Evaluating Partition Strategies for Mini-Bucket Elimination

Emma Rollon and **Rina Dechter**
Department of Information and Computer Science
University of California, Irvine
{erollon, dechter}@ics.uci.edu

Abstract

Mini-Bucket Elimination (MBE) is a well-known approximation algorithm for *graphical models*. It relies on a procedure to partition a set of funtions, called *bucket*, into smaller subsets, called *mini-buckets*. The impact of the partition process on the quality of the bound computed has never been investigated before. We take first steps to address this issue by presenting a framework within which partition strategies can be described, analyzed and compared. We derive a new class of partition heuristics from first-principles and demonstrate its impact on a number of benchmarks for probabilistic reasoning.

1 Introduction

Mini-Bucket Elimination (MBE) (Dechter & Rish 2003) is one of the most popular bounding techniques for reasoning tasks defined over *graphical models* such as *Bayesian networks* (Pearl 1988) or *soft Constraint Satisfaction Problems* (Bistarelli *et al.* 1999). The power of MBE has been extensively demonstrated for optimization tasks such as finding the most likely tuple of a probabilistic network, or finding the optimal solution for a weighted csp (Dechter & Rish 2003; Kask & Dechter 2001; Marinescu & Dechter 2007). In this paper we focus on the more challenging task of weighted counting which captures the problem of counting solutions of a constraint network, evaluating the probability of evidence over Bayesian networks, and computing the partition function over Markov networks. These tasks are $\#P$-complete and are central to both probabilistic and deterministic reasoning.

MBE provides an approximation by applying the exact Bucket Elimination (BE) algorithm (Dechter 1999; Bertele & Brioschi 1972) on a simplified version of the problem. In BE all the functions in the so-called *bucket* are processed together, yielding a single *bucket's function*, which is defined on the union of the variables of the individual functions. Since this processing can be computationally expensive, MBE partitions the bucket into smaller subsets called *mini-buckets*, such that the number of variables in each mini-bucket is bounded by $z+1$, for a given value of z. Then, MBE processes each mini-bucket independently, yielding a set of mini-bucket functions defined over smaller subsets of variables which together bound the bucket's function. The partitioning of a bucket into mini-buckets having a bound z can be carried out in many ways, each resulting in a different impact on the overall accuracy.

The MBE scheme was used extensively and very effectively for approximating optimizations tasks, generating upper and lower bounds, but mostly as a scheme for generating heuristic evaluation function for branch-and-bound or best-first search (Kask & Dechter 2001; Marinescu & Dechter 2007). Yet, in all these schemas little attention was paid to improving the partition process itself. In most (if not all) previous work, the partitioning heuristic used aims to minimize the number of mini-buckets in the partitioning. The heuristic relies solely on the scope of the functions and is therefore called *scope-based* heuristic. Its effectiveness compared against random partitioning heuristics was sporadically demonstrated, but no systematic study was ever carried out.

In this paper we present a framework within which different greedy heuristic schemes can be examined, analyzed and compared, including the scope-based greedy heuristic. We present a new class of greedy heuristics that look beyond the function's scope, focusing on the function's content aiming to minimize a distance measure between the target bucket function and its mini-bucket bound. This yields a set of local distance rules that can efficiently guide a greedy algorithm within our framework. Roughly, a local rule associates a pair of mini-buckets with the *error* of keeping the two separated. In this paper we evaluate our scheme for the task of upper bounding the probability of evidence on *noisy-or bayesian networks*, *coding networks* and *genetic linkage analysis* (Ott 1999). Earlier attempts to use the mini-bucket approximation for likelihood computation failed, often generating trivial upper bounds of 1 (Mateescu, Dechter, & Kask 2002). The results demonstrate the heuristic scheme potential.

2 Preliminaries

Let $\mathcal{X} = (x_1, \dots, x_n)$ be an ordered set of variables and $\mathcal{D} = (D_1, \dots, D_n)$ an ordered set of domains, where D_i is the finite set of potential values for x_i. The assignment of variable x_i with $a \in D_i$ is noted $(x_i = a)$. A *tuple* t is an ordered set of assignments to different variables $(x_{i_1} = a_{i_1}, \dots, x_{i_k} = a_{i_k})$. The *scope* of t, noted $var(t)$, is the set of variables that it assigns.

2.1 Belief Networks

A *Bayesian network* (Pearl 1988) is a quadruple $BN = (\mathcal{X}, \mathcal{D}, \mathcal{G}, \mathcal{P})$ where $\mathcal{G}$ is a directed acyclic graph over $\mathcal{X}$ and $\mathcal{P} = \{p_1, \ldots, p_n\}$, where $p_i = P(x_i|pa_i)$ denotes the conditional probability tables (CPTs). The set pa_i is the set of parents of the variable x_i in $\mathcal{G}$. A Bayesian network represents in a compact way a probability distribution over tuple t, $P(t) = \prod_{i=1}^{n} P(x_i|pa_i)$. Given a Bayesian network BN and evidence tuple e, the *probability of evidence* $P(e)$ is defined as: $P(e) = \sum_{\mathcal{X}-var(e)} \prod_{i=1}^{n} P(x_i|pa_i)_{|e}$ where $f(X)_{|e}$ is a new function h defined over $\mathcal{X} - var(e)$ such that $h(t) = f(t \cdot e)$, where $t \cdot e$ is a new tuple containing both assignments.

2.2 Bucket and Mini-Bucket Elimination

Bucket elimination (BE) (Dechter 1999; Bertele & Brioschi 1972) is an exact algorithm for answering a variety of queries over graphical models. In particular, given a Bayesian network BN, BE computes the probability of evidence e as shown in the following pseudo-code:

function BE($(\mathcal{X}, \mathcal{D}, \mathcal{G}, \mathcal{P}), e$)
1. $\mathcal{S} := \{f_{|e} \mid f \in \mathcal{P}\}$;
2. $\mathcal{X} := \mathcal{X} - var(e)$;
3. **while** $\mathcal{X} \neq \emptyset$ **do**
4. $\quad x_i := Select(\mathcal{X})$;
5. $\quad \mathcal{B}_i := \{f \in \mathcal{S} |\ x_i \in var(f)\}$;
6. $\quad g_i := \sum_{x_i} (\prod_{f \in \mathcal{B}_i} f)$;
7. $\quad \mathcal{S} := \mathcal{S} - \mathcal{B}_i \cup \{g_i\}$;
8. $\quad \mathcal{X} := \mathcal{X} - \{x\}$;
9. **endwhile**
10. **return** ($\prod_{f \in \mathcal{S}} f()$);

