variables increase exponentially, which makes the computation time of the junction-tree algorithm increase exponentially as well.

We illustrate the influence of cardinality on computation time with the multi-stage Bayesian network of Fig. 2. Each variable is conceptually real-valued, and so state-space cardinality is determined by the choice of granularity. In each stage, $C1$ and $C2$ are two variables that directly influence the states of the target variable T. The target variable is the variable that a decision maker would like to control by action A. This decision maker, however, does not have direct control of the states of T, and can directly influence the states of the variables $C1$ and $C2$. Also, the decision maker does not have the capability to collect information about the actual states of the target variable. The effects of the action are only partially observable to the decision maker via the actual observations O, and the choice of the decision maker's action is dependent on

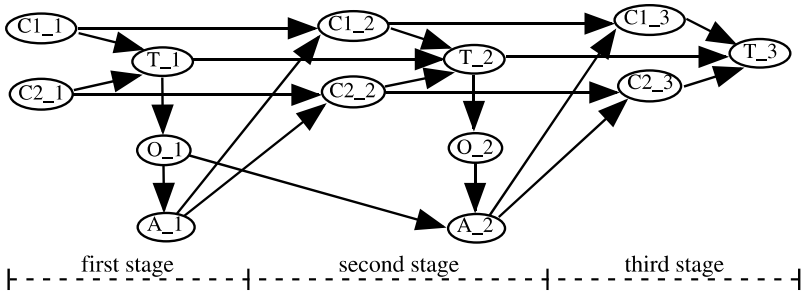


Fig. 2. A multi-stage Bayesian network.

O of the current stage and the previous stage if available. The task is to compute the marginal distribution of T_3 given O_1 instantiated.

As Fig. 3 indicates, computation time increases exponentially with cardinality. We test the junction-tree method implemented in HUGIN API 2.0, executed on SunSparc 20 machines running SunOS 4.1.3 with 32 megabytes RAM and 132 megabytes swap space. The chart shows the computation time for evaluating the network at cardinalities varying from 4 to 20 states per node. Networks with 22 or more states for each node require more computer memory than we have in the test environment, and are not executable.

This chart demonstrates a trade-off between computational time and solution precision. On the one hand, we may want to include more possible states to model the world more precisely. For instance, fine-grained discretization is desirable for variables representing readings from sensors and map locations in the robotics domain [48]. On the other hand, considering more states for variables increases the computation time, thereby potentially making the results of computation less useful in the face of urgent deadlines.

This observation suggests that we should apply the knowledge encoded in Bayesian networks in a more flexible way. We are not in general constrained to

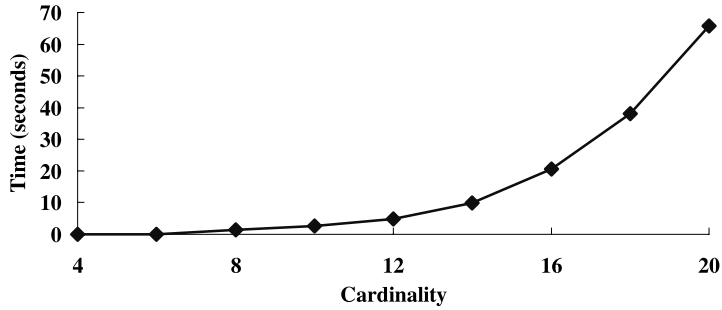


Fig. 3. Computation time surges as we gradually increase cardinality of variables in Fig. 2.

evaluating a network at the precision in which it is encoded. The degree to which we apply the knowledge contained in the available models should depend on the computational resources available for each individual application [2,21,28,43].

As indicated by Fig. 3, one way to reduce the evaluation time for a given Bayesian network is to reduce the number of states for variables. By sufficiently reducing the cardinality, we enable ourselves to compute a solution within the allocated computation time. If given more time for computation, we may run the algorithm at a finer grain, considering more states. Solutions obtained in later iterations are expected be more precise than those obtained in previous iterations. This iterative mechanism thus allows the inference method to adjust to unknown deadlines systematically. We present such *state-space abstraction methods* in the following sections.

3.2. Abstracting a Bayesian network: average policy

We abstract a Bayesian network by aggregating states of selected variables into *superstates*. We call the variables whose states are aggregated *abstracted variables*, the new network that contains abstracted variables an *abstract Bayesian network*, and the original states of the abstracted node *elementary states*. Each superstate is an aggregation of a consecutive subset of elementary states. We use $[a_{i,j}]$ to denote the superstate that is the aggregation of the elementary states $a_i, a_{i+1}, \ldots, a_j$ of an abstracted variable A.

In general, abstracting the state space of a random variable can induce conditional dependencies not present in the original network [3]. Incorporating the requisite new links, however, would defeat the motivation for abstraction in the first place. Therefore, we forego preservation of the joint distribution, and choose to preserve the graphical structure of the given network. With a preserved graphical structure and reduced cardinality, the abstract network requires less time to evaluate than the original network.

Given the decision to preserve the graphical structure, the remaining work for abstracting a Bayesian network is to assign the numbers in the CPTs of the abstracted variable and its children. We call the methods for assigning these probability values a probability reassignment *policy*. Let A be an abstracted variable, and Y be a child of A. A superstate is an aggregation of elementary states, so we set the conditional probability of a superstate to be the sum of the conditional probabilities of its constituent elementary states. Using $\widehat{\mathbf{Pr}}(\cdot)$ to denote probability values in the abstract Bayesian network

$$\widehat{\mathbf{Pr}}([a_{i,j}]|\boldsymbol{p}(A)) = \sum_{k=i}^{j} \Pr(a_k|\boldsymbol{p}(A)). \tag{1}$$

For instance, aggregating ta_2 and ta_3 of TA in Fig. 1 yields $\widehat{\mathbf{Pr}}([ta_{2,3}]) = 0.5$.

Assigning conditional probability values to the children of abstracted nodes is less straightforward, since we discard information in the conditional proposition. We adopt the *average policy*

$$\widehat{\Pr}(y|[a_{i,j}],\boldsymbol{px}(Y)) = \frac{1}{j-i+1}\sum_{k=i}^{j}\Pr(y|a_k,\boldsymbol{px}(Y)), \tag{2}$$

where $\boldsymbol{PX}(Y)$ denotes the parents of Y excluding A, that is, $\boldsymbol{PX}(Y) = \boldsymbol{P}(Y)\backslash\{A\}$. For instance, if we aggregate ta_2 and ta_3 of TA in Fig. 1, we will assign 0.4 as the conditional probability for the case that $KAB = 50$–60 given $TA = [ta_{1,2}]$.

The probability assignment task can be carried out locally since the operation relies only on information that is available in the CPTs of the abstracted nodes and their children. The average policy weighs the components of Y's conditioning state equally. If we have information about the relative importance of $\Pr(y|a_k,\boldsymbol{px}(Y))$, such as marginal probabilities $\Pr(a_k,\boldsymbol{px}(Y))$ for some a_k and $\boldsymbol{px}(Y)$, we may assign the new conditional probability tables more precisely [54]. However, information such as $\Pr(a_k,\boldsymbol{px}(Y))$ is generally not included in Bayesian networks, and computing such marginal probabilities at runtime would defeat the purpose of abstraction. As a result, we choose to assume that $\Pr(a_k,\boldsymbol{px}(Y))$ are equal among all elementary states a_k for a given $\boldsymbol{px}(Y)$.

4. Measuring quality of approximations

We measure the quality of approximations in terms of the distance between the approximate and exact distributions. With such a measure, we can characterize the properties of particular state-space abstraction methods (Section 6), and compare the effectiveness of alternative methods (Section 7).

We adopt a standard scoring rule to measure the divergence between two probability distributions. Let $\boldsymbol{e}$ be the observed values of a designated set of *evidence variables* $\boldsymbol{E}$. Let $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ denote the approximate probability distribution for $\boldsymbol{X}$ computed by approximating $\boldsymbol{A}$ given $\boldsymbol{E}$. The quality of the approximation is defined by the *Kullback score* [32]: [1]

$$K_{\boldsymbol{X}}^{\boldsymbol{A}} = \sum_{\boldsymbol{X}}\Pr(\boldsymbol{x}|\boldsymbol{e})\ln\frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})}, \tag{3}$$

where $\sum_{\boldsymbol{X}} h(\boldsymbol{x})$ means summing $h(\boldsymbol{x})$ over all possible values of $\boldsymbol{X}$. Notice that we do not explicitly show $\boldsymbol{E}$ in the notation for the Kullback score.

[1] This metric is also known as the Kullback–Leibler divergence, cross-entropy, or relative entropy.

The definition applies directly to any set of nodes $\boldsymbol{X}$ that does not include abstracted variables. For abstracted variables, we interpret the probability of the superstate as a uniform distribution over its constituent elementary state. That is, for an abstracted variable X

$$\widehat{\Pr}(x_k|\boldsymbol{e}) = \frac{\widehat{\Pr}([x_{i,j}]|\boldsymbol{e})}{j-i+1} \quad \text{for } i \leqslant k \leqslant j,$$

where $[x_{i,j}]$ is a superstate of X.

The range of the Kullback score is $[0,\infty]$, with lower Kullback scores corresponding to better approximations. The Kullback score is equal to zero if and only if the probability distributions being compared are exactly the same. Conventionally, the contribution of $\Pr(\boldsymbol{x}|\boldsymbol{e})\ln(\Pr(\boldsymbol{x}|\boldsymbol{e})/\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e}))$ is defined as zero whenever $\Pr(\boldsymbol{x}|\boldsymbol{e}) = 0$, so the only situation that could make Kullback scores infinite is that $\Pr(\boldsymbol{x}|\boldsymbol{e}) \neq 0$ and that $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e}) = 0$. We show in Appendix A that such a combination cannot occur when we apply the average policy in abstracting a Bayesian network. As a result, we do not have to consider the possibility of infinite Kullback scores. Since the Kullback scores are to be finite and non-negative in our settings, we assume that probability values considered are all positive in the following discussion to simplify presentation in the rest of this paper.

The Kullback score provides a theoretical upper bound on the absolute difference between the probability distributions being compared. Namely, for any state $\boldsymbol{x}$ of $\boldsymbol{X}$

$$|\Pr(\boldsymbol{x}|\boldsymbol{e}) - \widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})| \leqslant \sqrt{K_X^A/2}.$$

This bound, however, is usually much larger than the actual difference between the distributions.

The Kullback score is not a symmetric measure for comparing the divergence between two probability distributions. We will obtain a different value if we switch the roles of $\Pr(\boldsymbol{x}|\boldsymbol{e})$ and $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ in (3). The following measure, proposed by Jeffreys [24], is similar to the Kullback score, but it is symmetric. We touch upon this symmetric measure in Section 6.

$$J_X^A = \sum_X \Pr(\boldsymbol{x}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})} + \sum_X \widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e}) \ln \frac{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})}{\Pr(\boldsymbol{x}|\boldsymbol{e})} \tag{4}$$

5. Iterative state-space abstraction

One approach to tackle problems in time-constrained applications is to arrange the reasoning procedure so that it produces progressively more accurate results as more computation time is allocated [2,21]. The reasoning

procedures can be terminated any time, and the procedures simply return the most recent solution when terminated. In this section, we apply the state-space abstraction methods to design such *anytime algorithms* for evaluation of Bayesian networks. We present the algorithm and discuss the outcomes from an experimental study of its performance.

5.1. The algorithm

The design of this iterative abstraction algorithm is motivated by the chart shown in Fig. 3. Given the original Bayesian network (OBN), we may start the computation with a very abstract Bayesian network (ABN), and gradually refine the ABN for better approximations when time permits. The iterative state-space abstraction (ISSA) algorithm takes as input the OBN and query to compute the desired probability distribution. A query consists of the observed states of some nodes and the variables whose probability distributions are of interest. We call the nodes whose states are known *evidence nodes* and the nodes whose distributions are of interest *queried nodes*.

Algorithm (*The Iterative State-Space Abstraction Algorithm* (*OBN*, *query*)).
1. Construct an initial ABN with one superstate per abstracted node.
2. Evaluate the current ABN to obtain an approximation of the desired probability distribution.
3. If all states for all abstracted nodes are elementary, return.
4. Refine a selected superstate to construct a new ABN.
5. Go to Step 2.

