Journal of Artificial Intelligence Research 69 (2020) 231-295 Submitted 06/2020; published 09/2020

Improved High Dimensional Discrete Bayesian Network Inference using Triplet Region Construction

Peng Lin LINPENG@CUEB.EDU.CN
School of Statistics
Capital University of Economics and Business
Beijing 100070, P. R. China

Martin Neil M.NEIL@QMUL.AC.UK
Norman Fenton N.FENTON@QMUL.AC.UK
School of Electronic Engineering and Computer Science
Queen Mary University of London
London E1 4NS, and Agena Ltd, UK

Abstract

Performing efficient inference on high dimensional discrete Bayesian Networks (BNs) is challenging. When using exact inference methods the space complexity can grow exponentially with the tree-width, thus making computation intractable. This paper presents a general purpose approximate inference algorithm, based on a new region belief approximation method, called Triplet Region Construction (TRC). TRC reduces the cluster space complexity for factorized models from worst-case exponential to polynomial by performing graph factorization and producing clusters of limited size. Unlike previous generations of region-based algorithms, TRC is guaranteed to converge and effectively addresses the region choice problem that bedevils other region-based algorithms used for BN inference. Our experiments demonstrate that it also achieves significantly more accurate results than competing algorithms.

1. Introduction

Performing efficient inference to compute posterior densities on high dimensional Bayesian Network (BN) models, containing many densely connected variables, is a major computational challenge. Discovering the global optimum is known to be NP-hard (Dagum & Luby, 1993; Cooper, 1990) for exact and approximate inference algorithms, although recent work by (Weller & Jebara, 2013; Jebara, 2014) showed that polynomial time discovery is possible for some models. When using exact methods, such as the Junction Tree (JT) algorithm (Darwiche, 2009; Koller & Friedman, 2009), space complexity can grow exponentially with the tree-width (Murphy, 2012; Koller & Friedman, 2009) and the computation can quickly become intractable. Instead, we must rely on approximate inference. There are two categories of well known approaches for reducing the space complexity and performing efficient inference on BNs. These are a) variational inference based methods and b) the Bethe/Kikuchi free energy based approximation. We first briefly review these methods and illustrate their limitations when applied to BN inference.

Variational Inference (VI) (Jordan et al., 1998; Wainwright & Jordan, 2008; Beal, 2003) is a flexible deterministic framework to use factorized and tractable distributions to approximate Bayesian posterior densities (when they are analytically intractable) and can also provide a lower bound of the log evidence of the model when given data. VI can be applied to general models belonging to the conjugate exponential family by using the variational message passing (VMP) algorithm (Winn & Bishop, 2005). The VMP and its recent derivative distributed-VMP (Masegosa et al., 2016) localize the computations to maximize the lower bound of the log evidence (of the model) using message passing between the parent and child nodes in a BN. This means it can be applied to large networks and massive data sets. Alternatively, the lower bound of the log evidence can also be optimized using the gradient-based approach, such as that used by stochastic variational inference (SVI) (Hoffman et al., 2013). Both VMP and SVI suit Bayesian model selection tasks. However, the approximation accuracy of the posterior densities using these approaches relies heavily on how the posterior distributions over the latent variables are factorized. This factorization usually assumes a simple form, such as the mean-field (Jordan et al., 1998) which uses a fully factorized form of the latent variables. Despite this, the VI based methods are very efficient for the model selection task, even if the posterior densities are not necessarily well approximated (Hoffman & Blei, 2015; Blei, Kucukelbir, & McAuliffe, 2017). Certain improvements are available, such as the structured VMP (Winn, 2004) and structured SVI (Hoffman & Blei, 2015) which factorize the posterior distributions over the latent variables into clusters. In this way, dependency structures are retained and approximation accuracy is increased; however, a by-product is that they introduce more complex optimization problems that are difficult to solve (Blei et al., 2017).

Bethe/Kikuchi approximation: An alternative variational method to the previous VI based algorithms is the belief propagation algorithm (Yedidia et al., 2005; Heskes, 2006). Belief Propagation (BP) is an efficient algorithm to perform inference over the factor graph of the original model that can be either directed or undirected. BP is exact if the factor graph is a tree but only approximate if the factor graph contains cycles, and the approximation accuracy of the posterior marginal cannot be generally guaranteed. Yedidia et al. (2005) showed that the convergence of BP corresponds to the stationary points of the *Bethe approximation* of the free energy of a factor graph. Generalized BP (GBP) is a generalised form of the BP algorithm that uses the *region graph* instead of the factor graph and can be more accurate than the BP algorithm. The convergence of the GBP algorithm corresponds to the minimization of the Kikuchi cluster free energy (*Kikuchi approximation*), which generalizes the Bethe approximation by including higher-order interactions.

Compared to the VI based approaches, Kikuchi approximation requires only local consistency of the node beliefs hence does not ensure providing bounds of the free energy function (Beal, 2003; Yedidia et al., 2005) but can be more accurate and flexible to approximate the posterior densities. However, using the Kikuchi approximation involves choosing appropriate regions to construct a region graph, which is left to the model designer with only ad-hoc guidance available in the literature. Despite the success of existing region based approaches to approximate the posterior densities of undirected graphical models, such as the spin class/grid models (Yedidia et al., 2005), applying the region based approach for BN inference involves overcoming several limitations in existing algorithms. Most of these

relate to the question of how best to build a region graph that is both an efficient and accurate approximation.

Existing algorithms build a region graph based on either structural information (Dechter & Rish, 1997; Mateescu et al., 2010; Gelfand & Welling, 2012; Welling et al., 2005) of the graph model or by evaluating a cost function over a set of candidate regions generated by a greedy (or a guided) search (Komodakis & Paragios, 2008; Forouzan & Ihler, 2015; Sontag et al., 2008; Welling, 2004). However, the former approach does not ensure accuracy when given limited region size, because it ignores factor information in the model completely, and thus can incur locally very poor approximation (we call this the max variability problem). The latter approach is computationally expensive and the performance cannot be guaranteed.

We present a general inference algorithm called Triplet Region Construction (TRC) that overcomes these limitations, enabling us to perform robust inference on high tree width BNs. The TRC algorithm represents a major breakthrough, since the cluster size for high tree-width models is reduced from worst-case exponential to polynomial whilst ensuring accuracy.

Our approach involves four key novel contributions:

1. **Applies to directed graphs**: Whereas region-graph based belief propagation is typically applied to undirected graphical models with ad-hoc guidance on region selection, TRC is the first algorithm that can be applied systematically to directed graph models (sections 3 and 4).

2. **Optimum and localized region selection**: We present a region identification algorithm, called Outer Region Identification (ORI), that combines both structural and factor information to incorporate necessary local factor interactions as an effective way of identifying the outer regions in the region graph (sections 3). In contrast to function value based algorithms, ORI selects outer regions in an entirely localized way without the need to run message passing for scoring a cost function. ORI satisfies the perfect correlation property and the maxent-normal property (Yedidia et al., 2005; Gelfand & Welling, 2012), both desirable pre-conditions that improve accuracy.

3. **Improved numerical stability**: region-based belief propagation algorithms suffer from unavoidable numerical instability problems when performing inference on high tree-width factorized models. We propose a Region Graph Binary Factorization (RGBF) algorithm to decompose the region graph into an equivalent, but more numerically stable, alternative. We show that RGBF improves the robustness of region-based belief propagation algorithms (section 4).

4. **Improved accuracy**: The TRC algorithm, which combines the above ORI and RGBF algorithms, is guaranteed to converge. It achieves more accurate results when we compare the singleton and higher ordered marginal distributions of variables with those produced by other approximate algorithms when using bounded cluster size (section 5). TRC can also be extended to use larger cluster sizes for greater accuracy. Moreover, competing algorithms normally ignore the max variability problem associated with the approximation; this is simply because when results are reported

they are averaged and so inaccuracy is masked. We show this problem cannot be ignored, and argue that TRC effectively tackles the max variability problem better than competing approaches. In general, because TRC generates more regions than other algorithms to ensure the perfect correlation property is satisfied, its efficiency can be worse than other algorithms; however, if we relax the perfect correlation property, TRC can achieve similar efficiency as other algorithms.

The paper is structured as follows:

In section 2 we provide the necessary definitions and background for region-based approximation methods for BN models and explain why the previous region-based algorithms cannot effectively deal with the high tree-width BN inference. In section 3 we describe our region choice algorithm, which solves the complex region choice problem by using region graph based approximation for directed models. In section 4 we present our proposed TRC algorithm and explain how it can be extended using larger cluster size to obtain better accuracy. In section 5 we present experiments where we compare TRC with VI based algorithms and other region-based algorithms using challenging and high tree-width (also high dimensional) BNs as test cases. Section 6 concludes the paper and discusses possible extensions of the TRC algorithm.

2. Background and Definitions

This section provides necessary definitions, notation and background for the paper. In section 2.1 we introduce BNs and the region graph related background. In section 2.2 we review existing region graph based algorithms and illustrate their limitations. Section 2.3 introduces the complete Directed Acyclic Graph (DAG) and its equivalent binary factored model for which we will use the region-based approximation to perform inference. The rationale for using such a binary factored model is that it enables us to construct the region graph automatically and efficiently.

2.1 Converting BN to A Region Graph and Performing Inference

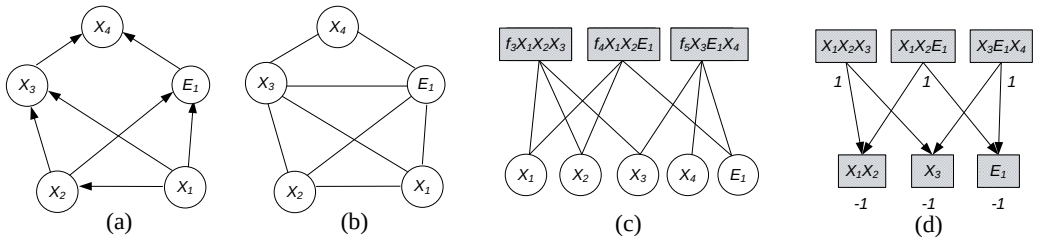

Figure 1: (a) a BN G; (b) moral graph $M[G]$ of (a); (c) factor graph of (a), with factor ϕ_{X_1} and $\phi_{X_1X_2}$ multiplied into $\phi_{X_1X_2X_3}$; (d) region graph $\mathcal{G}$ of (c) with counting numbers listed beside each region.

Bayesian Network (BN): A BN (such as that shown in Fig. 1 (a)) is a Directed Acyclic Graph (DAG), with nodes $X_1, ..., X_n$ representing random variables, together with a Conditional Probability Distribution (CPD) for each node conditional on its parent nodes, if

present. The absence of arcs between nodes encodes the Conditional Independence (CI) (Murphy, 2012) assumptions between variables when no evidence is entered in the BN. The BN represents the joint distribution, p of the random variables $X_1, ..., X_n$ as the product of its CPDs. Without using any CI assumptions, we can use the chain rule to factorize the joint distribution, as shown in Eq. (1).

$$p(X_1, X_2, ..., X_n) = p(X_1) \prod_{i=2}^{n} p(X_i | X_1, ..., X_{i-1}). \tag{1}$$

With CI assumptions this simplifies to Eq. (2)

$$p(X_1, ..., X_n) = \prod_{i=1}^{n} p(X_i | X_{pa(i)}), \tag{2}$$

where $pa(i)$ represents the parents of node i.

Next we introduce some fundamental definitions and the necessary notation related to region graphs.

Moral Graph: the BN needs to be converted into its equivalent Markov Networks [1] (MN) form. This is achieved by constructing the *moral graph* of a BN, G, denoted $M[G]$, where $M[G]$ is an undirected graph that contains an edge (X_i, X_j) if there is an edge between X_i and X_j in G, or if X_i and X_j are parents of the same child node. For example, Fig. 1 (b) is the moral graph $M[G]$ associated with the BN G of Fig. 1 (a). The added edge (such as the one linking X_3 and E_1 in Fig. 1 (b)) between the parents that share the same child node is called a *moral edge*.

Factor graph: A factor graph (shown in Fig. 1 (c)) is a bipartite graph representing the factorization of a function encoded by the Markov network, with a factor node per factor of the BN, a variable node per variable of the BN and an edge connecting a factor node to a variable node. The *factor size* is the number of variables included in a factor.

Cluster graph and cluster: A cluster graph for a set of factors Φ over variables V is an undirected graph, each of whose nodes i is associated with a subset of variables $C_i \in V$, where C_i is called a *cluster*. Each factor $\phi \in \Phi$ is associated with a cluster. Each edge between a pair of clusters C_i and C_j is associated with a sepset $S_{i,j} \in C_i \cap C_j$. The *cluster size* is the number of variables it contains. The *cluster space* is a product of cardinalities of all variables that belong to the cluster (Koller & Friedman, 2009).

Tree-width (*t.w.*): This is one less than the maximum cluster size produced using exact methods, such as Junction Tree (Lauritzen & Spiegelhalter, 1988).

We also need to convert the BN's CPDs into factors, such that: $\phi_{\{i\} \cup \{pa(i)\}}(X_i, X_{\{pa(i)\}}) = p(X_i | X_{\{pa(i)\}})$. However, connecting parent nodes X_i and X_j via a moral edge assumes that

1. A Markov Network is a set of random variables having a Markov property described by an undirected graph.

X_i and X_j are not conditionally independent and, if we do so, we may lose CI information (Murphy, 2012) contained in the BN (i.e. $I(M[G]) \subseteq I(G)$, where $I(\cdot)$ is the set of all CI information encoded by the graph).

We can now define region and region graph:

Definition 1: A *region* r is a set of variable nodes, V_r, and factor nodes, A_r, in a factor graph, such that if a factor node a is in A_r, all variable nodes linked to a are in V_r. The *region size* is the number of variables it contains.

Definition 2: A *region graph*, $\mathcal{G}$, is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ in which each vertex $v \in \mathcal{V}$, corresponding to a region r, is labelled with a subset of nodes in a factor graph, using label $l(v) \in \mathcal{L}$. We say v_p is a parent of v_c if $v_p \to v_c$ is a directed edge $e \in \mathcal{E}$, and $v_p \to v_c$ exists only if the label of v_c is a subset of the label of v_p. We say r_p is a parent region of r_c if $v_p \to v_c$. If there is a directed path from r_a to r_d, we say r_a is an ancestor region of r_d. Each region r is associated with a counting number $c_r = 1 - \sum_{r' \in Ancestor(r)} c_{r'}$, given $Ancestor(r)$ are all ancestor regions of r and c'_r is the number of degrees of freedom for the region r'.

Let $child(r)$ denote all child regions of region r and $parent(r)$ denote all parent regions of region r. Regions with no parents (and hence which have no incoming region edges) are called outer regions. All others are called inner regions. One classical method to construct a region graph is the Cluster Variation Method (CVM) (Yedidia et al., 2005). It can be used to produce a valid region graph given the outer regions are readily identified by other algorithms. CVM iterates over each level to generate the regions for subsequent levels using intersections between the regions declared at the previous level to define new regions at the current level. Fig. 1 (d) is a CVM region graph using only factors as outer regions. The accuracy of CVM cannot be guaranteed because the choice for the outer regions are left to the model designer.

The convergence of any region-graph based inference algorithm corresponds to the minimization of the Kikuchi cluster free energy $F_{\mathcal{G}}$ (Yedidia et al., 2005),

$$F_{\mathcal{G}} = \sum_{r \in R} c_r \{ \sum_{x_r} b_r(x_r) E_r(x_r) + \sum_{x_r} b_r(x_r) \log b_r(x_r) \} + L_{\mathcal{G}}, \tag{3}$$

where $E_r(x_r)$ is an energy term associated with each region and is computed using the factors $\phi_{\{i\} \cup \{pa(i)\}}(X_i, X_{\{pa(i)\}})$. The region belief term b_r, is an estimated distribution of the true distribution over a region (Yedidia et al., 2005).

The region-graph based approximation minimizes $F_{\mathcal{G}}$ over the locally consistent polytope $local[\mathcal{G}]$ (Murphy, 2012), which is a set of pseudo-marginal distributions over the variables in each region. This local consistency neglects higher ordered terms of consistencies, and hence achieves polynomial time complexity, provided that all regions are calibrated (Koller & Friedman, 2009).

Minimization of $F_{\mathcal{G}}$ is equivalent to the problem of constructing a fixed point for the constrained region belief equations. Hence, we need to consider the Lagrangian term $L_{\mathcal{G}}$,

$$L_{\mathcal{G}} = \sum_{r \in R} \sum_{c \in child(r)} \sum_{x_c} \lambda_{r,c}(x_c) \{ \sum_{x \in r \setminus c} b_r(x_r) - b_c(x_c) \} + \sum_{r \in R} \gamma_r (\sum_{x_r} b_r(x_r) - 1), \quad (4)$$

which incorporates two kinds of constraints: the normalization constraint for each region, $\sum_{x_r} b_r(x_r) = 1$ and the running intersection constraints between parent and child region beliefs, $\sum_{x \in r \setminus c} b_r(x_r) = b_c(x_c)$ $(c \in child(r))$. Solving for Lagrangian multipliers λ and γ, in Eq. (4), corresponds to an iterative message passing algorithm, such as GBP. We define a *constrained* region based free energy to be an approximate region based free energy subject to the constraints in Eq. (4).

The Kikuchi region based entropy, $H_{\mathcal{G}}$, is defined as.

$$H_{\mathcal{G}} = \sum_{r \in R} c_r H_r(b_r) = - \sum_{r \in R} c_r \sum_{x_r} b_r(x_r) \ln b_r(x_r), \quad (5)$$

$H_{\mathcal{G}}$ can be obtained when all beliefs are calibrated after the GBP update. Likewise, the *constrained* region based entropy is an approximate region based entropy subject to the constraints in Eq. (4).

In addition to GBP there are other region based message-passing algorithms (Meltzer et al., 2009; Yuille, 2002). One of these, called the Concave-Convex Procedure (CCCP) (Yuille, 2002; Yuille & Rangarajan, 2003) is particularly powerful since it guarantees convergence where GBP does not. However, both GBP and CCCP can be numerically unstable for large models. In general, GBP is more efficient than CCCP since CCCP runs using an outer-inner double loop and each inner loop involves recursively updating the Lagrangian multipliers. For simplicity, we present the CCCP updating equation in Appendix B.3.

2.2 Region Choice and The Limitations of Existing Approaches

Region graph based approximation reduces the maximum cluster size by constructing variational regions of limited size. However, the quality of the results produced by any region graph based algorithm is limited by the regions chosen, especially choices made about outer regions. We therefore introduce two desired properties that are relevant to region-based entropy and that subsequently influence region choices:

Definition 3: *Perfect-correlation property.* This property requires the sum of all regions' counting numbers to equal one, $\sum_R c_R = 1$.

This property is not necessary for all classes of models, but it does ensure that the region-based entropy is correct if all variables are perfectly correlated.

Definition 4: *Maxent-normal property.* This property holds if a constrained region-based free energy approximation is valid and the corresponding constrained region-based entropy, $H_{\mathcal{G}}$, achieves its maximum when all the beliefs are uniform.

These two properties ensure the region based entropy is correct when all variables are perfectly correlated or all beliefs are uniform.

All variables are perfectly correlated, in the global joint distribution, if all variables can only be in the same state with equal probability (Yedidia et al., 2005). For example, in Fig. 1 (a) (assuming variables are all binary) the BN has all variables perfectly correlated if node X_1 has a uniform distribution and other variables have diagonal CPDs, i.e. $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ for the CPD $X_2|X_1$ and $\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$ for the other CPDs. Given this, there are only two possible joint states (all variables will have state zero or one), and they have equal probability. The joint entropy of this example model is $\ln(2)$. We can verify this using region-based approximation and for this model every region will have entropy $\ln(2)$, which implies that the sum of the counting numbers over all regions must be one. If it is not equal to one the region based entropies must be incorrect. Similarly, we can verify the maxent-normal property for a region based approximation by setting all CPDs to uniform distributions and then compare the region based entropy to the true entropy, $N \times \ln(2)$ (where N is the number of variables in the model). Again, if it is not $N \times \ln(2)$ then it is not Maxent-normal.

Next, we review existing approaches that attempt to produce better region choice and describe their limitations with respect to the above properties. We classify these algorithms, with respect to how regions are chosen, as *structural information based* and *function value based*. Here we concentrate on comparing our algorithm against others that have used marginal inference and published results in marginal form. There is also much relevant research on MAP inference (Weiss, Yanover, & Meltzer, 2007; Batra, Nowozin, & Kohli, 2011), which is clearly related but is outside the scope of our and comparable algorithms.

Structural information based algorithms include:

- *Iterative Join Graph Propagation* (IJGP) (Mateescu et al., 2010) and its extensions. IJGP uses a bounded cluster size parameter (called i-bound) and connects all factor clusters in a loopy join graph. Accuracy in IJGP is increased by using higher i-bound values. There are different ways to define the join graph used in IJGP, including Mini Bucket Elimination (MB) (Dechter & Rish, 1997; Rollon & Dechter, 2010) and the weighted MB (WMB) approaches (Liu & Ihler, 2011) where the factors are grouped into partitions, and the i-bound parameter is used to control space complexity in each partition. However, accuracy is not guaranteed when using small i-bound parameter values. IJGP also ignores the counting number assignments with respect to the perfect correlation and maxent normal properties.

- *Loop Structured Region Graph* (LSRG) algorithm (Welling et al., 2005; Gelfand & Welling, 2012) and its extensions produces a three-level region graph that meets a number of pragmatic conditions. The first condition is to discover Fundamental Cycle Bases (FCBs) of G, such that each cycle has some edge that do not appear in any cycle preceding it in some predefined order. Given this, the FCB is tree-exact with respect to all spanning trees of G. If the FCBs also satisfy the Tree-Robustness condition, the inference result will be more accurate. The FCB approach can satisfy both desired properties provided that the fundamental cycles are correctly discovered, but there

are possibly many fundamental cycles to consider. Also, under a bounded cluster size, i.e. three, the use of structural information based approaches will be insufficient to choose regions. For example, in Fig. 1 (a) if we need to incorporate the pair-wise information of $\{E_1, X_3\}$, we would add either $\{X_2, E_1, X_3\}$ or $\{X_1, E_1, X_3\}$ by using FCB. However, these two regions are at symmetric positions in the moral graph in Fig. 1 (b) and it is not possible to determine which one should be chosen using only structural information.

Function value based approaches include:

- *Cluster pursuit* (Komodakis & Paragios, 2008; Sontag et al., 2008; Forouzan & Ihler, 2015). This is a class of score based algorithms that test the potential improvement of a defined cost function (usually upper bound of the partition function) by greedy or guided searching of the new regions. These methods involve region inference and message scheduling during the region selection process and can be computationally expensive.

- *Region pursuit* (Welling, 2004). This also requires region inference to test the improvement, in terms of free energy, by adding regions. Again, the difficulty in using function value to choose regions is that it may be computationally expensive when a large number of candidate regions need to be generated and tested. Also, which regions should be tested next is undefined.

