Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

Hierarchical Beam Search for Solving Most Relevant Explanation in Bayesian Networks

Xiaoyuan Zhu and **Changhe Yuan**
Queens College, City University of New York
65-30 Kissena Blvd., Queens, NY 11367
{xiaoyuan.zhu, changhe.yuan}@qc.cuny.edu

Abstract

Most Relevant Explanation (MRE) is an inference problem in Bayesian networks that finds the most relevant partial instantiation of target variables as an explanation for given evidence. It has been shown in recent literature that it addresses the overspecification problem of existing methods, such as MPE and MAP. In this paper, we propose a novel hierarchical beam search algorithm for solving MRE. The main idea is to use a second-level beam to limit the number of successors generated by the same parent so as to limit the similarity between the solutions in the first-level beam and result in a more *diversified* population. Three pruning criteria are also introduced to achieve further diversity. Empirical results show that the new algorithm typically outperforms local search and regular beam search.

Introduction

Bayesian networks represent the conditional independences between random variables as directed acyclic graphs and provide standard approaches for solving inference problems, such as Most Probable Explanation (MPE), Maximum a Posterior (MAP), and Most Relevant Explanation (MRE). MPE (Pearl 1988) is the problem of finding the most likely instantiation of a set of target variables given the remaining variables as evidence. While in an MAP (Pearl 1988) problem, there are auxiliary variables other then targets and evidence variables. Both MPE and MAP are solved by optimizing the joint posterior probability of the target variables given the evidence. As methods for explaining evidence (Nielsen, Pellet, and Elisseeff 2008; Lacave and Díez 2002), however, MPE and MAP often produce overspecified explanations (Pacer et al. 2013; Yuan et al. 2009), i.e., irrelevant target variables may also be included. This is because the criterion of maximizing a joint probability cannot exclude irrelevant target variables directly.

MRE (Yuan and Lu 2007; Yuan et al. 2009) is an inference method developed to address the limitations of MPE and MAP. The key idea of MRE is to find a partial instantiation of the target variables that maximizes the Generalized Bayes Factor (GBF) (Fitelson 2007; Good 1985) as an explanation for the evidence. GBF is a rational function of probabilities that is suitable for comparing explanations with different cardinalities. Theoretically, MRE is shown able to prune away independent and less relevant variables from the final explanation. The theoretical results have also been confirmed by a recent human study (Pacer et al. 2013). Because of the enlarged search space, however, MRE is shown to be extremely difficult to solve (Yuan, Lim, and Littman 2011). Existing algorithms for solving MRE are mainly based on local search and Markov chain Monte Carlo methods (Yuan et al. 2009).

In this paper, we propose an efficient *hierarchical beam search* algorithm for MRE inference in Bayesian networks. The key idea is to use two levels of beams to increase the *diversity* of the solution population under consideration. The first-level beam is used to limit the search to the most promising solutions, similar to the regular beam search (Rubin and Reddy 1977). The second-level beams are newly introduced to limit the number of successors generated by a current solution to prevent over-reproduction of similar offsprings. Three pruning criteria based on the theoretical properties of MRE are also introduced to achieve further diversity as well as efficiency. We applied the new algorithm to solve MRE problems in a set of benchmark diagnostic Bayesian networks. Empirical results show that the new algorithm typically outperforms local search and regular beam search.

Background

A Bayesian network (Pearl 1988; Koller and Friedman 2009) is represented as a Directed Acyclic Graph (DAG). The nodes in the DAG represent random variables. The lack of arcs in DAG define conditional independence relations among the nodes. If there is an arc from node Y to X, i.e., $Y \rightarrow X$, we say that Y is a parent of X, and X is a child of Y (We use upper-case letters to denote variables X or variable sets $\mathbf{X}$, and lower-case letters for values of scalars x or vectors $\mathbf{x}$.). A variable X is conditionally independent of its non-descendants given its parent set PA_X, which can be quantified by the conditional distribution $p(X|PA_X)$. The Bayesian network as a whole represents the joint probability distribution of $\prod_X p(X|PA_X)$.