The algorithm begins with the construction of a very abstract network where the abstracted nodes have only one superstate. The states of evidence nodes are not aggregated since this would obviously introduce unnecessary error into the approximate solutions.

The current ABN is then evaluated with an exact evaluation algorithm. In principle, we can use any exact evaluation algorithm at step 2. In this study, we use the junction-tree algorithm [27], which is by far the most popular exact evaluation algorithm for Bayesian networks.

Having computed an approximation, the algorithm can return a solution whenever the inference procedure needs to terminate. If, at step 3, all the superstates are refined to the finest grain, then the algorithm has evaluated the OBN and should just return the exact answer.

Refining a superstate is to recover a distinction among the states that are aggregated in the superstate. More specifically, refining a superstate $[a_{i,j}]$ of A is to replace $[a_{i,j}]$ with a pair of new states $[a_{i,k}]$ and $[a_{k+1,j}]$ in the state space of A, where $k = \lfloor (i+j)/2 \rfloor$. Conditional probability distributions related to these new states are translated from the OBN using the average policy. The refine-

ment operation introduces more states in the new ABN, and potentially improves the quality of the approximations. After constructing the new ABN, the algorithm returns to step 2.

The algorithm runs in this iterative fashion until it cannot continue. When there is no time for further computation, the algorithm terminates and returns the latest approximation obtained at step 2. The algorithm may also stop at the fourth step as just described.

The strategy used in deciding how the OBN should be approximated has a decisive influence on the performance of the ISSA algorithm. There are several degrees of freedom in designing the algorithm. We need to choose the nodes whose states are to be aggregated, and we need to decide how to aggregate the states of these selected nodes. When refining a network, we need to decide which superstate should be refined and how to refine it. Dividing a superstate into two new states that consist of almost the same number of elementary states is a convenient and easy-to-implement strategy, but might not provide the best performance possible. One may also wonder whether the average policy is a good choice for the probability reassignment task. We discuss these issues in Section 7.

5.2. Empirical study

We implement a baseline version of the ISSA algorithm for empirical study of its performance. The baseline ISSA algorithm aggregates the states of all nodes except those of the evidence nodes at step 1. Abstracted nodes in this initial ABN have only one state. At step 4 of the algorithm, the most probable superstate of each abstracted node is refined, where the most probable superstate is that with largest marginal probability among the superstates of an abstracted node.

The quality of the approximate probability distributions is measured by the Kullback score, with the exact probabilities calculated separately directly from the OBN. We have tested this baseline ISSA algorithm on several networks [54], including the network shown in Fig. 2.

Data reported in this section were collected from the experiment that used the network shown in Fig. 2 with 16 states per node. The goal was to compute $\Pr(t\_3|o\_1)$, the marginal distribution of target variable T_3 given the state of observable variable O_1 for all t_3. We conducted three sets of experiments Test1, Test2, and Test3, where the network instances differed in the skewness of probability distributions underlying the networks.

We focus on this distinction between test cases because skewness has a direct effect on quality of abstraction, and because skewed distributions are common in practical applications [9,44]. We assign the conditional probability distributions $\Pr(x|\boldsymbol{p}(X))$ using the following procedure. The assignment of $\Pr(x|\boldsymbol{p}(X))$ for different states of the parents of X is carried out indepen-

dently. The values of α are 0, 1, and 1.16 for Test1, Test2, and Test3, respectively.

1. Sample p_1 from the uniform distribution $U(0,1)$.
2. Sample q_i from $U(0,1)$ and assign $p_i = \alpha p_{i-1} + q_i$ for $i = 2,3,\ldots,n$.
3. Normalize the sequence $\{p_i\}$.
4. Let $\Pr(x_i|\boldsymbol{p}(X)) = p_i$ for $i = 1,2,3,\ldots,n$.

The ISSA algorithm was tested on 200 instances in each set of experiment. For each, the average Kullback score of $K^A_{T\_3}$ was plotted as a function of time, where each time point corresponds to a distinct iteration of the ISSA algorithm. The first point in each series represents the initial ABN, with one superstate per node. In this initial situation, the approximation is simply the uniform distribution, which serves as a baseline for comparison of the quality of approximations.

As we can see in the charts, the approximations improve monotonically, converging to the exact distribution when the refinement reaches the elementary states. The OBN had 16 states per node, so the curves have 16 points, each showing the average of $K^A_{T\_3}$ for iterations 1–16. Evaluation of these networks at full granularity takes of the order of 21 s, which is roughly equivalent to 11 abstraction iterations in the experiments. The approximations have a very small Kullback score (0.008 for Test3) at this point, with the extra advantage of having produced good approximations even earlier.

Note that the time per iteration increases substantially as we proceed. Since the proportion of time spent on early iterations becomes negligible, there is relatively little advantage in estimating the maximum granularity solvable in a given time and proceeding right to the level. Moreover, the earlier iterations determine the refinement pattern (i.e., which superstates to refine); uniform refinement at a pre-identified granularity would not be as accurate.

Test1 is a favorable case for the ISSA algorithm, as both the average policy and our interpretation of the probability of superstates make use of uniformity. Indeed, in this model the Kullback score is extremely low even before any refinement (0.00004)! Even starting from this level, however, the approximations improve steadily with refinement.

Distributions used in Test2 and Test3 are skewed, and these more skewed distributions are initially approximated much worse by the coarse-grained networks. However, the quality of the approximations for the more skewed distributions improves substantially in the first few iterations, and, within a few iterations, the quality of approximations for all cases becomes quite good. The parameters used in Test3 were the most skewed. For the case $n = 16$ in the experiments, p_{16} is at least nine times p_1, so the distributions are really skewed. The improvement with refinement in this case was more substantial, reaching a much better fit in just a few iterations (see Fig. 4).

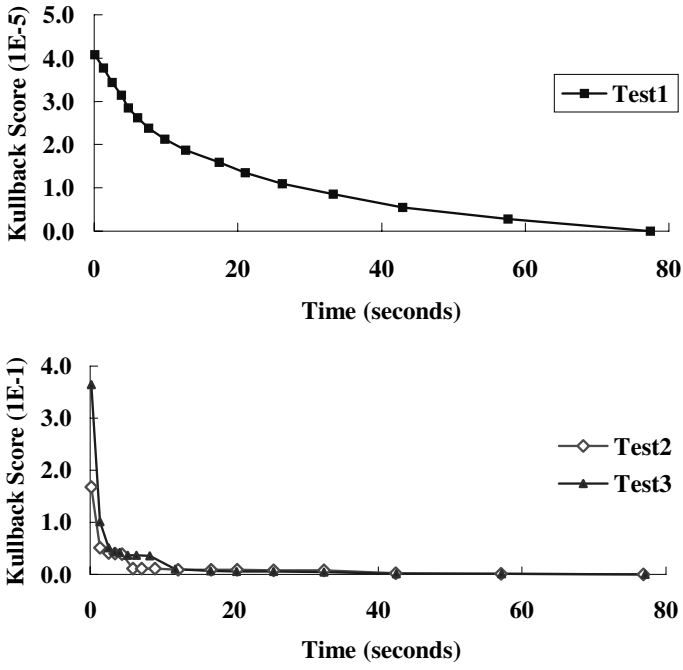


Fig. 4. ISSA returns approximate solutions that converge to exact solutions.

6. Properties of approximations

Given an approximate Bayesian network, can we tell what variables are affected by the approximation operations? Can we say anything about the quality of the approximations for those affected variables? Theorems reported in this section answer these questions by taking advantage of the conditional independence relationship among nodes. In the theorem statements, $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{Y})$ means that $\boldsymbol{X}$ is conditionally independent of $\boldsymbol{Y}$ given that $\boldsymbol{E} = \boldsymbol{e}$, and $CI(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{Y})$ means that $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{Y})$ holds for *all* instances $\boldsymbol{e}$ of $\boldsymbol{E}$. We continue to use the Kullback scores in presenting the theorems. However, we can show that the theorems presented in this section will hold if we had chosen the symmetric J_X^A, defined in (4), as the measure for quality of approximations.

To state our results, we require a bit more Bayesian network terminology. Let $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ be three disjoint subsets of nodes in a Bayesian network. $\boldsymbol{Z}$ is said to *d-separate* $\boldsymbol{X}$ from $\boldsymbol{Y}$ if there is no undirected path between a node in $\boldsymbol{X}$ and a node in $\boldsymbol{Y}$ along which the following two conditions hold: (1) every node with converging arrows is in $\boldsymbol{Z}$ or has a child node in $\boldsymbol{Z}$, and (2) every other node is outside $\boldsymbol{Z}$ [42]. Nodes that have no children and are neither evidence

nodes nor queried nodes are *barren* [49]. We generalize the definition a bit by considering nodes with only barren children as barren themselves.

Theorem 1 specifies those conditional distributions that cannot be affected by the approximation of $\boldsymbol{A}$. Roughly speaking, if nodes in $\boldsymbol{A}$ are irrelevant to the computation of $\Pr(\boldsymbol{x}|\boldsymbol{e})$, then approximating $\boldsymbol{A}$ should not have any impact on the probability. Notice that there might be some other conditional distributions that are not affected by approximation operations due to numerical coincidence.

Theorem 1. *The conditional distribution* $\Pr(\boldsymbol{x}|\boldsymbol{e})$ *is not affected by the abstraction of* $\boldsymbol{A}$ *if either of the following conditions holds. Namely,* $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e}) = \Pr(\boldsymbol{x}|\boldsymbol{e})$ *if*
1. $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{A})$, *or*
2. *nodes in* $\boldsymbol{A}$ *are barren for the computation of* $\Pr(\boldsymbol{x}|\boldsymbol{e})$.

Proof. All proofs for theorems are given in Appendix A.

Notice that d-separation is a stronger version of conditional independence. As a result, Theorem 1 and other theorems discussed in this section are readily applicable to situations where we replace the conditional independence conditions with their d-separation counterparts. When $\boldsymbol{E}$ d-separates $\boldsymbol{X}$ from $\boldsymbol{Y}$, $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ for any instance of $\boldsymbol{E}$, namely $CI(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{Y})$. Since we can easily determine whether a d-separation relation holds by inspecting the graphical structure, this interpretation provides an economical way of applying the theorems without conducting potentially intensive computation for determining the existence of conditional independence.

The following theorem asserts that the quality of approximations improves with increasing distance of the nodes from $\boldsymbol{A}$, where distance is defined by conditional independence relationships among variables.

Theorem 2. *Let* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *be two sets of nodes and* $\boldsymbol{A}$ *the abstracted nodes. If* $\boldsymbol{X} \cap \boldsymbol{Y} = \emptyset$, $(\boldsymbol{X} \cup \boldsymbol{Y}) \cap \boldsymbol{A} = \emptyset$, *and* $CI(\boldsymbol{Y}, \boldsymbol{X} \cup \boldsymbol{e}, \boldsymbol{A})$, *then* $K_{\boldsymbol{X}}^{\boldsymbol{A}} \geqslant K_{\boldsymbol{Y}}^{\boldsymbol{A}}$.

We may apply Theorem 2 to predict the relative quality of approximations in Bayesian networks that comprise duplications of subnetworks such as dynamic Bayesian networks [17] and temporal influence diagrams [46]. In these networks, it is common that variables representing state of the world at time stage T_i d-separate variables for T_{i+1} from variables for T_j. For prediction tasks where we have evidence about the current state of the world, if we only abstract variables for T_j, $j < i$, the approximations for variables for the distant future T_{i+1} must have better quality than those for variables for the near future T_i, according to the theorem. For diagnosis tasks, the reasoning is similar, and the approximations for variables representing the distant past are better than those for variables of the near past.

In addition, this theorem sheds light on the design of heuristics for controlling the approximation operations to maximize quality of approximations. We discuss this application in the following section.

The following theorem and its corollary specify properties of the quality of approximations for two conditionally independent sets of nodes. The Kullback score of the union of two independent sets of nodes is the sum of their individual Kullback scores. Such an additive property helps to reduce the computational cost of estimating the Kullback score of a large set of nodes, which is useful when we use estimated Kullback scores as heuristics for controlling how to approximate Bayesian networks.