Both the above function based methods offer no control of the counting number to guarantee the perfect correlation and maxent normal properties are satisfied. Hence, the region based entropy cannot be guaranteed to be correct under these conditions. The literature also classifies the region based algorithms as either top-down (IJGP and its extensions) or bottom-up (FCB, cluster pursuit) based on how regions are constructed. Top-down algorithms start with exact methods and split large regions into smaller ones and bottom-up algorithms start with small regions and then group them into larger ones. Both of them cannot usually ensure desired region graph properties.

In addition to the region choice problem associated with other algorithms, there are also numerical issues that can hinder the convergence of message passing for high dimensional models. Region graphs generated for large and complex models often involve multiple connections between regions located at different levels. Because one parent can have a large number of children in a model, the same variable will appear in many different regions across different region graph levels. This gives rise to a large counting number and multiple cycles associated with a single region, leading to under/overflows during the multiplication of multiple messages. This numerical instability can be encountered during message updating in both GBP and CCCP, hence preventing convergence for region graph based algorithms. We address a solution to the region choice problem in section 3 and the numerical instability in section 4.

2.3 Defining The Binary Factorized Model

Apart from its graphical representation, the most obvious attraction of using a BN is the fact that its joint probability distribution is the simplified factorization Eq. (2), rather

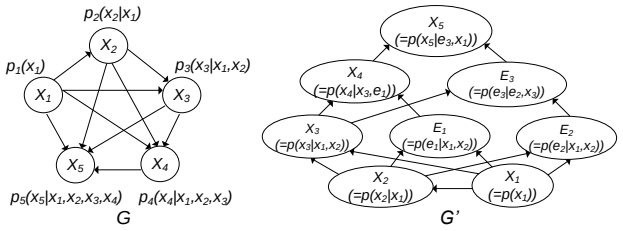

Figure 2: BF process of a 5-dimensional complete DAG G to its binary factorized model G'

than the normal Eq. (1). However, in the 'worst case' there may be no CI assumptions in the BN, and so Eq. (1) and (2) are equal. In other words, the BN graph is a *complete DAG* with n nodes, where every pair of nodes is connected by a directed edge. Performing inference on such a complete BN graph represents the worst case space complexity for exact algorithms and is usually intractable.

Given our inference task is to compute marginals, our proposed region-graph based algorithm will not operate on the original BN directly but requires that the BN be pre-processed using a Binary Factorization (BF) algorithm first. This converts the original BN G into a factored BN G', where each node has at most two parent nodes. The conversion will not change any ordered exact joint distributions for the original nodes in G and G'. The rationale for using such a binary factorized BN is that it helps generate a region graph involving only triplet outer regions so the resulting region graph can be more convenient to scale up and satisfy the desired region graph properties.

Our BF algorithm is different from other forms of BF algorithm in (Wainwright & Jordan, 2008); it extends the BF algorithm in (Neil et al., 2012; AgenaRisk, 2020) to accommodate discrete nodes. The core idea of our BF algorithm is to binary factorize the CPD defined on each node. In a BN any CPD containing more than three discrete parent nodes can be factorized using the method (by incrementally adding intermediate variables to combine a pair of parent nodes' CPDs each time). In this way each node in the resulting factorized BN has at most two parent nodes, and the maximum node indegrees are therefore reduced by adding intermediate nodes.

We show the BF process for a 5-dimensional complete DAG graph in Fig. 2. The *Binary Factorized Graph* (BFG) is a BN that is a BF version of the complete DAG. For example, the graph G' in Fig. 2 is a BFG.

In what follows we show that such a BFG model is uniquely defined if the parent child node ordering is defined by Theorem 1. We make use of a well-known result of graph theory (which can be proved by induction on the number of nodes) that asserts that any complete DAG of n nodes has a unique Hamiltonian path, and is hence uniquely defined up to a permutation of the n nodes. Specifically,

Theorem 1: *In any complete DAG of n nodes there is exactly one node with indegree $n-1$, exactly one node with indegree $n-2$, ..., exactly one node with indegree one, and exactly one node with indegree zero.*

Theorem 1 ensures the uniqueness of the chain rule factorization (Eq. (1)) for a complete DAG of n nodes subject to the order in which, for each $i = 1, \ldots, n$ node X_i is the (unique) node with indegree $i-1$. In what follows we will assume this ordering of the nodes in the complete graph.

Proposition 1: *A BN G can be transformed into a binary factorized BN G' whose nodes are a superset of G and which is 'equivalent' to G such that, for each node X_i in G, the CPD of X_i in G' is equivalent*[2] *after factorization to the CPD of X_i in G.*

The proof of Proposition 1 is given in Appendix A.1. Applying the BF process to an n node complete DAG results in a BFG model with κ_n nodes where $\kappa_n = n+(n-2)(n-3)/2 = (n^2 - 3n + 6)/2$.

We will use κ_n to denote the number of variables in a BFG throughout the rest of the paper. In a BFG all original nodes in an n-dimensional DAG at the left most path labeled as X_i, $i = 1, ..., n$, and the rest of the nodes are intermediate nodes.

We can transform any BN into a complete DAG by adding arcs, and then for every node X_i for which parents are added, the CPD of X_i in the complete DAG is defined as $p'(X_i|X_{\{pa(i)\}}) = p(X_i|X_{\{pa(i)\}})$ where $p(X_i|X_{\{pa(i)\}})$ is the CPD of X_i in the original BN. We can then binary factorize the complete DAG to obtain its BFG.

However, in a complete DAG the indegrees of each node is known (Theorem 1). For a BN that is not a complete DAG this conversion requires the BN nodes to be ordered, so that the indegree of each node can be determined. Here we present the detailed steps to convert a BN to a BFG.

(1) Triangulate the graph [3] $M[G]$ to obtain a triangulated graph $G_{\mathcal{T}}$ and a clique tree $C_{\mathcal{T}}$, and obtain an ordering $\prec$ of the cliques in an ancestor-descendent way, such that the clique C_i is ordered before C_j if the variables in C_i are ancestors of the variables in C_j.

(2) Using the order $\prec$ to define a valid parent-child ordering $\pi_G =: \{X_1 \rightarrow, ..., \rightarrow X_n\}$ for the n original variables in G, where for any X_j that is the successor to X_i $(i, j \in n)$, $X_j \notin X_{pa(i)}$. The ordering π_G uniquely define the indegrees of each original node.

(3) Add edges to G following the ordering π_G, i.e., each node links to its successors.

(4) Binary factorize G to obtain its BFG G'.

(5) Define the CPD for each original node X_i $(X_i \in G)$ in the BFG by reusing the CPD from G and replicating X_i or $X_{pa(i)}$ $(X_{pa(i)} \in G)$ via intermediate nodes in the specific path connecting X_i and $X_{pa(i)}$.

In step (1) nodes in each clique are ancestor-descendent ordered. If the nodes in a clique are ordered as neighbours in π_G the optimum regions can be obtained more efficiently in the corresponding BFG, otherwise more regions will be created.

2. Equivalent means the original CPD for node X_i in G can be rebuilt by the new CPDs for X_i and its associated intermediate nodes in G'. So any ordered exact joint distributions are the same between G and G'.
3. An undirected graph G is called *triangulated* if every cycle of length strictly greater than 3 possesses a chord.

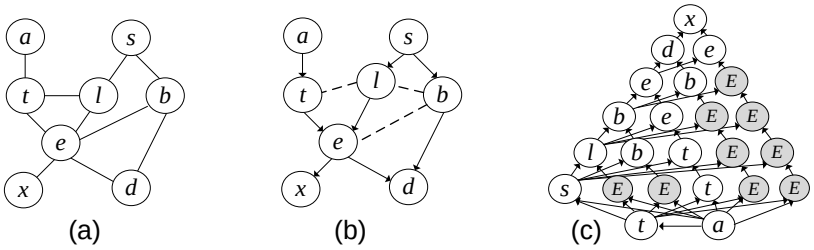

Figure 3: (a) moral graph $M[G]$ of the Asia BN G; (b) triangulated graph (with the parent-child directions preserved) of $M[G]$. The dashed lines are moral or chordal edges; (c) κ_8 BFG G' of G with the original nodes lie on the leftmost path and the replicated nodes labeled the same with the original nodes.

Step (1) ensures $M[G]$ is triangulated (Golumbic, 2004; Koller & Friedman, 2009) so large cycles in $M[G]$ can be split into triplet cycles. In practice, we can bypass step (3) given the BFG structure in step (4) will be determined once the node ordering in step (2) is defined. To reuse the CPD for X_i from G the parent nodes $X_{pa(i)}$ in G' must be the same as those in G. This is achieved by replicating the original variables via intermediate variables in G' if $X_{pa(i)}$ in G and G' differ.

As an example, consider the well-known Asia BN G whose moral graph is shown in Fig. 3 (a) and whose triangulated graph is shown in Fig. 3 (b). The associated BFG in Fig. 3 (c) shows intermediate nodes (in grey) and original node replicas. All original nodes lie on the leftmost path of the BFG structure, with a valid node ordering defined as $\pi_G =: \{a \to t \to s \to l \to b \to e \to d \to x\}$ (discussion of a different order is given in Appendix C.2). The node replicas are those variables with the same label as the original nodes but which do not lie on the left hand side of the model. The node replicas have CPDs that ensure consistency e.g. $p(t|s,t) = p(t)$. The $E_j, j = 1...9$ nodes are intermediate nodes that are not replicas and have uniform CPDs.

The triangulated graph in (b) shows that large cycle in $M[G]$ can be split into triplet cycles. Using the BFG G' (defined by π_G) we can verify all moral and chordal edges in (b) correspond to moral edges in (c), so the triplet cycles in (b) are contained in the BFG's moral graph $M[G']$. Hence $M[G']$ contains more sufficient triplet cycles than $M[G]$, i.e., $M[G]$ in (a) does not contain $\{s, l, b\}$. When given a bounded cluster size three we can benefit from using $M[G']$ rather than $M[G]$ to obtain better approximate accuracy.

However, if the triangulated graph $G_\mathcal{T}$ is not a planar graph not all triplet cycles in $G_\mathcal{T}$ are contained in $M[G']$. We will only select triplet cycles by following $M[G']$, because selecting all triplet cycles in $G_\mathcal{T}$ can even distort the approximation (as shown in section 3.3).

It is possible that multiple node orderings are valid, as long as they do not violate the conversion step (2). Different valid orderings will, of course, result in different BFGs, but they will not change the parent-child node relationships in BN G. So, despite using different BFGs, the parent-child node relationships can still be maintained in the BFG, but with a different configuration of node replicas. We can still obtain the same outer regions despite using different BFGs (as shown in section 3.4).

A κ_n BFG has reduced node indegree but the tree-width and the dimensionality of the original nodes remain the same as the n-dimensional complete DAG. So the space complexity of the BFG remains unchanged and the original high dimensional BN inference problem remains high dimensional. Since a BN's (tree-width+1) is greater than or equal to the maximum factor size, to reduce space complexity we can always restrict the maximum cluster size, bound it to the maximum factor size and find the optimum region graph based approximation. Next, we can increase the maximum cluster size subject to the memory constraint to improve the accuracy.

In addition, if a BN is already a binary factorized model the interaction information between the original nodes will be preserved in the BFG. Otherwise, the interaction information in the BFG will differ from the BN because we have binary factorized the BN and added intermediate nodes. The interaction change does not mean we lose information, because both the BN and the BFG have the same ground truth over the original nodes' marginals. However, there is a trade-off between the interaction change and the region identification efficiency. It may be less preferable for an interaction change when the optimum regions are identified easily in the BN. However, large, complex models will benefit from interaction change in our method as the region graph is built automatically and hence more efficiently (we show the interaction change comparisons for approximation quality in section 5.2).

Specifically, we gain crucial benefits when using a BFG model:

- It reduces the maximum node indegree and this allows our region-based algorithm to build regions efficiently involving only triplets, whilst other algorithms might have more candidate regions to consider when the node indegree increases (shown in section 3.3 and also by experiments in section 5);

- The triplet regions can be constructed more effectively and automatically in the BFG, as the moralized graph of a BN may contain fewer triplet cycles than its BFG (shown in Fig. 3 and section 5.2). This also helps scale up to using higher-ordered region sizes by merging smaller regions (shown in section 4.1).

- Given the BFG is uniquely defined it ensures that our corresponding region graph satisfies desirable properties, which are not guaranteed by other algorithms (shown in section 4.1).

3. Outer Region Identification

Our TRC algorithm uses an Outer Region Identification (ORI) algorithm and its efficiency optimization to identify outer regions, and the rest of the regions are generated by the CVM algorithm. ORI automatically identifies and adds outer regions to improve accuracy, while candidate regions that are not effective are removed. ORI is a two-stage region selection procedure. It uses conditional entropy to derive a structural property to directly select regions at the first stage. Then, at the second stage, we use each candidate region's factor information to select from those competing regions should they satisfy the same substructural property. All candidate regions identified as redundant regions are then removed from the region graph.

Section 3.1 introduces the maximally exhaustive property for adding candidate regions. Section 3.2 and 3.3 describe how redundant regions are identified for removal. Section 3.4 describes using different BFGs to obtain the same outer regions. Section 3.5 concludes the ORI algorithm.

3.1 Adding Interaction Triplets

Because all BNs are a subset of complete DAGs, the BNs factorized by the BF algorithm are also a subset of BFGs, i.e., the BFG contains more variables, but the ground truth of the original nodes' marginals is the same as the original BN. Hence, our ORI algorithm is established on the BFG model and can, therefore, be applied to all BNs. We can include all triplet factors as outer regions at the first level of a CVM region graph, which will then produce a valid region graph. However, using only triplet factors as outer regions will fail to incorporate some pair-wise information and hence will decrease accuracy. For example, the region graph in Fig. 1 (d) failed to generate an interaction for the pair wise information $\{E_1, X_3\}$ so node X_4 will not be approximated well because it depends on the pair wise information $\{E_1, X_3\}$. This means, to incorporate all necessary pair-wise information encoded in the triplet factors we need to add extra outer regions to generate intersections of pairs of outer regions to form the second level of the region graph. These intersections are node pairs in a triplet factor. Ideally, we should include all these node pairs as second level regions. Hence, we introduce the following maximal exhaustivity property to achieve that.

Definition 5: *Maximal Exhaustivity property*: A region graph satisfies this property if any *maximum membership subset*[4] of the outer region that contains a factor converted from the original BN is included in at least one second level region.

The ORI algorithm will first construct a region graph to satisfy the maximal exhaustivity property to ensure all higher ordered (with the order up to the outer region size -1) local interactions are incorporated. When outer regions are all triplets we will need to generate local pair-wise interactions and this can be achieved by defining two types of outer region members for our BFG models: primary triplets and interaction triplets.

Definition 6: A *primary triplet* $\mathcal{F} = (\mathcal{V}_{X_j}, \phi)$ is a triplet with nodes $\mathcal{V}_{X_j} = \{X_i, X_j, X_p\}$ in the moral graph $M[G']$ and a factor ϕ defined by the conversion from the CPD $p(X_j|X_i, X_p)$ in the BFG, G', as a child variable X_j and its two parents X_i and X_p.

Definition 7: An *interaction triplet* $\mathcal{U} = (\mathcal{V}, \mu)$ is a triplet with factor μ defined as a uniformly distributed factor, and triplet nodes $\mathcal{V} \in M[G']$ where $\mathcal{U}$ is not a primary triplet.

We also define:

Definition 8: The *maximum membership subset* of a primary triplet $\mathcal{F} = (\mathcal{V}_{X_j}, \phi)$, Ω_{X_j} (where X_j is a child node of X_i and X_p), is the set of combinations of all node pairs in $\mathcal{V}_{X_j}$:

4. A set of combinations of all nodes included in the outer region with the set size equal to the outer region size minus one.

$\{X_i, X_j\}$, $\{X_i, X_p\}$ and $\{X_j, X_p\}$.

The shared nodes between two primary triplets will mostly contain a single node only and this fails to satisfy the maximal exhaustivity property. However, by adding interaction triplets we can use node pairs that belong to different primary triplets and this creates a maximum subset via which two or more primary triplets can interact.

Consider again the Fig. 1 (d) region graph where pair-wise interaction is identified using only primary triplets factors but which does not meet the maximally exhaustive property. For example, the node pair $\{E_1, X_3\}$ is not shared by any two triplet factors but it influences the accuracy of node X_4. This lack of interaction can be resolved by adding the interaction triplet region $\{X_2, E_1, X_3\}$ to the region graph. Similarly, other pair-wise correlations will also be incorporated by adding other interaction triplets. Hence, to satisfy the maximal exhaustivity property second-level regions in our region graph will all be formed by an exhaustive set of all possible pair-wise interactions among all triplet factors.

As all primary triplet factors are CPD conversions from a BFG they are already identified in the moral graph and we can use the moral graph to help us identify interaction triplets, using what is called a coupled Markov Blanket:

Definition 9: *A coupled Markov Blanket* for edge nodes (X_i, X_j) is the set of nodes $\partial(X_i, X_j)$ composed of X_i and X_j's Markov blanket excluding nodes $\{X_i, X_j\}$. Therefore $\partial(X_i, X_j) = (\partial X_i \cup \partial X_j)_{\setminus(X_i, X_j)}$.

The coupled Markov Blanket limits the number of candidate nodes that will be used to generate candidate interaction triplets for a node pair $\{X_i, X_j\}$, while ensuring the maximal exhaustivity property is satisfied.

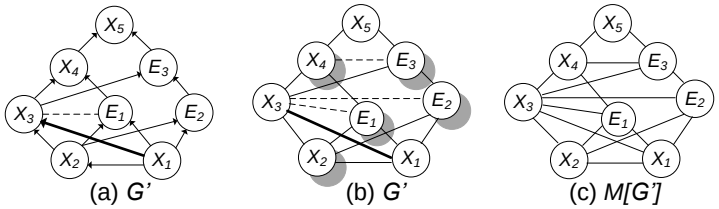

Figure 4: (a) κ_5 BFG with a moral edge shown as dashed line; (b) all nodes in the coupled Markov Blanket $\partial(X_1, X_3)$ in (a), shown with nodes shadowed; (c) moral graph of (a)

Fig. 4 (a) shows a primary triplet $\{X_1, X_2, X_3\}$ and an interaction triplet $\{X_1, X_3, E_1\}$ interacted via an edge (X_1, X_3) (shown as a bold solid line). In Fig. 4 (b), to identify interaction triplets for edge (X_1, X_3) (shown as a bold solid line), we first identify the coupled Markov Blanket $\partial(X_1, X_3) = \{X_2, E_1, E_2, X_4, E_3\}$ (shown with nodes shadowed). From this, the candidate interaction triplets are easily identified as: $\{X_1, X_3, E_1\}$, $\{X_1, X_3, E_2\}$, $\{X_1, X_3, X_4\}$, $\{X_1, X_3, E_3\}$ and $\{X_1, X_3, X_2\}$. Notice that $\{X_1, X_3, X_2\}$ is then excluded as it is a primary triplet. All maximum subsets, as node pairs, are identified as the edges in $M[G']$ as shown in Fig. 4 (c).

Therefore, adding interaction triplets within a coupled Markov blanket ensures all local pair-wise correlations are incorporated.

This method of adding new regions to create interactions is also used in (Welling, 2004) on the assumption that adding regions could improve the accuracy. However, we do not share this assumption because some regions, introduced by the maximal exhaustivity property, may be redundant and would not change all the singleton marginal inference results if retained. Also, unlike the cluster pursuit algorithm (which involves message passing to compute a cost function to test the potential improvement along with the newly added regions and which is very inefficient), we use both structural and factor information to determine redundancy in an entirely localized way.

Note that if we do not remove redundant regions the resulting region graph will not satisfy the perfect correlation property. Therefore, our ORI algorithm will first identify all outer regions that satisfy the maximally exhaustive property and then identify redundant regions to reject. We identify two kinds of redundant interaction triplet regions that must be rejected during ORI. We refer to them as an *incomplete interaction triplet* and a *competing interaction triplet*. The basic principle for deciding whether to include an interaction triplet is to determine if the triplet changes the entropy of any node in the model. Obviously, evaluating the entropy change of all nodes is computationally expensive. Therefore, we will instead use the CI structural information to identify redundancy and then use factor information to refine the redundancy identification further. Also, entropy information change is determined using all primary triplets, and no other interaction triplets, except the one under evaluation. We evaluate the interaction triplet without evidence because that will not change the factor information and also helps localize the computations.

Section 3.2 defines and discusses redundancy introduced by incomplete interaction triplets and section 3.3 defines and discusses redundancy introduced by competing interaction triplets.

3.2 Identifying Redundant Interaction Triplet Using Structural Information

Definition 10: *Incomplete interaction redundancy.* This is redundancy introduced by an interaction triplet that provides no singleton entropy information change for any node in $M[G']$.

To identify incomplete interaction redundancy, we use the following structural information:

Definition 11: *Incomplete interaction triplet* is an interaction triplet containing a node pair (edge) that does not exist in $M[G']$; If all node pairs in the interaction triplet are contained in $M[G']$ it is a *complete interaction triplet*.

Proposition 2: *Incomplete interaction redundancy can be identified by using incomplete interaction triplets.*

Proposition 2 is proved in Appendix A.2.

So, we change the original entropy evaluation problem (which is inefficient) to a structural identification problem (which is efficient). Proposition 2 means that all incomplete interaction triplets are redundant and hence the remaining (complete interaction triplets)

are retained. Note that, although all incomplete interaction triplets are removed, our proof of Proposition 2 is based on using a BFG model. Hence, evaluating one incomplete interaction triplet at a time is sufficient.

If a BN is sparser than its BFG version then it is possible that node entropies will change if two incomplete interaction triplets (that are rejected) are together derived from the BN rather than the BFG. An incomplete interaction triplet in a BN could become a complete interaction triplet in a BFG, so by using the BFG we will not "miss" such an interaction triplet.

For instance, if we use the Asia BN's moral graph (Fig. 3 (a)) to identify the incomplete interaction redundancy both $\{s, l, b\}$ and $\{l, b, e\}$ will be rejected since they are incomplete interaction triplets. However, these two triplets cannot be rejected because they together constitute a cycle path in the moral graph, resulting in node entropy changes. In contrast, using the BFG in Fig. 3 (c) we will obtain both ($\{s, l, b\}$ and $\{l, b, e\}$), as one of them is identified as a primary triplet and the other is a complete interaction triplet in the BFG. This result is also evident in our experiments (in section 5).

We use Fig. 4 (c) to illustrate how Proposition 2 is applied to our BFG model. We use the coupled Markov blanket to generate a list of interaction triplets to satisfy the maximum exhaustivity property. This involves adding both the incomplete and complete interaction triplets, which requires a greedy search over the moral graph and is inefficient. But using Proposition 2, we can simplify the search by only querying the complete interaction triplets, which is efficient. Note that all primary triplets are already identified and can thus be excluded, this leaves only five complete interaction triplets (which all contain a node pair connecting a moral edge) remaining post search for the Fig. 4 (c) model.

Next, we can select outer regions from the complete interaction triplets. We can verify that all complete interaction triplets contain a node pair that is connected by a moral edge, and each node pair connected by a moral edge is contained in one or two complete interaction triplets.