endfunction

After incorporating the evidence in the network (line 1), BE processes the remaining variables $\mathcal{X} - var(e)$, eliminating them one at a time. The elimination of variable x_i is as follows. First, the algorithm computes the so called *bucket* of variable x_i, noted $\mathcal{B}_i$, which contains all the functions in $\mathcal{S}$ having x_i in their scope (line 5). Next, BE computes the *function of bucket* $\mathcal{B}_i$, noted g_i, by multiplying all its functions and subsequently summing out x_i from the result (line 6). Then, $\mathcal{S}$ is updated by removing the functions in bucket $\mathcal{B}_i$ and adding g_i (line 7). The new $\mathcal{S}$ does not contain x_i (all functions mentioning x_i have been removed) but preserves the exact result. When all variables have been eliminated, $\mathcal{S}$ contains a set of empty-scope functions (*i.e.*, a set of constants). The multiplication of those functions is the probability of evidence $P(e)$. The time and space complexity of the algorithm is exponential in a structural parameter called *induced width*, which is the largest scope of all the functions computed.

Mini-bucket elimination (MBE) (Dechter & Rish 2003) is an approximation of full bucket elimination that can be used to bound the exact solution when the induced width is too large. Given a control parameter z and a bucket $\mathcal{B}_i = \{f_1 \ldots, f_m\}$, MBE generates a partition $Q = \{Q_1, \ldots, Q_p\}$ of $\mathcal{B}_i$, where each subset $Q_j \in Q$ is called *mini-bucket*. Abusing notation, the scope of a set of functions $\mathcal{F}$, noted $var(\mathcal{F})$, is the union of scopes of the functions it contains. Given an integer parameter z, MBE restricts the size of the scopes of each mini-bucket by $z + 1$. Then, each mini-bucket is processed independently. The pseudo-code of MBE is obtained by replacing lines 6 and 7 in algorithm BE by,

6. $\{Q_1, \ldots, Q_p\} :=$ Partition($\mathcal{B}_i, z$);

6b. **for each** $j = 1 \ldots p$ **do** $g_{i_j} := \sum_{x_i} (\prod_{f \in \mathcal{Q}_j} f)$;

7. $\mathcal{S} := (\mathcal{S} \cup \{g_{i_1}, \ldots, g_{i_p}\}) - \mathcal{B}_i$;

Definition 1 (function of a partition) *Given a partition $Q = \{Q_1, \ldots, Q_p\}$ of a bucket $\mathcal{B}_i$, the* function represented by the partition Q *is* $g_i^Q = \prod_{j=1}^{p} \sum_{x_i} \prod_{f \in Q_j} f$.

It can be proved (Dechter & Rish 1997) that given a bucket $\mathcal{B}_i$ and a partition Q of $\mathcal{B}_i$, $\forall t$, $g_i(t) \leq g_i^Q(t)$. Consequently, the upper bound computed in each bucket accumulates yielding an upper bound of $P(e)$.

The time and space complexity of MBE is $O(d^{z+1})$ and $O(d^z)$, respectively, where d is the maximum domain size. The parameter z allows trading time and space for accuracy: greater values of z allow larger mini-buckets yielding tighter bounds.

3 Partitioning Framework

As we have seen, line 6 of MBE algorithm computes a partition of bucket $\mathcal{B}_i$. Different partitions will result in different upper bounds. In the following, we formalize the task of finding the optimal bucket partitioning and present a partitioning structure within which partition strategies can be described.

3.1 The Optimal Partitioning Task

We consider partitions that are parameterized by the maximum arity of each mini-bucket.

Definition 2 (z-partition) *Given a bucket $\mathcal{B}_i$ and a control parameter z, a partition $Q = \{Q_1, \ldots, Q_p\}$ of $\mathcal{B}_i$ is a z-*partition *if* $\forall Q_j \in Q, |var(Q_j)| \leq z$.

The goal of the partition process is to find a z-partition Q such that g_i^Q is the *closest* to the bucket function g_i. The closeness of two functions defined over the same scope can be evaluated in terms of a distance measure $dist$. Formally,

Definition 3 (partition task) *Given a bucket $\mathcal{B}_i$, a parameter z and a distance measure $dist$, the partition task is to find a z-partition Q^* of $\mathcal{B}_i$ such that*

$$Q^* = \arg\min_{Q} \{dist(g_i^Q, g_i)\}$$

where Q is a z-partition of $\mathcal{B}_i$.

In the probabilistic context, there are several common distance measures between probability distributions f and g which are relevant:

- *Relative error*: $RE(f, g) = \sum_t (\log(f(t)) - \log(g(t)))$.
- *Maximum relative error*: $MRE(f, g) = \max_t \{\log(f(t)) - \log(g(t))\}$.
- *Kullback-Leibler (KL) divergence*: $KL(f, g) = \sum_t f(t) \times \log(\frac{f(t)}{g(t)})$.
- *Absolute error*: $AE = \sum_t |f(t) - g(t)|$.

To gain insight into the partition task we propose a *partitioning framework*.

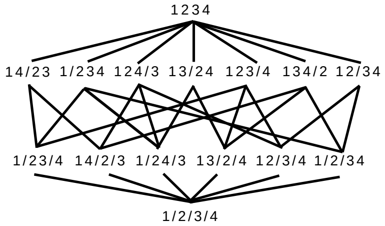

Figure 1: Hasse diagram of the partition lattice of $\mathcal{B}_6 = \{f_1, f_2, f_3, f_4\}$. We specify each function by its subindex.

3.2 Partitioning Lattice

We will organize the space of partitionings in a lattice using the *refinement* relation between partitions.

Definition 4 (refinement relation) *Partition Q is a* refinement *of partition Q', noted $Q \sqsubset Q'$, iff $Q \neq Q'$ and every element of Q is a subset of some element of Q'. Formally, $Q \sqsubset Q' \iff Q \neq Q' \,\wedge\, \forall Q_j \in Q,\, \exists Q'_k \in Q',\, Q_j \subseteq Q'_k$*

We say that $Q \sqsubseteq Q'$ iff $Q \sqsubset Q'$ or $Q = Q'$. In that case, we say that Q is *finer* than Q' or, conversaly, that Q' is *coarser* than Q. It is easy to see that the *refinement* relation yields a partial order and, indeed, a complete lattice that can be represented as a Hasse diagram.