MRE (Yuan et al. 2009) has been developed to find a partial instantiation of the target variables as an explanation for given evidence in a Bayesian network. Here, *explana-*

tion refers to the explanation of evidence, whose goal is to explain why some observed variables are in their particular states using the target variables in the domain. The complexity of MRE is conjectured to NP^{PP} complete (Yuan, Lim, and Littman 2011). Formally, the MRE problem is defined as follows.

Definition 1. *Let* $\mathbf{M}$ *be a set of targets, and* $\mathbf{e}$ *be the given evidence in a Bayesian network. Most Relevant Explanation is the problem of finding a partial instantiation* $\mathbf{x}$ *of* $\mathbf{M}$ *that has the maximum generalized Bayes factor score* $GBF(\mathbf{x};\mathbf{e})$ *as the explanation for* $\mathbf{e}$*, i.e.,*

$$MRE(\mathbf{M};\mathbf{e}) = \arg\max_{\mathbf{x},\emptyset\subset\mathbf{X}\subseteq\mathbf{M}} GBF(\mathbf{x};\mathbf{e}), \quad (1)$$

where GBF is defined as

$$GBF(\mathbf{x};\mathbf{e}) = \frac{p(\mathbf{e}|\mathbf{x})}{p(\mathbf{e}|\bar{\mathbf{x}})}. \quad (2)$$

In the above equations, $\mathbf{x}$ is an instantiation of $\mathbf{X}$. We use $\bar{\mathbf{x}}$ to represent all of the alternative explanations of $\mathbf{x}$. To further study the properties of generalized Bayes factor, we can reformulate GBF as follows.

$$\begin{aligned} GBF(\mathbf{x};\mathbf{e}) &= \frac{p(\mathbf{e}|\mathbf{x})}{p(\mathbf{e}|\bar{\mathbf{x}})} = \frac{p(\mathbf{x}|\mathbf{e})p(\bar{\mathbf{x}})}{p(\mathbf{x})p(\bar{\mathbf{x}}|\mathbf{e})} \\ &= \frac{p(\mathbf{x}|\mathbf{e})(1-p(\mathbf{x}))}{p(\mathbf{x})(1-p(\mathbf{x}|\mathbf{e}))}. \end{aligned} \quad (3)$$

Different from MPE and MAP which maximize the joint posterior probability function, MRE maximizes the rational probability function in Equation 3. This makes it possible for MRE to compare the explanations with different cardinalities. Several basic concepts related to GBF are useful for developing efficient pruning rules for MRE algorithms. The Conditional Bayes Factor (CBF) (Yuan, Lim, and Lu 2011) is a measure of the quality of explanation $\mathbf{x}$ for given evidence $\mathbf{e}$ conditioned on explanation $\mathbf{y}$, and is formulated as follows.

$$CBF(\mathbf{x};\mathbf{e}|\mathbf{y}) = \frac{p(\mathbf{e}|\mathbf{x},\mathbf{y})}{p(\mathbf{e}|\bar{\mathbf{x}},\mathbf{y})} = \frac{p(\mathbf{x}|\mathbf{y},\mathbf{e})(1-p(\mathbf{x}|\mathbf{y}))}{p(\mathbf{x}|\mathbf{y})(1-p(\mathbf{x}|\mathbf{y},\mathbf{e}))}. \quad (4)$$

The belief update ratio of $\mathbf{x}$ given $\mathbf{e}$ is defined as follows.

$$r(\mathbf{x};\mathbf{e}) = \frac{p(\mathbf{x}|\mathbf{e})}{p(\mathbf{x})}. \quad (5)$$

From Equations 3 and 5, we can use belief update ratio to represent GBF as follows.

$$GBF(\mathbf{x};\mathbf{e}) = \frac{r(\mathbf{x};\mathbf{e})}{r(\bar{\mathbf{x}};\mathbf{e})}, \quad (6)$$

where $r(\bar{\mathbf{x}};\mathbf{e}) = \frac{1-p(\mathbf{x}|\mathbf{e})}{1-p(\mathbf{x})}$ is the belief update ratio of the alternative explanations $\bar{\mathbf{x}}$.