Theorem 3. *Let* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *be two sets of nodes and* $\boldsymbol{A}$ *the abstracted nodes. If* $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{Y})$ *then* $K^{\boldsymbol{A}}_{\boldsymbol{X}\cup\boldsymbol{Y}} = K^{\boldsymbol{A}}_{\boldsymbol{X}} + K^{\boldsymbol{A}}_{\boldsymbol{Y}}$.

When two independent variables jointly d-separate a third from the approximated nodes, Corollary 1 dictates that the quality of the approximations for the two independent variables is no better than the quality of approximation for the third variable.

Corollary 1. *For three sets of nodes,* $\boldsymbol{X}$, $\boldsymbol{Y}$, *and* $\boldsymbol{Z}$, *if* $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{Y})$, $(\boldsymbol{X} \cup \boldsymbol{Y}) \cap \boldsymbol{Z} = \emptyset$, $(\boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{Z}) \cap \boldsymbol{A} = \emptyset$ *and* $CI(\boldsymbol{Z}, \boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{e}, \boldsymbol{A})$, *then* $K^{\boldsymbol{A}}_{\boldsymbol{X}} + K^{\boldsymbol{A}}_{\boldsymbol{Y}} \geqslant K^{\boldsymbol{A}}_{\boldsymbol{Z}}$.

Properties of approximated probability distributions are discovered elsewhere. Kjærulff approximately evaluates Bayesian networks by removing weak dependences in the junction trees of the given Bayesian networks [29]. He reports that the quality of approximation for the clique potential functions improves with the increasing distance from the clique from which weak links are removed. When we carry out the state-space abstraction in a Bayesian network, the potential functions of the cliques that contain the abstracted nodes will be affected. The deviation between the original and the approximate probability information decreases with increasing distance, where the distance is defined based on d-separation in Bayesian networks and on path distance in junction trees.

Theorem 2 has a close relationship with the property of decreasing relative entropy in Markov chains [5]. Let π_n and π'_n represent two probability distributions of a Markov chain at time n. The property states that the relative entropy between π_n and π'_n is larger than that between π_{n+1} and π'_{n+1}, where π_{n+1} and π'_{n+1} are, respectively, computed from π_n and π'_n, along with the transition probability from n to $n+1$ of the Markov chain. We can relate this property of Markov chains to Theorem 2, when we interpret $\boldsymbol{X}$ and $\boldsymbol{Y}$ as sets of variables representing the states of the Markov chain at time n and $n+1$. The condition $CI(\boldsymbol{Y}, \boldsymbol{X} \cup \boldsymbol{e}, \boldsymbol{A})$ in Theorem 2 guarantees that the transition probability from $\boldsymbol{X}$

to $\boldsymbol{Y}$ is preserved after abstraction of $\boldsymbol{A}$. Therefore, by virtue of the property, K_X^A must be larger than K_Y^A.

7. Algorithmic variations

An understanding of the relative quality of approximations may bear on the design issues raised in Section 5.1. Refining a selected superstate will increase the time necessary to evaluate the new network, and is expected to improve the quality of approximations. We would prefer to refine the superstate such that we increase evaluation time as little as possible, and improve quality as much as possible.

The increase in computation time will depend on the exact methods used in ISSA. It is not difficult to compute how much more computation would be needed to evaluate the refined network, and we can readily apply the methods discussed in [29] for the junction-tree methods. Therefore we will focus on the effects of refining superstates on improving the quality of approximations in the following discussion.

To understand how refining superstates would affect quality of approximations, we seek approaches to two key problems. The first problem is how to assign conditional probabilities once some states are selected for aggregation, and the second is how to determine which distinctions between states are least relevant for answering the query.

We confine ourselves to solutions that apply only information that is locally available. As we explain below, locality is defined in terms of the Markov boundary of the abstracted nodes. It is possible that we could obtain better approximations if we employ more information in deciding how to approximate the OBNs. However, doing so would also increase the computational costs for each iteration of ISSA, and would potentially reduce the usefulness of the approximate solutions. By employing only local information, we attempt to keep down computational costs for deciding how to approximate the OBNs.

7.1. Probability assignment policies

Conceivably, there are many ways to assign the conditional probabilities related to superstates. Among all choices, policies that employ more information in determining the new probability values should provide better approximate solutions in general.

We consider Eq. (1) discussed in Section 3.2 a standard method for assigning the probability values in the CPTs of the abstracted node. The conditional probability of a superstate is the sum of the conditional probability of its constituent states.

The average policy employs Eq. (2) for assigning the probability values in the CPTs of the children of the abstracted node. A closer examination shows that the average policy uses probability values that are available in the CPTs of the abstracted nodes and their children. We repeat Eq. (2) for easier comparison below

$$\widehat{\Pr}(y|[a_{i,j}], \boldsymbol{px}(Y)) = \frac{1}{j-i+1} \sum_{k=i}^{j} \Pr(y|a_k, \boldsymbol{px}(Y)).$$

Chang and Fung [3] suggest Eq. (1) for assigning the CPT of the abstracted nodes, and the following formula for assigning the CPTs of the children of the abstracted nodes, where $|\boldsymbol{p}(A)|$ denotes the number of possible states of the parents of node A. Henceforth, we refer to this method as the *CF policy*

$$\widehat{\Pr}(y|[a_{i,j}], \boldsymbol{px}(Y)) = \frac{1}{|\boldsymbol{p}(A)|} \sum_{\boldsymbol{P}(A)} \frac{\sum_{k=i}^{j} \Pr(y|a_k, \boldsymbol{px}(Y)) \Pr(a_k|\boldsymbol{p}(A))}{\widehat{\Pr}([a_{i,j}]|\boldsymbol{p}(A))}. \tag{5}$$

The CF policy uses only local information, and weights the importance of all possible states of the parents of the abstracted nodes uniformly. In contrast, the average policy uniformly weights the constituent states aggregated in the superstate in (2). Comparison between Eqs. (2) and (5) shows that the CF policy is computationally more expensive than the average policy.

We have compared the performance of implementations of ISSA with the average and CF policies in experiments [34,54]. As we have expected, empirical results indicate that using the CF policy leads to a better average performance of ISSA.

The choice of probability assignment policy also affects what types of approximations we would compute. The average and CF policies are devised for computing point-valued approximations. We have reported another probability assignment policy for computing bounds of probability distributions [36].

7.2. Control heuristics for ISSA

We explore heuristics for determining the relevance of the distinction of states to the accuracy of the approximate solution. We apply the properties of approximations discussed in Section 6 to search for heuristics that we can compute locally.

Theorem 2 provides a foundation for our approach to determining the degree of relevance of states. When we have a chance to refine the state space of an abstracted node, we are essentially changing from an abstraction to another of the abstracted nodes. Theorem 2 dictates that, as long as $CI(\boldsymbol{Q}, \boldsymbol{X} \cup \boldsymbol{e}, \boldsymbol{A})$ holds, we will have $K^{\boldsymbol{A}}_{\boldsymbol{X}} \geqslant K^{\boldsymbol{A}}_{\boldsymbol{Q}}$ for any abstraction of $\boldsymbol{A}$. Therefore, minimizing $K^{\boldsymbol{A}}_{\boldsymbol{X}}$ when we refine the state space of abstracted nodes will provide a good prospect of reducing $K^{\boldsymbol{A}}_{\boldsymbol{Q}}$.

The choice of such an $\boldsymbol{X}$ is arbitrary. However, we may prefer to have the smallest one among alternatives when the algorithm computes the heuristics at runtime under time constraint. Computing $K_{\boldsymbol{X}}^{\boldsymbol{A}}$ for a larger $\boldsymbol{X}$ will be computationally more expensive than computing the score for a smaller $\boldsymbol{X}$. The Markov boundary of $\boldsymbol{A}$, denoted $\boldsymbol{MB}(\boldsymbol{A})$, is such a minimal set. $\boldsymbol{MB}(\boldsymbol{A})$ is the minimal set of nodes such that $CI(\boldsymbol{V} \setminus \boldsymbol{MB}(\boldsymbol{A}) \setminus \boldsymbol{A}, \boldsymbol{MB}(\boldsymbol{A}), \boldsymbol{A})$ holds, that is, no proper subset of $\boldsymbol{MB}(\boldsymbol{A})$ exists such that the conditional independence holds [42]. The Markov boundary of a node Y consists of the parents, children, and parents of children of Y.

The following theorem provides a starting point for design and selection of control heuristics, where $K_{\boldsymbol{U}}^{\boldsymbol{A}}$ is the Kullback score of all unabstracted variables, $\boldsymbol{V} \setminus \boldsymbol{A}$.

Theorem 4. *Let* $\boldsymbol{U} = \boldsymbol{V} \setminus \boldsymbol{A}$

$$K_{\boldsymbol{U}}^{\boldsymbol{A}} = \sum_{\boldsymbol{MB}(\boldsymbol{A})} \Pr(\boldsymbol{mb}(\boldsymbol{A})) \ln \frac{g(\boldsymbol{A})}{\hat{g}(\boldsymbol{A})}, \tag{6}$$

where

$$g(\boldsymbol{A}) \equiv \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}) \cup \boldsymbol{A}} \Pr(v^i | \boldsymbol{p}(V^i)) \quad \textit{and}$$

$$\hat{g}(\boldsymbol{A}) \equiv \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}) \cup \boldsymbol{A}} \widehat{\Pr}(v^i | \boldsymbol{p}(V^i)).$$

This theorem tells us that we could make $K_{\boldsymbol{U}}^{\boldsymbol{A}}$ equal to zero by preserving the values of $g(\boldsymbol{A})$ for all possible values of $\boldsymbol{A}$. Keeping $K_{\boldsymbol{U}}^{\boldsymbol{A}} = 0$ means that we would have preserved the joint distribution of all unabstracted nodes. As a result, any conditional distribution $\Pr(\boldsymbol{q}|\boldsymbol{e})$ for any $\boldsymbol{Q} \subset \boldsymbol{U}$ and any $\boldsymbol{E} \subset \boldsymbol{U}$ would also be preserved. However, when discussing techniques for approximating one variable, Chang and Fung [3] have reported that it is generally impossible to compute exact probability of interest *and* to save the cost of computing the probability by using the approximate network. Theorem 4 shows why it is difficult, if ever possible, to preserve the joint distribution of $\boldsymbol{U}$ while abstracting a set of nodes $\boldsymbol{A}$.

Consider the simplest case in which we approximate only one node, Y. [2] In this case, $K_{\boldsymbol{U}}^{Y}$ is equal to

$$\sum_{\boldsymbol{MB}(Y)} \Pr(\boldsymbol{mb}(Y)) \ln \frac{g(Y)}{\hat{g}(Y)},$$

[2] For clarity, we assume that only one node is approximated in much of the following discussion.

and $g(Y)$ is equal to $\Pr(\boldsymbol{c}(Y)|\boldsymbol{p}(Y),\boldsymbol{p}(\boldsymbol{C}(Y)))$. Since it is difficult to preserve $g(Y)$, what we would like to do is to minimize K_U^Y to maximize the quality of the approximations for $\boldsymbol{U}$. To this end, we need to have methods to compute K_U^Y at run time to minimize it.

Computing K_U^Y exactly is a costly task. Split the Markov boundary of Y into two parts: (1) the children of Y and (2) the parents of Y and the parents of children of Y, and denote $\boldsymbol{P}(Y) \cup \boldsymbol{P}(\boldsymbol{C}(Y))$ by $\boldsymbol{PC}(Y)$. We rewrite K_U^Y as follows:

$$K_U^Y = \sum_{\boldsymbol{PC}(Y)} \Pr(\boldsymbol{pc}(Y)) \sum_{\boldsymbol{C}(Y)} g(Y) \ln \frac{g(Y)}{\hat{g}(Y)}.$$

The inner summation can be calculated locally using the CPTs of $\boldsymbol{MB}(Y)$. The quantity $\Pr(\boldsymbol{pc}(Y))$, however, is not included in the specification of the Bayesian networks and requires exact evaluation of the Bayesian networks to obtain its value. Consequently, it is generally impractical to compute K_U^Y exactly at run time.