Definition 5 (Hasse diagram) *Given a bucket $\mathcal{B}_i$, and the partial order $\sqsubseteq$, the Hasse diagram of all the possible partitions of $\mathcal{B}_i$ is defined as follows. Each partition Q is a vertex in the diagram. There is an upward edge from Q to Q' if $Q \sqsubset Q'$ and there is no partition Q'' such that $Q \sqsubset Q'' \sqsubset Q'$. In this case we say that Q' is a* parent *of Q, and we denote by $pa(Q)$ the set of all its parents. The* bottom partition, *noted $Q^{\perp}$, corresponds to having one mini-bucket for each function in $\mathcal{B}_i$, and the* top partition, *noted $Q^{\top}$, has only one mini-bucket containing all functions, which is the input bucket itself. Namely, $g_i^{Q^{\top}} = g_i$.*

Example 1 *Consider a bucket $\mathcal{B}_6 = \{f_1, f_2, f_3, f_4\}$. Its Hasse diagram in depicted in Figure 1. As observed, the finest partition is $Q^{\perp} = \{\{f_1\}, \{f_2\}, \{f_3\}, \{f_4\}\}$ (depicted in the bottom of the diagram). The coarsest partition is $Q^{\top} = \{\{f_1, f_2, f_3, f_4\}\}$ (depicted in the top of the diagram).*

As shown in (Dechter & Rish 1997), functions of coarser partitions always improve the upper bound:

Theorem 1 *Given a bucket $\mathcal{B}_i$ and two partitions Q and Q' of $\mathcal{B}_i$, $Q \sqsubset Q' \Rightarrow \forall t,\ g_i(t) \leq g_i^{Q'}(t) \leq g_i^{Q}(t)$*

Therefore, for any two arbitrary nodes Q and Q' connected in the partition lattice by an upward path, the bound $g_i^{Q'}$ is tighter than g_i^{Q}. However, in general, it is not possible to establish any tighter-than relation among functions of partitions that are not upward connected in the lattice. We can clearly conclude that,

Corollary 1 *Given a bucket $\mathcal{B}_i$ and two partitions Q and Q' of $\mathcal{B}_i$, $Q \sqsubset Q' \Rightarrow dist(g_i^{Q'}, g_i) \leq dist(g_i^{Q}, g_i)$, where* dist *is any of the distance funtions defined in Section 3.1.*

Corollary 1 is central to our partitioning framework. It states that the distance $dist$ to the top partition is always non-increasing along any upward path in the partitioning lattice. As a consequence, it is easy to see that the optimal z-partition Q^* is *maximal*, that is, all its parents in the lattice are l-partitions where $z < l$.

Theorem 2 *Given a bucket $\mathcal{B}_i$, the time complexity of finding the optimal z-partition of $\mathcal{B}_i$ as defined in Definition 3 is $O(T \times D)$ where T is the number of maximal z-partitions of $\mathcal{B}_i$, and D is the complexity of computing $dist(g^Q, g_i)$.*

The number of maximal z-partitions of a bucket $\mathcal{B}_i$ is upper bounded by the size of the lattice, which is $O(|\mathcal{B}_i|^{|\mathcal{B}_i|})$. However, a tigther bound is the maximum among the number of partitions in each level of the partitioning lattice given by the *Stirling number of the second kind* (Comtet 1974).

Proposition 1 *Given a bucket $\mathcal{B}_i$, its number of maximal z-partitions T for any z is bounded by: $T \leq \max_{1 \leq k \leq |\mathcal{B}_i|}\{S_2(|\mathcal{B}_i|, k)\}$ where $S_2(|\mathcal{B}_i|, k)$ is the Stirling number of the second kind, $S_2(|\mathcal{B}_i|, k) = \sum_{j=1}^{k} (-1)^{k-j} \frac{j^{|\mathcal{B}_i|-1}}{(j-1)!(k-j)!}$*

We can view any partition-seeking algorithm as a traversal algorithm of the partition lattice. An optimal partition-seeking algorithm would need to traverse the partition lattice bottom-up along all paths leading to a maximal z-partition. Since this is computationally hard, we approximate this task.

4 Heuristic Partitioning

Our focus is on depth-first greedy traversals of the partitioning lattice, going bottom-up to a maximal z-partition that are guided by a heuristic evaluation function, called $next$. The following pseudo-code describes our *Partition* scheme.

function Partition($\mathcal{B}_i, z$)
1. $Q := \emptyset$; **for each** $f \in \mathcal{B}_i$ **do** $Q := Q \cup \{f\}$;
2. **while** $\exists Q' \in pa(Q)$ s.t. Q' is a z-partition **do** $Q := next(Q)$;
3. **return** Q;
endfunction

Starting with the bottom partition of $\mathcal{B}_i$ (line 1), the algorithm traverses the path depth-first until a maximal z-partition is found (line 2). Different $next$ functions will likely lead to different final maximal z-partitions. Yet, each iteration is guaranteed to tighten the resulting bound (see Corollary 1).

In the following, we describe the scope-based heuristic discussed earlier, and propose a new class of partition heuristics, called *content-based* heuristics. It is important to note that the space complexity of MBE using any of these partition heuristics remains exponential in z.

4.1 Scope-based Heuristic

The partition heuristic proposed in (Dechter & Rish 1997) and used throughout in subsequent work with MBE, tries to minimize the number of mini-buckets in the partition, by including in each mini-bucket as many functions as possible. In our lattice, any bottom-up traversal of bucket partitioning reduce the number of mini-buckets. The only other guidance in selecting a parent partition is to favor unbalanced mini-buckets (see the pseudo-code below). First, the mini-buckets in the current partition are decreasingly ordered from left to right according to their arity. Then, we try to find the first mini-bucket in the ordering that can be merged with one of its next right-hand mini-buckets.

```
function next_SCP(Q)
    sort_by_arity(Q);
    j := 1;
    while ∄k, j < k < |Q|, |var(Q_j ∪ Q_k)| ≤ z do
        j := j + 1;
    endwhile
    if j < |Q| then return Q − Q_j − Q_k ∪ {Q_j ∪ Q_k};
    else return Q;
endfunction
```

Proposition 2 *Given a partition Q of a bucket $\mathcal{B}_i$, the time complexity of* $\text{next}_{SCP}(Q)$ *is* $O(|Q|\log(|Q|) + |Q|^2)$.