Local search based methods have been applied to solve MRE in Bayesian networks (Yuan, Lim, and Littman 2011), such as forward search and tabu search (Glover 1990). The forward search algorithm starts by first generating multiple initial singleton solutions. Then it greedily improves current solutions by adding one target variable or changing the state of one existing variable. The algorithm is quite sensitive to the initialization. The tabu search algorithm starts with an empty solution set. At each step, it generates the neighbors of the current solution by adding, changing, or deleting one target variable. Then it selects the best neighbor which has the highest GBF score and has not been visited before. Duplicate detection is thus necessary in the search process. Unlike in forward search, the best neighbor in tabu search can be worse than the current solution. To stop the tabu search, upper bounds are set on both the total number of search steps L and the number of search steps since the last improvement M as the stopping criteria.

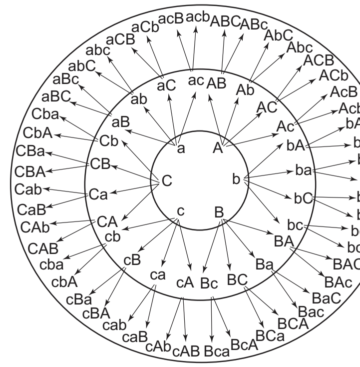

Figure 1: The search space of an MRE problem.

In this paper, we consider applying beam search (Rubin and Reddy 1977) to solve MRE. Beam search uses an open list to store the search frontier. At each step, it generates all successors of the solutions in the open list, but only retains the best solutions to create a new open list for the next step. A potential limitation of beam search is that most of the remaining solutions may originate from just a few solutions and are similar to each other. This reduces the diversity of the incumbent solution set, and the algorithms are still easily stuck in local optima. We address the limitation by developing a novel *hierarchical beam search* algorithm. We also propose three pruning criteria to speed up the hierarchical beam search in MRE problem.

Hierarchical Beam Search for Solving MRE

Solving MRE exactly is difficult, especially for the Bayesian networks having a large number of targets. This section introduces a hierarchical beam search algorithm that demonstrates excellent accuracy and scalability.

Search space formulation

Previous local search methods (Yuan, Lim, and Littman 2011) allow the use of some or all of the following operations in exploring the search space of MRE: adding a variable, changing a variable, and deleting a variable. These operations are all needed because the methods use greedy search steps that only consider the best neighbor of each solution. Because we use beam search, we have a lower risk of missing important parts of the search space compared with local search methods. We can therefore simplify the formulation of the solution space using the search graph in Figure 1. This is an example with three variables, each with two values, e.g., the first variable has two values “a” and “A”. The nodes represent possible solutions, and an arc from $\mathbf{U}$ to $\mathbf{V}$ means $\mathbf{V}$ is a successor of $\mathbf{U}$. The search starts with an open list containing all the singleton solutions in the innermost circle. We use only the operation of *adding* a variable to explore the search space, i.e., moving from inner circles to outer circles.

Hierarchical beam search

The search space exemplified in Figure 1 is huge. For n variables with d states each, the graph contains $(d+1)^n - 1$ solutions, even larger than the size of the search space of MAP (d^n). For large Bayesian networks with dozens or even hundreds of target variables, finding the optimal solution is challenging. We therefore apply beam search to solve the MRE search problem. Beam search was designed to address both the greediness of hill climbing search and the high computational complexity of exhaustive search. It uses an open list to limit the size of the search frontier so that only the most promising solutions get visited and expanded. However, beam search has the following limitation that has been overlooked to the best of our knowledge. The successors of a solution tend to not only look similar but also have similar scores; they also often lead the search to the same neighborhood in the search space. By retaining the best successors, the open list may end up containing the successors of just a few solutions that seem promising presently; other parts of the search space get thrown away. The search thus becomes too greedy often too soon and fails to find an excellent solution.