One way to estimate K_U^Y is to assume that $\Pr(\boldsymbol{pc}(Y))$ is equiprobable for all possible instances of $\boldsymbol{PC}(Y)$. If $\boldsymbol{PC}(Y)$ has 16 states, we simply set $\Pr(\boldsymbol{pc}(Y))$ to 1/16 for each of these 16 states. We define this heuristic function as the REMB score for the approximated node, where REMB stands for relative entropy of the Markov boundary. Namely, we define

$$\mathrm{REMB}(Y) = \frac{\sum_{\boldsymbol{MB}(Y)} g(Y) \ln(g(Y)/\hat{g}(Y))}{|\boldsymbol{P}(Y)| \cdot |\boldsymbol{P}(\boldsymbol{C}(Y))|}, \tag{7}$$

where $|\boldsymbol{J}|$ denotes the number of possible states of a set $\boldsymbol{J}$.

Notice that the derivation of the REMB heuristic aims at maximizing the quality of the approximations for $\Pr(\boldsymbol{q}|\boldsymbol{e})$ for *any* $\boldsymbol{Q} \subset \boldsymbol{U}$ and *any* $\boldsymbol{E} \subset \boldsymbol{U}$. As a result, the REMB heuristic may not perform well for a specific combination of $\boldsymbol{Q}$ and $\boldsymbol{E}$.

For a specific $\boldsymbol{Q}$, we may use heuristics that attempt to maximize the quality of the approximations for a $\boldsymbol{W}$, where $\boldsymbol{W}$ is the minimal subset of $\boldsymbol{MB}(\boldsymbol{A})$ such that $CI(\boldsymbol{Q}, \boldsymbol{W} \cup \boldsymbol{e}, \boldsymbol{A})$ holds. This could lead to a simpler heuristic function than $\mathrm{REMB}(\boldsymbol{A})$.

Now that we have heuristic control functions that help us to improve the quality of approximate solutions of $\Pr(\boldsymbol{q}|\boldsymbol{e})$ for *any* $\boldsymbol{Q} \subset \boldsymbol{U}$ and *any* $\boldsymbol{E} \subset \boldsymbol{U}$. We attempt to find heuristics tuned for a particular $\boldsymbol{E}$. The following theorem provides a perspective on this problem.

Theorem 5. *Let* $\boldsymbol{U} = \boldsymbol{V} \setminus \boldsymbol{A} \setminus \boldsymbol{E}$

$$K_U^A = \ln \frac{\widehat{\Pr}(\boldsymbol{e})}{\Pr(\boldsymbol{e})} + \sum_{\boldsymbol{MB}(\boldsymbol{A}) \setminus \boldsymbol{E}} \Pr(\boldsymbol{mb}(\boldsymbol{A})|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A}\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A}\|\boldsymbol{e})}, \tag{8}$$

where $g(\boldsymbol{A}\|\boldsymbol{e})$ *and* $\hat{g}(\boldsymbol{A}\|\boldsymbol{e})$ *are, respectively, defined the same as* $g(\boldsymbol{A})$ *and* $\hat{g}(\boldsymbol{A})$ *in Theorem* 4, *except that we need to set the states of variables involved in calculating* $g(\boldsymbol{A}\|\boldsymbol{e})$ *and* $\hat{g}(\boldsymbol{A}\|\boldsymbol{e})$ *according to the evidence* $\boldsymbol{e}$ *when* $\boldsymbol{E} \cap (\boldsymbol{C}(\boldsymbol{A}) \cup \boldsymbol{P}(\boldsymbol{C}(\boldsymbol{A}))) \neq \emptyset$.

Theorem 5 is a generalized version of Theorem 4 in that we consider the existence of evidence nodes. $K_U^{\boldsymbol{A}}$ defined in Theorem 4 is just an approximation of the second term of the right-hand side of (8). It would be preferable to take the first term into consideration when we design control heuristics for ISSA. The first term, however, contains $\Pr(\boldsymbol{e})$ and $\widehat{\Pr}(\boldsymbol{e})$, and is very computationally costly to compute. Ignoring the first term, we can derive $\mathrm{REMB}(\boldsymbol{A})$ as we did for 7, and use the function in ISSA. However, as we see in the experimental study, the performance delivered by this control heuristic function depends on whether the marginal probability $\Pr(\boldsymbol{e})$ is preserved.

The following corollary identifies a condition under which the Kullback score $K_U^{\boldsymbol{A}}$ is decomposable, thereby making $\mathrm{REMB}(\boldsymbol{A})$ decomposable and easier to compute when possible. In addition, Corollary 2 is useful for selecting the part of the approximate network that should be refined. Assume that we have $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ that conform to the conditions specified in the corollary, and that refining either $\boldsymbol{A}_1$ or $\boldsymbol{A}_2$ will lead to the same amount of reduction in $K_U^{\boldsymbol{A}_1}$ and $K_U^{\boldsymbol{A}_2}$. In this case, we should refine the alternative that leads to the smallest increase in the computational cost of inference, if we are to refine only one set from $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$.

Corollary 2. *Let* $\boldsymbol{A} = \boldsymbol{A}_1 \cup \boldsymbol{A}_2$ *be the abstracted nodes. If* $\boldsymbol{MB}(\boldsymbol{A}_1) \cap \boldsymbol{MB}(\boldsymbol{A}_2) = \emptyset$, *then the Kullback score of the nodes* $\boldsymbol{U} = \boldsymbol{V} \setminus \boldsymbol{A} \setminus \boldsymbol{E}$ *is additive when* $\boldsymbol{E} = \emptyset$ *or the probability* $\Pr(\boldsymbol{e})$ *is preserved in the approximate networks. Namely,* $K_U^{\boldsymbol{A}} = K_U^{\boldsymbol{A}_1} + K_U^{\boldsymbol{A}_2}$.

7.3. Experimental comparison

To compare the effectiveness of some selected heuristics, we implemented the ISSA algorithm with different control heuristics. We ran the experiments with the network shown in Fig. 5, and let the algorithm run without interrupting its execution. All nodes in the network had four states, except that A had 16 states. We assigned the parameters of the probability distributions

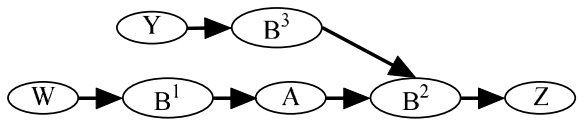


Fig. 5. A Bayesian network for comparing control heuristics of ISSA.

with the procedure discussed in Section 5, and we set $\alpha = 0$ in the procedure. We abstracted A, and computed the approximations for specific conditional probability distributions in three experiments. In the first iteration, A had two superstates, and in each following iteration, a superstate was selected and split into two other states. Heuristics were used to select the superstate to split. Since A had two superstates initially and a new state was introduced in each iteration, we had 15 iterations in the experiments. Notice that the network was chosen such that we did not consider variables W, Y, and Z when we computed REMB(A).

We plot the average normalized Kullback scores against the iterations in the ISSA algorithm. The Kullback scores are normalized by dividing the score obtained in each iteration by the score obtained in the first iteration. Typically, we got the highest Kullback scores in the first iterations, so we decide to use results from the first iterations as reference for comparing speed of improvement in the Kullback scores. In addition, we collected data from 1000 tests that differed only in the parameters chosen for the probability distributions of the Bayesian networks, so each point on the curves shows the averages of normalized scores collected from 1000 tests.

We include the RAND and BEST curves for a comparison purpose. The RAND curves were achieved by a method that randomly picks a superstate to split. Hence, we expect that a useful control heuristic should deliver better performance than this random method. The BEST curves illustrate a gold standard achieved by a myopic method that compares the results of *all* alternatives of refining the abstracted variable and *always* picks the alternative that *actually* maximizes the quality of the approximation in the next iteration. Notice that the BEST curves do not necessarily have the best performance in all iterations due to the existence of local optima.

We also compare the REMB heuristic with the VAR and SIM heuristics. Using the REMB heuristic, we select the superstate with the lowest REMB score. The REMB score, REMB($[a_{i,j}]$) of a superstate $[a_{i,j}]$, is set to the REMB(A) defined in (7) assuming that $[a_{i,j}]$ is chosen to split and is already split. The VAR heuristic selects the superstate with the highest VAR score defined as follows:

$$\mathrm{VAR}([a_{i,j}]) = \frac{\sum_{k=i,j} \sum_{l=1,|B^3|} V(B^2|a_k, b_l^3)}{(j-i+1)|B^3|}, \tag{9}$$

where $V(B^2|a_i, b_l^3)$ denotes the variance of the probability distribution of B^2 given a_i and b_l^3. The VAR score thus represents the average variance of probability distributions related to a superstate. If $V(B^2|a_i, b_l^3)$ and $V(B^2|a_{i+1}, b_l^3)$ are large, it may be intuitively preferable not to aggregate states a_i and a_{i+1} to keep the impact on the distribution of B^2 as small as possible. Therefore, we choose to refine the superstate with the highest VAR score.

The SIM heuristic selects the superstate that has the highest similarity score defined as

$$\mathrm{SIM}([a_{i,j}]) = \frac{\sum_{k=i,j} \sum_{l=1,|B^3|} S(a_k, b_l^3)}{(j-i+1)|B^3|}, \tag{10}$$

where

$$S(a_k, b_l^3) = \sum_{r=1}^{|B^2|} \Pr(b_r^2|a_k, b_l^3) \ln \frac{\Pr(b_r^2|a_k, b_l^3)}{\widehat{\Pr}(b_r^2|a_k, b_l^3)} \tag{11}$$

and

$$\widehat{\Pr}(b_r^2|a_k, b_l^3) = \sum_{k=i}^{j} \Pr(b_r^2|a_k, b_l^3) \Big/ (j-i+1). \tag{12}$$

Notice that formula (11) is a Kullback score, and that formula (12) is an incarnation of Eq. (2) for the network shown in Fig. 5. Therefore, we can interpret formula (11) as a measure of how the assigned distribution, $\widehat{\Pr}(b^2|a_k, b_l^3)$, is similar to the actual conditional distributions, $\Pr(b^2|a_k, b_l^3)$, for a particular combination of a_k and b_l^3. We can also interpret the formula as a measure of how the actual conditional distributions are similar to each other, using their average as the standard for comparison. This SIM score is thus a measure of similarity between the probability distributions $\Pr(b_r^2|a_k, b_l^3)$ for all a_k and b_l^3. In the extreme, if all these distributions are the same, this score will be zero and it should be fine to leave the superstate as aggregated. Therefore, using this control heuristic, we choose refine the superstate with the highest SIM score.

The REMB heuristic outperforms the SIM and VAR heuristics most of the time as indicated in the charts of Fig. 6, and the SIM and VAR heuristics perform better than the method that randomly selects superstates to split. One might have expected the REMB heuristic to perform better than SIM and VAR since it uses more information in reaching a selection decision. In the network used in the experiments, we use $\Pr(a|b^1)$ and $\Pr(b^2|a, b^3)$ in the computation of the REMB score, while we use only $\Pr(b^2|a, b^3)$ in the computation of the SIM and VAR scores. The other difference is that the REMB score is essentially a projected score in that we compute the scores by assuming that the superstate being considered is split already.