If a node pair connected by a moral edge is only contained in one complete interaction triplet then the interaction triplet is retained, such as $\{X_3, X_4, E_3\}$ in Fig. 4 (c). If the node pair is contained in two complete interaction triplets, which must be symmetric in the moral graph, such as $\{X_1, E_1, X_3\}$ and $\{X_2, E_1, X_3\}$, it is unclear whether we should keep them both or reject one or other of them.

Selecting them both would induce a message exchange over the same moral edge and might then distort the approximation, as none of them contains exact information. Rejecting one of them is also difficult using structural information alone given they are symmetric in the moral graph. We refer to these as "competing interaction triplets". We will use the conditional entropy encoded in the region belief to select an optimal region directly from the competing regions without performing message passing.

After selecting one of the two competing interaction triplets, each node pair connected by a moral edge will be exclusively included in only one interaction triplet, so the number of the interaction triplets finally obtained equals the number of the moral edges. Therefore, our region selection procedure is very efficient because we only need to iterate each moral edge in the moral graph of a BFG and choose one interaction triplet optimally for each moral edge.

3.3 Identifying Redundant Interaction Triplets Using Factor Information

In this section, we use factor information to reject one of the competing interaction triplets as using structural information alone is insufficient. We will use the conditional entropy information encoded in the region belief to select an optimal region directly from competing regions without performing region graph inference. The competing interaction regions are formally defined below:

Definition 12: *Competing interaction triplets* are two complete interaction triplets $\{i, j, k\}$ and $\{p, j, k\}$ composed using the same shared nodes $\{j, k\}$ connecting a moral edge, and a pair of parent nodes $\{i, p\}$ of the nodes $\{j, k\}$.

Definition 13: *Competing interaction redundancy.* This is redundancy introduced by an interaction triplet that has a less accurate approximation to the pair-wise joint distribution of the moral edge than its competing interaction triplet.

Note that it is not possible for the two nodes connecting a moral edge to share more than two parent nodes because this would violate the BF process. Therefore, each time the number of competing interaction triplets is equal to two.

Proposition 3: *Competing interaction redundancy cannot be identified using structural information alone.*

The proof of Proposition 3 is given in Appendix A.3.

Adding both the competing interaction triplets breaks the perfect correlation property, so our best option is to choose one of them (assuming the maximum cluster size is bound to equal the maximum factor size). Given proposition 3, we instead use factor information to determine the choice of the competing regions.

Rather than analysing the factor information by optimizing the bounds of the partition function globally, we use a local method that rejects one candidate region and accepts the other.

We use conditional entropy decomposition for the exact joint entropy (Cover & Thomas, 2006). Given an elimination order $\mathbf{e}$, the exact joint entropy can be decomposed to:

$$H(X_1, ..., X_n) = \sum_i H(X_{e(i)} | X_{e(i+1),...,} X_{e(n)}). \tag{6}$$

The r.h.s of Eq. (6) includes all conditional distributions over the variables X_1 to X_n defined by the elimination order, so the size is exponential. Restricting the number of conditioning variables cannot decrease the entropy and gives us an upper bound on $H(X_1, ..., X_n)$. The Conditional Entropy Decomposition (CED)(Globerson & Jaakkola, 2007; Hazan et al., 2012) approach constructs an upper bound for the partition function by restricting the number of conditioning variables to be in the predefined clusters. The CED approach solves an optimization problem on the partition function but it does not determine which region to select. We do not compute the actual upper bound for the exact joint entropy, but only compute a relative value. This involves little extra computation, given

that the maximum cluster size is three, and can be done without the need to perform region inference. We can also restrict the conditioning variables for our entropy decomposition by using the pre-defined regions.

We only need to consider the κ_4 BFG structure, shown in Fig. 1 (a) because that is where the competing interaction redundancy occurs in the BFG model. Any higher order BFG contains multiple κ_4 BFG structures. For example, Fig. 4 (c) contains two κ_4 BFG structures, so we can easily extend the approach.

Next, using the Fig.1 (a) model, and assuming a cluster size of three, we cannot compute the marginal $p(E_1, X_3)$ exactly so choosing either $\{X_1, E_1, X_3\}$ or $\{X_2, E_1, X_3\}$ as the interaction gives an approximation of the true marginal $p(E_1, X_3)$. Hence, our task is to better approximate $p(E_1, X_3)$ by using a tighter upper bound of $H(X_1, X_2, X_3, X_4, E_1)$ based on the pre-defined regions.

Because the marginal $p(E_1, X_3)$ is not dependent on node X_4, we can restrict the conditioning variables to be in the set $\{X_1, X_2, X_3, E_1\}$ with a fixed elimination order $E_1 \rightarrow X_3 \rightarrow X_2 \rightarrow X_1$. We can therefore simplify the problem by tightening the upper bound of $H(X_1, X_2, X_3, E_1)$, which involves fewer variables and gives a more accurate entropy approximation.

Furthermore, we do not need to calculate the actual upper bound of $H(X_1, X_2, X_3, E_1)$ given that we are only interested in the relative difference between the two choices. We can also further restrict the conditioning variables to be in the marginal set of our pre-defined regions, R.

Thus, based on Eq. (6) for a four-node distribution we have:

$$\sum_i H(x_{e(i)}|x_{e(i+1)}, ..., x_{e(4)}) \leq \sum_i H^j(x_i|x_{r\backslash i}; b_r), \tag{7}$$

where $i = 1, ..., 4$ and $j = 1$ or 2 is the choice of the competing interaction triplet. $H^j(x_i|x_{r\backslash i}; b_r)$ is the conditional entropy of the j^{th} choice, and b_r is the belief of region r ($r \in R$) containing the variable i. Note that Eq. (7) is the sum of the restricted conditional entropies in the marginal set of the pre-defined regions.

The difference between the two choices is dependent on the interaction triplets only since other regions are simply the same set of primary triplets (with the same elimination order and region counting number). Thus, we can use the following equation to simplify the problem:

$$\Delta H^j = \sum_i H^j(x_i|x_{r_j\backslash i}; b_{r_j}), i = 1...4, \tag{8}$$

where r_j is the interaction region by the j^{th} choice, and ΔH^j is the conditional entropy corresponding to r_j.

Therefore, to calculate ΔH^j we need to compute the belief b_{r_j}. For our example (Fig. 1 (a)) we start with the following equations:

$$p(X_1, X_3) = \sum\nolimits_{X_2} \phi_{X_1}\phi_{X_1X_2}\phi_{X_1X_2X_3},$$
$$p(X_1, E_1) = \sum\nolimits_{X_2} \phi_{X_1}\phi_{X_1X_2}\phi_{X_1X_2E_1},$$

$$p(X_2, X_3) = \sum\nolimits_{X_1} \phi_{X_1}\phi_{X_1X_2}\phi_{X_1X_2X_3},$$

$$p(X_2, E_1) = \sum\nolimits_{X_1} \phi_{X_1}\phi_{X_1X_2}\phi_{X_1X_2E_1}.$$

These values are computed exactly. Next, we can compute the belief:

$$b_{r_j} = \sum_{X_j} b(X_j, E_1, X_3) = \sum_{X_j} p(X_j, X_3)p(X_j, E_1)/p(X_j), \tag{9}$$

where the value of b_{r_j} is the same value under the choice of $\{X_j, E_1, X_3\}$ as the interaction triplet, by performing region inference. Finally, we will use the conditional entropy ΔH^j to identify the competing interaction redundancy in the competing interaction triplets and choose the one with the smaller value. The other is therefore redundant and is removed.

In BNs that are not BFGs, the singleton and pair-wise marginals associated $p(X_j)$ in Eq. (9) are not directly obtainable if X_j is not a leaf node. However, we can obtain the pseudo-marginal[5] instead for the marginals associated with X_j by using the CPD $p(X_j|X_{pa(j)})$ and also the CPDs of these parents' $p(X_{pa(j)}|X_{pa\{pa(j)\}})$ i.e. the grand-parents. In our experiments, we used the CPDs $p(X_j|X_{pa(j)})$ and $p(X_{pa(j)}|X_{pa\{pa(j)\}})$, and set $p(X_{pa\{pa(j)\}}) = \mathbf{1}$ when computing the pseudo-marginal of X_j. We found that the error rate for computing the conditional entropy ΔH^j by using the pseudo-marginal in the above setting compared to using the exact $p(X_j)$ is around 6% over 150 random factors [6]. Clearly the calculation of ΔH^j is localized.

We randomly generated a list of CPDs in Appendix D.1 for the κ_4 BFG model in Fig. 1 (a), and use the model as an example to demonstrate how the factor information is applied. The difference in results of choosing different interaction regions is shown in Table 1.

Table 1: Pair wise joint probability results for moral edge $\{E_1, X_3\}$ compared by choosing different interaction regions

	Joint probabilities for $\{E_1, X_3\}$				*KL*	ΔH
Exact	0.488	0.122	0.077	0.313		
$\{X_1,E_1,X_3\}$	0.473	0.137	0.092	0.298	0.0028	0.727
$\{X_2,E_1,X_3\}$	0.343	0.267	0.222	0.168	0.190	1.33

In Table 1, if $\{X_2, E_1, X_3\}$ is chosen as the interaction triplet the *KL* error is 0.19, compared to 0.0028 when $\{X_1, E_1, X_3\}$ is chosen. Also if we use $\{X_2, E_1, X_3\}$ the resulting pairwise joint probability values are nearly opposite to the true values. Despite this problem the marginal approximation for each node does not look as bad as the pair-wise joint approximation, for node X_4 *KL* is 4.0e-5 when $\{X_1, E_1, X_3\}$ is chosen and 4.3e-3 when $\{X_2, E_1, X_3\}$ is chosen.

5. The marginal distribution computed using local factors.
6. In future work we can improve the error rate by incorporating more ancestor CPDs.

When we use extreme factors (near zero and one) instead for the triplet CPDs in Table 1 the difference in results can be very large. The *KL* results for the pair-wise and singleton marginal are 7.0e-12 and 6.0e-16 if $\{X_1, E_1, X_3\}$ is chosen as the interaction triplet. In contrast, if $\{X_2, E_1, X_3\}$ is chosen the results are 0.11 and 0.04 respectively. If we retain both interaction triplets without removing competing interaction redundancy the results are 0.92 and 0.11.

In (Yedidia et al., 2005) the region-based approximation is computed by belief propagation subject to constraints using Eq. (3). The moral edge is not regularised ($\phi_{j,k} = 1$) in the BN, and multiple pair-wise solutions are possible to satisfy the region based constraints in the message passing procedure. So, even if singleton beliefs are approximated well, the pair-wise joint beliefs may not be well approximated at all, as shown in Table 1. Without the control for competing interaction regions the probabilities approximated could be "flipped", i.e., $\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$ is approximated to $\begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix}$, compared to the true values. Obviously this problem is unacceptable especially when higher-ordered marginals are needed. We refer to the problem of very poor local approximation as the *max variability* problem and it can be quantified by the max *KL* error. By using conditional entropy, ΔH, we can effectively reduce the max variability problem.

3.4 Different Node Orderings and Replacement Interaction Triplet

We use different node orderings, hence different BFGs, to illustrate how we can obtain the same interaction triplets. The two examples in Fig. 5 (a) and (c) present situations where competing interaction triplets and external nodes are used to define different valid node orderings.

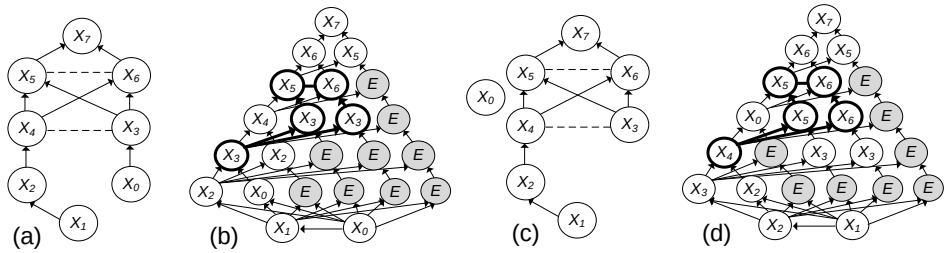

Figure 5: (a) BN G with the dashed line representing a moral edge; (b) BFG G' of (a) with the original nodes on the leftmost path and the replicated nodes labeled the same as the original nodes; (c) BN G_1 with node X_0 ordered between X_4 and X_5; (d) BFG G'_1 of (c).

In Fig. 5 (a), the BN G encodes competing interaction triplets $\{X_3, X_5, X_6\}$ and $\{X_4, X_5, X_6\}$. Fig. 5 (b) shows a BFG G' of G with a pre-defined valid node ordering $\pi_G : \{...X_3 \rightarrow X_4 \rightarrow X_5 \rightarrow X_6...\}$ where X_4 is ordered after X_3 in a clique. Suppose we had identified $\{X_3, X_5, X_6\}$ as the one to be retained from the competing interaction triplets in G. Given X_4 blocks the path from X_3 to X_5, using G' we cannot obtain the interaction triplet $\{X_3, X_5, X_6\}$ directly. Despite this, we can check if there exists another parent-child

path that introduces an interaction triplet containing the node pair $\{X_5, X_6\}$, such as the bold paths in (b). In the bold paths, X_3 and its replicas are parent nodes of both X_5 and X_6. The original node replicas are actually the same as the original nodes. This indicates both the competing interaction triplets are maintained in the BFG. In this case, we should replace the $\{X_4, X_5, X_6\}$ by $\{X_3, X_5, X_6\}$ without the need to change the node ordering for X_3 and X_4. Therefore, we have obtained the same interaction triplet even though we used different node orderings. Also, given that we are replacing one interaction triplet with another interaction triplet, the total number of interaction triplets is unchanged.

Next, in (c) we change a valid node ordering $\pi_G : \{X_1 \rightarrow ...X_4 \rightarrow X_5... \rightarrow X_7\}$ by inserting an additional node X_0 between X_4 and X_5 to the BN G_1, such that X_0 and G_1 belong to different cliques for a large BN. Here, we have defined a different valid node ordering $\pi_G : \{X_1 \rightarrow ...X_4 \rightarrow X_0 \rightarrow X_5... \rightarrow X_7\}$ and obtained the BFG G_1' in (d). Suppose we identified $\{X_4, X_5, X_6\}$ as an interaction triplet to be retained from the competing interaction triplets. Then we cannot directly obtain it using G_1' as X_0 is ordered after X_4 and blocks the path. Likewise, there is a bold path in G_1' introducing an interaction triplet $\{X_4, X_5, X_6\}$ through the original node replicas. So now we can repeat the approach and replace the interaction triplet $\{X_0, X_5, X_6\}$ with $\{X_4, X_5, X_6\}$.

We now define a replacement interaction triplet for interaction triplet $\{i, j, k\}$:

Definition 14: *Replacement interaction triplet.* For a BN G and associated BFG G', this is an interaction triplet $\{j, k, p\}$ in $M[G']$ introduced by a cycle path that is composed by a node pair $\{j, k\}$ connecting a moral edge, a shared parent node p of j and k, and the replicas of j, k, and p.

The replacement interaction triplet results from the definition of the BFG (that is factorized from a complete DAG) such that any two original nodes have a unique path to reach one from the other. So a parent-child relationship in G can always be preserved in G' through node replication. Thus, the different valid node orderings will not change the parent-child relationships of a BN.

An interaction triplet that does not have a replacement interaction triplet in the BFG will be selected directly. Therefore, despite using different BFGs, we only need to check each moral edge for an interaction triplet and its replacement interaction triplet. If the replacement interaction triplet corresponds to a triplet cycle in a triangulated graph of the BN G we should select it rather than using the original interaction triplet. Using the replacement interaction triplet also ensures the total number of outer regions will not increase and can avoid generating interactions that may distort the approximation.

3.5 Optimization and Summary of The ORI Algorithm

Converting a BN to a BFG may introduce some intermediate nodes that do not replicate any original node (such as the E nodes in Fig. 5 (b)). Outer regions created by these nodes can be safely removed. For the intermediate nodes replicating the original nodes, such as the original node replicas that have the same label with the original nodes in Fig. 5 (b), we can reuse the original nodes to replace these nodes for all outer regions as the replicas are the same as the original nodes.

We can then optimize the ORI algorithm by *node reuse* and *unused triplets removal* to reduce the number of outer regions.

Node reuse: *If the intermediate node E_j is a replicated node for the original node X_i we can change E_j to X_i directly in the outer regions (this includes both the primary and the interaction triplets) selected, since node E_j is the same as the node X_i.*

Unused triplet removal: *If E_j does not replicate any original node and is a child node of a primary triplet we can also remove this primary triplet and its connected interaction triplet sharing the same moral edge from the outer regions.*

For example, in Fig. 5 (b) we can identify a primary triplet $\{X_5, X_6, X_6\}$ that is composed by an original node X_5, an original node X_6 and its replicated node being reused by original node X_6. This primary triplet can be further simplified to $\{X_5, X_6\}$ and the region $\{X_5, X_6\}$ then becomes a subset of other outer regions. As a result, the outer regions in the region graph will be reduced as fewer outer regions are created. The outer regions associated with the E_j nodes not reused by any original node will also be removed during unused triplet removal. So there will be no E_j nodes represented in the outer regions. We provide optimization tests in section 5.2.

We summarize the ORI algorithm in Algorithm 1.

If a BN is a BFG, ORI simply collects all primary triplets and iterates through each moral edge to identify interaction triplets. If the node pair connecting a moral edge shares a single parent node then the unique interaction triplet is retained. If the node pair shares two parent nodes a competing interaction triplet with smaller ΔH is kept. If a BN is not a BFG ORI first caches all competing interaction triplets and identifies competing interaction redundancies using ΔH. Then ORI converts the BN to a BFG using the predefined node ordering. Next, it iterates through each moral edge in the BFG to find the interaction triplet or its replacement.

From Algorithm 1 we can summarize the ORI into four sub-algorithms: i. BF algorithm; ii. Selecting all primary and complete interaction triplets using the BFG of a BN; iii. Localized conditional entropy test; iv. Efficiency optimization by node reuse and unused triplet removal.

From Algorithm 1 we can verify that the number of interaction triplets equals the number of moral edges in a BFG, with a total number $(n-2)(n-3)/2$. The primary triplets are selected directly, with a total number of $n-2+(n-2)(n-3)/2$. So the space complexity for the ORI is proportional to the BFG's dimension, $[(n-2)(n-3)/2]+[n-2+(n-2)(n-3)/2] = (n-2)^2$ for κ_n, which is polynomial $\mathcal{O}(n^2)$. As all outer regions are determined by ORI we can obtain our TRC region graph, as described in section 4.

Algorithm 1: ORI algorithm

Input: a BN G
Output: Outer regions $\mathcal{R}$;
1: Perform BF to G to obtain G' (with variables $X_i, i = 1, ..., n$);
2: **if** G' *is a* κ_n *BFG* **then**
  $M[G'] \leftarrow$ parametrizing G';
  Merge all primary triplets in $M[G']$ to $\mathcal{R}$;
  for *each node pair* $\{j, k\}$ *connecting a moral edge in* $M[G']$ **do**
    Merge $\{i, j, k\}$ to $\mathcal{R}$ if j and k share a single parent i;
    Merge a competing interaction triplet to $\mathcal{R}$ if j and k share two parents;
3: **if** G' *is not a BFG* **then**
  Identify all competing interaction triplets in G';
  Define a valid node ordering $\pi_{G'}$ and convert G' to BFG $\tilde{G}'$;
  $M[\tilde{G}'] \leftarrow$ parametrizing $\tilde{G}'$;
  Merge all primary triplets in $M[\tilde{G}']$ to $\mathcal{R}$;
  for *each node pair* $\{j, k\}$ *connecting a moral edge in* $M[\tilde{G}']$ **do**
    Identify interaction triplet $\{i, j, k\}$;
    Check the replacement interaction triplet $\{j, k, p\}$ for $\{i, j, k\}$;
    Merge $\{i, j, k\}$ or $\{j, k, p\}$ to $\mathcal{R}$;
4: Perform node reuse and unused triplet removal to $\mathcal{R}$;
5: return $\mathcal{R}$;

4. The Triplet Region Construction Algorithm

The TRC algorithm is composed of three sub-algorithms:

- Outer Region Identification (ORI) with further efficiency optimization
- Region Graph Binary Factorization (RGBF)
- Concave-Convex Procedure (CCCP)

Section 4.1 presents our TRC region graph which results from applying ORI and CVM. To avoid numerical instability, in Section 4.2 we propose the Region Graph Binary Factorization (RGBF) algorithm to ensure each region has exactly two parent regions. Section 4.3 summarises the TRC algorithm and introduce the worse case BNs for testing in section 5.

4.1 TRC Region Graph and Its Extension

Now that all outer regions are determined by primary and interaction triplets, the CVM algorithm can generate the corresponding valid region graph. The resulting region graph for our BFG models contains three levels, with all first level regions having counting numbers equal to one (as all factors are included in the first level). The resulting region graph is our TRC region graph which we now show satisfies both the perfect correlation and maxent normal properties.

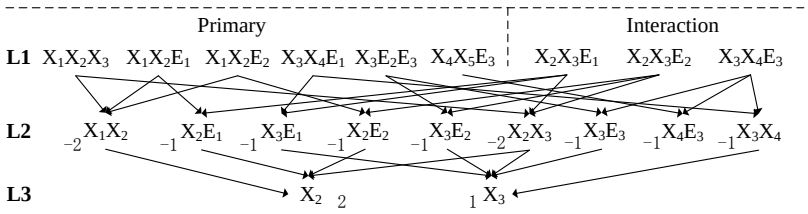

Figure 6: TRC region graph for Fig. 4 (a)

An example of a TRC region graph for Fig. 4 (a) the κ_5 BFG is shown in Fig. 6. There are two κ_4 BFGs included in the κ_5 BFG so there are two competing interaction redundancy tests involved. In Fig. 6 we have assumed our choice is X_2 (another choice is X_1) after applying the competing interaction redundancy test for each of the κ_4 BFG. The region graph constructed when a different choice for the competing interaction redundancy test is given in Appendix A.5.

There are six primary triplets and three moral edges in a κ_5 BFG. Each moral edge is exclusively contained in one interaction triplet. So there are three interaction triplets. The L_1 level of the TRC region graph is composed of all the outer regions, and the remaining levels are generated by CVM.

The number of interaction triplets and primary triplets for a κ_n BFG is determined in section 3.5. So we can summarize the properties for the TRC region graph, as shown in Table 2 (the proof of these results is given in Appendix A.5).

Table 2: Properties for κ_n $(n > 3)$ BFG region graph $\mathcal{G}$, where $v(r)$ is the number of variables included in region r, "total" is the total number of regions involved in each level of the region graph, $max(c_r)$ and $min(c_r)$ are the maximum and minimum value of the counting number c_r

Levels	$v(r)$	$total$	$max(c_r)$	$min(c_r)$
1^{st}	3	$(n-2)^2$	1	1
2^{nd}	2	$(n-2)^2$	-1	$3-n$
3^{rd}	1	$(n-3)$	$n-3$	1

Table 2 shows the region membership size $v(r)$, the number of regions contained in each level, and max, min of counting numbers in each level regions. For convenience, we summarize these result by assuming we have chosen the node X_2 after the competing interaction redundancy test for all competing interaction triplets in all κ_4 BFGs (that are included in the κ_n BFG). The different choice of the competing interaction triplet will not affect the result of our proof (shown in Appendix A.5).