We propose a hierarchical beam search algorithm to address the above limitation of beam search. There are two levels of beams in the proposed search strategy; the hierarchical beams are organized as in Figure 2. The first-level beam is the same as the open list used by beam search. It stores the top N solutions as the search frontier. A second-level beam is newly introduced to store the best successors of *each* solution in the current open list. By limiting the second-level beam size to be K, we limit the over reproduction of offsprings from the same solution, and prevent the open list of the next step from being dominated by the successors of just a few solutions. This hierarchical beam search strategy has the effect of enhancing the *diversity* of the solutions maintained in the first-level beam, and preventing the search from being easily stuck in local optima.

Another important aspect of the new algorithm is duplicate detection. Different orderings of variable additions may lead to the same solutions. Reexpanding these solutions will lead to exponential explosion of the search space. It is therefore important to detect duplicates to avoid redundant work. A closed list is commonly used to store all solutions that have already been expanded. Whenever a successor is generated, it is checked against both the open and closed lists to see whether it has been visited before. Given the way we formulate the search space and organize the beams, the successors of the solutions in a circle can only appear in the very

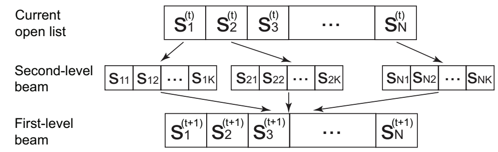

Figure 2: An illustration of the two-level hierarchical beams.

next outer circle; no expanded nodes will be ever generated again. There is thus no need to maintain a closed list. It is only necessary to keep a *working* open list for duplicate detection. Each solution in any second-level beams is checked against this open list for duplicates. This working open list eventually becomes the new open list for the next iteration by retaining the top N solutions.

Pruning criteria

To further enhance the diversity of solutions and improve the efficiency of hierarchical beam search, we derived three pruning criteria based on the theoretical properties of GBF and greedy search. The first pruning criterion is based on the following Theorem (Yuan, Lim, and Lu 2011).

Theorem 1. *Let the conditional Bayes factor (CBF) of explanation y given explanation $\mathbf{x}_{1:n} = x_1, x_2, \ldots, x_n$ be less than the inverse of the belief update ratio of the alternative explanations $\bar{\mathbf{x}}_{1:n}$, i.e.,*

$$CBF(y; \mathbf{e}|\mathbf{x}_{1:n}) < \frac{1}{r(\bar{\mathbf{x}}_{1:n}; \mathbf{e})}, \quad (7)$$

then we have

$$GBF(\mathbf{x}_{1:n}, y; \mathbf{e}) < GBF(\mathbf{x}_{1:n}; \mathbf{e}). \quad (8)$$

Therefore when adding a new variable, we do not add target y into $\mathbf{x}_{1:n}$, if we have $CBF(y; \mathbf{e}|\mathbf{x}_{1:n}) < \frac{1}{r(\bar{x}_{1:n}; \mathbf{e})}$ (*Pruning criterion 1*). Theorem 1 guarantees that no single-variable additions can improve the GBF score of the current solution. Note that this pruning rule intentionally ignores the possibility that adding multiple variables at the same time may increase the GBF score, even though each individual variable brings no improvement. It is also worth mentioning that as shown in Theorem 1, for adding (Equation 7) a variable to target sequence $\mathbf{x}_{1:n}$, we need to compute CBF and $r(\bar{x}_{1:n}; \mathbf{e})$. $r(\bar{x}_{1:n}; \mathbf{e})$ only needs to be computed once for each state in the open list. CBF can be efficiently evaluated using only conditional probabilities at each expansion. These properties of Theorem 1 make the pruning process very efficient.