The REMB curves also support our interpretation of Theorems 4 and 5. When $\boldsymbol{E} = \emptyset$ as in the first chart, the REMB heuristic performs rather well, but, when $\boldsymbol{E} \neq \emptyset$ as in the other charts, the REMB heuristic may or may not perform well. In the second experiment, the marginal probability of the evidence node, $\Pr(w)$, is preserved in the abstracted network according to Theorem 1. In the third experiment, the marginal probability of the evidence node, $\Pr(z)$, may

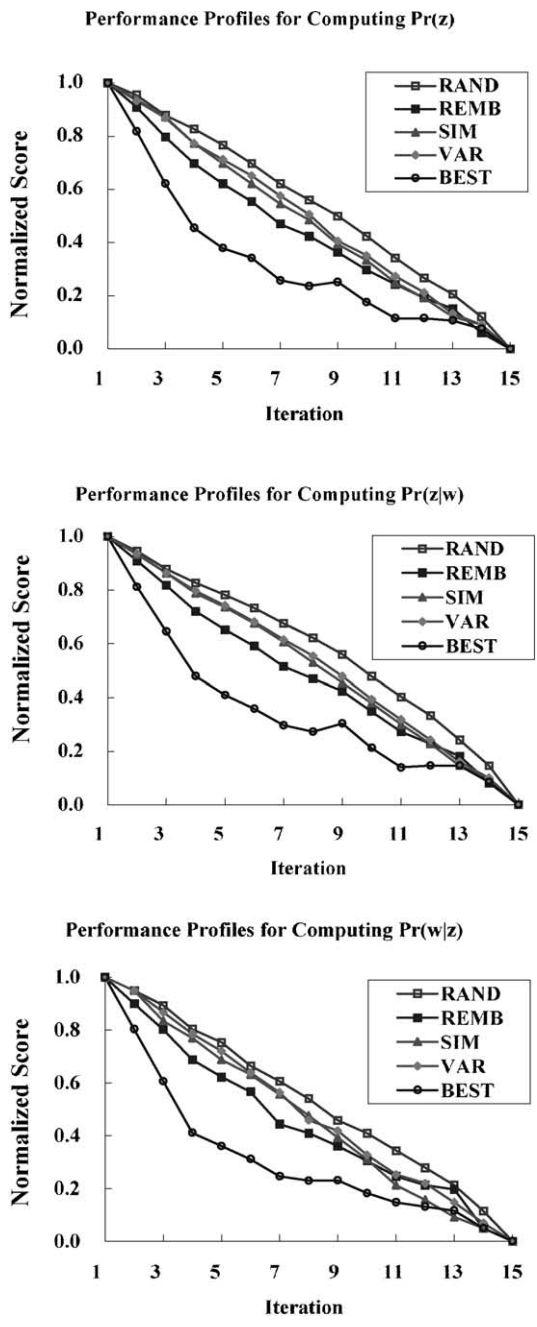


Fig. 6. Comparison of performance profiles.

not be preserved. Therefore, the REMB heuristic is more likely to be a good heuristic in computing $\Pr(z|w)$ than in computing $\Pr(w|z)$, and this is supported by the observation that the REMB curve shifts closer to the RAND curve in the third experiment.

8. Conclusions

We present abstraction methods for an approximate evaluation of the Bayesian networks. Our methods are distinguished in that we attempt to temporarily aggregate states of variables while computing approximations. By extremely reducing the number of states of variables, we dramatically reduce the computation time necessary to obtain approximate solutions. Applying this idea, we propose an iterative state-space abstraction algorithm for anytime evaluation of Bayesian networks. We discover some useful properties of the approximations which then form the basis of our searching for control heuristics for improving the performance profiles of the baseline ISSA algorithm. We identify the so-called REMB heuristic for selecting the superstate to refine. The REMB heuristic performs better than other simple intuitive heuristics. The state-space abstraction methods have been applied to resolving ambiguous qualitative relationships in probabilistic qualitative networks [35] and computing bounds of probability distributions [36,37].

Three important factors influence the computational costs of evaluating Bayesian networks: The number of variables, the connectivity among these variables, and the number of states in these variables. Localized partial evaluation carries out the approximation by temporarily ignoring selected variables [16], and link removal algorithms temporarily remove weak dependences in Bayesian networks [53] and junction trees [29]. Our methods and those proposed by Kozlov and colleague approximate Bayesian networks by reducing the number of states of variables. Kozlov and Singh aggregate states that are *similar* in BN2O networks [9], and they define similarity based on linear dependence between conditional probability distributions [31]. Kozlov and Koller work on the dynamic discretization of continuous variables in Bayesian networks [30]. Their method partitions the ranges of continuous variables based on information collected outside of the Markov boundary.

A common challenge to designing approximate evaluation algorithms for Bayesian networks is how to determine relevancy between the probability distributions of interest and parameters that specify the networks. For state-space abstraction methods, we would like to recover distinction between original states that is most relevant to the probability distribution being computed. We tackle this problem with the REMB score function to estimate the relevance in ISSA. Computation of REMB scores uses information that is

available in the Markov boundary of the abstracted nodes. Therefore it is cheaper to compute REMB scores at run time than computing score functions that consider information that is not local to the abstracted nodes. The iterative nature of ISSA also provides a chance for the algorithm itself to incrementally tune the abstraction of abstracted nodes to maximizing quality of approximations, producing the anytime property as a by-product.

Acknowledgements

This work was supported in part by Grant NSC-89-2213-E-004-007 from the National Science Council of Taiwan, and in part by Grant F49620-94-1-0027 from the Air Force Office of Scientific Research of the United States.

Appendix A. Proof for Section 4: Measuring quality of approximations

The following theorem states that the approximated probability should not be zero unless the exact probability is zero. We show the theorem with the guidance of two observations. First, we do not introduce extra zeros into the computation process when we apply the average policy and when we interpret the approximated probability. Also, we assign zeros to new conditional probability distributions in the ABN only if the original condition probability values with which we compute the new conditional probability are all zeros. Therefore, intuitively, the theorem should hold.

Theorem 6. *Let* $\boldsymbol{Q}$ *and* $\boldsymbol{E}$ *be the queried and evidence nodes, respectively. As a result of our using the average policy in assigning probability values and our interpreting approximated probability uniformly, the approximated probability cannot be zero unless its corresponding exact probability is zero, namely*:

$$\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \;\Rightarrow\; \Pr(\boldsymbol{q}|\boldsymbol{e}) = 0.$$

Proof. Let $\boldsymbol{V}$ denote the set of nodes $\{X^1, X^2, \ldots, X^n\}$ in a Bayesian network. Without loss of generality, we assume that only node X^k is abstracted. The proof consists of the following steps:

1. We show that, roughly speaking, the probability in the original joint distribution must be zero if the probability in the approximated joint distribution is zero

$$\widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \;\Rightarrow\; \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0$$
$$\text{for all } i^k \in [r, s].$$

2. We show that for any unabstracted node X in the network

$$\sum_{X \text{ in ABN}} \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \Rightarrow \sum_{X \text{ in OBN}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0 \quad \text{for all } i^k \in [r,s].$$

3. We prove that for any abstracted node X^k in the network

$$\sum_{X^k \text{ in ABN}} \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \Rightarrow \sum_{X^k \text{ in OBN}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0.$$

4. Finally, we show the theorem using results from previous steps.
Step 1: In the ABN, we have

$$\widehat{\Pr}(\boldsymbol{v}) = \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = \prod_{i=1}^{n} \widehat{\Pr}(x^i | \boldsymbol{p}(X^i)).$$

Since X^k is the sole abstracted node, only the conditional probability tables of X^k and its successors are modified when we aggregate the states of X^k. Namely, $\widehat{\Pr}(x^i|\boldsymbol{p}(X^i)) = \Pr(x^i|\boldsymbol{p}(X^i))$ for all X^i that is neither X^k nor a successor of X^k. For simplicity of notation, I use $\boldsymbol{S}(X^k)$ and $\boldsymbol{R}(X^k)$ to denote $\mathrm{SUCC}(X^k) \cup X^k$ and $\boldsymbol{V} - \boldsymbol{S}(X^k)$, respectively, in the following derivation. Notice that $\boldsymbol{S}(X^k) \cup \boldsymbol{R}(X^k) = \boldsymbol{V}$

$$\begin{aligned}
\widehat{\Pr}(\boldsymbol{v}) &= \prod_{i=1}^{n} \widehat{\Pr}(x^i|\boldsymbol{p}(X^i)) \\
&= \left(\prod_{i=1, X^i \in \boldsymbol{R}(X^k)}^{n} \Pr(x^i|\boldsymbol{p}(X^i)) \right) \left(\prod_{i=1, X^i \in \boldsymbol{S}(X^k)}^{n} \widehat{\Pr}(x^i|\boldsymbol{p}(X^i)) \right) \\
&= C_k \prod_{i=1, X^i \in \boldsymbol{S}(X^k)}^{n} \widehat{\Pr}(x^i|\boldsymbol{p}(X^i))
\end{aligned}$$

Notice that $\prod_{i=1, X^i \in \boldsymbol{R}(X^k)}^{n} \Pr(x^i|\boldsymbol{p}(X^i))$, denoted C_k, is a common factor of $\widehat{\Pr}(\boldsymbol{v})$ and $\Pr(\boldsymbol{v})$. The state-space abstraction operations do not change this value. Thus, if $C_k = 0$, then $\widehat{\Pr}(\boldsymbol{v}) = 0$ and $\Pr(\boldsymbol{v}) = 0$ are true, making the implication $\widehat{\Pr}(\boldsymbol{v}) = 0 \Rightarrow \Pr(\boldsymbol{v}) = 0$ true at the same time.

If $\widehat{\Pr}(\boldsymbol{v}) = 0$ but $C_k \neq 0$, the factor $\prod_{i=1, X^i \in \boldsymbol{S}(X^k)}^{n} \widehat{\Pr}(x^i|\boldsymbol{p}(X^i))$ must be 0. Clear, if a product is zero, then at least one of the terms involved in the product must be zero. There are two possibilities for this situation to occur: This term must come from either the abstracted node or the children of the abstracted node.

Consider the case that $\widehat{\Pr}([x_{r,s}^k]|\boldsymbol{p}(X^k)) = 0$. Recall that we set $\widehat{\Pr}([x_{r,s}^k]|\boldsymbol{p}(X^k))$ to $\sum_{i=r}^{s} \Pr(x_i^k|\boldsymbol{p}(X^k))$. Since the only occasion that a sequence of non-negative

numbers can add up to 0 is that all the numbers are zero, we have that for all $i \in [r,s]$, $\Pr(x_i^k|\boldsymbol{p}(X^k))$ must be zero when $\widehat{\Pr}([x_{r,s}^k]|\boldsymbol{p}(X^k)) = 0$. As a result, $\Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n)$ must be 0 for $i^k \in [r,s]$ in this case.

Consider the case that $\widehat{\Pr}(y|[x_{r,s}^k], \boldsymbol{px}(Y)) = 0$ for a successor Y of X^k. Since we set

$$\widehat{\Pr}(y|[x_{r,s}^k], \boldsymbol{px}(Y)) = \frac{\sum_{i=r}^{s} \Pr(y|x_i^k, \boldsymbol{px}(Y))}{s-t+1}$$

in the average policy, it must be true that $\Pr(y|x_i^k, \boldsymbol{px}(Y)) = 0$ for all $i \in [r,s]$, when $\widehat{\Pr}(y|[x_{r,s}^k], \boldsymbol{px}(Y)) = 0$. As a result, $\Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0$ holds for all $i \in [r,s]$ again, and we have proved that

$$\widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \;\Rightarrow\; \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0 \quad \text{for all } i^k \in [r,s].$$

Step 2: Assume that $X = X^j$ is unabstracted node with m states. We derive that

$$\begin{aligned}
&\sum_{X^j \text{ in ABN}} \widehat{\Pr}(\boldsymbol{v}) = 0 \\
&\Rightarrow \sum_{l=1}^{m} \widehat{\Pr}(x_{i^1}^1, \ldots, x_l^j, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \\
&\Rightarrow \widehat{\Pr}(x_{i^1}^1, \ldots, x_l^j, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \quad \text{for all } l \in [1,m] \\
&\quad \text{Apply the result from step 1.} \\
&\Rightarrow \Pr(x_{i^1}^1, \ldots, x_l^j, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0, \quad \text{for all } l \in [1,m],\ i^k \in [r,s].
\end{aligned}$$

Therefore, when X is unabstracted, we have that

$$\sum_{X \text{ in ABN}} \widehat{\Pr}(\boldsymbol{v}) = 0 \;\Rightarrow\; \sum_{X \text{ in OBN}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0 \quad \text{for all } i^k \in [r,s].$$

Step 3: Assume that X^k has t states in the OBN

$$\begin{aligned}
&\sum_{X^k \text{ in ABN}} \widehat{\Pr}(\boldsymbol{v}) = 0 \\
&\Rightarrow \sum_{X^k} \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \\
&\Rightarrow \forall [x_{r,s}^k],\ \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \\
&\quad \text{Apply the result from step 1} \\
&\Rightarrow \text{for all } i^k \in [1,t],\ \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0 \\
&\Rightarrow \sum_{X^k \text{ in OBN}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0.
\end{aligned}$$

Step 4: We have

$$\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \;\Rightarrow\; \frac{\widehat{\Pr}(\boldsymbol{q}\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{e})} = 0 \;\Rightarrow\; \widehat{\Pr}(\boldsymbol{q}\boldsymbol{e}) = 0 \;\Rightarrow\; \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \widehat{\Pr}(v) = 0.$$

We continue the proof by considering two cases: $X^k \in \boldsymbol{Q}$ and $X^k \notin \boldsymbol{Q}$. Consider the first case. This is equivalent to assuming that X^k is not in $\boldsymbol{V} - \boldsymbol{Q} - \boldsymbol{E}$. Namely, the summation $\sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \widehat{\Pr}(v)$ sums out a set of unabstracted nodes, so we may apply the result from step 2 and conclude that:

$$\begin{aligned}
&\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \\
&\quad\Rightarrow \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \widehat{\Pr}(x_{i^1}^1, \ldots, [x_{r,s}^k], \ldots, x_{i^n}^n) = 0 \\
&\qquad \text{Apply result from step 2 for unabstracted nodes} \\
&\overset{X^k \in \boldsymbol{Q}}{\Rightarrow} \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) = 0 \quad \text{for all } i^k \in [r,s] \\
&\quad\Rightarrow \Pr(\boldsymbol{q}', x_{i^k}^k, \boldsymbol{e}) = 0 \quad \text{for all } i^k \in [r,s], \text{ where } \boldsymbol{Q}' = \boldsymbol{Q} - \{X^k\} \\
&\quad\Rightarrow \Pr(\boldsymbol{q}|\boldsymbol{e}) = 0 \quad \text{for } i^k \in [r,s].
\end{aligned}$$

Therefore, when $X^k \in \boldsymbol{Q}$ and $\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0$, the corresponding $\Pr(\boldsymbol{q}|\boldsymbol{e}) = 0$ must also be zero.