Proposition 3 *Given a bucket $\mathcal{B}_i$, the time complexity of* Partition($\mathcal{B}_i$,z) *using $next_{SCP}$ as heuristic function is* $O(|\mathcal{B}_i|\log(|\mathcal{B}_i|) + |\mathcal{B}_i|^2)$.

The main advantage of the scope-based heuristic is its simplicity which amounts to small overhead. Its main disadvantage is that it does not consider the actual information contained in each function.

4.2 Content-based Heuristics

We propose a new partition heuristic derived from the global optimization task in Definition 3. Given the current partition Q, we seek the parent partition of Q in the lattice that minimizes the distance $dist$ to the top target partition. Formally, given a bucket $\mathcal{B}_i$, the *optimal* (opt) $next$ is

$$next^{opt}_{dist}(Q) = \arg\min_{Q' \in pa(Q)} \{dist(g_i^{Q'}, g_i)\} \qquad (1)$$

subject to Q' being a z-partition.

In general, computing $dist(g_i^{Q'}, g_i)$ is exponential in the arity of g_i. The only exception is when RE is used as $dist$. In that case, we can derive a measure which is time exponential in z only. Let the number of complete assignments over a set of variables $\mathcal{Y}$ be $\mathcal{W}(\mathcal{Y})$. Then,

Proposition 4 *The number of extensions of a tuple t' such that $var(t') \subseteq \mathcal{Y}$ to the full scope $\mathcal{Y}$ is,* $\mathcal{W}(\mathcal{Y} - var(t')) = \frac{\mathcal{W}(\mathcal{Y})}{\mathcal{W}(var(t'))}$

Theorem 3 *Given a partition Q of a bucket $\mathcal{B}_i$, $next^{opt}_{RE}(Q)$ equals*

$$\arg\max_{Q^{jk} \in pa(Q)} \{\frac{1}{\mathcal{W}(var(Q_j \cup Q_k))} \times RE(g_i^{\{Q_j,Q_k\}}, g_i^{Q_j \cup Q_k})\} \qquad (2)$$

where Q^{jk} is the parent z-partition of Q that merges mini-buckets $Q_j, Q_k \in Q$.

Proof. Let $Q' = \{Q_1, \ldots, (Q_j \cup Q_k), \ldots, Q_p\}$ and $Q'' = \{Q_1, \ldots, (Q_l \cup Q_m), \ldots, Q_p\}$ be two parent partitions of $Q = \{Q_1, \ldots, Q_p\}$. First, let us suposse that $(Q_j \cup Q_k) \cap (Q_l \cup Q_m) = \emptyset$ (i.e., Q' and Q'' merge different mini-buckets). Q' is closer to $Q^\top$ than Q'' iff $RE(g_i^{Q'}, g_i) < RE(g_i^{Q''}, g_i)$, namely,

$$\sum_t \log\left[g_i^{Q_1}(t) \times \ldots \times g_i^{Q_j \cup Q_k}(t) \times \ldots \times g_i^{Q_p}(t)\right] \leq \sum_t \log\left[g_i^{Q_1}(t) \times \ldots \times g_i^{Q_l \cup Q_m}(t) \times \ldots \times g_i^{Q_p}(t)\right]$$

where $var(t) = var(\mathcal{B}_i)$. Using properties of $\log$ function, reordering and cancelling, the previous expression yields:

$$\sum_t (\log\left[g_i^{Q_l}(t) \times g_i^{Q_m}(t)\right] - \log\left[g_i^{Q_l \cup Q_m}(t)\right]) \leq \sum_t (\log\left[g_i^{Q_j}(t) \times g_i^{Q_k}(t)\right] - \log\left[g_i^{Q_j \cup Q_k}(t)\right])$$

Instead of summing over all tuples in the bucket's scope, we can sum over the tuples in the scopes of the mini-buckets involved in each side of the inequality and weigh each side by its number of extensions to the full scope. Then, the previous expression can be rewritten as,

$$\frac{1}{\mathcal{W}(var(t'))} \overbrace{\sum_{t'} (\log\left[g_i^{Q_l}(t) \times g_i^{Q_m}(t)\right] - \log\left[g_i^{Q_l \cup Q_m}(t)\right])}^{RE(g_i^{\{Q_l,Q_m\}}, g_i^{Q_l \cup Q_m})} \leq \frac{1}{\mathcal{W}(var(t''))} \underbrace{\sum_{t''} (\log\left[g_i^{Q_j}(t) \times g_i^{Q_k}(t)\right] - \log\left[g_i^{Q_j \cup Q_k}(t)\right])}_{RE(g_i^{\{Q_j,Q_k\}}, g_i^{Q_j \cup Q_k})}$$

where $var(t') = var(Q_l \cup Q_m)$ and $var(t'') = var(Q_j \cup Q_k)$. The heuristic will prefer Q' over Q'' if the averaged $RE(g_i^{\{Q_j,Q_k\}}, g_i^{Q_j \cup Q_k})$ is greater than the averaged $RE(g_i^{\{Q_l,Q_m\}}, g_i^{Q_l \cup Q_m})$. The derivation when the new mini-bucket in Q' and Q'' have one mini-bucket in common is very similar and leads to the same expression. Therefore, we can conclude that the theorem holds.

□

We will denote Expression 2 in Theorem 3 as $next_{RE}$. Note that the distance measure RE in $next_{RE}$ only refers to functions in the two candidate mini-buckets to be merged. This pairwise internal distance can be interpreted as the penalty or error due to keeping them separated. Namely, it defines a *local heuristic* which only has to consider the mini-buckets to be merged.