We can verify that the properties in Table 2 apply to the Fig. 6 region graph, i.e., the 3^{rd} level $max(c_r) = 5 - 3 = 2$, the 2^{nd} level $min(c_r) = 3 - 5 = -2$, and both the total number of regions contained in the 1^{st} and the 2^{nd} level is $(n-2)^2 = (5-2)^2 = 9$. Using the summaries in Table 2 we can derive the sum of all regions counting numbers in

a κ_n BFG region graph, which equals one, so the TRC region graph satisfies the perfect correlation property. The TRC region graph also satisfies the maxent-normal property. We provide proofs of both properties in Appendix A.5 and A.6.

If a BN is not a BFG, ORI converts the BN to a BFG and applies the node reuse and unused triplet removal to the outer regions. These reductions will not break the perfect correlation property because node reuse does not directly remove outer regions but simplifies and merges the outer regions. The unused triplet removal process removes the primary and interaction triplets in pairs for the unused intermediate nodes without affecting other variables' local structures.

However, the perfect correlation property is not compulsory for all models. For many situations, we won't have strong correlations for all variables. Even with the relaxation of the perfect correlation property the accuracy for the marginals will not degrade. For large models, relaxing the perfect correlation property can help reduce the number of outer regions significantly and, hence improve efficiency.

If the perfect correlation property is relaxed efficiency can be optimized further by *interaction triplet removal* for the outer regions: after the previous optimization used in ORI, the number of primary triplets will be close to the number of factors of the original model. This means many interaction triplets will be left without connecting to the existing primary triplets; the role of such interaction triplets is to maintain the counting number to satisfy the perfect correlation property. Therefore, we can remove them provided that other interaction triplets connecting existing primary triplets are not affected.

Interaction triplet removal: *an interaction triplet can be removed if it is not connected to any existing primary triplet and also does not affect other interaction triplets connecting existing primary triplets.*

This optimization is explored as extended work in Appendix B.2. It is important because we show that by using these optimizations we can achieve similar efficiency to other algorithms while still achieving greater accuracy.

The cluster size of the current TRC region graph is three. We can extend the TRC region graph to use higher cluster size if higher accuracy and higher ordered marginals are needed. For a cluster size four region graph, we can merge the triplet regions from the current region graph to produce new outer regions with cluster size four, and remove unnecessary size four outer regions to optimize efficiency (and also avoid creating unnecessary interactions which might be harmful). In a BFG, we know the exact number of parent nodes that each node pair connecting a moral edge is dependent on. So we can guide the merge process and remove those outer region that introduce no new interaction information for a given moral edge. The resulting region graph will still satisfy the perfect correlation and maxent normal properties (proven in Appendix A.7). We emphasize that to build the size four cluster TRC region graph we still have to build the triplet TRC region graph first. The size four TRC region graph or the higher ordered region graph is only obtained by the guided merging. Therefore, the TRC region graph can use arbitrary (≥ 3) cluster size which is flexible.

We continue using the κ_5 BFG in Fig. 4 (a) as an example. Suppose we have obtained the TRC region graph with all the outer regions being triplets, shown in Fig. 6. To obtain a size four TRC region graph, we can iterate each node pair $\{j, k\}$ connecting a moral edge

in the BFG and find all the parent nodes on which the node pair $\{j, k\}$ is dependent to guide the merging.

There are three moral edges in Fig. 4 (a): $\{X_3, E_1\}$, $\{X_3, E_2\}$ and $\{X_4, E_3\}$. The node pair connecting the moral edge $\{X_3, E_1\}$ is dependent on the pair-wise information of $\{X_1, X_2\}$. Thus, to compute $p(X_3, E_1)$ exactly we need to include four nodes $\{X_1, X_2, X_3, E_1\}$. So we should merge the primary triplet $\{X_1, X_2, X_3\}$ to the interaction triplet X_2, X_3, E_1. The resulting size four cluster has introduced the pair-wise information of $\{X_1, X_2\}$, which is new information compared to the singleton information of X_1 or X_2 introduced by only triplets before.

Likewise, we can merge the primary triplets $\{X_3, X_4, E_1\}$ and $\{X_3, E_2, E_3\}$ to the interaction triplet $\{X_3, X_4, E_3\}$ respectively to obtain two size four clusters $\{X_3, X_4, E_1, E_3\}$ and $\{X_3, X_4, E_2, E_3\}$, which will introduce new pair-wise information $\{X_3, E_1\}$ and $\{X_3, E_2\}$ for better approximating $p(X_4, E_3)$. If we are given a cluster size five, we can then keep merging the two size four clusters to obtain a size five cluster $\{X_3, X_4, E_1, E_2, E_3\}$, and $p(X_4, X_3)$ can be computed exactly. We did not merge any region to another region without guiding, so the total number of outer regions does not increase after the merge. The first level of the region graph now contains a mixture of size three, size four and size five outer regions, and the total number of outer regions is 4 (compared to 9 initially).

TRC can be also viewed as a bottom-up approach that finds all the optimum interaction triplets while maintaining the desired region graph properties. So, approximation quality is ensured by using the smallest possible (triplets) cluster size. Therefore, TRC can be more accurate than other algorithms given a bounded cluster size. Accuracy is then improved by guided merging when a higher-ordered cluster size is used. In contrast, other bottom-up algorithms, such as FCB, do not justify how to improve accuracy and nor can they automatically decide how to generate regions under bounded cluster size. Also, during the merge TRC does not add as many large regions as the greedy approaches; these approaches, such as cluster pursuit (Sontag et al., 2008) suffer from a greedy merging and, as a consequence, the computation can quickly become a bottleneck.

4.2 Region Graph Binary Factorization Algorithm

We use CCCP (Yuille, 2002) to perform message passing over the TRC region graph. However, both GBP and CCCP can be numerically unstable, so may prevent convergence if the region graph contains large values for the counting numbers and many cycles associated with the child regions. There is a so called damping technique (Yedidia et al., 2005; Jaimovich et al., 2010) using a weighted message of the old and the new messages to help improve numerical instability but it is not very effective when the region graph present large counting numbers and multiple cycles. Therefore, one has to avoid creating a cyclic region graph and perform optimizations of the message updating procedure to improve the numerical stability. These methods are tied to the message passing procedure and may not be generally applicable.

We instead solve both the large counting number and multiple cycles problems directly from the region graph by proposing a Region Graph Binary Factorization (RGBF) algorithm, which can be applied to both GBP and CCCP to effectively improve numerical

instability.

Definition 15: A *Region Graph Binary Factorization* (RGBF) algorithm is one that ensures that each region in a region graph, originally with more than two parents, has exactly two parents without changing the validity of a region graph.

RGBF solves the numerical problem by avoiding excessive message passing from altering the region graph structure. This differs from other methods that are tied to message passing algorithms. So, RGBF can effectively improve the numerical instability problem for both CCCP and GBP message passing.

Recall that $c_r = 1 - \sum_{r' \in Ancestor(r)} c_{r'}$, and so a large absolute value of counting number also implies a global multiplicity of connected regions. For the BFG model, in Table 2, the connections between first and second level regions grow because the $min(c_r)$ is linearly decreasing (conversely $max(c_r)$ is linearly increasing), which means the number of multiple connections grow and we are guaranteed to encounter a numerical instability problem from multiple cycles in the region graph. To reduce the absolute value of the counting number and decompose the multiple connections within a region graph we apply the RGBF algorithm that is described in Algorithm 2.

Algorithm 2: RGBF algorithm

Input: k-level CVM region graph $\mathcal{G}$;
Output: k-level CVM region graph $\mathcal{G}'$;
1: $\mathcal{G}' \leftarrow \emptyset$;
2: Copy all levels regions in $\mathcal{G}$ to $\mathcal{G}'$ without region connections;
3: **for** i = *level* 2 *to level* k **do**
  for *each region* r *in level* i **do**
    if *the number of parent regions* $p_r > 2$ **then**
      create $p_r - 1$ copies r'_z of region r, $z = 1...p_r - 1$ in level i;
      Connect each region r'_z to two parent regions of r and ensure the neighboring r'_z share only a single parent region of r;
      Connect all child regions of r to all copies r'_z;
    else *the number of parent regions* $p_r <= 2$
      Copy all parent and child region connections for r in $\mathcal{G}$ to $\mathcal{G}'$;
4: Assign counting numbers to all regions in $\mathcal{G}'$;
5: return $\mathcal{G}'$;

The main idea of Algorithm 2 is to create copies for regions that have more than two parent regions. Then each copy is restricted to connect to two parent regions while ensuring the neighboring copies are connected through one shared parent region, so the copies of regions are consistent. The benefit of applying the RGBF algorithm is that large counting numbers no longer occur and multiple connections are decomposed into local connections. Therefore, the number of cycles in the region graph is reduced to a minimum.

This RGBF algorithm will be used to generate an equivalent region graph $\mathcal{G}'$ from the original region graph $\mathcal{G}$, with the equivalence properties described in the following proposition:

Proposition 4: *By applying the RGBF algorithm we transform a k-level CVM region graph $\mathcal{G}$ with all factors included in the 1^{st} level, into an equivalent k-level region graph $\mathcal{G}'$, such that each region r in $\mathcal{G}'$ ($r \in R, r \notin R_{1^{st}{}_{level}}$) is connected to two parents. The counting numbers for all regions are 1, -1 and 0. This does not change the consistency and unity* [7] *properties of $\mathcal{G}$.*

We provide the proof in Appendix A.4.

The time and space complexity for RGBF is proportional to $\sum_{j=1}^{t} c_{r_j}$, where t is the number of regions with a counting number $|c_{r_j}| \geq 2$. The RGBF algorithm simply splits the region into copies when the counting number $|c_{r_j}| \geq 2$ and ensures each region is connected to at most two parent regions.

An example is shown in Fig. 7.

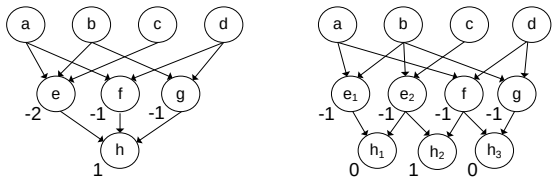

Figure 7: (a) region graph $\mathcal{G}$ (all 1st level regions' counting numbers 1); (b) region graph $\mathcal{G}'$ resulting from (a) by RGBF

In this example, regions e and h are copied twice and three times respectively. The counting numbers for each 2nd level region becomes -1, and for each 3rd level region becomes 0 or 1. It does not matter if 1 is placed on h_1 or h_2 since it does not change the consistency and unity conditions, but it will influence the convergence speed. All regions h_1 to h_3 have the same belief as they are still connected through their parent regions. Likewise, region e_1 to e_2 also have the same belief.

If we use the RGBF algorithm for a multiply connected CVM region graph, the CCCP updating will be numerically robust. For example, in Fig. 7 (a) updating the parent-child message $\lambda_{a\to e}$ in CCCP equations (Yuille & Rangarajan, 2003) involves the belief calculations of seven regions at a time: a, b, c, e, f, g and h. This number grows with multiple connections for h and the number of cycles associated with h also grows (there are three cycles associated with h in Fig. 7 (a)). But after applying RGBF to (a), as shown in (b), to update $\lambda_{a\to e_1}$ there are now 5 regions (a, b, e_1, e_2, h_1) and this number does not increase with the number of connections because there are no multiple connections and only one cycle, maximum, for each level three region. The large counting number for all levels of the region no longer exist and therefore under/overflow problems are avoided.

7. A variable has unity when the sum of all regions counting numbers associated with that variable is one.

4.3 TRC Algorithm and The Worst Case BN Examples

The TRC algorithm (Algorithm 3) is a sequential combination of ORI and its optimization, RGBF, and CCCP.

Algorithm 3: TRC algorithm

Input: a parameterized BN G with variables V;
Output: G with marginal distributions;
1: Obtain outer regions $\mathcal{R} \leftarrow ORI(G)$;
2: Interaction triplet removal of $\mathcal{R}$ if relaxed the perfect correlation property;
3: Obtain region graph $\mathcal{G} \leftarrow CVM(\mathcal{R})$;
4: Obtain region graph $\mathcal{G}' \leftarrow RGBF(\mathcal{G})$;
5: Perform CCCP message passing (in parallel) to $\mathcal{G}'$;
6: Return marginal distribution for variables V;
7: Guided merge of $\mathcal{R}$ if given higher ordered cluster size and repeat 3-6;

TRC can use the CCCP message updating in parallel because to update each CCCP message only limited regions are involved in the computation. The space complexity of TRC is the sum of all levels' regions and is polynomial and proportional to $\sum_{3\ levels} v(r) \cdot total$ (as shown in Table 2) for BFGs. This contrasts with the exponential complexity for exact methods. Therefore, for a κ_n BFG with all binary variables the space complexity for TRC is $\mathcal{O}(n^2)$ while the exact method is $\mathcal{O}(2^n)$.

The time complexity is proportional to the number of the 1^{st} to the 2^{nd} level region edges, which is the sum of all 2^{nd} level regions degree of freedom, $\sum_{j=1}^{(n-2)^2} (|c_{r_j}| + 1)$, and is polynomial. Proof of these results is given in Appendix A.8.

As mentioned in section 2.3, when the BN is a binary factorized model, the interactions between original nodes in the BN will be preserved in the BFG, and the interaction information will be captured without change when using TRC.

If the BN undergoes a binary factorization then additional nodes are added, transforming the interaction information in the original nodes into another form via the new intermediate nodes in the BFG. This transformation will not change the exact distributions for the original nodes between the BN and the BFG. When restricted by a bounded cluster size, the interactions in the BN may produce a more accurate result than using the BFG, so there is a trade off between the efficiency of region construction and accuracy. For instance, when the BN is complex, identifying the optimum regions is very inefficient as there are many choices to consider, but by using the BFG it can be made very efficient. Accuracy is then improved by a guided merge presented in section 4.1.

Therefore, we test the interaction change introduced by the BF for the "worst-case" BNs, which are complex and containing competing interaction triplets that encode strong interaction information. We also show that, compared to these worst cases, applying the interaction change to BNs that do not encode competing interaction triplets will have minimal or zero side effects when given bounded cluster size.

Here we present a "worst-case" BN containing dense structures in $M[G]$ and with induced maximum cluster size (under exact methods) larger than maximum factor size. So we

will bound our maximum cluster size to the maximum factor size to ensure the maximum cluster space is bounded.

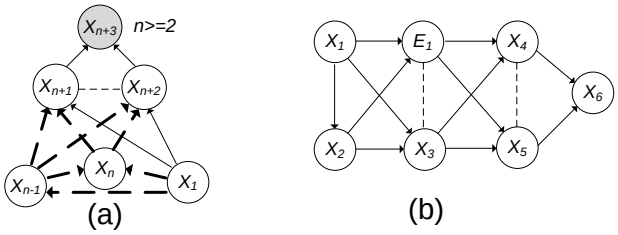

Figure 8: (a) a BN with the node pair in the moral edge (shown as a dashed line) dependent on n parent nodes (called $n+2$ dimensional dense BN with *t.w.* $n+1$). The child node in grey is observed; (b) BF model of the 5-dimensional dense BN

In Fig. 8 we present a class of BNs encoding multiple competing interaction triplets. These BNs are either used in practical applications or embedded in other BNs as sub-structures. They are the worst cases as their moral graph contains dense sub-structures and also reflect how moral edges are involved in competing interaction triplets. If the moral edge is not approximated well the subsequent parent-child distributions in the BN will also be poorly approximated. The node pair in the moral edge in Fig. 8 (a) ($(n+2)$ dimensional dense BN) depends on n parent nodes and the parent nodes are all densely connected. When n increases the BN encodes a multiplicity of competing interaction triplets. If we bound the cluster size to the maximum factor size the gap between the cluster size and the (tree-width+1) is always a constant value of one. Fig. 8 (b) is the binary factorized model of (a) with $n = 3$.

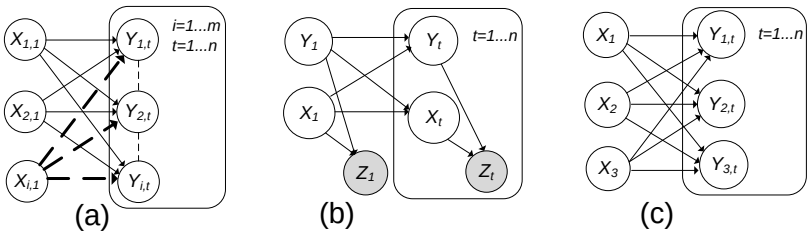

Figure 9: (a) Dynamic BN with n time slices with all child nodes dependent on m parent nodes (called $m \times n$ DBN with *t.w.* $2m-1$); (b) a coupled HMM model (Barber, 2012); (c) a dynamic BN of a Hopfield network (Barber, 2012).

Fig. 9 (a) ($m \times n$ DBN) is a dynamic BN with $m \times n$ dimensions where m determines the tree-width and n determines the temporal length of the BN. Here the child node depends on all m parent nodes. The difference of the bounded cluster size and the (tree-width+1) increases linearly with m. The moral graph of Fig. 9 (a) contain dense substructures and an increase in the number of parent nodes results in a corresponding increase in the number of competing interaction triplets and this makes manual region identification difficult if

not impossible (given a bounded cluster size). Fig. 9 (b) is a coupled HMM and (c) is a Hopfield network (Barber, 2012), which can be applications of (a) in practice. So, these model classes (Fig. 8 and 9) reflect the most critical situation for the interaction change for TRC and also present the worst case for all algorithms. We contrast TRC with other algorithms using these tests in section 5.

5. Experiments

To demonstrate the importance of the region choice dilemma and numerical instability problems that we have highlighted, we concentrate on experiments involving public, challenging BNs and associated BFGs with increased dimensionality and tree-width. We compare algorithms by using marginals in our tests.

Section 5.1 explains the experimental settings. In Section 5.2 we test the ORI algorithm in an incremental way to present its advantage over competing algorithms. Section 5.3 tests the RGBF algorithm using the high dimensional BFG models. Section 5.4 tests a wide range of models in practice and compare TRC with the state of the art algorithms.

5.1 Experimental Settings

For a fair comparison, except where mentioned explicitly, in our tests the maximum cluster size is bounded to equal the maximum factor size in a BN for each algorithm. So all algorithms have the same maximum cluster space. Hence, we compared the singleton, pair-wise, and triplet marginals under the following conditions: 1) Except for TRC, which executes a BF model of the BN, the other algorithms use the BN. 2) in a BN the parent-child joint distribution can be always computed by the joint distribution of the moral edge multiplying with the parent-child CPD. So except where mentioned explicitly, if the maximum factor size is three, we will compare the pair-wise marginals of the moral edge. Likewise, we compare the triplet marginal (composed from nodes belonging to the moral edges) of the parent nodes if the maximum factor size is four.

Except for the BFGs introduced in this paper, all other test models are publicly obtainable. For simplicity we use binary variables for all the original nodes. For random generated factors we define "normal factors" being generated by a uniform distribution over $[0, 1]$ and "extreme factors" (marked by "*") are random factors near zero and one. The closer factor values (normalized) to zeros and ones, the stronger the factor strength.

Except when comparing with the competing algorithms, for the "worst case BNs" we also contrast TRC with the best approximation result obtainable by trying the exhaustive set of the interaction regions post inference. We denote this result the "best region choice" (short for "Best") under a fixed cluster size constraint, despite the fact it is also not exact, and use it as a reference to justify the approximation accuracy of all competing algorithms.

The environment for testing was Java JDK 1.8, Intel i5 4300m. We also used the existing software package fastInf (Jaimovich et al., 2010), merlin (Marinescu, 2019) and runGBP (Gelfand, 2011) for references of the testing. The convergence threshold is 1e-08. The TRC code, test cases and random factors are publicly available in our code repository (Lin, 2020).

5.2 Incremental Tests of The ORI Algorithm

The ORI algorithm can be divided into four sub-algorithms: i. BF algorithm; ii. selecting all primary and complete interaction triplets using the BFG of a BN; iii. the localized conditional entropy test; iv. efficiency optimization by node reuse and unused triplet removal.

We compare ORI with the other algorithms using the following incremental steps to demonstrate the effectiveness of each sub-algorithm. 1. Test all the competing algorithms from using lower ordered to higher ordered interactions, and from fewer to more interactions. 2. Under the same ordered node interactions test if the competing algorithms can find the most efficient regions. 3. Introduce interaction change (by BF) for BN and test the ground truth found by the ORI using the BFG. 4. Introduce interaction change for the worst-case BNs to test the approximation quality of ORI. 5. Efficiency comparison for ORI with/without efficiency optimization.

Step 1 and 2 use BNs that are already binary factorized. Step 3 and 4 use BNs that need to be binary factorized for ORI, which will incur an interaction change.

In step 1, we test ORI with VI based approaches and other Bethe/Kikuchi based approaches using the well known Asia model (*t.w.* 2) (Lauritzen & Spiegelhalter, 1988). The maximum factor size in the Asia model is three, which equals the induced maximum cluster size, so the algorithm needs to find the exact solution. The Asia model is already a binary factorized model and without the competing interaction triplets, so it is only testing the sub-algorithm ii (selecting all primary and complete interaction triplets using the BFG of a BN) of the ORI.

Table 3: *KL* comparisons using *mean* (*s.d./max.*) between ORI and the competing algorithms for the Asia model

model	VI			Bethe/Kikuchi			
	MF	VMP	SVMP	Bethe	CVM	FCB	ORI
Asia	0.39	0.27	0.02	2.1E-04	2.1E-04	3.2E-06	**2.0E-12**
	(0.5/1.26)	(0.36/0.96)	(0.04/0.13)	(5.2E-4/1.5E-3)	(5.2E-4/1.5E-3)	(7.9E-6/2.3E-5)	**(2.1E-12/5.0E-12)**

Table 3 shows from left to right the order of accuracy of the different algorithms (so MF is the least accurate and ORI the most accurate, while the max variability problem decreases). The mean field (MF) (Jordan et al., 1998; Jaimovich et al., 2010) and the VMP (Winn & Bishop, 2005; Masegosa et al., 2017) algorithms use fully factorized forms for the latent variables thus they only encode singleton interactions. The structured VMP (SVMP) (Winn, 2004) captures some of the pair-wise interactions between nodes and hence its accuracy is better compared to VMP. All the three VI based approaches factorize the original joint distribution of the model to the tree-structured distributions.

In contrast, the Bethe approximation runs on a factor graph with loops, which performed better than the VI based approaches in our tests. The CVM algorithm uses primary triplets only as outer regions and the interactions generated at the second level of the region graph involve only singleton interactions. So it is equivalent to the Bethe approximation running on a factor graph for this test. FCB introduces interaction triplets for this model and performs better than the CVM. But, given a bounded cluster size three, FCB generates

less sufficient interaction triplets compared to ORI, because the cycle length found by the FCB is larger than three. As a result, if we increase the factor strength the *KL* error for FCB increases from 1.0E-06 to 1.0E-03. In contrast, ORI will not "miss" these smaller interactions as it runs on the BFG. And it also will not miss the higher ordered interaction clusters because it can merge the smaller clusters. ORI has found the exact solution for this test. So, when given a bounded cluster size, ORI has obtained better interactions than other algorithms.