When $X^k \notin \boldsymbol{Q}$, the summation $\sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \widehat{\Pr}(v)$ sums out a set of unabstracted nodes and the abstracted node X^k. (Recall that we do not abstract evidence nodes, so $X^k \notin \boldsymbol{E}$.) In this case, we apply the results from step 3, and conclude that $\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \;\Rightarrow\; \Pr(\boldsymbol{q}|\boldsymbol{e}) = 0$. The detail follows:

$$\begin{aligned}
&\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \\
&\quad\Rightarrow \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}} \widehat{\Pr}(v) = 0 \\
&\overset{X^k \notin \boldsymbol{Q}}{\Rightarrow} \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}-X^k} \left(\sum_{X^k} \widehat{\Pr}(v) \right) = 0 \\
&\qquad \text{Each } \sum_{X^k \text{ in ABN}} \widehat{\Pr}(v) \text{ within the large parentheses must be zero.} \\
&\qquad \text{Apply the result from step 3 for abstracted nodes.} \\
&\quad\Rightarrow \sum_{\boldsymbol{V}-\boldsymbol{Q}-\boldsymbol{E}-X^k} \left(\sum_{X^k \text{ in OBN}} \Pr(x_{i^1}^1, \ldots, x_{i^k}^k, \ldots, x_{i^n}^n) \right) = 0 \\
&\quad\Rightarrow \Pr(\boldsymbol{q}, \boldsymbol{e}) = 0 \\
&\quad\Rightarrow \Pr(\boldsymbol{q}|\boldsymbol{e}) = 0.
\end{aligned}$$

In both cases, we have shown that $\widehat{\Pr}(\boldsymbol{q}|\boldsymbol{e}) = 0 \Rightarrow \Pr(\boldsymbol{q}|\boldsymbol{e}) = 0$, and this concludes the proof. □

A.1. Proof for Theorem 1

Theorem 1. *The conditional marginal distribution* $\Pr(\boldsymbol{x}|\boldsymbol{e})$ *of* $\boldsymbol{X}$ *is not affected by the state-space abstraction of the abstracted node* $\boldsymbol{A}$, *if either of the following conditions holds. Namely, if*

1. $CI(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{A})$, *or*
2. *Nodes in* $\boldsymbol{A}$ *are barren nodes with respect to the computation of* $\Pr(\boldsymbol{x}|\boldsymbol{e})$ *in the OBN*;

then $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e}) = \Pr(\boldsymbol{x}|\boldsymbol{e})$

Proof. *Condition* 1: When we coarsen the state space of the abstracted nodes, we modify the conditional probability distributions $\Pr(\boldsymbol{a}|\boldsymbol{p}(\boldsymbol{A}))$ and $\Pr(\boldsymbol{c}(A)|\boldsymbol{a}, \boldsymbol{p}(\boldsymbol{C}(\boldsymbol{A})))$. If the probability $\Pr(\boldsymbol{x}|\boldsymbol{e})$ does not depend on values in these conditional probability tables, then $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ must be equal to $\Pr(\boldsymbol{x}|\boldsymbol{e})$.

Consider the case of $\Pr(\boldsymbol{a}|\boldsymbol{p}(\boldsymbol{A}))$. In computing the probability $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$, we may remove nodes $\boldsymbol{A}$ from the network without affecting the probability $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$, given the conditional independence between $\boldsymbol{X}$ and $\boldsymbol{A}$. Therefore, the modification of conditional probability table of $\boldsymbol{A}$ does not affect $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ since the existence of the nodes in $\boldsymbol{A}$ is irrelevant to the computation of $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ in the first place.

Let Y be a successor of nodes in $\boldsymbol{A}$. If Y is not in $\boldsymbol{E}$, then $\boldsymbol{X}$ must be conditionally independent of Y, given the conditional independence between $\boldsymbol{X}$ and $\boldsymbol{A}$. Hence, the modification of $\Pr(y|\boldsymbol{a}, \boldsymbol{p}(Y))$ will not affect $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$. The other situation is that Y is in $\boldsymbol{E}$, in addition to that $\boldsymbol{X}$ is conditionally independent of $\boldsymbol{A}$. In this case, the only way that Y can influence $\boldsymbol{X}$ is via successors of Y. Thus, the probability $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ is not affected by the state-space abstraction of $\boldsymbol{A}$ since the conditional probability tables of successors of Y are not modified at all.

Condition 2: If $\boldsymbol{A}$ are barren nodes with respective to the calculation of $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$, then we can remove nodes in $\boldsymbol{A}$ from the network without changing the joint probability distribution of the remaining nodes in the network [49]. As a result, $\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})$ must also be preserved and is equal to $\Pr(\boldsymbol{x}|\boldsymbol{e})$. □

A.2. Proof for Theorem 2

Theorem 2. *For two disjoint sets of nodes,* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$, *if* $(\boldsymbol{X} \cup \boldsymbol{Y}) \cap \boldsymbol{A} = \emptyset$, *and* $CI(\boldsymbol{Y}, \boldsymbol{X} \cup \boldsymbol{e}, \boldsymbol{A})$, *then* $K_{\boldsymbol{X}}^{\boldsymbol{A}} \geqslant K_{\boldsymbol{Y}}^{\boldsymbol{A}}$.

Proof. The Kullback score of $\boldsymbol{X}$ when we abstract $\boldsymbol{A}$, $K_{\boldsymbol{X}}^{\boldsymbol{A}}$, can be expanded as follows:

$$
\begin{aligned}
K_X^A &= \sum_X \Pr(\boldsymbol{x}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})} \\
&= \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})}, \quad \because \boldsymbol{X} \cap \boldsymbol{Y} = \emptyset \\
&= \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \ln \frac{\Pr(\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})\Pr(\boldsymbol{e})}.
\end{aligned}
$$

Similarly, we have

$$
K_Y^A = \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \ln \frac{\Pr(\boldsymbol{y},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})\Pr(\boldsymbol{e})}.
$$

Now, we show that $K_X^A \geqslant K_Y^A$ by showing $K_X^A - K_Y^A \geqslant 0$ as follows:

$$
\begin{aligned}
K_X^A - K_Y^A &= \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \ln \frac{\Pr(\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})\Pr(\boldsymbol{y},\boldsymbol{e})} \\
&\quad \because \ln(x) = -\ln(1/x) \geqslant 1 - 1/x \\
&\geqslant \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} \left(1 - \frac{\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})\Pr(\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})}\right) \\
&= \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} - \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})\Pr(\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})\Pr(\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})} \\
&\quad \because \sum_X \sum_Y \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} = 1 \quad \text{and} \quad \Pr(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e}) = \frac{\Pr(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{x},\boldsymbol{e})} \\
&= 1 - \sum_X \sum_Y \frac{\Pr(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})\Pr(\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})} \\
&\quad \because \widehat{\Pr}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e}) = \Pr(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e}) \text{ due to Theorem 1 and } \boldsymbol{X} \cap \boldsymbol{Y} = \emptyset \\
&= 1 - \sum_Y \frac{\Pr(\boldsymbol{y},\boldsymbol{e}) \sum_X \widehat{\Pr}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e})}{\Pr(\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y},\boldsymbol{e})} \\
&\quad \because \sum_X \widehat{\Pr}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{x},\boldsymbol{e}) = \widehat{\Pr}(\boldsymbol{y},\boldsymbol{e}) \quad \text{and} \quad \sum_Y \frac{\Pr(\boldsymbol{y},\boldsymbol{e})}{\Pr(\boldsymbol{e})} = 1 \\
&= 0. \qquad \square
\end{aligned}
$$

A.3. Proof for Theorem 3

Theorem 3. *Let* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *be two disjoint sets of nodes, and* $\boldsymbol{A}$ *be the set of abstracted nodes. If* $CI(\boldsymbol{X},\boldsymbol{e},\boldsymbol{Y})$ *then* $K_{\boldsymbol{X}\cup\boldsymbol{Y}}^{\boldsymbol{A}} = K_{\boldsymbol{X}}^{\boldsymbol{A}} + K_{\boldsymbol{Y}}^{\boldsymbol{A}}$.

Proof.

$$
\begin{aligned}
K^{A}_{X\cup Y} &= \sum_{X\cup Y} \Pr(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x},\boldsymbol{y})}{\widehat{\Pr}(\boldsymbol{x},\boldsymbol{y})} \\
&= \sum_{x}\sum_{y} \Pr(\boldsymbol{x}|\boldsymbol{e}) \Pr(\boldsymbol{y}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e}) \Pr(\boldsymbol{y}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})\widehat{\Pr}(\boldsymbol{y}|\boldsymbol{e})}, \quad \text{since } CI(\boldsymbol{X},\boldsymbol{e},\boldsymbol{Y}) \\
&= \sum_{x}\sum_{y} \Pr(\boldsymbol{x}|\boldsymbol{e}) \Pr(\boldsymbol{y}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{y}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{y}|\boldsymbol{e})} \\
&\quad + \sum_{x}\sum_{y} \Pr(\boldsymbol{x}|\boldsymbol{e}) \Pr(\boldsymbol{y}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})} \\
&= \sum_{x} \Pr(\boldsymbol{x}|\boldsymbol{e}) \sum_{y} \Pr(\boldsymbol{y}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{y}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{y}|\boldsymbol{e})} \\
&\quad + \sum_{y} \Pr(\boldsymbol{y}|\boldsymbol{e}) \sum_{x} \Pr(\boldsymbol{x}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{x}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{x}|\boldsymbol{e})} \\
&= K^{A}_{X} + K^{A}_{Y}, \quad \text{since } \sum_{Y} \Pr(\boldsymbol{y}|\boldsymbol{e}) = \sum_{X} \Pr(\boldsymbol{x}|\boldsymbol{e}) = 1
\end{aligned}
$$

If some of the nodes in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are abstracted, then we will divide $\widehat{\Pr}(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{e})$ by a constant that depends on how $\boldsymbol{X}$ and $\boldsymbol{Y}$ are abstracted [34]. The proof for this situation is almost identical to the proof shown above, with a slight adjustment to account for the uniform interpretation of approximate probability. □

A.4. Proof for Corollary 1

Corollary 1. *For three sets of nodes,* $\boldsymbol{X}$, $\boldsymbol{Y}$, *and* $\boldsymbol{Z}$, *if* $CI(\boldsymbol{X},\boldsymbol{e},\boldsymbol{Y})$, $(\boldsymbol{X}\cup\boldsymbol{Y})\cap\boldsymbol{Z}=\emptyset$, $(\boldsymbol{X}\cup\boldsymbol{Y}\cup\boldsymbol{Z})\cap\boldsymbol{A}=\emptyset$ *and* $CI(\boldsymbol{A},\boldsymbol{X}\cup\boldsymbol{Y}\cup\boldsymbol{e},\boldsymbol{Z})$, *then* $K^{A}_{X}+K^{A}_{Y}\geqslant K^{A}_{Z}$.