The other distance measures do not yield a local rule like $next_{RE}$, which is easy to compute. Therefore, we must resort to approximation. The derivation from Expression (1) of those local heuristics is similar to the one presented in Theorem 3, the only difference being the use of some approximation to transform an expression exponential in the arity of g_i to an expression exponential in z. Let f and g be two functions, and t be a tuple such that $var(t) = var(f) \cup var(g)$. When $dist$ in Expression (1) is KL or AE, an expression of the form $\sum_t (f(t) \times g(t))$ is approximated by $\sum_t f(t) \times \max_t\{g(t)\}$, while when $dist$ is MRE, $\max_t\{f(t) - g(t)\}$ is approximated by $\max_t\{f(t)\} - \max_t\{g(t)\}$. Note that in the KL and AE approximations, the role of f and g can be interchanged, leading to different local heuristics. For lack of space we will only list and experiment over a subset of the resulting derived heuristics. We refer to them as $next_{MRE}(Q)$, $next_{KL}(Q)$, and $next_{AE}(Q)$, defined, respectively, as follows:

$$\arg \max_{Q^{jk} \in pa(Q)} \{\max_t \{\log \left[g_i^{\{Q_j, Q_k\}}\right]\} - \max_t \{\log \left[g_i^{Q_j \cup Q_k}\right]\}\}$$

$$\arg \max_{Q^{jk} \in pa(Q)} \{MRE(g_i^{Q_j \cup Q_k}, g_i^{\{Q_j, Q_k\}})\}$$

$$\arg \max_{Q^{jk} \in pa(Q)} \{\mathcal{W}(var(Q_j \cup Q_k)) \frac{\max_t \{g_i^{\{Q_j, Q_k\}}\}}{\sum_t g_i^{Q_j \cup Q_k}}\}$$

where $var(t) = var(Q_j \cup Q_k)$.

We also consider another type of content-based heuristics, derived from Expression (2). Instead of ranking the partitions according to RE, these heuristics called $next_{dist'}(Q)$, rank them according to the other distance measures proposed in Section 3.1.

Proposition 5 *Given a partition Q of a bucket $\mathcal{B}_i$, and a distance measure $dist$, the time complexity of $next_{dist}(Q)$ is $O(|\mathcal{B}_i|^2 d^z)$, where d is the maximum domain size of the variables and z is the control parameter.*

Proposition 6 *Given a bucket $\mathcal{B}_i$, the time complexity of* Partition($\mathcal{B}_i$,z) *using $next_{dist}(Q)$ as heuristic function is $O(|\mathcal{B}_i|^3 d^z)$.*

5 Empirical Evaluation

In this section we evaluate the performance of each of the mini-bucket partition heuristics individually in order to determine, first, whether there exist any that is either clearly superior or inferior. We report results with the scope-based heuristic, which was used in previous work, and with only a subset of the content-based partition heuristics presented in the previous section (due to space reasons). As we will see, none of the partition strategy dominates. But, when combined (by taking their minimum upper bound) they yield a far superior bound to any single strategy. This combined heuristic, while superior and more robust strategy for using the MBE scheme is also more time consuming; linear in the number of participating partition heuristics. Note, however, that the MBE scheme is restricted by the space and not by the time. Therefore, as the experiments show, having a collection of combined heuristics can increase its power without increasing the needed memory, at the cost of only a constant factor (depending on the number of partitioning schemes we use) to its time. In our experiments we consider the combination of all the individual partition heuristics reported.

We compare the individual and combined MBE schemes with two alternative approaches available in the literature: the any-time bounding scheme ATB (Bidyuk & Dechter 2006) and Box-Propagation (Mooij & Kappen 2008). ATB is based on the cutset-conditioning schema and applies exact computation over a subset of the cutset search space, controlled by a parameter h, while applying Bound-Propagation (Leisink & Kappen 2003) to the rest of the space. Both Bound-Propagation (and as a consequence ATB), and Box-Propagation were derived for bounding posterior probabilities. The probability of evidence can be obtained by applying the chain rule to individual bounds on posteriors. We could not compare with Tree-Reweighted (Wainwright, Jaakkola, & Willsky 2005), the other alternative approach to compute upper bounds on P(e), because all its implementations are only available for binary graphical models.

We conduct our empirical evaluation on three benchmarks: *noisy-or bayesian networks*, *coding networks* and *linkage analysis*. All instances are included in the UAI08 evaluation [1]. For comparison, we always report upper bound on P(e) (UB) and cpu time in seconds. For MBE, we report UB and cpu time as a function of the control parameter z, while for ATB we report this information as a function of h. In all the tables, we box and underline the best and second best upper bound computed by any partition heuristic as a function of z, respectively. The highest value of z reported is the highest feasible value given the available memory. MBE uses the variable ordering established by the *min-fill* heuristic (Dechter 2003) after instantiating evidence variables. We run all experiments in a Pentium Core Duo 2.6 GHz and 2GB ram.

Coding Networks. Table 1 reports the results. The exact P(e) is not available. First, let us consider the impact of the partition heuristic on the upper bound (columns '*SCP-Based Heur.*' and '*CTNT-Based Heur.*'). For $z = 22$, there is no dominating partition heuristic. Each heuristic computes the best UB on two instances. The only exception is $next_{RE}$, that computes the best UB on one instance. The improvement of the best upper bound with respect to the second best ranges from 25% (e.g., see *BN_129*) to orders of magnitude (e.g., see *BN_126*, *BN_130*, *BN_132* and *BN_133*).

Regarding cpu time, the content-based heuristics are 2 to 3 times slower than the scope-based heuristic. The reason is that during the traversal of the partition lattice content-based partioning heristics have to compute intermediate functions. It is important to note that it is the space and not the time that bounds the maximum feasible z. As a consequence, that constant increase in time is not that significant as the space complexity remains the same.

When $z = 22$ and each partition heuristic is considered independently, ATB($h = 150$) outperforms at least one of them on four instances (i.e., *BN_128*, *BN_129*, *BN_131* and *BN_133*). However, ATB only outperforms MBE$_{Combined}$($z = 22$) on instance *BN_126*. MBE$_{Combined}$($z = 22$) obtains UBs from 1 order (e.g., see *BN_131*) to 5 orders (e.g., see *BN_130*) of magnitude better than ATB($h = 150$) while 3 to 5 times faster. Box-Propagation is the least accurate approach, computing upper bounds up to 27 orders of magnitude worse than MBE$_{Combined}$ (see *BN*_131).