In step 2, we use the BayesGrid BN (den Broeck et al., 2014) to test the ability of ORI and the competing algorithms to find the most efficient regions. A 2×2 nodes BayesGrid contains a leaf node X_1, two child nodes X_2 and X_3 depending on X_1, and a child node X_4 depending on X_2 and X_3. Our test model is a 5×5 BayesGrid (*t.w.* 5) with the last child node observed. The BayesGrid BN is effective for testing the interactions because there is only one interaction triplet that is optimum for each of the primary triplets. Adding more or different regions rather than the optimum one could degrade the approximation. We test ORI with competing algorithms that use the same ordered interactions so the VI based approaches are not compared. Again this test only verifies the sub-algorithm ii of the ORI.

Table 4: *KL* comparisons using *mean* (*s.d./max.*) between ORI and the competing algorithms for the BayesGrid model

model	Join graph/Top down		Region graph/Bottom up		
	IJGP	WMB	CVM	FCB	ORI
BayesGrid	1.5E-01	5.9E-03	5.1E-04	2.8E-08	**2.8E-08**
	(3.4E-01/1.44E+00)	(7.7E-03/2.9E-02)	(1.4E-03/7.0E-03)	(6.5E-08/2.9E-07)	**(6.5E-08/2.9E-07)**

The results in Table 4 show, from left to right, the order of accuracy (so IJGP is the least accurate and ORI is the most accurate). Increasing the i-bound for the top-down algorithms can increase the accuracy of the results, which implies the top-down algorithms introduce a loss of accuracy when constrained by a limited cluster size. In contrast, the bottom-up algorithms try to find the optimum solution from using limited cluster size. Both IJGP and WMB (run using merlin (Marinescu, 2019)) are significantly worse than the bottom-up algorithms in this test. The CVM uses only primary triplets to generate the outer regions hence the interactions generated are not sufficient. The FCB has found the optimum interaction triplets for this test, which are complete interaction triplets sharing the same moral edge with the primary triplets. The BayesGrid is already a binary factorized model so the interactions in the BN are preserved in the BFG. ORI has also obtained the optimum interaction triplets using the BFG (given in Appendix C.4) and it achieves the same accuracy with FCB in this test.

Next, we use the coupled HMM in Fig. 9 (b) to test the sub-algorithms ii and iii (localized conditional entropy test) of the ORI. The coupled HMM encodes competing interaction triplets, and to approximate the model each moral edge's joint distribution needs to be accurately approximated or the error will be propagated to the next time slice. Although the tree-width for this model is only three, the number of the interaction triplet region choices is $2 \times 2 \times 2 = 8$ for four-time slices and exponentially increases with the number of time slices. Results are shown in Fig. 10 (a) where FCB 1 or 2 indicates that

FCB chooses the y_1 or y_2 node as a root node for generating the fundamental cycles. We simulated 100 instances of random factors for the results. ORI significantly improves the accuracy of all singletons and pair-wise marginal compared to FCB. ORI achieves almost the same accuracy as the "Best" result in this test, indicating that ORI selects the correct competing regions and effectively solves the max variability problem.

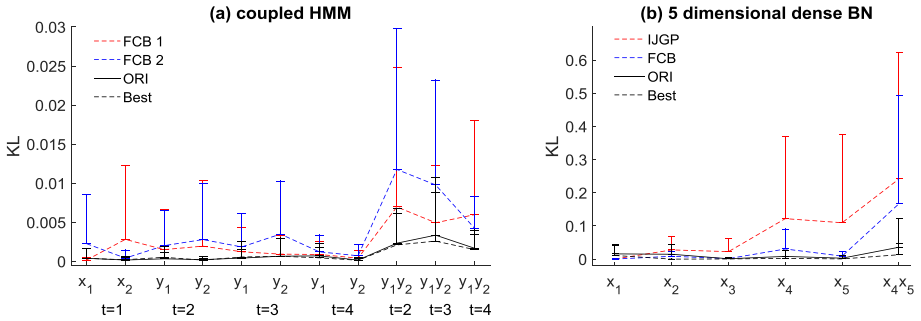

Figure 10: (a) ORI vs. FCB using the coupled HMM model with four-time slices. FCB 1 or 2 indicates it uses either y_1 or y_2 as the root node to generate the fundamental cycles. (b) ORI vs. IJGP and FCB using the 5-dimensional dense BN

In step 3 we test combination of sub-algorithms i (BF) and ii of the ORI algorithm using a BN containing cycles (in the moral graph), induced by a child node dependant with a number of parent nodes where the parent nodes share a single parent node. Because the induced maximum cluster size equals the maximum factor size in a BN, this test has a simple ground truth solution which is a cluster composed of all membership variables in the child node's factor. There is an interaction change after the BN is binary factorized, so for a successful test the ORI algorithm is required to find the exact solution using the BFG without increasing the cluster space bounded by the BN. We provide this verification in Appendix D.2, and show that ORI finds the ground truth using the bounded cluster space, as expected and the test verifies that the result using ORI is not altered by the interaction change introduced by BF.

In step 4, we test the combination of sub-algorithms i, ii and iii of the ORI algorithm using one of the worst-case BNs shown in Fig. 8 (a) (an $n+2$ dimensional dense BN). The induced maximum cluster size of the BN is greater than the maximum factor size, so if we bound the maximum cluster size to the maximum factor size our solution is approximate.

Assuming all nodes are binary and $n = 3$, there are six nodes in a 5 dimensional dense BN. The maximum cluster size is four so the maximum cluster space is bounded to 16. Likewise, we assume the BN contains extreme factors. With this setup the ORI algorithm runs on the binary factorized BN shown in Fig. 8 (b), where node E_1 is an intermediate node with cardinality four. So, given that the bounded cluster space is 16, ORI uses only triplets. We can compare the ORI result with that produced by the top-down algorithms (IJGP with maximum cluster size four) and the bottom-up (FCB) algorithms, and also test all possible size four clusters post inference to find the best outer regions.

We simulated ten instances of random factors for the result shown in Fig. 10 (b). The results obtained by ORI significantly outperformed both IJGP and FCB. The "Best" solution is slightly better than ORI, which indicates there is a relative loss of interaction information of the original nodes caused by the interaction change. After BF the interaction change introduces a relative loss of interaction information compared to the interactions obtained by the "Best" solution. ORI will not lose interaction information if the BN is already a binary factorized model, such as the coupled HMM. Nevertheless, the relative loss of the interaction information can be compensated by a guided merge of the TRC outer regions if higher ordered clusters are allowed. In addition to this test, we also stress tested ORI using the $n+2$ dimensional dense BN and the $m \times n$ DBNs with different tree-width and factor strength, as discussed in Appendix D.3. For these worst cases BNs ORI has also outperformed competing algorithms.

In step 5, we use the Asia model's BFG in Fig. 3 (c) and Fig. 5 (b) models to test sub-algorithm iv (efficiency optimization) of the ORI. They are both κ_8 BFGs so without employing node reuse and unused triplet removal the number of outer regions will be a fixed number $(n-2)^2$ where $n = 8$. After removing the unused triplets for the Asia model's BFG the number of outer regions is reduced from 36 to 22, given there are 14 outer regions containing unused intermediate nodes. After node reuse, the number of outer regions reduces further from 22 to 7, which equals the number of factors in the Asia model. After removing unused triplets, 20 outer regions remain in the model shown in Fig. 5 (b), and after reusing nodes this is reduced to 6. These optimizations will not alter the perfect correlation property and clearly with the efficiency optimization ORI has achieved comparable efficiency to that achieved using other competing algorithms. Efficiency comparison test results are provided in Appendix D.4.

We have incrementally shown the effectiveness of the sub-algorithms of ORI and how each optimisation addresses the region choice problem using simple and complex BNs. All the sub-algorithms of ORI have the potential to be used to optimise other algorithms.

5.3 RGBF and High Dimensional BFG Experiments

This section focuses on testing the RGBF algorithm and then testing TRC performance when we scale up to higher ordered BFG models and models containing a greater number of states for each node.

To investigate the effectiveness of the RGBF algorithm, Fig. 11 (a)-(c) shows the results of using the GBP updating algorithms (implemented in (AgenaRisk, 2020)) on a TRC region graph containing multiple cycles (on the κ_{10}, κ_{11}, κ_{12} BFG models) with and without RGBF. Without RGBF (but with damping) GBP demonstrates significant inaccuracy, especially evident in the lowest seven dimensions. In contrast, GBP results with RGBF achieve high accuracy in these tests. These tests also show that the low dimensional variables of a BFG model are more likely to experience numerical problems than high dimensional variables since low dimensional variables are connected to more children than high dimensional variables.

We implemented the CCCP algorithm in (AgenaRisk, 2020) without applying any particular optimization to the messages, and tested it using a κ_{12} BFG model, with the factors

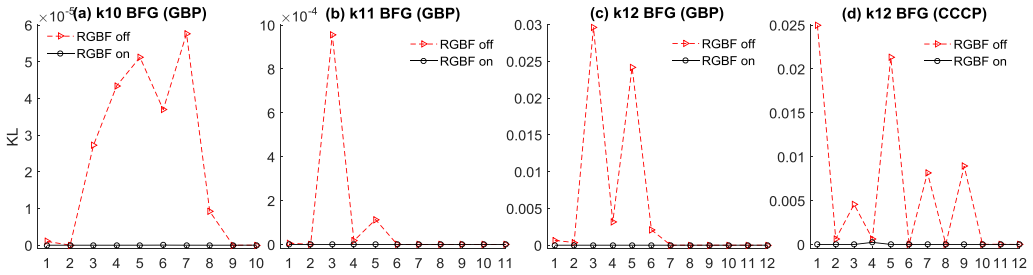

Figure 11: (a)-(c): RGBF test by using the KL of marginal in κ_{10}, κ_{11}, κ_{12} BFGs for GBP update, the Y-axis is the KL of marginals and X-axis is the variable ID; (d) RGBF test by using the KL of marginal in κ_{12} BFG for CCCP update.

listed in our code repository. Fig. 11 (d) shows the CCCP results when not using RGBF, with a convergence threshold of 3.0E-03 and the number of inner loops set at 3. When higher settings are used CCCP did not converge at all if RGBF was not used. When RGBF is used with CCCP, and also using an even more challenging convergence threshold of 1.0E-08 and with the number of inner loops was set at 4, the results show significant improvement.

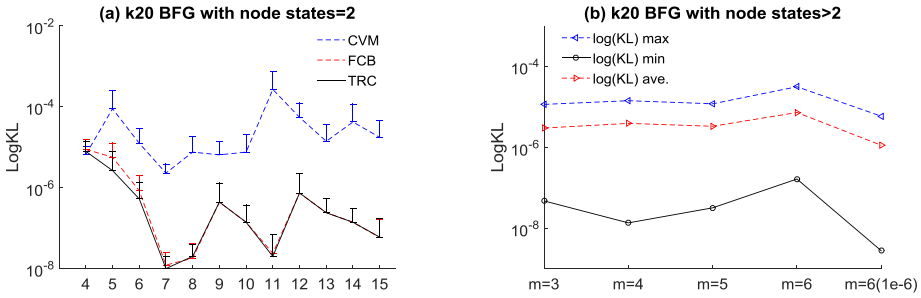

Figure 12: (a) TRC vs. CVM and FCB by comparing the $log(KL)$ of marginal in a κ_{20} BFG; (b) TRC performance on $log(KL)$ of marginal in a κ_{20} BFG with a different number of node states m.

In Fig. 12 (a), we compared ORI with other region graph-based algorithms using ten instances of random factors of κ_{20} BFG (*t.w.* 19) using singleton marginal for nodes X_4 to X_{15}, which lie on the leftmost path of the BFG. We used the RGBF algorithm in these tests.

CVM shows significant errors compared to TRC. The lower-dimensional variables accuracy in the κ_{20} BFG is determined by the competing interaction triplets so FCB is worse than the TRC for variables X_4 to X_7. Variables X_8 to X_{15} are less affected by the competing interaction triplets, and the FCB has chosen the same complete interaction triplets as TRC from the BFG, so the results for these variables are very close. Note that, for simplicity, we only show the accuracy of the variables lying on the leftmost path of the BFG. There are 17 κ_4 BFGs included in a κ_{20} BFG, and FCB will fail to test all the competing interaction redundancies. Hence, many intermediate nodes will not be approximated well by FCB.

IJGP (run by (Gelfand, 2011)) does not converge in this test when setting i-bound < 7 when running using GBP.

We also scaled the κ_{20} BFG to κ_{100} BFG (*t.w.* 99, variables 4853) with random factors and found that the accuracy of TRC is not degraded. The results of these high dimensional BFG tests are provided in Appendix D.5. These results show that the space complexity is reduced from exponential (with exact methods) to polynomial and TRC shows robust and accurate performance.

In Fig. 12 (b), we tested the TRC accuracy on a κ_{20} BFG by increasing the variables' number of states under the convergence threshold of 1e-5. The KL statistics degrade slightly with the increase in the number of states, but when the convergence threshold is set to 1e-6, the KL is reduced. So, as the number of discrete states increases, we can set a higher converge threshold to guarantee accuracy.

Next, we test TRC efficiency optimization using interaction triplet removal, and as a consequence the perfect correlation property will be relaxed. We use two kinds of models in the directed model category from the PASCAL challenge 2011 (Elidan, Globerson, & Heinemann, 2011), These are the Pedigree and Promedas models (these are the only directed models in the contest for computing marginals). These models contain extreme factors and the Promedas models also encode multiple competing interaction redundancies. TRC will convert these test models into BFGs and the BFGs for Pedigree and Promedas are equal to κ_{517} (for network ID 3000) and κ_{434} (for network ID 4021) dimensional BFGs.

Table 5: Efficiency comparison for TRC by relaxing/retaining the perfect correlation property

Perfect Correlation	Pedigree (ID 3000)			Promedas (ID 4021)		
	KL	time	iteration	KL	time	iteration
Relaxed	1.0E-14 (8.3E-14)	372s	796	1.5E-11 (2.2E-11)	78s	1289
Retained	5.0E-11 (1.2E-10)	300m	24700	3.2E-10 (4.8E-10)	48m	6783

Table 5 shows the KL for singletons of the Pedigree and Promedas models obtained by TRC. There are many more regions involved if the perfect correlation is retained, producing minor numerical differences compared to the results obtained otherwise.

For the Promedas model (*t.w.* 4), There are 1700 outer regions in the TRC region graph when the interaction triplet removal is not used, and that is around 3.6 times more regions than optimized after the interaction triplets are removed. For the Pedigree model (*t.w.* 19) there will be 13000 outer regions, without using interaction triplet removal, which is 8 times more than the number of outer regions optimized after the interaction triplets are removed.

In general, the higher the tree-width of the BN the more outer regions will be introduced. Our tests also verified that large BNs can be converted to BFGs and TRC can converge on high dimensional models with extreme factors, but at a computational time cost if the perfect correlation property is an absolute requirement. Note that the marginal results of the PASCAL test models are not sensitive to the perfect correlation property. In contrast, we found that by trying different outer regions using CVM, the results will be less accurate

if the perfect correlation property is not satisfied for the $n+2$ dimensional dense BN and the $m \times n$ DBNs.

5.4 Experiments on A Wide Range of BNs in Practice

We have tested a wide range of BNs (and synthetic data sets) including:

- Well known BNs Asia (Lauritzen & Spiegelhalter, 1988) (*t.w.* 2), Student (Murphy, 2012) (*t.w.* 3), BayesGrid (den Broeck et al., 2014) (*t.w.* 5), coupled HMM (Murphy, 2012) (*t.w.* 3) and a Hopfield network (Murphy, 2012) (*t.w.* 5).
- PASCAL challenge models (Elidan et al., 2011) including Promedas (*t.w.* 4 to 28, variables 400 to 900), Pedigree (*t.w.* 19, variables 385) and Protein [8] (*t.w.* 33, variables 14306).
- BNs introduced in this paper including a κ_{20} BFG (*t.w.* 19, variables 173), $n+2$ dimensional dense BN and $m \times n$ DBNs.
- BNs hosted in the Bayesian nets repository (Elidan, 1998) including Barley (Kristensen, 1998) (*t.w.* 5, variables 48), Pedigree Pigs (Jensen, 1998) (*t.w.* 11, variables 441), Diabetes (Andreassen et al., 1991) (*t.w.* 5, variables 413), the linkage analysis model (Jensen & Kong, 1996) (*t.w.* 13, variables 714), and the Munin model (Andreassen et al., 1989) (*t.w.* 8, variables 1041).

The number of variables ranges from 7 to 14306 and the tree-width ranges from 2 to 33 in these test models. We run TRC by optimizing the efficiency and relaxation of the perfect correlation property (if interaction triplet removal is possible). We contrast TRC with WMB (by merlin (Marinescu, 2019)), IJGP and FCB (by runGBP (Gelfand, 2011)) as they are representative of the top-down (IJGP, WMB) and bottom-up (FCB) classes of algorithm respectively. All experimental results comparing IJGP, FCB and TRC are summarized in Table 8.

We can divide all the test models into two categories by either encoding the competing interaction triplets or not. When models do not encode the competing interaction triplets we only showed the singleton marginal (other algorithms perform no better than TRC on the higher ordered marginal so we omit the results), shown in Table 8 for the 1^{st} to the 6^{th} models. Here TRC has outperformed IJGP in test cases run on four out of six test models and achieved almost the same accuracy for the remaining model. To obtain better accuracy when using IJGP the i-bound is set subject to the allocated memory, but when we use the bounded cluster size IJGP generated less sufficient interactions than TRC. The WMB is also worse than the TRC in these test cases so we omit the results.

TRC has also outperformed FCB on five out of six models and achieved exactly the same results for the other model (BayesGrid). If we increase the factor strength the result discrepancy between FCB and TRC will be increased, such as the 2^{nd} test model in Table 8. TRC has produced the same KL mean statistics with FCB for the Protein model but

8. The Protein model in (PASCAL 2011) is a large challenging MN model for MAP inference task. We convert it to a BN by removing the cyclic factors associated to child nodes, so the model structures, CPDs, and all the variables are retained, the test model can be downloaded in (Lin, 2020).

the *KL* (*s.d.*) and *max.* statistics are better than FCB. When given a bounded cluster size, FCB could miss necessary interactions because it is reliant on the cycle length found in the BN, which could be larger than the bounded cluster size. Also, compared to FCB which does not use a control parameter to improve the accuracy, TRC is more flexible as it can use different ordered cluster sizes to improve accuracy.

When models encode the competing interaction triplets TRC shows clear advantage over the competing algorithms, especially in higher ordered marginal results. TRC outperforms competing algorithms in all the 15 test models (from the 7^{th} to the 21^{st}), as shown in Table 8. Moreover, some tests result in Table 8 are obtained using normal factors only, the discrepancy of the results between TRC and the competing algorithms will increase along with the increasing factor strength.

IJGP and FCB obtained accurate approximations for the singletons marginal in the Promedas test models, but exhibit significant inaccuracy when we examine the pair-wise marginal involved in the competing interaction triplets, such as the node pair $\{X_4, X_5\}$ in Fig. 8 (b). This is because these node pairs do not have a shared child node in the Promedas model so they are not connected as a moral edge, which affect none of the singleton marginal results. However, once they have a shared child node the singleton marginals will be inaccurate.

Table 6: TRC vs. other algorithms using *KL mean* (*s.d.*) for the pair-wise marginal for Promedas test models

ID	IJGP	WMB	FCB	TRC
4000	9.0E-02	5.5E-02	6.0E-03	**1.7E-12**
	(9.0E-02)	(2.4E-02)	(1.3E-02)	**(2.8E-12)**
4021	0.14	0.5	3.0E-02	**1.8E-07**
	(0.15)	(0.1)	(7.0E-02)	**(2.7E-07)**
4034	6.6E-02	0.39	5.6E-02	**1.6E-02**
	(0.13)	(0.35)	(0.12)	**(2.1E-02)**
4069	5.9E-02	2.7	1.8E-02	**2.7E-12**
	(0.1)	(1.1)	(3.5E-02)	**(3.0E-12)**
4083	9.7E-03	1.9	1.0E-02	**7.0E-04**
	(1.4E-02)	(2.0)	(1.4E-02)	**(1.6E-03)**

In Table 6, we show the Promedas results for the pair-wise marginals of the parent nodes which are included in the competing interaction triplets, and demonstrate that these pair-wise marginals cannot be ignored as they are poorly approximated by competing algorithms. The original factors in these models are all extreme factors, so we simulated 10 random instances for each model using extreme factors. Clearly, TRC outperforms other algorithms significantly on these pair-wise marginals. We omit the results for pair-wise marginal for the moral edges because TRC performed equally well with the competing algorithms.

In Table 7 we have listed the maximum *KL* results for 10 models encoding competing interaction triplets in our random factor tests. TRC has achieved better accuracy compared to the other algorithms. In many cases, especially where *KL* was greater than 0.1, the

Table 7: *KL max.* of singleton, pair-wise and triplet marginals for models with competing interaction triplets

models	IJGP	FCB	TRC
(Singleton)			
1. * 6 dimensional dense BN	0.41	0.38	**2.0E-02**
2. * 3x3 DBN	0.19	0.24	**8.6E-02**
3. Munin	0.15	0.15	**3.8E-03**
(Pair-wise)			
4. κ_{20} BFG	0.18	0.2	**0.07**
5. Pigs	0.05	0.03	**1.0E-03**
6. coupled HMM	0.1	0.08	**0.02**
7. * Promedas	0.4	0.4	**3.8E-02**
(Triplets)			
8. Diabetes	0.08	0.05	**8.0E-03**
9. Barley	0.15	0.11	**0.04**
10. * Hopfield	0.35	0.21	**0.09**

competing algorithms "flipped" the pair-wise and triplet marginal probabilities compared to the true values, leading to misleading inaccurate results. Even for singleton marginals, the max variability problem is serious and evident for the competing algorithms. For example, in the Munin test model, there is a node pair affected by competing interaction triplets sharing no children. So the node pair will not be connected as a moral edge in the moral graph. This means the node pair will not be approximated well because competing algorithms will not generate interactions for it given the cluster size is limited. The error will be propagated to many other variables, especially if the node pair is included in a cycle path. In contrast, TRC runs on a BFG and the results will be accurate using only triplet outer regions which is smaller than required by other algorithms for this test case. FCB will often experience multiple region choices for the model encoding competing interaction triplets. For example, there are six choices to select as root node for the set of fundamental cycles in the substructure $\{X_1, X_2, X_3, X_4, X_5, X_6\}$ using FCB for the 6-dimensional dense BN and the number of choices increases with dimensionality. Also, the results can vary significantly by different choices.