Proof. We have $K^{A}_{X\cup Y}\geqslant K^{A}_{Z}$ by Theorem 2 and $K^{A}_{X\cup Y}=K^{A}_{X}+K^{A}_{Y}$ by Theorem 3; therefore, we have $K^{A}_{X}+K^{A}_{Y}\geqslant K^{A}_{Z}$. □

A.5. Proofs for Theorems 4 and 5

We introduce a special structure, called *EABN*, that will be used in proving a lemma which in turn will be used in the final proofs for the theorems.

A.5.1. Equivalent networks of abstract Bayesian networks

The concept of equivalent networks of ABNs, denoted *EABN*, is a useful tool for comparing the probability distributions specified in the OBNs and ABNs. EABNs are useful for analyzing the control heuristics for ISSA.

An EABN of an ABN uses the state space of the OBN in specifying its probability distributions, and preserves the joint distribution of unabstracted nodes in the ABN. For any ABN with only one abstracted node, we may construct such an EABN using the following procedure.

A.5.2. EABN construction procedure

Assume that the abstracted node A^1 has α states in the OBN; that these α states are aggregated into β states, $[a^1_{s_1,t_1}], [a^1_{s_2,t_2}], \ldots, [a^1_{s_\beta,t_\beta}]$, in the ABN; that A^1 has γ children, $Y^1, Y^2, \ldots, Y^\gamma$. We can construct an EABN that recovers the state space of A^1 and preserves the joint distribution of the nodes in the ABN, excluding A^1. The procedure follows:

1. Copy all information specifying the ABN to the EABN, except the CPTs of A^1 and its children.
2. Let A^1 in the EABN inherit the states of A^1 in the OBN.
3. Set the CPTs of A^1 and its children in the EABN by the following formula when $l \in [s_i, t_i]$:

$$\widetilde{\mathbf{Pr}}(a^1_l|\boldsymbol{p}(X)) = \Pr(a^1_l|\boldsymbol{p}(X)) \tag{13}$$

and

$$\widetilde{\mathbf{Pr}}(y^j|a^1_l, \boldsymbol{px}(Y^j)) = \widehat{\mathbf{Pr}}(y^j|[a^1_{s_i,t_i}], \boldsymbol{px}(Y^j)), \tag{14}$$

where $\Pr(a^1_l|\boldsymbol{p}(X))$ and $\widehat{\mathbf{Pr}}(y^j|[a^1_{s_i,t_i}], \boldsymbol{px}(Y^j))$ are values copied from the OBN and ABN, respectively.

Proof. I show the joint distributions of the unabstracted nodes, $\boldsymbol{V}' = \boldsymbol{V} \setminus \{A^1\}$, are equal in the ABN and its EABN as follows. Let $\boldsymbol{S} = \{A^1\} \cup \boldsymbol{C}(A^1)$ and $\widehat{\mathbf{Pr}}(\boldsymbol{v}')$ and $\widetilde{\mathbf{Pr}}(\boldsymbol{v}')$ be the probability distribution of $\boldsymbol{V}'$ in the ABN and EABN, respectively.

$$\widehat{\mathbf{Pr}}(\boldsymbol{v}') = \sum_{A^1} \prod_{i=1}^{n} \widehat{\mathbf{Pr}}(v^i|\boldsymbol{p}(V^i)) = \prod_{V^i \in \boldsymbol{V}\setminus\boldsymbol{S}} \Pr(v^i|\boldsymbol{p}(V^i)) \cdot \sum_{A^1} \prod_{V^i \in \boldsymbol{S}} \widehat{\mathbf{Pr}}(v^i|\boldsymbol{p}(V^i)).$$

The first factor is not a function of the state of A^1, so we denote the factor as κ and rewrite the equation

$$\begin{aligned}\widehat{\mathbf{Pr}}(\boldsymbol{v}') &= \kappa \sum_{A^1} \prod_{V^i \in \boldsymbol{C}(A^1)\cup\{A^1\}} \widehat{\mathbf{Pr}}(v^i|\boldsymbol{p}(V^i)) \\ &= \kappa \sum_{i=1}^{\beta} \widehat{\mathbf{Pr}}([a^1_{s_i,t_i}]|\boldsymbol{p}(A^1)) \prod_{j=1}^{\gamma} \widehat{\mathbf{Pr}}(y^j|[a^1_{s_i,t_i}], \boldsymbol{px}(Y^j))\end{aligned}$$

We rewrite the formula based on the fact that

$$\widehat{\Pr}([a^1_{s_i,t_i}]|\boldsymbol{p}(A^1)) = \sum_{l=s_i}^{t_i} \Pr(a^1_l|\boldsymbol{p}(A^1)) \text{ (average policy).}$$

$$= \kappa \sum_{i=1}^{\beta} \sum_{l=s_i}^{t_i} \Pr(a^1_l|\boldsymbol{p}(A^1)) \prod_{j=1}^{\gamma} \widehat{\Pr}(y^j|[a^1_{s_i,t_i}], \boldsymbol{px}(Y^j))$$

Since $s_1 = 1$ and $t_\beta = \alpha$,

we can rewrite the summation as follows.

$$= \kappa \sum_{l=1}^{\alpha} \Pr(a^1_l|\boldsymbol{p}(A^1)) \prod_{j=1}^{\gamma} h(y^j, l, \boldsymbol{px}(Y^j)),$$

where $h(y^j, l, \boldsymbol{px}(Y^j)) = \widehat{\Pr}(y^j|[a^1_{s_i,t_i}], \boldsymbol{px}(Y^j))$ if $l \in [s_i, t_i]$

$$= \kappa \sum_{l=1}^{\alpha} \widetilde{\Pr}(a^1_l|\boldsymbol{p}(A^1)) \prod_{j=1}^{\gamma} \widetilde{\Pr}(y^j|a^1_l, \boldsymbol{px}(Y^j))$$

$$= \widetilde{\Pr}(\boldsymbol{v}') \qquad \square$$

When multiple nodes are abstracted, we can iteratively apply a procedure that is similar to the procedure to construct the EABN. Assume that we abstract a set of m nodes $\boldsymbol{A}$, and that $[A^1, A^2, \ldots, A^m]$ is an ancestral ordering of $\boldsymbol{A}$ defined below.

Definition 1. [39]. Let $\boldsymbol{J} = \{J^1, \ldots, J^n\}$ be a set of nodes in a Bayesian network. $[J^1, J^2, \ldots, J^n]$ is an ancestral ordering of the nodes in $\boldsymbol{J}$ if for every $J^i \in \boldsymbol{J}$ all the ancestors of J^i are ordered before J^i.

We recover the state space of the abstracted nodes one node at a time in the ancestral ordering. Let EABN^1 denote the network we construct from the EABN construction after recovering the state space of A^1. We can recover the state space of A^2 by the procedure to construct another network EABN^2, using the conditional probability $\Pr(a^1_l|\boldsymbol{p}(X))$ of EABN^1 in the place of $\Pr(a^1_l|\boldsymbol{p}(X))$ in (13). We may prove that EABN^2 preserve the joint distribution of nodes $\boldsymbol{V} \setminus \{A^2\}$ in EABN^1 with the same method for proving the procedure. As a result, EABN^2 preserves the joint distribution of $\boldsymbol{V} \setminus \{A^1, A^2\}$ of the ABN, since EABN^1 preserves the joint distribution of $\boldsymbol{V} \setminus \{A^1\}$ of the ABN. By induction, we may continue the procedure to recover the state space of the remaining abstracted nodes, and the resulting EABN will preserve the joint distribution of $\boldsymbol{V} \setminus \boldsymbol{A}$ of the ABN.

The fact that the OBNs and EABNs use the same state space in specifying probability distributions provides a convenient way for comparison of probability distributions. We can refer to the conditional probability distribution of a child node Y given an instantiation a_i of its abstracted parent node A in the EABN, where a_i is a state aggregated in a superstate $[a_{j,k}]$ in the ABN. We cannot do so with an ABN because we have $\Pr(y|[a_{j,k}], \boldsymbol{px}(Y))$ rather than

$\Pr(y|a_i, \boldsymbol{px}(Y))$ in the ABN. Also, by comparing the CPTs of nodes in the OBN and EABN, we can easily find that the effects of abstracting a node are equivalent to modifying only the CPTs of the children of the abstracted node.

We now prove the following lemma that relates the joint distributions of the OBN and ABN with the idea of EABN. This lemma will be used in proving the theorems.

Lemma 1. *Let* $\boldsymbol{V}' = \boldsymbol{V} \setminus \boldsymbol{A}$, $\boldsymbol{E} \subset \boldsymbol{V}'$, $\boldsymbol{S} = \boldsymbol{C}(\boldsymbol{A}) \cup \boldsymbol{A}$. *Then*

$$\Pr(\boldsymbol{v}')/\widehat{\Pr}(\boldsymbol{v}') = g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})/\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e}),$$

where

$$g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e}) = \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{S}} \Pr(v^i|\boldsymbol{p}(V^i)), \quad \textit{and}$$

$$\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e}) = \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{S}} \widehat{\Pr}(v^i|\boldsymbol{p}(V^i)),$$

and $\widetilde{\Pr}(\cdot)$ *denote probability values specified in the EABN of the ABN. The double vertical bars used in* $g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})$ and $\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})$ *signify that we need to set the states of the variables in the calculation of* $g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})$ *and* $\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})$ *according to* $\boldsymbol{e}$. *This may be necessary when some children of* $\boldsymbol{A}$ *or some parent nodes of children of* $\boldsymbol{A}$ *are instantiated.*

Proof.

$$\widetilde{\Pr}(\boldsymbol{v}') = \sum_{\boldsymbol{A}} \prod_{i=1}^{n} \widetilde{\Pr}(v^i|\boldsymbol{p}(V^i)) = \prod_{V^i \in \boldsymbol{V}\setminus\boldsymbol{S}} \Pr(v^i|\boldsymbol{p}(V^i)) \cdot \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{S}} \widehat{\Pr}(v^i|\boldsymbol{p}(V^i))$$

The first factor does not depend on the value of $\boldsymbol{A}$ and is denoted as κ

$$\widetilde{\Pr}(\boldsymbol{v}') = \kappa \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{S}} \widehat{\Pr}(v^i|\boldsymbol{p}(V^i)).$$

Similarly, $\Pr(\boldsymbol{v}') = \kappa \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{S}} \Pr(v^i|\boldsymbol{p}(V^i))$. Now recall that $\widehat{\Pr}(\boldsymbol{v}') = \widetilde{\Pr}(\boldsymbol{v}')$, since the EABN preserves the joint distribution of unabstracted nodes. Divide $\Pr(\boldsymbol{v}')$ by $\widehat{\Pr}(\boldsymbol{v}')$ the lemma is proved. □

A.5.3. Proof for Theorem 4

This theorem is a special case of Theorem 5 when $\boldsymbol{E} = \emptyset$. □

A.5.4. Proof for Theorem 5

Theorem 5. *Let* $\boldsymbol{U} = \boldsymbol{V} \setminus \boldsymbol{A} \setminus \boldsymbol{E}$.

$$K_U^{\boldsymbol{A}} = \ln \frac{\widehat{\Pr}(\boldsymbol{e})}{\Pr(\boldsymbol{e})} + \sum_{\boldsymbol{MB}(\boldsymbol{A})\setminus\boldsymbol{E}} \Pr(\boldsymbol{mb}(\boldsymbol{A})|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}.$$

Proof. We expand K_U^A as follows:

$$\begin{aligned}
K_U^A &= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u}|\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u}|\boldsymbol{e})} \\
&= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})\widehat{\Pr}(\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})\Pr(\boldsymbol{e})} \\
&= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})} + \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\widehat{\Pr}(\boldsymbol{e})}{\Pr(\boldsymbol{e})} \\
&= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})} + \ln \frac{\widehat{\Pr}(\boldsymbol{e})}{\Pr(\boldsymbol{e})} \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \\
&\quad \text{Given that } \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) = 1, \\
&\quad \text{we rewrite the second term as follows.} \\
&= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})} + \ln \frac{\widehat{\Pr}(\boldsymbol{e})}{\Pr(\boldsymbol{e})} \qquad (15)
\end{aligned}$$