[1] http://graphmod.ics.uci.edu/uai08/Software

Inst.	w*	z	SCP-Based Heur.		CTNT-Based Heur.								MBE$_{Combined}$		ATB			BoxProp	
			$next_{SCP}$		$next_{RE}$		$next_{KL'}$		$next_{AE}$		$next_{MRE'}$								
			UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	h	UB	Time	UB	Time
BN_126	55	20	5.49E-43	7.94	8.64E-44	17.72	5.61E-44	22.00	2.24E-45	16.47	**1.50E-45**	18.91	1.50E-45	83.05	4	1.52E-41	50.14	3.74E-30	62.47
		21	1.85E-43	14.43	**4.31E-46**	40.22	1.22E-42	40.92	1.88E-42	41.89	6.31E-45	43.60	4.31E-46	181.06	50	2.54E-42	631.52		
		22	1.57E-44	32.45	2.33E-43	88.25	**1.67E-45**	84.70	1.55E-44	82.84	3.13E-44	86.39	1.67E-45	374.63	150	1.25E-42	1441.75		
BN_127	54	20	9.10E-46	9.55	1.29E-45	24.04	2.49E-47	26.40	**1.24E-47**	25.33	6.63E-45	22.92	1.24E-47	108.25	4	2.27E-43	54.85	3.86E-31	63.61
		21	1.94E-44	19.83	**2.41E-47**	46.66	1.58E-45	46.92	4.91E-45	39.09	6.30E-45	49.31	2.41E-47	201.82	50	2.25E-44	426.26		
		22	2.42E-47	37.49	1.35E-47	74.72	2.01E-47	81.32	1.98E-48	86.72	**1.00E-48**	85.83	1.00E-48	366.09	150	1.90E-44	946.33		
BN_128	49	20	3.76E-42	9.45	1.07E-41	25.02	6.98E-42	26.88	4.14E-41	23.79	**9.01E-43**	28.13	9.01E-43	113.28	4	1.63E-42	85.71	1.98E-31	63.02
		21	1.91E-41	18.02	**4.49E-44**	45.37	2.57E-42	52.26	7.88E-42	45.76	4.49E-43	51.31	4.49E-44	212.72	50	7.19E-43	637.23		
		22	5.14E-43	32.90	3.00E-41	81.56	8.47E-45	88.21	2.03E-43	90.65	**5.64E-45**	81.65	5.64E-45	374.97	150	1.44E-43	1225.00		
BN_129	53	20	2.34E-44	8.95	1.44E-44	20.37	9.22E-44	24.27	4.04E-46	20.84	**3.12E-47**	25.02	3.12E-47	99.46	4	8.15E-45	50.48	1.78E-29	62.4
		21	2.46E-46	16.63	1.22E-44	41.68	1.36E-45	47.13	1.17E-44	40.62	**1.23E-46**	43.87	1.23E-46	189.94	50	2.11E-45	585.19		
		22	1.39E-44	36.78	1.43E-44	80.84	3.91E-45	87.34	**1.26E-45**	90.01	1.55E-45	102.23	1.26E-45	397.21	150	5.43E-46	1400.44		
BN_130	53	20	3.42E-45	8.53	2.87E-46	20.02	4.22E-45	20.06	**7.06E-48**	16.15	2.95E-46	18.38	7.06E-48	83.13	4	2.87E-44	47.22	5.99E-29	63.29
		21	**5.52E-50**	17.61	3.08E-48	35.80	7.95E-48	44.38	1.61E-48	34.60	3.44E-47	36.66	5.52E-50	169.06	50	2.96E-45	619.17		
		22	**2.35E-50**	30.83	1.58E-47	60.98	5.65E-47	76.75	3.49E-48	64.35	3.20E-47	68.50	2.35E-50	301.41	150	2.28E-45	1299.18		
BN_131	53	20	1.34E-42	8.99	**3.90E-46**	21.40	3.73E-44	22.66	2.25E-45	21.45	7.88E-45	20.34	3.90E-46	94.85	4	1.25E-44	52.81	1.06E-30	63.16
		21	**1.27E-47**	16.27	6.77E-45	35.02	1.76E-44	43.01	6.96E-45	38.47	1.96E-46	41.59	1.27E-47	174.36	50	3.68E-45	484.84		
		22	9.22E-45	30.81	2.44E-45	65.36	8.91E-45	83.95	**3.09E-46**	60.65	4.70E-46	61.78	3.09E-46	302.56	150	1.00E-45	1276.21		
BN_132	51	20	3.32E-49	9.69	1.47E-46	20.44	2.42E-48	25.62	7.18E-49	22.34	**9.19E-50**	20.94	9.19E-50	99.03	4	6.32E-44	50.84	3.28E-32	63.32
		21	6.94E-47	15.96	2.38E-46	39.89	8.13E-50	43.59	1.93E-48	39.77	**1.04E-51**	42.98	1.04E-51	182.19	50	1.03E-44	689.28		
		22	**1.97E-50**	29.23	3.12E-48	73.44	3.56E-48	78.00	2.17E-49	69.00	1.78E-46	67.99	1.97E-50	317.67	150	8.09E-45	1627.15		
BN_133	55	20	1.80E-43	8.18	7.42E-46	22.34	2.04E-44	24.52	5.35E-43	19.47	**3.43E-47**	23.67	3.43E-47	98.18	4	2.74E-42	53.20	2.89E-29	62.26
		21	6.00E-44	17.24	4.73E-45	37.89	**1.92E-45**	46.85	1.99E-44	39.84	4.68E-45	41.80	1.92E-45	183.63	50	2.37E-43	671.80		
		22	1.47E-44	35.26	3.81E-44	62.31	**1.80E-45**	85.02	2.68E-43	70.68	2.92E-43	77.00	1.80E-45	330.27	150	9.50E-44	1846.83		
BN_134	55	20	4.87E-44	9.53	2.12E-44	20.62	2.18E-44	22.54	**4.57E-45**	20.47	2.13E-44	22.98	4.57E-45	96.14	4	1.80E-43	47.69	1.59E-30	63.81
		21	1.45E-43	17.07	2.68E-45	42.40	1.22E-43	43.15	4.13E-44	35.22	**3.37E-46**	45.29	3.37E-46	183.13	50	8.62E-45	606.47		
		22	3.80E-47	39.00	**1.66E-47**	74.36	9.56E-46	105.81	3.96E-47	95.75	4.99E-47	82.74	1.66E-47	397.67	150	4.82E-45	1412.53		

Table 1: Results on coding networks. The best and the second best upper bound computed by any partition heuristic as a function of z is boxed and underlined, respectively.