Based on the analysis here TRC demonstrates clear advantage over other algorithms on both singleton and higher ordered marginals, especially when the models encode competing interaction triplets. TRC also effectively reduced the max variability problem compared to others. We provide efficiency comparisons against competing algorithm for Table 8 test models in Appendix D.4. By using efficiency optimizations TRC can achieve comparable efficiency to that achieved by competing algorithms.

Table 8: Summary statistics of *KL mean* (*s.d./max.*) among IJGP, FCB and TRC. The empty entries mean there are no pairwise or triplets for parent nodes (connecting a moral edge) that are involved in the competing interaction triplets in the model.

	Singleton			**Pair-wise**		
Models	IJGP	FCB	TRC	IJGP	FCB	TRC
1. Asia	**1.6E-12 (2.0E-12/5.0E-12)**	3.2E-06 (7.9E-06/2.3E-05)	2.0E-12 (2.1E-12/5.0E-12)	-	-	-
2. * Asia	**5.8E-08 (3.52E-08/1.3E-07)**	2.4E-04 (5.8E-04/1.7E-03)	3.2E-07 (3.2E-07/8.4E-07)	-	-	-
3. Student	5.0E-06 (1.3E-05)	5.3E-06 (9.6E-06)	**3.4E-06 (8.9E-06)**	-	-	-
4. BayesGrid	3.4E-04 (1.1E-03/5.8E-03)	**2.8E-08 (6.5E-08/2.9E-07)**	2.8E-08 (6.5E-08/2.9E-07)	-	-	-
5. Linkage	2.6E-14 (1.3E-13)	3.7E-14 (1.7E-13)	**1.1E-14 (3.9E-14)**	-	-	-
6. Protein	2.8E-05 (1.2E-04/3.2E-03)	2.0E-05 (7.6E-05/2.4E-03)	**2.0E-05 (6.4E-05/1.3E-03)**	-	-	-
7. * 4 dimensional dense BN	7.5E-02 (1.6E-01/5.8E-01)	4.7E-02 (1.4E-01/5.8E-01)	**1.1E-03 (4.0E-03/1.6E-02)**	3.2E-01 (8.5E-01/3.4E+00)	3.0E-01 (8.6E-01/3.4E+00)	**1.1E-03 (3.7E-03/1.5E-02)**
8. * 5 dimensional dense BN	1.9E-02 (3.9E-02/1.5E-01)	7.9E-03 (2.4E-02/1.5E-01)	**4.3E-03 (1.2E-02/8.6E-02)**	1.5E-01 (3.0E-01/9.7E-01)	1.0E-01 (2.5E-01/9.9E-01)	**2.0E-02 (6.5E-02/2.6E-01)**
9. * 6 dimensional dense BN	3.4E-02 (9.6E-02/4.1E-01)	1.6E-02 (5.0E-02/3.8E-01)	**1.0E-03 (2.7E-03/2.0E-02)**	5.1E-02 (1.2E-01/4.1E-01)	4.6E-02 (1.1E-01/4.2E-01)	**4.7E-03 (7.7E-03/2.9E-02)**
10. * 2 × 3 DBN	2.2E-03 (5.4E-03/2.1E-02)	1.1E-03 (4.7E-03/2.5E-02)	**3.1E-07 (2.0E-06/1.4E-05)**	1.0E-02 (1.6E-02/6.2E-02)	5.5E-03 (2.0E-02/1.1E-01)	**3.1E-04 (1.6E-03/9.1E-03)**
11. * 3 × 3 DBN	1.2E-02 (3.1E-02/1.9E-01)	2.1E-02 (4.6E-02/2.4E-01)	**6.3E-03 (1.7E-02/8.6E-02)**	-	-	-
12. coupled HMM	2.1E-03 (2.5E-03)	1.2E-03 (3.9E-03)	**3.9E-04 (1.0E-03)**	1.0E-02 (7.8E-03/0.1)	6.0E-03 (1.2E-02/0.08)	**2.5E-03 (4.7E-03/0.02)**
13. κ_{20} BFG	1.4E-04 (2.7E-04)	3.6E-06 (3.5E-06)	**2.7E-06 (3.1E-06)**	5.0E-03 (1.6E-02/0.18)	6.0E-03 (2.1E-02/0.2)	**1.8E-03 (7.7E-03/0.07)**
14. Diabetes	2.1E-04 (2.8E-03/5.7E-02)	1.3E-05 (1.4E-04/2.1E-03)	**4.5E-06 (4.3E-05/8.3E-04)**	5.8E-03 (5.1E-03/0.01)	3.4E-03 (3.1E-03/8.0E-3)	**5.5E-04 (4.9E-04/1.0E-3)**
15. Hopfield	8.6E-04 (1.7E-03)	3.6E-04 (2.9E-04)	**2.9E-04 (3.1E-04)**	9.7E-03 (1.4E-02/0.04)	6.0E-03 (5.4E-03/0.02)	**3.0E-03 (2.0E-03/6.0E-3)**
16. *Hopfield	2.2E-03 (2.1E-03)	5.5E-03 (1.0E-02)	**7.9E-04 (9.6E-04)**	3.8E-02 (1.7E-02/0.18)	1.6E-02 (1.7E-02/0.13)	**1.3E-02 (1.4E-02/0.08)**
17. Barley	5.8E-04 (3.6E-03/2.5E-02)	1.0E-03 (6.3E-03/4.4E-02)	**6.0E-05 (3.4E-04/2.6E-03)**	1.6E-02 (1.6E-02/0.04)	2.2E-02 (1.6E-02/0.04)	**2.6E-03 (2.6E-03/6E-3)**
18. Pedigree PASCAL	1.2E-13 (5.4E-13)	1.2E-13 (5.5E-13)	**1.0E-14 (8.3E-14)**	-	-	-
19. Pigs	3.8E-07 (6.4E-06/1.3E-04)	3.8E-07 (6.4E-06/1.3E-04)	**1.6E-07 (2.2E-06/3.6E-05)**	1.3E-02 (1.6E-02/4.5E-02)	1.2E-02 (1.3E-02/3.3E-2)	**3.7E-04 (4.1E-04/1.0E-3)**
20. Promedas PASCAL	**2.8E-13 (4.1E-13)**	2.1E-11 (3.1E-11)	1.5E-11 (2.2E-11)	4.6E-02 (7.2E-02/0.4)	3.4E-02 (6.8E-02/0.4)	**1.8E-03 (3.0E-03/3.8E-2)**
21. Munin	3.3E-04 (6.8E-03/1.5E-01)	3.4E-04 (6.9E-03/1.5E-01)	**1.5E-05 (2.2E-04/3.8E-03)**	5.5E-04 (9.0E-03/0.15))	7.3E-04 (1.0E-2/0.15)	**2.0E-05 (2.4E-04/3.9E-03)**
	Singleton			**Triplet**		
Models	IJGP	FCB	TRC	IJGP	FCB	TRC
11. * 3 × 3 DBN	1.2E-02 (3.1E-02/1.9E-01)	2.1E-02 (4.6E-02/2.4E-01)	**6.3E-03 (1.7E-02/8.6E-02)**	1.1E-01 (0.18/0.77)	1.9E-01 (0.3/1.2E+00)	**4.7E-02 (6.6E-02/0.22)**
12. coupled HMM	2.1E-03 (2.5E-03)	1.2E-03 (3.9E-03)	**3.9E-04 (1.0E-03)**	2.3E-02 (1.4E-02/0.12)	1.7E-02 (1.3E-02/0.09)	**1.0E-02 (6.6E-03/0.02)**
14. Diabetes	2.1E-04 (2.8E-03/5.7E-02)	1.3E-05 (1.4E-04/2.1E-03)	**4.5E-06 (4.3E-05/8.3E-04)**	3.5E-02 (3.0E-02/0.08)	2.1E-02 (1.8E-02/0.05)	**3.2E-03 (3.2E-03/8.0E-3)**
15. Hopfield	8.6E-04 (1.7E-03)	3.6E-04 (2.9E-04)	**2.9E-04 (3.1E-04)**	2.8E-02 (2.0E-02/0.07)	2.6E-02 (1.8E-02/0.07)	**1.6E-02 (8.0E-03/0.02)**
16. *Hopfield	2.2E-03 (2.1E-03)	5.5E-03 (1.0E-02)	**7.9E-04 (9.6E-04)**	1.1E-01 (2.9E-02/0.35)	2.3E-01 (4.5E-01/0.21)	**3.8E-02 (2.5E-02/0.09)**
17. Barley	5.8E-04 (3.6E-03/2.5E-02)	1.0E-03 (6.3E-03/4.4E-02)	**6.0E-05 (3.4E-04/2.6E-03)**	5.6E-02 (5.6E-02/0.15)	5.5E-02 (3.5E-02/0.11)	**1.9E-02 (1.5E-02/0.04)**

6. Conclusion and Future Work

We have presented a general purpose approximate Bayesian Network inference algorithm – Triplet Region Construction (TRC) – that overcomes the computational complexity barrier presented by exact algorithms. Specifically, whereas exact algorithms are exponential for BNs with increasing tree-width, the TRC algorithm reduces the space complexity from exponential to polynomial for factorized models. The TRC algorithm provides systematic improvements over previous approximate methods for region-based approximate belief propagation (namely relating to region choice, convergence, and accuracy). It guarantees convergence and the maxent-Normal and perfect-correlation properties are preserved.

The binary factorization (BF) process is a necessary first step for our algorithm as it reduces the node indegree required when building a region graph involving only triplet outer regions and reduces the number of interaction regions to consider. Also, the ORI algorithm solves the problem of identifying effective triplet interaction regions by using both structural and factor information in an entirely localized way. The RGBF algorithm is then applied to improve the stability when using GBP/CCCP. All these sub-algorithms can be used or extended separately with other algorithms. We also demonstrate how further optimizations, specifically node reuse and interaction triplet removal, can produce results that achieve similar efficiency as competing algorithms.

The various and extensive experiments show that, given a bounded cluster size, TRC is more accurate than competing algorithms, in both the high (also high dimensional) and low tree-width models presented in the paper. TRC is an automated algorithm and is relatively easy to extend to use different maximum cluster size to improve accuracy. TRC also effectively addressed the max variability problem when other competing algorithms cannot.

Future extensions of this work will focus on using TRC for high tree-width model parameter learning and improved computation efficiency. When using region-based approximation for parameter learning the competing interaction redundancy problem is more critical, i.e., in each EM iteration the factors will change and may result in different competing interaction redundancies between each EM iteration. We will also combine TRC with discretization or sampling methods to approximate continuous variables. Lastly, we will look at paralellisation of the algorithms.

Acknowledgments

M. Neil is the corresponding author of the paper. This work was supported in part by: the European Research Council under project, ERC-2013-AdG339182- BAYES-KNOWLEDGE; the Leverhulme Trust under Grant RPG-2016-118 CAUSAL-DYNAMICS; the EPSRC under project EP/P009964/1: PAMBAYESIAN: Patient Managed decision-support using Bayes Networks; The Alan Turing Institute under the EPSRC grant EP/N510129/1. We also thank our anonymous reviewers for their useful comments and suggestions.

Appendix A. Proofs

This section contains all the proofs associated with the paper.

A.1 Proof of Proposition 1

Given our inference task is to compute marginals we ensure any ordered joint distributions for the original nodes in a BN G are identical to the corresponding nodes in its BFG G'. This can be achieved by rebuilding the original CPD for each node X_i in G using the new CPD for X_i and its associated intermediate nodes in G'. Hence, we need to define the new CPD for X_i and its associated intermediate nodes in G' such that if the new CPDs in G' reproduce the original CPDs in G then Proposition 1 is proven.

In what follows we assume the unique ordering of the complete graph G from Theorem 1, and apply the structural factorization of G by introducing a set of intermediate variables E_t that are not in the original BN ($E_t \in G', E_t \notin G$). For example, in the case of a 5-dimensional complete DAG G, the structure of the binary factorized version is as shown in Fig. 13 G'. While the BF algorithm is guaranteed to produce a uniquely structured BFG G' for each complete DAG G, we show that the CPDs in G' for each node X_i in G, the CPD of X_i in G' is equivalent, after factorization, to the CPD of X_i in G.

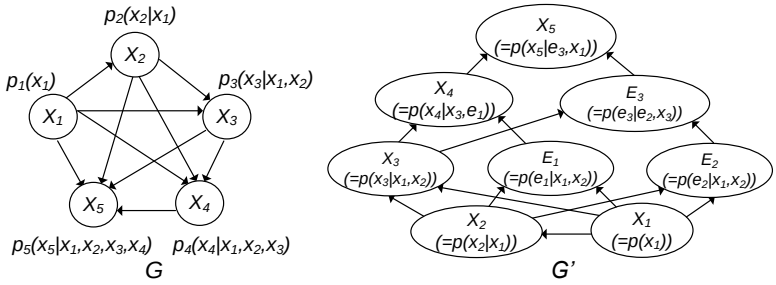

Figure 13: BF process of a five-dimensional dense graph G to its binary factorized model G'.

In general, a discrete node D with three discrete parents A, B and C can be transformed into an equivalent binary factorized form by introducing an intermediate node E (with parents A and B) that has $n \times m$ states e_{ij} ($i = 1, \ldots, n$ and $j = 1, \ldots, m$) where A has n states $a_1, \ldots, a_n$ and B has m states $b_1, \ldots, b_m$. The CPD for E is defined as:

$$p(E = e_{ij}|a_k, b_l) = \begin{cases} 1 & if\ k = i\ and\ l = j \\ 0 & otherwise \end{cases}$$

The CPD for node D in G' (with parents C and E) is defined as: $p_{G'}(D|e_{ij}, c_k) = p_G(D|a_i, b_j, c_k)$.

A.2 Proof of Proposition 2

All interaction triplets are generated by using the coupled Markov blanket. We can verify that there is at least one node pair in the interaction triplet $\{i, j, k\}$ that corresponds

to a moral edge, or which is not directly connected as an edge in $M[G']$. This means all interaction triplets being evaluated are composed from incomplete and complete interaction triplets. So, to prove Proposition 2 we need to prove all incomplete interaction triplets are incomplete interaction redundant in the BFG, and hence the remaining (the complete interaction triplets) are retained.

We first use Fig. 14 as an example to illustrate the rationale that the incomplete interaction triplets can be removed and the complete interaction triplets need to be retained.

We introduce an incomplete interaction triplet $\{i, j, k\}$ from Fig. 14 (a) and construct the region graph in (b) by including all the primary triplets and the interaction triplet $\{i, j, k\}$ as outer regions. By introducing the incomplete interaction triplet we create the pair-wise interactions $\{i, j\}$ and $\{i, k\}$ in L_2 of (b), and all regions will be consistent for the pair-wise regions $\{i, j\}$ and $\{i, k\}$. However, these two pair-wise regions are exclusively contained in the primary triplets $\{i, j, p\}$ and $\{i, k, q\}$ respectively, which makes these pair-wise interactions redundant.

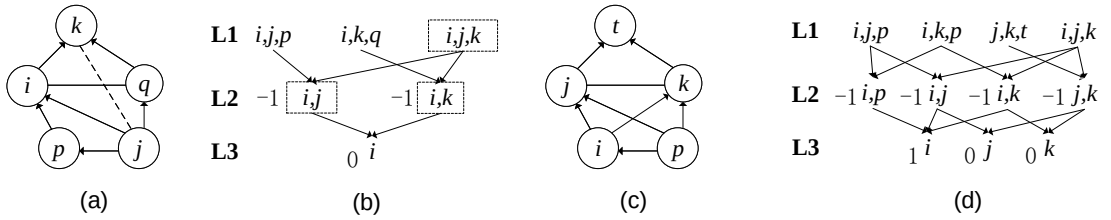

Figure 14: (a) $\{i, j, k\}$ is an incomplete interaction triplet in a portion structure of the κ_4 BFG, edges without arrow are moral edges; (b) region graph of (a) where the regions with the dashed box can be removed; (c) $\{i, j, k\}$ is a complete interaction triplet in a κ_4 BFG; (d) region graph of (c).

Therefore, introducing $\{i, j, k\}$ will not change the pair-wise belief of $b_{i,j}$ and $b_{i,k}$, but it will change the belief of $b_{j,k}$ because $\{i, j, k\}$ forces an update of the pair-wise information $\{j, k\}$ and changes the region belief $b_{j,k}$ from $b_{j,k} = b_j b_k$ to $b_{j,k} = \sum_i b_{i,j} b_{i,k} / b_i$. The change of the pair-wise belief $b_{j,k}$, however, will not change any of the singleton belief in the model, because the nodes j and k are not connected as an edge in the BN in (a), indicating there is no child node depending on the node pair $\{j, k\}$. Given no singleton entropy change in the model the introduction of the incomplete interaction triplet is redundant and so it can be removed.

Fig. 14 (c) shows a complete interaction triplet $\{i, j, k\}$, and we construct a region graph in (d) by using the complete interaction triplet with all the primary triplets as outer regions. Introducing the complete interaction triplet creates three pair-wise interactions $\{i, j\}$, $\{i, k\}$ and $\{j, k\}$ in L_2 of (d). Again, the pair-wise interactions $\{i, j\}$ and $\{i, k\}$ are exclusively included in the corresponding primary triplets, and so they are fixed. The pair-wise belief of $b_{j,k}$ will change by introducing the $\{i, j, k\}$. Given the node pair $\{j, k\}$ is connected as a moral edge, there must be a child node $t$dependant on it and which causes a singleton entropy change. As a result, the complete interaction triplet will need to be retained.

Proof. For convenience, we denote region entropy as $H(b_r) \equiv -\sum_i b_r(x_i) \ln b_r(x_i)$, and mutual information for two region beliefs are $I(b_r; b_s) = H(b_s) - H(b_s|b_r)$. The redundancy of an interaction triplet can be determined by determining if the singleton entropy of any variable t in $M[G']$ changes when the interaction triplet is introduced. The variable t can be a member of $\{i, j, k\}$ or not.

1. Supposing variable t has entropy change and $t \in \{i, j, k\}$, so we need to determine the entropies of i, j, and k as at least one of them changes:

   The interaction triplet contains three node pairs, $\{i, j\}$, $\{i, k\}$ and $\{j, k\}$, in which a node pair is either connected as a moral edge or not connected as an edge in the moral graph. Assuming this node pair is $\{j, k\}$ then the other two node pairs must be connected as edges in the moral graph, which means they must belong to two primary triplets and are regularized. Here we can assume the two primary triplets are $\{i, j, p\}$ and $\{i, k, q\}$ and they must share the node i.

   The node pair $\{j, k\}$ is not regularised and can be associated with a pair-wise factor $\phi_{j,k} = 1$ implying $j \perp k|pa_{(j,k)}$ (we do not need to consider the evidence case as we are evaluating the interaction triplet without evidence) and hence the entropies of j and k are separately determined by other node pairings: $\{i, j\}$ and $\{i, k\}$. Next, we can compute the mutual information of these two node pairs, given they contain non-uniform factors. So, given $j \perp k|pa_{(j,k)}$, we have:

   $$I(b_{i,j}; b_{i,k}) = H(b_{i,k}) - H(b_{i,k}|b_{i,j}) = H(b_{i,k}) - H(b_k|b_i) = H(\tilde{b}_i), \tag{10}$$

   where $b_{i,j}$ and $b_{i,k}$ are region beliefs associated with the region $\{i, j\}$ and $\{i, k\}$, $H(\tilde{b}_i)$ is the entropy of the marginal belief over variable i, which means the entropy change of $i$will change the entropy of both j and k.

   We can compute the $H(\tilde{b}_i)$ by the GBP equation:

   $$\begin{aligned} H(\tilde{b}_i) \propto \tilde{b}_i &= \sum_{j,p} \tilde{f}_{i,j,p} \prod m_{i,j} \prod m_{i,p}, \\ &= \sum_{k,q} \tilde{f}_{i,k,q} \prod m_{i,k} \prod m_{i,q}, \end{aligned} \tag{11}$$

   where $\tilde{f}_{i,j,p}, \tilde{f}_{i,k,q}$ are triplet factors associated with the two primary triplets that contain variable i. In (11) all pair-wise messages are incoming messages to the two primary triplet regions. Messages $m_{i,p}$ and $m_{i,q}$ do not result from the introduction of $\{i, j, k\}$ so they are fixed, but the $m_{i,j}$ and $m_{i,k}$ messages vary. Based on GBP message updating (Yedidia et al., 2005), messages $m_{i,j}$ and $m_{i,k}$ are determined by all messages sent from factor regions to regions for node pairs $\{i, j\}$ and $\{i, k\}$. In summary, the entropies of all the three variables in $\{i, j, k\}$ are determined by the regions of node pairs $\{i, j\}$ and $\{i, k\}$.

2. Supposing variable t has entropy change and $t \notin \{i, j, k\}$.

Since t is not a member of $\{i,j,k\}$, there is no singleton entropy change in $\{i,j,k\}$, otherwise we can refer to case 1. The entropy of t will possibly change only if t depends on the pair-wise information of two variables in $\{i,j,k\}$, otherwise the entropy of variable t will not change given that there is no singleton entropy change in $\{i,j,k\}$. Therefore, $pa_{(t)} \in \{i,j,k\}$, and $\{i,j,k\}$ must contain a node pair that corresponds to a moral edge. Suppose the moral edge is $\{j,k\}$, so the entropy of $t$depends on the region belief $b_{j,k}$, and $b_{j,k}$ changes from $b_{j,k} = b_j b_k$ to $b_{j,k} = \sum_i b_{i,j} b_{i,k}/b_i$ (because of the interaction triplet $\{i,j,k\}$). In summary, the entropy change from t is also determined by the regions of node pairs $\{i,j\}$ and $\{i,k\}$.

As a result, whenever t is a member of $\{i,j,k\}$ or not, the entropy change of t depends on all the factors containing node i and then sending messages to regions for node pairs $\{i,j\}$ and $\{i,k\}$. We denote these factors as a set $\Psi_{i,j,k}$. We also denote a set of factors $\Phi_{i,j,k}$ that is composed of all primary triplets connected by the interaction triplet $\{i,j,k\}$. No matter whether $\{i,j,k\}$ is an incomplete or complete interaction triplet the belief $b_{j,k}$ will be changed from $b_{j,k} = b_j b_k$ to $b_{j,k} = \sum_i b_{i,j} b_{i,k}/b_i$. So we have the following conditions:

If $\Psi_{i,j,k} \subseteq \Phi_{i,j,k}$, and if $\{j,k\}$ does not correspond to an edge in the moral graph, there will be no variable dependent on the pair-wise belief of $b_{j,k}$, and the interaction triplet is therefore redundant. We can verify that all incomplete interaction triplets satisfy this condition, such as in Fig. 14 (a). Otherwise if $\{j,k\}$ corresponds to a moral edge then there must exist a child variable dependent on the pair-wise belief of $b_{j,k}$, and so $\{i,j,k\}$ must be a complete interaction triplet that needs to be retained, such as Fig. 14 (c).

If $\Psi_{i,j,k} \not\subset \Phi_{i,j,k}$, then $\{i,j,k\}$ can never be an incomplete interaction triplet in a BFG. It must contain a node pair $\{j,k\}$ corresponding to a moral edge and there must also exist a child variable depending on the pair-wise of $b_{j,k}$, such as $\{X_3, X_4, E_3\}$ in Fig. 13 G'. Thus $\{i,j,k\}$ is a complete interaction triplet that needs to be retained.