We also propose two other pruning criteria to enhance solution diversity and reduce search space dramatically. In one criterion, we set an upper bound T on $CBF(y; \mathbf{e}|\mathbf{x}_{1:n})$, and do not add target y into $\mathbf{x}_{1:n}$ if $CBF(y; \mathbf{e}|\mathbf{x}_{1:n}) < T$ (*Pruning criterion 2*). The intuition of this pruning criterion is that $CBF(y; \mathbf{e}|\mathbf{x}_{1:n})$ measures the amount of extra information brought by adding variable y to the existing explanation $\mathbf{x}_{1:n}$ given evidence $\mathbf{e}$. If $CBF(y; \mathbf{e}|\mathbf{x}_{1:n})$ is less or only slightly higher than 1.0 (measured using T), the extra information is so little that it can be ignored.

Networks	Nodes	Leaves	States	Arcs
Alarm	37	11	105	46
Hepar	70	41	162	123
Emdec6h	168	117	336	261
Tcc4g	105	69	210	193
Insurance	27	6	89	52
Win95	76	1	152	112

Table 1: Benchmark diagnostic Bayesian networks used to evaluate the proposed algorithms.

Our studies show that in many cases single target values may have the maximum GBF scores. However, during the searching process, the states expanded by adding these target values can not be pruned by using the above two criteria, since this kind of target value can bring relevant information to the current state. However, in most of the cases we have $GBF(y, \mathbf{x}_{1:n}; \mathbf{e}) < GBF(y; \mathbf{e})$, which means $\mathbf{x}_{1:n}$ does not add information given y and $\mathbf{e}$, i.e., $(y, \mathbf{x}_{1:n})$ is dominated by y (Yuan, Lim, and Lu 2011). Thus in the proposed algorithm, we recorded the GBF of individual target values, and add y into target sequence $\mathbf{x}_{1:n}$, only if $GBF(y, \mathbf{x}_{1:n}; \mathbf{e}) > GBF(y; \mathbf{e})$ (*Pruning criterion 3*).

Finally, when one explanation is a subset of another explanation, but both have the same GBF score, we say the first explanation dominates the second explanation; only the first explanation is considered. This is designed to favor more concise explanations.

Experiments

We tested the proposed algorithms on six benchmark diagnostic Bayesian networks listed in Table 1, i.e., Alarm (Ala), Hepar (Hep), Emdec6h (Emd), Tcc4g (Tcc), Insurance (Ins), and Win95 (Win) (Beinlich et al. 1989; Binder et al. 1997; Onisko 2003). The nodes in these Bayesian networks are classified into two categories: target and observation (evidence). A *target* node represents a diagnostic interest. An *observation* node represents a symptom or a test. The experiments were performed on a 3.40GHz Intel Core i7 CPU with 8G RAM running a 3.11.0 linux kernel.

Experimental design

In the experiments, we compared the hierarchical beam search with/out pruning (Hbp/Hb) to four different algorithms, i.e., forward search (Fwd), tabu search (Tabu), beam search with/out pruning (Bmp/Bm). In beam search, there is no second-level beam which bounds the number of successors of each state. Thus, the proposed hierarchical beam search converges to beam search, when the second-level beam size K approaches infinity (or larger than the maximum number of successors generated by individual states). In Tabu search, we set $L = 400$ and $M = 20$. In the hierarchical beam search algorithm, we set the upper bound $T = 1 + 10^{-8}$ on CBF. We tested different combinations of beam sizes for the algorithms. For example, in Hbp160-2, we set the first-level beam size $N = 160$, and set the second-level beam size $K = 2$. In Bm320, we set the beam size $N = 320$. In the experiemnts, we highlighted the overall best accuracy and running time using bold, and the best results of each first-level beam size using bold and italic.

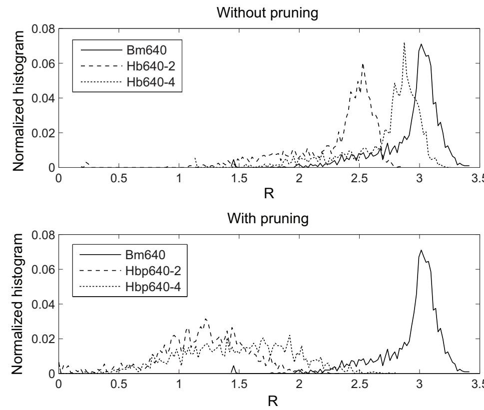

Figure 3: Normalized histograms of the ratios between the numbers of *duplicates* and the numbers of *open solutions* of all test cases for different algorithms on Win95.