Inst.	P(e)	z	SCP-Based Heur.		CTNT-Based Heur.								MBE$_{Combined}$		ATB			BoxProp	
			$next_{SCP}$		$next_{RE}$		$next_{KL'}$		$next_{AE}$		$next_{MRE'}$								
			UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time	h	UB	Time	UB	Time
bn2o-30-15-150, nb. vars. = 45, evidence = 15, $w^* = 23$																			
1a	5.85E-05	16	3.29E-03	0.14	1.38E-03	0.453	**3.46E-04**	0.36	5.96E-04	0.31	3.81E-04	0.48	3.46E-04	1.75	4	5.32E-01	1.98	9.93E-01	352.08
		18	1.88E-03	0.53	1.84E-03	1.279	2.26E-04	1.39	**1.39E-04**	1.01	3.13E-04	1.19	1.39E-04	5.40	200	5.70E-02	103.39		
1b	0.565652	16	7.59	0.14	7.51E-01	0.406	8.46E-01	0.39	**7.35E-01**	0.37	7.55E-01	0.50	7.35E-01	1.81	4	9.26E-01	2.01	9.92E-01	351.81
		18	1.44	0.53	7.32E-01	1.42	8.19E-01	1.50	7.74E-01	1.37	**6.67E-01**	1.05	6.67E-01	5.87	200	8.26E-01	101.75		
2a	4.02E-07	16	2.11E-05	0.16	7.53E-06	0.422	5.46E-06	0.45	3.33E-06	0.33	**2.28E-06**	0.36	2.28E-06	1.72	4	1.25E-01	2.01	9.68E-01	352.02
		18	**1.79E-06**	0.47	3.39E-06	1.279	5.62E-06	1.62	2.50E-06	1.58	4.29E-06	1.19	1.79E-06	6.13	200	4.04E-03	88.81		
2b	0.541111	16	4.75	0.14	**7.03E-01**	0.515	7.12E-01	0.44	7.17E-01	0.47	7.92E-01	0.45	7.03E-01	2.01	4	7.99E-01	2.01	9.68E-01	352.2
		18	9.96	0.56	7.28E-01	1.372	7.16E-01	1.79	7.58E-01	1.06	**6.67E-01**	0.94	6.67E-01	5.72	200	7.51E-01	89.93		
3a	1.18E-04	16	1.00E-02	0.14	8.73E-04	0.484	1.50E-03	0.41	**8.14E-04**	0.39	1.19E-03	0.41	8.14E-04	1.83	4	1.68E-01	1.98	9.85E-01	351.84
		18	9.63E-04	0.61	**4.68E-04**	1.357	1.16E-03	1.58	2.81E-03	1.36	7.62E-03	1.26	4.68E-04	6.16	200	2.67E-02	73.77		
3b	0.188686	16	8.82E-01	0.14	4.53E-01	0.375	5.29E-01	0.53	**3.61E-01**	0.38	3.80E-01	0.39	3.61E-01	1.81	4	7.70E-01	1.99	9.68E-01	351.98
		18	7.35E-01	0.50	4.89E-01	1.279	4.00E-01	1.54	3.30E-01	1.67	**3.27E-01**	1.20	3.27E-01	6.19	200	5.36E-01	69.35		
bn2o-30-25-250, nb. vars. = 55, evidence = 25, $w^* = 25$																			
1a	2.96E-09	16	7.06E-05	0.30	6.31E-06	1.138	4.32E-06	1.25	1.41E-05	1.28	**4.31E-06**	1.50	4.31E-06	5.46	4	6.60E-02	5.78	9.82E-01	584.54
		18	1.98E-05	0.83	4.75E-06	5.054	3.63E-07	4.46	1.86E-06	4.21	**9.07E-08**	4.79	9.07E-08	19.34	200	1.10E-03	395.51		
1b	0.151829	16	18.42	0.28	5.99E-01	1.076	5.28E-01	1.36	**4.99E-01**	1.09	6.05E-01	1.31	4.99E-01	5.12	4	8.11E-01	5.83	9.81E-01	585.01
		18	2.75	1.01	**3.69E-01**	4.742	3.94E-01	3.64	4.09E-01	5.30	5.72E-01	4.38	3.69E-01	19.08	200	6.48E-01	380.79		
2a	2.44E-07	16	6.21E-04	0.30	3.67E-04	1.217	**1.21E-05**	0.97	1.64E-04	1.30	4.44E-04	1.28	1.21E-05	5.06	4	1.98E-01	5.80	9.90E-01	584.52
		18	2.80E-05	0.94	5.86E-05	5.18	**5.89E-06**	5.41	5.02E-05	3.82	1.90E-05	5.15	5.89E-06	20.50	200	2.17E-02	401.59		
2b	0.308949	16	4.58	0.30	7.92E-01	1.014	**6.39E-01**	1.05	6.67E-01	1.11	6.92E-01	1.50	6.39E-01	4.96	4	7.63E-01	5.78	9.85E-01	584.39
		18	2.51	0.91	6.53E-01	5.054	6.36E-01	3.85	6.85E-01	4.38	**5.89E-01**	4.56	5.89E-01	18.75	200	7.07E-01	367.17		
3a	2.76E-10	16	1.26E-06	0.30	6.46E-07	1.544	2.45E-07	1.15	2.19E-07	1.45	**2.55E-08**	1.23	2.55E-08	5.68	4	1.14E-01	5.81	9.94E-01	585.01
		18	8.88E-08	1.01	7.33E-08	5.975	**3.74E-08**	4.60	4.30E-08	4.23	4.96E-08	4.87	3.74E-08	20.69	200	7.10E-03	409.05		
3b	0.468007	16	4.23	0.30	8.42E-01	1.154	**7.88E-01**	1.33	7.98E-01	1.20	8.22E-01	1.25	7.88E-01	5.23	4	7.96E-01	5.85	9.90E-01	584.79
		18	1.52	0.89	7.83E-01	5.148	7.27E-01	4.65	7.26E-01	4.60	**6.84E-01**	4.28	6.84E-01	19.57	200	7.47E-01	336.55		

Table 2: Results on bn2o networks. The best and the second best upper bound computed by any partition heuristic as a function of z is boxed and underlined, respectively.

Noisy-or Bayesian Networks. Table 2 reports the results. For $z = 18$, $next_{MRE'}$ outperforms the other partition heuristics on six instances. The improvement of the best upper bound with respect to the second best ranges from 5% to 1 order of magnitude (e.g., *bn2o-30-25-250-1a*). It is important to note that $next_{SCP}$ is not able to compute a better upper bound than the trivial bound of 1 on five instances, while all content-based heuristics do.