A.3 Proof of Proposition 3

Note that competing interaction redundancy occurs in a κ_4 BFG, as shown in Fig. 14 (c) and any higher-ordered BFG must contain multiple κ_4 BFGs. The competing interaction triplets $\{i,j,k\}$ and $\{p,j,k\}$ in Fig. 14 (c) that both contain the same node pair $\{j,k\}$ which corresponds to a moral edge in $M[G']$. Given that the entropy of an individual node, except node $t$in Fig. 14 (c), can be calculated exactly by using only two primary triplets, then the only difference in contribution between the two competing interaction triplets is the differing approximation quality of the pair-wise belief $b_{j,k}$. Also, the accuracy of the belief at node t depends on the accuracy of the pair-wise joint belief $b_{j,k}$. The exact marginal p_t equals $p(t) = \sum_{j,k} f_{j,k,t} p_{j,k}$, so we need to minimize the KL $D(b_{j,k}||p_{j,k})$. Hence, we need at least one interaction triplet to exchange messages for the node pair $\{j,k\}$ associated with primary triplet $\{j,k,t\}$. Given node j, k share a pair of parent nodes and the parent nodes are also dependent, to calculate $p_{j,k}$ exactly we need to involve four nodes. But we only have cluster size of three, and under this condition, we only have two competing interaction triplets to consider: $\{i,j,k\}$ and/or $\{p,j,k\}$.

We can choose to either add both of them or one of them, but adding both of them will not help because of the following counter-examples:

1. Suppose the node pair $\{j, k\}$ connecting a moral edge shares a pair of parent nodes $\{p, i\}$. To compute the pair-wise joint of $p_{j,k}$ exactly we need the pair-wise joint of $p_{p,i}$, but both competing interaction triplets contain only the singleton information of p or i. Adding both does not incorporate the exact joint of $p_{p,i}$. Likewise, adding both of them forces the message exchange of $\{j, k\}$ over the two competing interaction triplets, but given none of them is computed exactly it risks distorting the approximation.

2. Adding both of them breaks the perfect correlation property.

3. If we choose one of them, we cannot use the structural information alone as well, given the competing interaction triplets are symmetric.

So under the cluster size three constraint our best option is to choose only one of them. Clearly we cannot determine this choice using structural information alone.

A.4 Proof of Proposition 4

Given p_r the number of parents, Algorithm 2 RGBF will produce $p_r - 1$ copies of the region r in $\mathcal{G}'$ when c_r is not equal to 1, -1 or 0 in $\mathcal{G}$. Each region r in $\mathcal{G}'$ will share one parent with its neighboring copy of r. Equivalence between $\mathcal{G}$ and $\mathcal{G}'$ can be proven by using the consistency and unity conditions [9] for a region graph.

1. Consistency: Since the first level has not changed, the consistency of all r ($r \notin R_{1st\ level}$) and its copies with their parents in $\mathcal{G}'$ must be maintained. This is satisfied given each region r is connected to its neighboring copy by sharing one parent, such that parents and all regions r are connected and hence consistent.

2. Unity: Unity for each variable must be the same in $\mathcal{G}$ and $\mathcal{G}'$. As $\mathcal{G}'$ does not contain any new regions compared to $\mathcal{G}$ but only copies of regions, r, from $\mathcal{G}$, the counting number for each variable will only be influenced by the region r and its copies. So the unity condition can be satisfied by integer accumulation of r and its copies' counting numbers in $\mathcal{G}'$ to c_r in $\mathcal{G}$, $\sum_{i=1}^{p_r-1} c_{r_i} = c_r$ ($r_i \in \mathcal{G}'$, $r \in \mathcal{G}$), which will not change the unity condition for each variable. Note that, in this way, the cumulative counting number is not unique but can be specified by using 1, -1, and 0 as these work for any integer.

A.5 Proof that the TRC region graph satisfies the perfect correlation property

The counting number properties for BFG model is already summarized in Table 2. We prove the results in Table 2 below:

1. Let n be the number of original nodes in a BFG, G' so the number of intermediate nodes in G' is $1 + 2 + ... + n - 3 = (n-2)(n-3)/2,\ n > 3$.

9. A variable has unity when the sum of all regions counting numbers associated with that variable is one.

2. From the parent to child relationships in G' the number of primary triplets is determined by the sum of the number of original variables and intermediate variables minus 2, as there are two factors absorbed into triplets. So we have $n-2+(n-2)(n-3)/2$ primary triplets.

3. The number of interaction triplets is the number of moral edges and it is also the number of intermediate nodes, so we have $(n-2)(n-3)/2$ interaction triplets.

4. The number of first level triplets is then: $L_1 = n-2+(n-2)(n-3)/2+(n-2)(n-3)/2 = (n-2)^2$.

5. The number of second-level intersections is determined by the number of first level triplets and is $(n-2)^2$.

6. There are $n-3$ intersections with the form $\{X_i, X_j\}$ which has counting number -1 to $3-n$, so $\min(c_r) = 3-n$ at the second level. All other intersections with the form $\{X_i, E_t\}$ have counting number -1.

7. The third level regions are all single-variable regions and are original variables X_i, with the counting number 1 to $n-3$ sequentially, so $\max(c_r) = n-3$.

Now we can prove the *perfect correlation property*:

The sum of all first level region counting numbers is $(n-2)^2 \times 1$. The second and third level regions' counting numbers are cancelled by each other, which will leave one region with counting number $3-n$ (there are two regions at the second level with counting $3-n$ and one is cancelled) and $(n-2)^2-(n-3)-1$ regions with counting -1. So, we sum them all to obtain $(n-2)^2+3-n+((n-2)^2-(n-3)-1)\times -1 = 1$.

An example of a TRC region graph satisfying the perfect correlation property when different root nodes are selected after the competing interaction redundancy test is shown in Fig. 15.

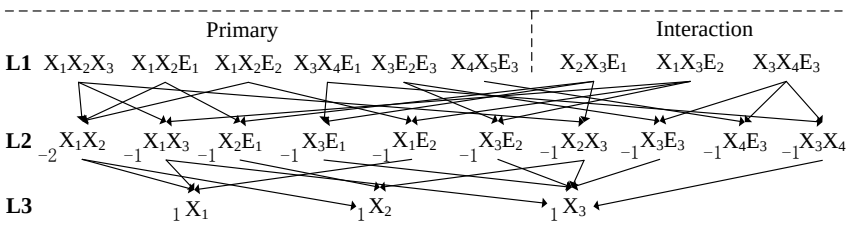

Figure 15: TRC region graph for a κ_5 BFG (in Fig. 13) when X_1 and X_2 are selected (after competing interaction redundancy test) for each κ_4 BFG it contains

We can verify in Fig. 15 that the sum of all regions' counting number remains 1 $(9-11+3=1)$.

A.6 Proof that a TRC region graph satisfies the maxent-entropy normal property

The Bethe approximation is maxent-normal (Yedidia et al., 2005), and so the entropy of the region graph, $H_{\mathcal{G}}$, can be written as $H_{\mathcal{G}} = \sum_{i=1}^{N} H(b_i) - \sum_{a=1}^{M} I(b_a)$ where N is the number of variables in the region graph, X_i, and M is the number of factors, a, ($\mathbf{X}_a$ are the variables defined by the factor a). $H(b_i) \equiv -\sum_{X_i} b_i(X_i) \ln b_i(X_i)$ is the sum of entropies from all variables X_i in the region graph, and $I(b_a) \equiv \sum_{\mathbf{X}_a} b_a(\mathbf{X}_a) \ln b_a(\mathbf{X}_a) - \sum_{i \in N(a)} H(b_i)$ is the mutual information, which is the entropy for a region containing factor a, minus the entropies of all variables contained in factor a. $H_{\mathcal{G}}$ is maximal, equalling $\sum_{i=1}^{N} H(b_i)$, when all beliefs, $b_i(X_i)$ and $b_a(\mathbf{X}_a)$ are uniform, and under these circumstances, the mutual information, $I(b_a)$ equals zero. In our region graph we can always construct $H_{\mathcal{G}}$ in the form of $H_{\mathcal{G}} = \sum_{i=1}^{N} H(b_i) - \sum_{a=1}^{M} I(b_a)$ because the mutual information for each triplet can be constructed by its connected second-level regions and the single variables the triplet contains, resulting in minimal I terms and maximal entropy $H_{\mathcal{G}}$ when all beliefs are uniform. The rest of the proof is omitted for brevity because the verification can be done directly on the TRC region graph.

A.7 Proof that a size four TRC region graph satisfies the perfect correlation property

We use induction to prove this property. Consider the size four TRC region graph for a κ_6 BFG. For convenience, we merged all triplets into size four outer regions, so the first level will not contain any triplet outer regions. We merge all primary triplets into the interaction triplets and merge all interaction triplets. The first level regions are all size four regions with counting number 1. We have the following counting numbers for each level in the region graph:

Table 9: size 4 TRC region graph counting numbers for a κ_6 BFG

1^{st} level	22×1		
2^{nd} level	5×-3	6×-2	7×-1
3^{rd} level	3×2	2×3	1×1
4^{th} level	-3	2	1

There are 22 1^{st} level regions with counting number 1 so the overall count for the 1^{st} level is 22×1. There are 5 regions with counting number -3, 6 regions with counting number -2 and 7 regions with counting number -1 at the 2^{nd} level. The counts for the remaining levels are also shown in Table 9. The sum of all the fourth level regions is zero. And we will have the same counting numbers pattern for the κ_n BFG's region graph in Table 10.

In Table 10, $t = n-3$, C_{level1} = sum of the 1^{st} level region counts, and C_{level4} = sum of the 4^{th} level region counts. Let C_{level2} and C_{level3} are the sum of the 2^{nd} and the 3^{rd} level region counts respectively we have $C'_{level2} = C_{level2} - [(2t-1) \times -(n-3)] - [(2t) \times -(n-4)]$,

Table 10: size 4 TRC region graph counting numbers for a κ_n BFG, $n \geq 6$

1^{st} level	C_{level1}		
2^{nd} level	$(2t-1) \times -(n-3)$	$2t \times -(n-4)$	C'_{level2}
3^{rd} level	C'_{level3}		1×1
4^{th} level	$C_{level4} = 0$		

and $C'_{level3} = C_{level3} - 1$. The 2^{nd} and the 3^{rd} level regions have different signs and we have the following pattern, $C'_{level2} + C'_{level3} = 5 \times (n-5)$. So overall, we have the following equations:

$$C_{level4} = 0, \tag{12}$$

$$C_{level2} + C_{level3} = [(2t-1) \times -(n-3) + 2t \times -(n-4) + 5 \times (n-5)] + 1, \tag{13}$$

$$C_{level1} + [(2t-1) \times -(n-3) + 2t \times -(n-4) + 5 \times (n-5)] = 0, \tag{14}$$

where (14) is verified by induction and so by combining (12) to (14) the sum of all levels region counts equals one.

For efficiency optimization we can remove specific size four outer regions produced by merging the triplet outer regions. If a size four outer region is only composed from a primary and an interaction triplet (sharing a moral edge) we can remove it (or not merge the two triplets), given that adding it will not introduce new interaction information for the associated moral edge. In general, as we know the exact number of parent nodes for each node pair connecting a moral edge, we can evaluate whether the merged region introduces new interaction information for the moral edge or not.

After applying the reduction operations we can remove more than half the size four outer regions. The resulting 1^{st} level of the region graph will contain a mixture of the size four and the triplet outer regions. We can verify that the optimized size four region graph will still satisfy the perfect correlation property (using the induction proof). And we only need to analyze the region removal process for once for the BFG during the efficiency optimization.

A.8 Proof that TRC time and space complexity is polynomial

For all BFGs the space complexity is proportional to the sum of all levels regions in a TRC region graph. The space complexity is proportional to $\sum_{3levels} v(r) \cdot total$ ("total" is the number of regions involved in each level of the region graph), which is polynomial.

Time complexity is proportional to the number of edges from the first to second level regions, which is the sum of all second level's degree of freedoms $\sum_{j=1}^{(n-2)^2} (|c_r| + 1)$ and is polynomial.

Appendix B. Methods

This section contains all the methods supplement to the algorithms in the paper.

B.1 ORI efficiency optimization by node reuse and unused triplets removal

Converting a BN to a BFG may introduce some intermediate nodes that do not replicate any original nodes in the BN. Outer regions created by these nodes can be safely removed. Given the replicated nodes are actually the same as the original nodes, then, for intermediate nodes that replicate original nodes, we can reuse the original nodes to replace these nodes in all outer regions. We explain the rationale of ORI efficiency optimization by node reuse and unused triplets removal for BNs below:

1. ***Node reuse:*** Replication is applied to reconstruct original node parent-child relationships along a specific path in the BFG. This means the primary and interaction triplets associated with these replicated nodes are only copying information from one replication to the other. The resulting outer region can be reduced from a triplet to a node pair if two nodes in the region become the same. The number of outer regions is therefore reduced since these node pairs are subsets of other triplet regions. This method does not alter the perfect correlation property because the reduction is achieved by merging subsets rather than removing them.

2. ***Unused triplets removal:*** We can increase computation efficiency further by removing primary and interaction triplets in pairs, but only when the child node in the primary triplet is an unused intermediate variable (i.e. does not replicate any original variable). This means this primary triplet does not influence any original variables in the model. In this case, removing the primary triplet r and an interaction triplet r' (r and r' share the same node pair that is connected by a moral edge), as a pair, won't change the local structure of other variables in the region graph, and the perfect correlation property will be preserved.

B.2 TRC efficiency optimization by relaxing the perfect correlation property

Following Algorithm 1 (ORI) with node reuse and unused triplets removal optimizations, we can keep reducing the number of interaction triplets but only if we relax the perfect correlation property. We call this method ***interaction triplet removal***. After node reuse and unused triplets removal, the number of remaining primary triplets is close to the number of factors in the original model. As a result, some interaction triplets remain but are disconnected from primary triplets and instead are only connected to neighboring interaction triplets. We can remove these interaction triplets provided that other interaction triplets connected to primary triplets are not affected. The perfect correlation property cannot be guaranteed given we are removing interaction triplets directly.

All replicated nodes are now replaced by original nodes and the unnecessary interaction triplets are removed; hence we will eventually remove all intermediate nodes. An intermediate node that does not replicate any original node can be removed. As a result, the remaining triplets are triplet cycles containing original nodes only.

B.3 CCCP updating equations

TRC uses CCCP or GBP for message passing. We omit the GBP message passing equations (Yedidia et al., 2005). The CCCP updating equations (Yuille, 2002) are given below:

$$h_r(x_r) = e^{-\frac{c_r}{c_{\max}}\{E_r(x_r)+1\}}\{b_r(x_r)\}^{\frac{c_{\max}-c_r}{c_{\max}}}, \tag{15}$$

$$g_r(x_r) = e^{-\gamma_r - \sum\limits_{s\in child(r)} \lambda_{r\to s}(x_s) + \sum\limits_{v\in parent(r)} \lambda_{v\to r}(x_r)}, \tag{16}$$

$$b_r(x_r) = h_r(x_r)g_r(x_r), \tag{17}$$

$$e^{2\lambda_{r\to u}(x_u;\tau+1)} = e^{2\lambda_{r\to u}(x_u;\tau)}\frac{\sum_{x\in r\setminus u} b_r}{b_u}. \tag{18}$$

Where: λ and γ are Lagrangian multipliers. c_{max} is the max value of all regions' counting numbers in a region graph. h_r and g_r are pre-calculated parameters for computing belief terms b_r. In the CCCP algorithm, updating each $\lambda_{r\to u}$ is a recursive process that involves calculating the beliefs over all u's parents and children, and its children's parents. CCCP can be updated in parallel provided that neighboring parent-child region messages are not affected.

Appendix C. Examples

This section contains four examples supplemental to the examples in the paper.

C.1 Example of the BF process applied to a BN

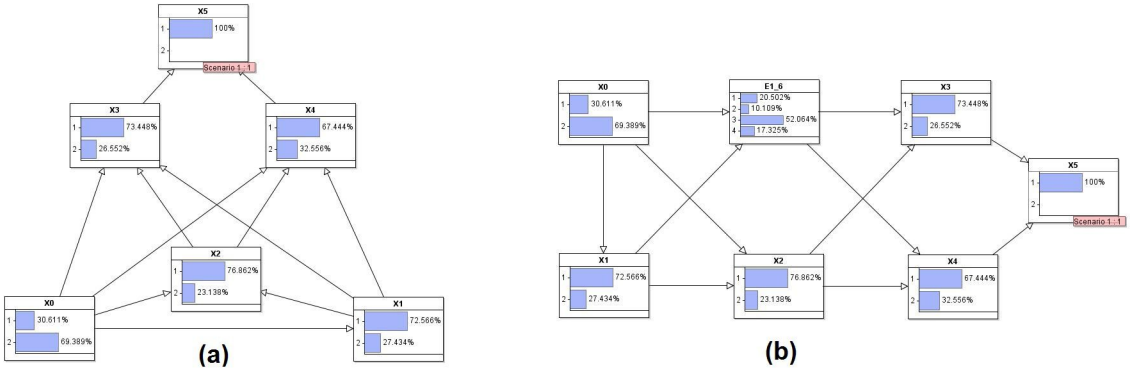

Figure 16: (a) 5-dimensional dense BN; (b) BF model of (a).

The 5-dimensional dense BN shown in Fig. 16 (a) is also introduced in the paper. To binary factorize the model, we can add an intermediate node E_1 that combines the X_0 and X_1 with a CPD: $p(E_1|X_0, X_1) = diag(1, 1, 1, 1)$ and obtain the BF model in (b).

C.2 Example of BFG conversion and obtaining TRC outer regions

We use the Asia BN as an example below to define the appropriate node ordering π_G for the BFG conversion.

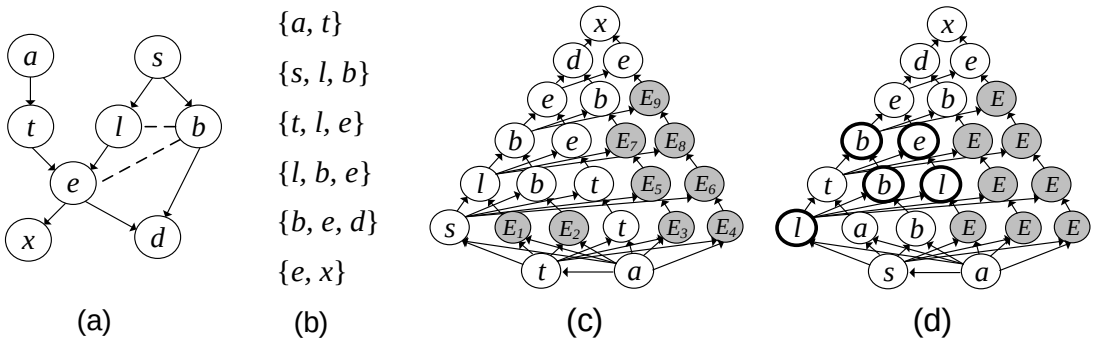

Figure 17: (a) the Asia BN with dashed lines a moral edge and a chordal edge; (b) cliques obtained by exact method; (c) BFG G_1; (d) BFG G_2.

We can obtain a node ordering: $\{a \to t \to s \to l \to b \to e \to d \to x\}$ where nodes in a clique ($\{s, l, b\}$ and $\{l, b, e\}$ shown in (b)) are ordered as neighbours, which is valid and then uniquely define a BFG G_1 in (c). In (d) we define another valid node ordering (i.e. obtained by breadth first search) that does not follow strictly from the node neighbouring order in cliques in (b), but we can obtain the same interaction triplets by using the replacement interaction triplets. In BFG G_2 we can still obtain $\{l, b, e\}$ (shown as bold face) and $\{s, l, b\}$ (shown as an interaction triplet), which are necessary triplet cycles. However, using (c) is more efficient than (d) as more intermediate nodes are left without replicating any original node in (c), so less regions will be created.

We can then obtain TRC outer regions as follows. Firstly, we can discard pairs of primary and interaction triplets if a primary triplet's child node is not used (unused triplet removal), such as all the E nodes in Fig. 17 (c). Next, we can reuse the original node to replace the replicated node to get the following regions (primary triplet followed by its interaction triplet) in Table 11.

Table 11: Primary triplets in the Asia BN's BFG in Fig. 17 (c)

$\{a, t, s\}$	with no interaction triplet
$\{a, t, t\} -> \{a, t\}$	can be simplified as node t appear twice
$\{s, t, t\} -> \{s, t\}$	with interaction triplet $\{t, s, t\} -> \{s, t\}$
$\{s, b, E_2\}$	with interaction $\{t, s, E_2\}$
$\{s, l, E_1\}$	with interaction $\{t, s, E_1\}$
$\{s, l, b\}$	with interaction $\{s, l, s\} -> \{s, l\}$
$\{t, l, e\}$	with interaction $\{s, l, t\}$
$\{b, e, e\} -> \{b, e\}$	with interaction $\{l, b, e\}$
$\{b, b, E_7\} -> \{b, E_7\}$	with interaction $\{l, b, E_7\}$
$\{b, e, d\}$	with interaction $\{b, b, e\} -> \{b, e\}$
$\{e, d, x\}$	with interaction $\{e, e, d\} -> \{e, d\}$

We then merge the repeated and subset regions to obtain the 1^{st} level of the TRC region graph containing 12 triplets, as shown in Table 12. There are three intermediate nodes E_1, E_2 and E_7 in the Table 12 list. The resulting TRC region graph satisfies the perfect correlation and maxent-normal properties. After the optimization, the number of 1^{st} level region is reduced from 36 to 12, which is close to the number of outer regions generated by FCB. We can convert the original CPDs to factors for the corresponding primary triplets. The interaction triplets will be uniform factors.

Table 12: TRC outer regions for the Asia BN's BFG in Fig. 17 (c)

$\{a, t, s\}$	primary triplet
$\{s, b, E_2\}$	primary triplet
$\{t, s, E_2\}$	interaction triplet
$\{s, E_1, l\}$	primary triplet
$\{t, s, E_1\}$	interaction triplet
$\{s, l, b\}$	primary triplet
$\{t, l, e\}$	primary triplet
$\{s, l, t\}$	interaction triplet
$\{b, e, l\}$	interaction triplet
$\{l, b, E_7\}$	interaction triplet
$\{b, e, d\}$	primary triplet
$\{e, d, x\}$	primary triplet

If we relax the perfect correlation property we can further reduce the above-listed interaction triplets using interaction triplet removal. For example, we can remove the interaction triplet for those regions containing E_2 and E_7 as they are not connected to primary triplets sharing the same node pair (connected by a moral edge). Finally, we can also remove the interaction triplet containing E_1 as from the original model the nodes in this triplet are independent. So after a set of reduction operations, we are left with the triplet cycles shown in Fig. 17 (a), which is to select $\{s, l, b\}$ and $\{b, e, l\}$ as interaction triplets.