Furthermore, by Lemma 1 we just showed, we have

$$\sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})} = \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}.$$

We continue the derivation as follows:

$$\begin{aligned}
\sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{\Pr(\boldsymbol{u},\boldsymbol{e})}{\widehat{\Pr}(\boldsymbol{u},\boldsymbol{e})} &= \sum_U \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})} \\
&= \sum_{\boldsymbol{MB}(\boldsymbol{A})} \left(\sum_{\boldsymbol{U}\setminus\boldsymbol{MB}(\boldsymbol{A})} \Pr(\boldsymbol{u}|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})} \right) \\
&\quad \because \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})} \text{ is a function of only } \boldsymbol{MB}(\boldsymbol{A}) \\
&= \sum_{\boldsymbol{MB}(\boldsymbol{A})} \left(\ln \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})} \sum_{\boldsymbol{U}\setminus\boldsymbol{MB}(\boldsymbol{A})} \Pr(\boldsymbol{u}|\boldsymbol{e}) \right) \\
&= \sum_{\boldsymbol{MB}(\boldsymbol{A})} \Pr(\boldsymbol{mb}(\boldsymbol{A})|\boldsymbol{e}) \ln \frac{g(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})}{\hat{g}(\boldsymbol{A},\boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e})} \qquad (16)
\end{aligned}$$

Plug this result back to (15), and the proof is done. □

A.6. Proof for Corollary 2

Corollary 2. *Let* $\boldsymbol{A} = \boldsymbol{A}_1 \cup \boldsymbol{A}_2$ *be the approximated nodes. If* $\boldsymbol{MB}(\boldsymbol{A}_1) \cap \boldsymbol{MB}(\boldsymbol{A}_2) = \emptyset$, *then the Kullback score of the nodes* $\boldsymbol{U} = \boldsymbol{V} \setminus \boldsymbol{A} \setminus \boldsymbol{E}$ *is additive when* $\boldsymbol{E} = \emptyset$ *or the probability* $\Pr(\boldsymbol{e})$ *is preserved in the approximate networks. Namely,* $K_U^{\boldsymbol{A}} = K_U^{\boldsymbol{A}_1} + K_U^{\boldsymbol{A}_2}$.

Proof. Given $\boldsymbol{A} = \boldsymbol{A}_1 \cup \boldsymbol{A}_2$ and $\boldsymbol{MB}(\boldsymbol{A}_1) \cap \boldsymbol{MB}(\boldsymbol{A}_2) = \emptyset$, we can expand $g(\boldsymbol{A}\|\boldsymbol{e})$ as follows:

$$
\begin{aligned}
g(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e}) &= \sum_{\boldsymbol{A}} \prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}) \cup \boldsymbol{A}} \Pr(v^i|\boldsymbol{p}(V^i)) \\
&= \sum_{\boldsymbol{A}_1 \cup \boldsymbol{A}_2} \left(\left(\prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}_1) \cup \boldsymbol{A}_1} \Pr(v^i|\boldsymbol{p}(V^i)) \right) \right. \\
&\quad \left. \cdot \left(\prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}_2) \cup \boldsymbol{A}_2} \Pr(v^i|\boldsymbol{p}(V^i)) \right) \right) \\
&= \left(\sum_{\boldsymbol{A}_1} \prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}_1) \cup \boldsymbol{A}_1} \Pr(v^i|\boldsymbol{p}(V^i)) \right) \\
&\quad \cdot \left(\sum_{\boldsymbol{A}_2} \prod_{V^i \in \boldsymbol{C}(\boldsymbol{A}_2) \cup \boldsymbol{A}_2} \Pr(v^i|\boldsymbol{p}(V^i)) \right) \\
&= g(\boldsymbol{A}_1, \boldsymbol{mb}(\boldsymbol{A}_1)\|\boldsymbol{e}) g(\boldsymbol{A}_2, \boldsymbol{mb}(\boldsymbol{A}_2)\|\boldsymbol{e})
\end{aligned}
$$

Similarly, we can show that $\hat{g}(\boldsymbol{A}, \boldsymbol{mb}(\boldsymbol{A})\|\boldsymbol{e}) = \hat{g}(\boldsymbol{A}_1, \boldsymbol{mb}(\boldsymbol{A}_1)\|\boldsymbol{e}) \hat{g}(\boldsymbol{A}_2, \boldsymbol{mb}(\boldsymbol{A}_2)\|\boldsymbol{e})$. Given these two equalities, this corollary follows from Theorem 5. □

References

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

[2] M. Boddy, T.L. Dean, Deliberation scheduling for problem solving in time-constrained environments, Artificial Intelligence 67 (1994) 245–285.

[3] K.-C. Chang, R. Fung, Refinement and coarsening of Bayesian networks, Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence (1990) 475–482.

[4] G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artificial Intelligence 42 (1990) 393–405.

[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.

[6] P. Dagum, M. Luby, Approximating probabilistic inference in Bayesian belief networks is NP-hard, Artificial Intelligence 60 (1993) 141–153.

[7] P. Dagum, M. Luby, An optimal approximation algorithm for Bayesian inference, Artificial Intelligence 93 (1997) 1–27.

[8] B. D'Ambrosio, Incremental probabilistic inference, in: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, Washington, DC, 1993, pp. 301–308.
[9] B. D'Ambrosio, Symbolic probabilistic inference in large BN2O networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 128–135.
[10] A. Darwiche, G. Provan, Query DAGs: A practical paradigm for implementing belief-network inference, Journal of Artificial Intelligence Research 6 (1997) 147–176.
[11] R. Dechter, Bucket elimination: A unifying framework for probabilistic inference, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 1996, pp. 211–219.
[12] R. Dechter, Topological parameters for time-space tradeoff, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 1996, pp. 220–227.
[13] R. Dechter, Mini-buckets: A general scheme for generating approximations in automated reasoning, in: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 1997, pp. 1297–1302.
[14] A.L. Delcher, A.J. Grove, S. Kasif, J. Pearl, Logarithmic-time updates and queries in probabilistic networks, Journal of Artificial Intelligence Research 4 (1996) 37–59.
[15] F.J. Díez, Local conditioning in Bayesian networks, Artificial Intelligence 87 (1996) 1–20.
[16] D.L. Draper, S. Hanks, Localized partial evaluation of belief networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 170–177.
[17] J. Forbes, T. Huang, K. Kanazawa, S. Russell, The BATmobile: Towards a Bayesian automated taxi, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1995, pp. 1878–1885.
[18] P. Haddawy, An overview of some recent developments in Bayesian problem-solving techniques, AI Magazine 20 (2) (1999) 11–19.
[19] D.E. Heckerman, A. Mamdani, M.P. Wellman, Real-world applications of Bayesian networks, Communications of the ACM 38 (3) (1995) 24–26.
[20] M. Henrion, Propagating uncertainty in Bayesian networks by probabilistic logic sampling, in: J.F. Lemmer, L.N. Kanal (Eds.), Uncertainty in Artificial Intelligence 2, Elsevier, Amsterdam, 1988, pp. 149–163.
[21] E. Horvitz, Computation and action under bounded resources, Ph.D. thesis, Stanford University, Stanford, California, USA, 1990.
[22] E. Horvitz, H.J. Suermondt, G.F. Cooper, Bounded conditioning: Flexible inference for decisions under scarce resources, in: Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence, 1989, pp. 182–193.
[23] T.S. Jaakkola, M.I. Jordan, Computing upper and lower bounds on likelihoods in intractable networks, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 1996, pp. 340–348.
[24] H. Jeffreys, Theory of Probability, Clarendon Press, Oxford, 1948.
[25] F. Jensen, S.K. Andersen, Approximations in Bayesian belief universes for knowledge-based systems, in: Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence, 1990, pp. 162–169.
[26] F.V. Jensen, An Introduction to Bayesian Networks, Springer, New York, 1996.
[27] F.V. Jensen, K.G. Olesen, An algebra of Bayesian belief universes for knowledge-based systems, Networks 20 (5) (1990) 637–660.
[28] N. Jitnah, A.E. Nicholson, A best-first search method for anytime evaluation of belief networks, in: S. Zilberstein, L. Hoebel (Eds.), Proceedings of the AAAI-97 Workshop on Building Resource-Bounded Reasoning Systems, WS-97-08, AAAI Press, 1997, pp. 69–73.
[29] U. Kjærulff, Reduction of computational complexity in Bayesian networks through removal of weak independences, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 374–382.

[30] A.V. Kozlov, D. Koller, Nonuniform dynamic discretization in hybrid networks, in: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 314–325.
[31] A.V. Kozlov, J.P. Singh, Computational complexity reduction for BN2O networks using similarity of states, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 1996, pp. 357–364.
[32] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1951) 76–86.
[33] V. Lepar, P.P. Shenoy, A comparison of Lauritzen–Spiegelhalter, Hugin, and Shenoy–Shafer architectures for computing marginals of probability distributions, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 328–337.
[34] C.-L. Liu, State-space abstraction methods for approximate evaluation of Bayesian networks, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, USA, 1998.
[35] C.-L. Liu, M.P. Wellman, Incremental tradeoff resolution in qualitative probabilistic networks, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 338–345.
[36] C.-L. Liu, M.P. Wellman, Using qualitative relationships for bounding probability distributions, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 346–353.
[37] C.-L. Liu, M.P. Wellman, Using stochastic-dominance relationships for bounding travel times in stochastic networks, in: Proceedings of the IEEE International Conference on Intelligent Transportation Systems, 1999, pp. 55–60.
[38] R.M. Neal, Probabilistic inference using Markov chain Monte Carlo methods, Ph.D. thesis, University of Toronto, Toronto, Ontario, Canada, 1993.
[39] R.E. Neapolitan, Probabilistic Reasoning in Expert Systems: Theory and Algorithms, Wiley, New York, 1990.
[40] A.E. Nicholson, N. Jitnah, Belief network algorithms: A study of performance using domain characterisation, in: G. Antoniou, A.K. Ghose, M. Truszczynski (Eds.), Learning and Reasoning with Complex Representations, Lecture Notes in Artificial Intelligence, vol. 1359, Springer, Berlin, 1998, pp. 169–188.
[41] J. Pearl, Evidential reasoning using stochastic simulation of causal models, Artificial Intelligence 32 (1987) 245–257.
[42] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, Los Altos, CA, 1988.
[43] M. Pittarelli, Anytime algorithms and deliberation scheduling, in: M. Pittarelli (Ed.), ACM SIGART Bulletin: Special Issue on Anytime Algorithms and Deliberation Scheduling, vol. 7(2), ACM, New York, 1996, p. 2.
[44] D. Poole, Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities, in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, 1993, pp. 606–612.
[45] D. Poole, Context-specific approximation in probabilistic inference, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 447–454.
[46] G.M. Provan, Tradeoffs in constructing and evaluating temporal influence diagrams, in: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, 1993, pp. 40–47.
[47] D. Roth, On the hardness of approximate reasoning, Artificial Intelligence 82 (1996) 273–302.
[48] J.E. Santos, On linear potential functions for approximating Bayesian computations, Journal of the ACM 43 (3) (1996) 399–430.
[49] R.D. Shachter, Probabilistic inference and influence diagrams, Operation Research 36 (4) (1988) 589–604.
[50] P.P. Shenoy, G. Shafer, Axioms for probability and belief-function propagation, in: R.D. Shachter, T. Levitt, J. Lemmer, L. Kanal (Eds.), Uncertainty in Artificial Intelligence 4, North-Holland, Amsterdam, 1990, pp. 169–198.

[51] S.E. Shimony, Finding MAPs for belief networks is NP-hard, Artificial Intelligence 68 (1994) 399–410.
[52] H.J. Suermondt, G.F. Cooper, Probabilistic inference in multiply connected belief networks using loop cutsets, International Journal of Approximate Reasoning 4 (1990) 283–306.
[53] R.A. van Engelen, Approximating Bayesian belief networks by arc removal, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (8) (1997) 916–920.
[54] M.P. Wellman, C.-L. Liu, State-space abstraction for anytime evaluation of probabilistic networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 567–574.
[55] N.L. Zhang, D. Poole, Exploiting causal independence in Bayesian network inference, Journal of Artificial Intelligence Research 5 (1996) 301–328.