If we disregard upper bounds greater than the trivial one and each partition heuristic is considered independently setting $z = 18$, ATB($h = 200$) is superior to at least one of them in three instances (i.e., *bn2o-30-15-150-2b, bn2o-30-15-150-3b* and *bn2o-30-25-250-3b*). However, MBE$_{Combined}$($z = 18$) outperforms ATB in all instances. MBE$_{Combined}$ is able to compute upper bounds up to 5 orders of magnitude smaller than ATB requiring 1 order of magnitude less computation time (e.g., see *bn2o-30-25-250-1a* and *bn2o-30-25-250-3a*). As in the previous benchmark, Box-Propagation is the least accurate approach.

Linkage Analysis. Table 3 shows the results. Comparison with ATB and Box-Propagation was not possible. Both algorithms require a Bayesian network and an independent set of evidence. However, the pedigree instances we have already incorporate the evidence into the definition of the network. Since we do not compare with alternative approaches, we omit the *combined* MBE.

Regarding accuracy on the upper bound, there is no clear dominating partition heuristic. However, $next_{MRE'}$ seems to be inferior in this benchmark. The improvement of the best upper bound with respect to the second best ranges from 6% (e.g., see *pedigree13*) up to 2 orders of magnitude (e.g., see *pedigree37* and *pedigree41*).

Regarding cpu time, the content-base heuristics are typically 2 to 3 times slower than the scope-based heuristic. However, there exist exceptions where the content-based heuristics are 1 to 2 orders of magnitude slower (e.g., see *pedigree23* and *pedigree37*).

6 Conclusions

The paper investigates a new heuristic scheme for mini-bucket partitioning and applies it to the probability of evidence in Bayesian networks. We derive the new heuristic from first-principles and demonstrate its impact on a series of benchmarks. Our experimental results suggest that, in general, none of the partitions heuristics dominate all the others. Interestingly, the combination of all heuristics to compute the final bound as the best among them results in a very effective method both in terms of accuracy and time.

Acknowledgements

This work was supported by the NIH grant 5R01HG004175-03.

References

Bertele, U., and Brioschi, F. 1972. *Nonserial Dynamic Programming*. Academic Press.

Bidyuk, B., and Dechter, R. 2006. An anytime scheme for bounding posterior beliefs. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, 1095–1100.

Bistarelli, S.; Fargier, H.; Montanari, U.; Rossi, F.; Schiex, T.; and Verfaillie, G. 1999. Semiring-based CSPs and valued CSPs: Frameworks, properties and comparison. *Constraints* 4:199–240.

Comtet, L. 1974. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. Dordrecht-Holland/Boston-U.S.A. Reidel Publishing Company.

Dechter, R., and Rish, I. 1997. A scheme for approximating probabilistic inference. In *Proceedings of the 13th UAI-97*, 132–141.

Dechter, R., and Rish, I. 2003. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM* 50(2):107–153.

Dechter, R. 1999. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence* 113:41–85.

Dechter, R. 2003. *Constraint Processing*. San Francisco: Morgan Kaufmann.

Kask, K., and Dechter, R. 2001. A general scheme for automatic generation of search heuristics from specification dependencies. *Artificial Intelligence* 129:91–131.

Leisink, M. A. R., and Kappen, H. J. 2003. Bound propagation. *Journal of Artificial Intelligence Research* 19:139–154.

Marinescu, R., and Dechter, R. 2007. Best-first and/or search for graphical models. In *AAAI*, 1171–1176. AAAI Press.

Mateescu, R.; Dechter, R.; and Kask, K. 2002. Tree approximation for belief updating. In *AAAI/IAAI*, 553–559.

Mooij, J. M., and Kappen, H. J. 2008. Bounds on marginal probability distributions. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, 1105–1112.

Ott, J. 1999. *Analysis of Human Genetic Linkage*. Baltimore, Maryland: The Johns Hopkins University Press.

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

Wainwright, M. J.; Jaakkola, T.; and Willsky, A. S. 2005. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory* 51(7):2313–2335.

Inst. nb.	P(e)	w*	z	SCP-Based Heur.		CTNT-Based Heur.							
				$next_{SCP}$		$next_{RE}$		$next_{KL'}$		$next_{AE}$		$next_{MRE'}$	
				UB	Time	UB	Time	UB	Time	UB	Time	UB	Time
7	-	35	20	**2.27E-35**	65.85	4.57E-35	151.01	9.16E-34	98.05	8.64E-34	98.08	5.27E-35	104.80
9	-	28	20	1.49E-67	20.72	1.55E-68	30.05	**2.57E-69**	30.72	**2.57E-69**	30.36	1.91E-68	30.50
13	-	40	20	3.96E-16	21.36	3.97E-16	34.88	1.52E-14	35.51	**9.23E-17**	34.27	1.85E-15	36.47
19	E-59.79	27	18	1.13E-43	60.34	**1.06E-43**	92.57	4.85E-43	112.92	5.51E-43	90.21	3.63E-43	327.91
23	E-39.69	31	18	**3.02E-28**	6.54	9.29E-27	797.08	5.58E-28	524.81	2.21E-27	489.70	6.15E-28	509.92
30	-	23	20	**8.68E-81**	15.05	1.71E-80	16.26	1.71E-80	15.99	3.18E-80	15.91	1.71E-80	16.01
33	E-54.27	33	20	1.55E-43	44.60	8.11E-46	87.27	9.24E-47	62.71	1.19E-46	61.79	**1.03E-47**	67.30
34	-	38	18	2.74E-37	56.96	4.26E-34	79.08	5.99E-35	100.23	**2.01E-37**	77.06	6.01E-36	77.84
37	E-116.57	22	18	5.60E-103	43.27	**9.68E-105**	637.54	4.67E-103	560.73	3.76E-103	524.25	3.91E-103	530.56
40	-	34	18	8.81E-67	6.80	**3.50E-68**	27.57	2.82E-65	23.14	1.38E-66	22.09	5.31E-68	31.39
41	-	39	18	6.25E-45	106.66	2.19E-43	184.47	**5.01E-48**	198.59	**5.01E-48**	156.41	1.21E-46	148.82
44	-	31	20	2.52E-56	35.69	5.02E-56	69.09	**4.53E-57**	54.01	1.02E-56	54.40	1.02E-56	54.66
51	-	49	18	9.10E-44	12.76	1.77E-43	15.74	**1.49E-44**	17.66	1.17E-43	17.96	1.15E-43	17.94

Table 3: Results on pedigree networks. The best and the second best upper bound computed by any partition heuristic as a function of z is boxed and underlined, respectively. A '-' means unknown $P(e)$.