For Alarm, Hepar, Emdec6h, and Tcc4g, we set all of the leaf nodes as evidence and the rest nodes as targets. Then, we randomly generated 1000 test cases of each Bayesian network by sampling from the prior distribution represented by the network. For Insurance and Win95, since the number of leaves is small, we randomly generated five test settings by selecting 60% and 30% non-leaf nodes as targets, respectively. The rest of nodes in these two networks are set as evidence. Then, we randomly generated 200 test cases of each test setting, thus totally 1000 test cases of each network. We restricted each test case to have at least one abnormal observation. The complexity of a test case is mainly determined by the number of targets. For all the test Bayesian networks, we cannot find the optimal solutions, so we generated the baselines by using the *best* solutions found by all tested algorithms. Thus, the solutions in the Bayesian networks are not necessarily the optimal solutions of the test cases, but these results are meaningful when we compare the performance between the different algorithms.

Hierarchical beam search vs. local search and regular beam search

In the experiments, we compared two versions of the hierarchical beam search (with/out pruning) with local search (i.e., forward search and tabu search) and two versions of beam search (with/out pruning). The size of the first-level beam of all the algorithms is set to be 160 and 320, respectively. The results are shown in Table 2 with two numbers in each entry. The number on top is the number of cases whose solutions found by the algorithm are consistent with the baselines. The number at the bottom is the average running time for solving the test cases in seconds.

The results show that regular beam search and hierarchical beam search usually have higher accuracy than forward and tabu search, but are slower, especially on large networks. The accuracy of regular beam search was consistently improved by using hierarchical beam strategy and

Top (#) Bot (s)	Fwd	Tabu	First-level 160						First-level 320					
			Bm	Bmp	Hb2	Hb4	Hbp2	Hbp4	Bm	Bmp	Hb2	Hb4	Hbp2	Hbp4
Ala (26)	902	920	966	962	***967***	963	***967***	963	979	981	979	**983**	977	979
	0.10	0.84	2.27	***0.40***	2.18	2.36	0.44	***0.40***	4.50	0.82	4.34	4.27	***0.73***	0.78
Hep (29)	935	906	954	951	959	947	***963***	950	976	975	**985**	976	**985**	979
	0.79	2.66	4.83	1.52	5.17	3.85	1.21	***0.86***	8.99	2.92	10.31	7.68	***2.30***	2.56
Emd (51)	957	993	998	**1000**	996	998	**1000**	**1000**	995	**1000**	995	997	**1000**	**1000**
	0.28	0.72	17.67	1.03	20.62	14.91	1.02	***0.65***	22.12	1.95	32.15	29.95	1.70	***1.56***
Tcc (36)	956	927	954	**1000**	990	984	**1000**	**1000**	960	**1000**	998	986	**1000**	**1000**
	0.39	1.16	7.72	0.73	8.20	6.22	0.84	***0.69***	12.76	1.36	15.95	12.41	***1.30***	1.44
Ins (17)	888	868	877	879	886	879	***897***	886	907	911	922	917	**926**	921
	0.20	1.40	1.42	***0.83***	1.38	1.43	0.89	0.86	2.80	1.60	2.72	3.00	***1.54***	1.57
Win (24)	903	875	930	960	949	939	***966***	962	955	973	967	962	974	**979**
	0.34	1.45	2.71	1.19	2.91	2.90	1.14	***0.95***	5.31	2.34	5.64	5.57	2.37	***1.86***

Table 2: Comparing hierarchical beam search with local search and regular beam search.

Top (#) Bot (s)	Hbp160			Hbp320			Hbp640		
	1	2	4	1	2	4	1	2	4
Ala (26)	928	**967**	963	928	977	**979**	928	981	**994**
	0.13	0.44	0.40	**0.13**	0.73	0.78	**0.13**	1.50	1.56
Hep (29)	**976**	963	950	976	**985**	979	976	**991**	988
	0.30	1.21	0.86	**0.33**	2.30	2.56	**0.32**	4.90	5.02
Emd (51)	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**
	0.37	1.02	0.65	**0.42**	1.70	1.56	**0.40**	3.27	3.19
Tcc (36)	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**	**1000**
	0.26	0.84	0.69	**0.29**	1.30	1.44	**0.28**	2.33	2.38
Ins (17)	801	**897**	886	801	**926**	921	801	**946**	943
	0.23	0.90	0.89	**0.19**	1.54	1.57	**0.20**	2.75	2.81
Win (24)	925	**966**	962	925	974	**979**	925	980	**984**
	0.22	1.14	0.95	**0.23**	2.37	1.86	**0.23**	5.19	3.67

Table 3: Testing the effect of different sizes for the first- and second-level beams on the hierarchical beam search.

pruning criteria at different sizes of first-level beam. Hierarchical beam search (including Hb and Hbp) achieved the best results on all the test networks. Somewhat surprisingly, increasing the size of the second-level beam from 2 to 4 did not significantly increase the accuracy of hierarchical beam search. The results show that introducing the second-level beams does help beam search to find better solutions, and the second-level beam does not need to be large to achieve the improvement. Moreover, for both beam search and hierarchical beam search, pruning boosted the speed of the search algorithms significantly. For large Bayesian networks, running time was decreased by 10 times. The results suggest that the pruning criteria prevented beam search from wasting time on exploring similar search spaces.

Diversity of solution population

We have claimed in several places that the second-level beams increase the diversity of the solutions in the open list. It is nontrivial to define a formal metric to measure the diversity of the solutions. Instead, we used R, the ratio between the total number of *duplicates* and the total number of *solutions retained in the open lists* for a test case, as an indicator for the solution diversity. We plotted the normalized histogram of the Rs of all test cases for each algorithm on Win95 in Figure 3. The higher the ratio, the more expansions led to the same solutions, indicating the lack of diversity between the solutions in the open list. The top graph in Figure 3 shows the comparison for Bm640, Hb640-2, and Hb640-4. Clearly Hb640-2 and Hb640-4 have a higher degree of diversity in their open lists than Bm640. Furthermore, Hb640-2 has a higher diversity than Hb640-4, which is consistent with the results in Table 2 showing the second-level beam size does not need to be large in order to achieve diversity. The bottom graph shows the results for Hbp640-2 and Hbp640-4. Theirs Rs are mostly between 1.0 and 2.0, unlike Bm640 whose Rs are peaked around 3.0. The results indicate that the pruning criteria helped the search algorithms avoid much redundant explorations of search space and achieved higher diversity.

Effect of beam sizes

To further test the effect of beam sizes on the performance of hierarchical beam search, we tried first-level beam sizes of 160, 320, and 640, and second-level beam sizes of 1, 2, and 4. The results are shown in Table 3. All those algorithms are with the pruning criteria. The results show that as the first-level beam size gets larger, the algorithms typically achieves better accuracy but worse efficiency. The accuracy of the algorithms may be negatively affected if the second-level beam size is too low or too large, especially when the first-level beam size is modest. Intuitively, when the second-level beam size is too low, the ability of the search to explore promising search spaces is negatively impacted; when too high, the search tends to focus too much on some search spaces but not the others. A moderate second-level beam size thus provides a nice balance.

Top(#) Eff(#/ms)	Hb	Hbp1	Hbp2	Hbp3	HbpAll
Ala (26)	**995**	**995**	**995**	993	994
	0	5.34	59.81	1.56	**159.26**
Hep (29)	986	987	986	987	**988**
	0	38.43	33.90	0.49	**58.33**
Emd (51)	994	997	**1000**	994	**1000**
	0	2.47	101.37	0.74	**141.88**
Tcc (36)	990	**1000**	**1000**	989	**1000**
	0	18.36	93.41	0.63	**112.90**
Ins (19)	940	941	941	**944**	943
	0	14.84	24.80	8.43	**55.51**
Win (24)	974	981	977	975	**984**
	0	12.82	27.27	0.95	**48.85**

Table 4: Comparing individual pruning criteria of the hierarchical beam search algorithms with beam size 640-4.

Effect of individual pruning criteria

We evaluated the performance of individual pruning criteria in hierarchical beam search algorithms with the first-level beam size 640 and the second-level beam size 4, e.g., Hbp1 denotes hierarchical beam search with pruning criterion 1. We further compared the results with Hb (i.e., Hb640-4) and HbpAll (i.e., Hbp640-4) (with all the pruning criteria) in Table 4. The top number in each entry still represents the number of cases with highest quality. The bottom number is now the *pruning efficiency*, defined as the number of pruned states divided by running time (ms) in each test case. The results show that the over-all accuracies of the three pruning criteria are similar with HbpAll. By comparing individual pruning criteria, we found that criterion 2 is the most efficient one, and criterion 3 is the least efficient one. The HbpAll algorithm which integrates the three pruning criteria together achieves the best pruning efficiency, which is larger than the individual criteria combined on all networks except Hep.

Conclusions

The main contribution of this paper is a novel hierarchical beam search algorithm for solving MRE inference problems in Bayesian networks. The algorithm uses two levels of beams to control the exploration of the search space. The first-level beam is used to limit the search to the most promising solutions. A second-level beam is used for each solution being expanded to store the best successors. Limiting the sizes of the second-level beams has the effect of increasing the diversity of solution population in the first-level beam. Three pruning criteria based on the theoretical properties of MRE are also introduced to achieve further diversity and efficiency. The experimental results also showed that the proposed algorithm performed competitive or better than the state-of-the-art algorithms, such as forward, tabu, and regular beam search, both in accuracy and speed. Based on the empirical study, we believe that the hierarchical beam search algorithms are especially suitable for the search problems with large branch factors and high correlation among the generated states.

Acknowledgements This work was supported by NSF grants IIS-0953723, IIS-1219114, and a PSC-CUNY enhancement award.

References

Beinlich, I.; Suermondt, G.; Chavez, R.; and Cooper, G. 1989. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on AI and Medicine*, 247–256.

Binder, J.; Koller, D.; Russell, S.; and Kanazawa, K. 1997. Adaptive probabilistic networks with hidden variables. *Machine Learning* 29:213–244.

Fitelson, B. 2007. Likelihoodism, bayesianism, and relational confirmation. *Synthese* 156:473–489.

Glover, F. 1990. Tabu search: A tutorial. *Interfaces* 20:74–94.

Good, I. J. 1985. Weight of evidence: A brief survey. *Bayesian statistics* 2:249–270.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press.

Lacave, C., and Díez, F. J. 2002. A review of explanation methods for Bayesian networks. *Knowledge Eng. Review* 17:107–127.

Nielsen, U. H.; Pellet, J.-P.; and Elisseeff, A. 2008. Explanation trees for causal Bayesian networks. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, 427–434.

Onisko, A. 2003. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*. Ph.D. Dissertation, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science.

Pacer, M.; Lombrozo, T.; Griffiths, T.; Williams, J.; and Chen, X. 2013. Evaluating computational models of explanation using human judgments. In *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, 498–507.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Rubin, S. M., and Reddy, R. 1977. The locus model of search and its use in image interpretation. In *IJCAI*, 590–595.

Yuan, C., and Lu, T.-C. 2007. Finding explanations in Bayesian networks. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, 414–419.

Yuan, C.; Liu, X.; Lu, T.-C.; and Lim, H. 2009. Most relevant explanation: Properties, algorithms, and evaluations. In *Proceedings of 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, 631–638.

Yuan, C.; Lim, H.; and Littman, M. L. 2011. Most relevant explanation: computational complexity and approximation methods. *Ann. Math. Artif. Intell.* 61:159–183.

Yuan, C.; Lim, H.; and Lu, T.-C. 2011. Most relevant explanation in Bayesian networks. *J. Artif. Intell. Res.* 42:309–352.