C.3 Example to show how to obtain the TRC region graph for a coupled HMM

For simplicity in Fig. 18 (a), for the coupled HMM model shown, we removed three observed nodes. To construct the BFG for the BN we need to incrementally test the competing interaction redundancies for the BN and determine the list of competing interaction triplets to retain. We then obtain the BFG in (b). The optimized TRC region graph is then built using the same procedure used for the Asia model. In summary, we obtain the TRC region graph by these four steps: (1) detect the competing interaction triplets in the BN and define a node ordering; (2) construct the BFG; (3) construct the outer regions of the region graph using the BFG; (4) optimize the outer regions by a set of reduction operations and then generate the region graph using CVM.

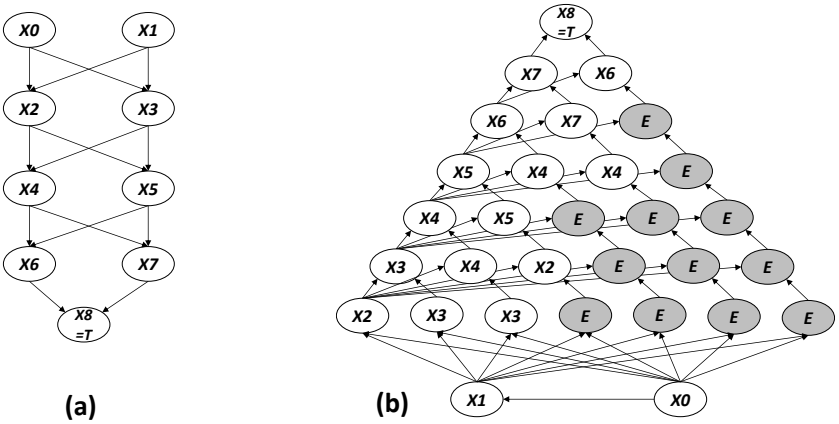

Figure 18: (a) four-time slices coupled HMM model with the last child node observed; (b) The corresponding κ_9 BFG of (a) where the E nodes in grey are intermediate nodes not replicating any original nodes.

C.4 Example to show how to merge triplet outer regions to size four outer regions

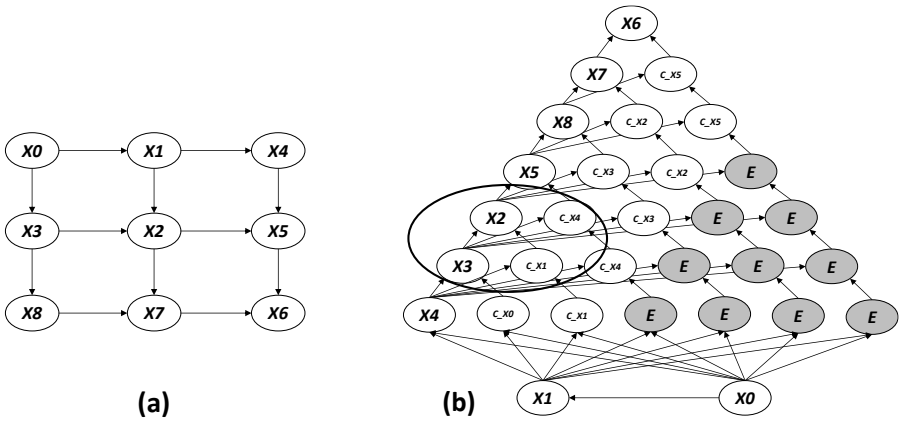

Figure 19: (a) a 3×3 Grid BN; (b) The corresponding κ_9 BFG where the C nodes are the replicated nodes and the E nodes in grey are intermediate nodes not replicating any original nodes.

In Fig. 19 (a) we show a Bayes Grid BN and its corresponding BFG (b) using 3×3 Grids. Firstly, we obtain the TRC triplet outer regions using the BFG in Fig. 19 (b). To obtain size four outer regions we will merge the primary triplets into the interaction triplets provided that the merged regions introduce new interaction information (compared to not merging) for the moral edges they are associated with. To determine whether the merged region introduces new interaction information for the moral edge we check what parent nodes are involved that can compute the pair-wise distribution of the moral edge

exactly. Obviously this information can be obtained directly from the BFG model. For instance, we will discard the merged region $\{C_{X_0}, X_1, X_3, X_4\}$ given it introduces no new interaction information for the moral edge $\{C_{X_0}, X_4\}$, compared to using the primary triplet $\{C_{X_0}, X_3, X_4\}$ and the interaction triplet $\{C_{X_0}, X_1, X_4\}$ without merging. To compute the pair-wise distribution $P_{C_{X_0}, X_4}$ exactly we need to incorporate the pair-wise information of $\{X_0, X_1\}$, which are the two parent nodes of the node pair $\{C_{X_0}, X_4\}$ (connecting the moral edge). But the region $\{C_{X_0}, X_1, X_3, X_4\}$ we've merged does not contain the information.

We will retain the merged region $\{C_{X_1}, X_2, X_3, C_{X_4}\}$ (shown in the bold circle in (b)) given it introduces new interaction information for the moral edge $\{X_2, C_{X_4}\}$. To compute $P_{X_2, C_{X_4}}$ exactly we need the joint distribution of $\{C_{X_1}, X_3, C_{X_4}\}$ (the three parent nodes of the node pair $\{X_2, C_{X_4}\}$), so the merged region $\{C_{X_1}, X_2, X_3, C_{X_4}\}$ is retained. Likewise, we know the exact number of parent nodes for each node pair connecting a moral edge, so the merging and removal process is guided, and we can achieve arbitrary outer region sizes > 3.

We have used this test to illustrate the merging process but alternatively we could use the replacement interaction triplet to obtain the target interaction triplet without the need to merge. This is evident in the BFG in (b) where there exists replacement interaction triplets for the moral edges $\{X_3, C_{X_1}\}$, $\{X_2, C_{X_4}\}$, $\{X_8, C_{X_2}\}$ and $\{X_7, C_{X_5}\}$.

Appendix D. Experiments

This section contains experiments supplement to the paper.

D.1 CPDs for Fig. 1 (a)

Table 13: CPDs for Fig. 1 (a)

X_1	.54	.46						
$X_2\|X_1$	.996	.004	.67	.33				
$X_3\|X_1, X_2$	.86	.14	.86	.14	.03	.97	.60	.4
$E_1\|X_1, X_2$	.98	.02	.996	.004	.09	.91	.35	.65
$X_4\|E_1, X_3$	.99	.01	.98	.02	.96	.04	.8	.2

D.2 Interaction change

This experiment is supplement to section 5.2 of the paper, which is a test of the ORI approximation quality when introducing the interaction change.

Fig. 20 (a) is a BN with a child node dependent on a number of parent nodes where all the parent nodes share a single ancestor node. Assuming they are all binary nodes, the exact solution for the induced cluster size is seven (by including variables X_1 to X_7 as a single cluster) and the induced cluster space is 128. Compared to the BN in (a), there is an interaction change caused by the BF algorithm for the model in (b). ORI needs to find the exact solution under the bounded cluster space 128 for the BF model in (b). There are different BF models available for (a), but the BF model in (b) ensures the maximum factor space (product of cardinalities of all variables in a factor) is also under 128. The cardinality

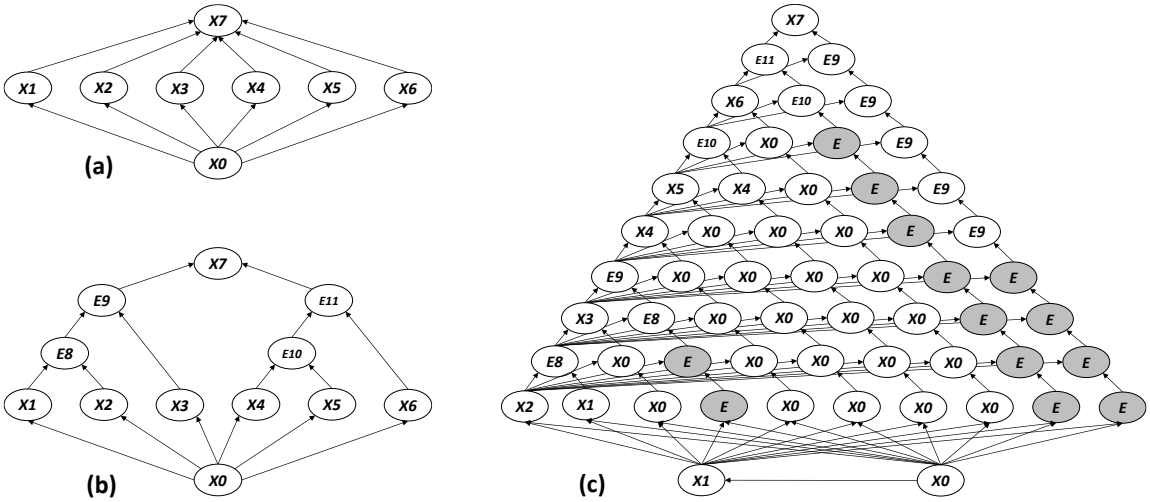

Figure 20: (a) BN G; (b) binary factorized model of (a); (c) BFG of (a).

is four for variables E_8 and E_{10}, and eight for variables E_9 and E_{11}. The corresponding BFG model of (b) is shown in (c).

Next, the ground truth of the original nodes for (a) and (b) is the same under exact methods (Neil et al., 2012). So we only need to verify the clusters generated by the exact method for (b) can be also generated using ORI from (c). If we index variables X_0 to X_7 to be $0, ..., 7$ and variables E_8 to E_{11} to be $8, ..., 11$, the clusters generated by the exact and the ORI solutions are given in Table 14.

Table 14: interaction change test

size	Exact	ORI	space
4	0 1 2 8	(1 2 8 + 0 2 8)	32
4	0 4 5 10	(0 4 5 + 4 5 10)	32
4	0 3 8 9	(3 8 9 + 0 3 9)	128
4	0 6 10 11	(6 10 11 + 0 6 10)	128
4	0 6 9 10	(0 6 10 + 6 9 10)	128
3	6 9 11	(6 9 11)	128
3	7 9 11	(7 9 11)	128

In Table 14 the ORI rebuilds the exact clusters by merging the primary and interaction triplets found in the BFG, without exceeding the bounded cluster space 128. This means, despite an interaction change, this does not incur a loss of information from the original nodes, and ORI has found the ground truth of the BN in (a) under the bounded cluster space (as obtained by exact methods).

D.3 Stress testing

Additional to section 5.2 experiments provided in the paper, we stress test all competing algorithms using the $(n+2)$ dimensional dense BN (Fig. 8) and $m \times n$ DBN (Fig. 9). These

BNs are either used in practical applications or embedded in other BNs as sub-structures. They are the worst cases as their moral graph contains dense sub-structures that reflect how moral edges are involved in competing interaction triplets. The experiments also test the approximation quality of TRC when interaction change occurs.

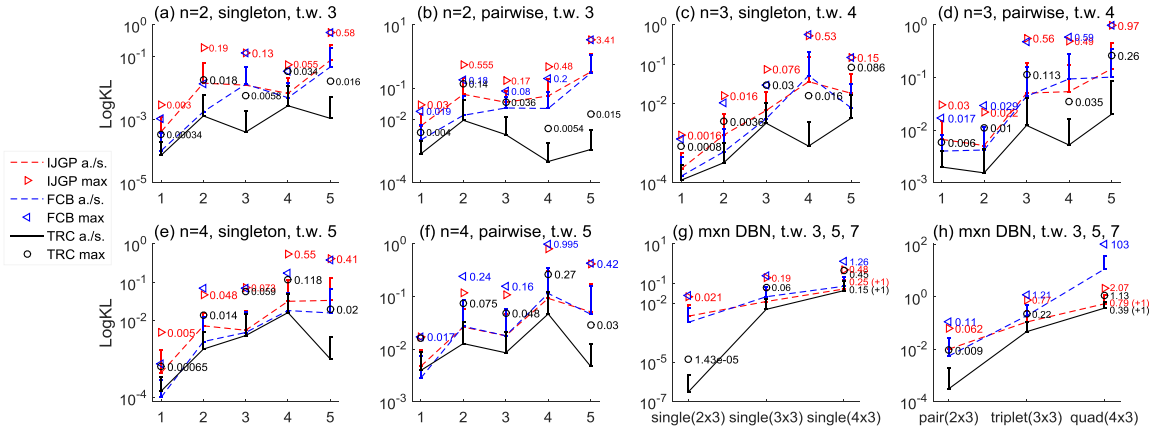

Figure 21: (a) to (f): TRC vs. competing algorithms when the difference between the cluster size and the tree-width is fixed but the factor strength (x-axis) is increasing, by using the $(n+2)$ dimensional dense BN, where $n = 2$, 3, or 4. The average/s.d. of the result is short for a./s.; (g) and (h): the factor strength is fixed but the difference between the cluster size and the (tree-width+1) is increasing for the $m \times n$ DBN, where $m \times n = 2 \times 3$, 3×3 and 4×3. The x-axis represents different ordered nodes' marginals being compared. The last child node in the model is observed. "(+1)" means the results are obtained with one increased cluster size and restrict the max cluster space equal to all compared algorithms.

In Fig. 21, we show the performance of each algorithm using the $(n+2)$ dimensional dense BN and the $m \times n$ DBN. Fig. 21 (a)-(f) is a factor strength test when fixing the difference (=1) between the cluster size and the (tree-width+1). The factors are sampled from a zero scaled log-normal distribution with s.d. from 1 to 5. The higher the s.d. the closer factor values (normalized) to zeros and ones, hence this generates stronger factor strength. The stronger the factor strength the worse the approximation and more severe the max variability problem can be. Fig. 21 (g)-(h) is a tree-width test where the factor strength is fixed (using extreme factors 0.99 and 0.01) but with increased difference between the cluster size and the tree-width. So, under these two settings, for the worst test models, we can evaluate how the averaged results and the max variability of each algorithm performs. We optimized the efficiency by node reuse and the perfect correlation property is retained for these tests.

We obtain each data point in Fig. 21 by 20 instances of random factors. In Fig. 21 (a)-(f) TRC significantly outperformed the competing algorithms for both singleton and pair-wise marginals. When the factor strength increases the performance of the competing algorithms generally decreases but TRC does not necessarily degrade (below or close to 10^{-2} while others exceed 10^{-1}). Also, the max variability of TRC is significantly lower

than the others.

In Fig. 21 (g)-(h), TRC also improves the results over others but the performance for all algorithms degrade as a function of the model dimension m. This is because the difference between the cluster size and the tree-width increases with the dimension m. If we increase the cluster size for IJGP and TRC both results for max values are improved, but TRC improves more significantly, as shown in (g) and (h) for the worst test case $m \times n = 4 \times 3$.

TRC simply merges the smaller clusters and those merges do not distort the approximation [10] but improve the result significantly. For a fair comparison we did not merge certain regions for TRC if they might exceed the maximum cluster space of others, by increasing one cluster size. But in practice we should merge to obtain higher accuracy. There are many choices for FCB to generate the fundamental cycles for these tests and the results vary significantly depending on which ones are used. In these tests the max variability statistics for FCB is worse than produced by the others. We also tried different outer regions for the Fig. 21 test models using CVM, and found the results are less accurate if the perfect correlation property is relaxed.

These tests verified that under extreme settings and with the interaction change the TRC algorithm still achieves better results than competing algorithms.

D.4 Efficiency comparison

Table 15 compares the efficiency of each algorithm for the test models presented in the paper. We obtain the results in Table 15 by running each algorithm under the CCCP message passing and convergence threshold 1.0e-08 for all algorithms. The CPU processor was an i5 4300m.

We performed the TRC efficiency optimization by node reuse and interaction triplet removal. There are many models contained in the PASCAL categories, so the values we compare are listed in Table 15 as single tests. We have bounded all algorithms with the same cluster space. Compared to IJGP and FCB, TRC can achieve similar efficiency when the perfect correlation property is relaxed.

TRC generally produced more interaction regions than the others as the outer regions found by TRC are more sufficient, but for many test cases TRC achieved similar efficiency. However, it is still possible to reduce outer regions further from the remaining interaction triplets, and we will explore it in future work. In addition, parallel computation for CCCP is another option to improve efficiency. We have implemented the parallel CCCP in Agena (AgenaRisk, 2020), which can improve the efficiency for at least 30% depending on the number of processors.

10. As discussed in (Welling, 2004), adding larger regions can distort the approximation for certain models, i.e. merging regions for the dense models could be harmful.

Table 15: Efficiency comparison of the competing algorithms using time (second) and cccp iterations

Models	IJGP		FCB		TRC	
	Time (s)	Iter.	Time (s)	Iter.	Time (s)	Iter.
Asia	0.1	58	0.1	41	0.22	70
Student	0.1	51	0.1	46	0.2	70
BayesGrid 5×5	1.1	95	1.7	127	3	170
Linkage	132	669	89	532	410	1065
4 d. dense BN	0.05	30	0.1	28	0.2	74
5 d. dense BN	0.1	56	0.2	51	0.5	113
*6 d. dense BN	0.3	141	0.3	115	1.7	235
*2×3 DBN	0.14	77	0.3	94	0.3	84
*3×3 DBN	0.4	105	1.6	223	1.8	207
coupled HMM	0.3	104	0.5	100	1.1	180
κ_{20} BFG	26	466	74	744	75	747
Diabetes	73	844	30	358	216	764
*Hopfield	0.6	150	1.5	176	2.3	258
Barley	3	201	3	176	18	478
Pedigree	55	318	14	154	210	658
Pigs	92	1080	44	672	217	1198
Promedas	151	2107	55	1226	65	1289
Munin	526	2316	362	1774	1290	2253
Protein	1326	584	1004	420	5600	757

D.5 High dimensional BFG tests

Table 16: KL distance for high dimensional BFG models (binary variables with normal random factors) test

t.w.	19 (κ_{20})	39 (κ_{40})	79 (κ_{80})	99 (κ_{100})
JT $O(2^n)$	8 Mb	8E3 Gb	9E15 Gb	9E21Gb
TRC $O(n^2)$	.06 Mb	.11 Mb	.47 Mb	.73 Mb
iterations	782	1567	3561	3963
max.(KL)	1.5E-04	1.9E-05	3.8E-05	2.8E-05
min.(KL)	3.7E-12	2.5E-13	2.1E-09	4.5E-08
ave.(KL)	1.5E-05	5.2E-06	7.5E-06	2.9E-06

Table 16 is a summary of the test results for κ_{20} (t.w. 19, 173 variables), κ_{40} (t.w. 39, 743 variables), κ_{80} (t.w. 79, 3083 variables) and κ_{100} (t.w. 99, 4853 variables) BFGs respectively, cluster space complexity for each model is compared to a JT solution. These models are very large and the results show that the cluster space complexity is reduced from

exponential to polynomial (from gigabytes to less than one megabyte). As the dimensions increase the accuracy does not notably decrease; all KL statistics show a robust and accurate performance and we can increase the convergence threshold to obtain higher accuracy. Because exact computation for all variables is computationally expensive we compare the accuracy of the first 20 dimensions produced under TRC. We would argue that if the first 20 dimensions are accurately approximated then higher dimension variables must also be accurate by the same approximation, otherwise the inaccuracy will be revealed in the lower dimensions, in priority order.

References

AgenaRisk (2020). Agena. `https://www.agenarisk.com`.

Andreassen, S., Jensen, F. V., Andersen, S. K., Falck, B., Kjærulff, U., Woldbye, M., Sørensen, A. R., Rosenfalck, A., & Jensen, F. (1989). MUNIN — an expert EMG assistant. In Desmedt, J. E. (Ed.), *Computer-Aided Electromyography and Expert Systems*, chap. 21. Elsevier Science Publishers, Amsterdam.

Andreassen, S., Hovorka, R., Benn, J., Olesen, K. G., & Carson, E. R. (1991). A model-based approach to insulin adjustment. In Stefanelli, M., Hasman, A., Fieschi, M., & Talmon, J. (Eds.), *Proceedings of the Third Conference on Artificial Intelligence in Medicine*, pp. 239–248. Springer-Verlag.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Batra, D., Nowozin, S., & Kohli, P. (2011). Tighter relaxations for map-mrf inference: A local primal-dual gap based separation algorithm. In Gordon, G., Dunson, D., & Dudík, M. (Eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proceedings of Machine Learning Research*, pp. 146–154, Fort Lauderdale, FL, USA. PMLR.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Ph.D. thesis, University College London, UK.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artif. Intell.*, *42*(2-3), 393–405.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory (2. ed.)*. Wiley.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in bayesian belief networks is np-hard. *Artif. Intell.*, *60*(1), 141–153.

Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks* (1st edition). Cambridge University Press, New York, NY, USA.

Dechter, R., & Rish, I. (1997). A scheme for approximating probabilistic inference. In *UAI '97: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Brown University, Providence, Rhode Island, USA, August 1-3, 1997*, pp. 132–141.

den Broeck, G. V., Mohan, K., Choi, A., & Pearl, J. (2014). Efficient algorithms for bayesian network parameter learning from incomplete data. *CoRR*, *abs/1411.7014*.

Elidan, G. (1998). Bn repository. https://www.cse.huji.ac.il/~galel/Repository/.

Elidan, G., Globerson, A., & Heinemann, U. (2011). Pascal 2011 probabilistic inference challenge. http://www.cs.huji.ac.il/project/PASCAL/,2012.

Forouzan, S., & Ihler, A. T. (2015). Incremental region selection for mini-bucket elimination bounds. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pp. 268–277.

Gelfand, A. (2011). Generalised belief propagation. https://github.com/aegelfand/GBP/,2011.

Gelfand, A., & Welling, M. (2012). Generalized belief propagation on tree robust structured region graphs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pp. 296–305.

Globerson, A., & Jaakkola, T. (2007). Approximate inference using conditional entropy decompositions. In Meila, M., & Shen, X. (Eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, Vol. 2 of *Proceedings of Machine Learning Research*, pp. 131–138, San Juan, Puerto Rico. PMLR.

Golumbic, M. C. (2004). Chapter 4 - triangulated graphs. In Golumbic, M. C. (Ed.), *Algorithmic Graph Theory and Perfect Graphs*, Vol. 57 of *Annals of Discrete Mathematics*, pp. 81 – 104. Elsevier.

Hazan, T., Peng, J., & Shashua, A. (2012). Tightening fractional covering upper bounds on the partition function for high-order region graphs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pp. 356–366.

Heskes, T. (2006). Convexity arguments for efficient minimization of the bethe and kikuchi free energies.. *Journal of Artificial Intelligence Research*, *26*.

Hoffman, M., & Blei, D. (2015). Stochastic Structured Variational Inference. In Lebanon, G., & Vishwanathan, S. V. N. (Eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, Vol. 38 of *Proceedings of Machine Learning Research*, pp. 361–369, San Diego, California, USA. PMLR.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14* (4), 1303–1347.

Jaimovich, A., Meshi, O., McGraw, I., & Elidan, G. (2010). Fastinf: An efficient approximate inference library. *J. Mach. Learn. Res.*, *11*, 1733–1736.

Jebara, T. (2014). *Tractability: Practical Approaches to Hard Problems*, chap. Perfect graphs and graphical modeling. Cambridge Press.

Jensen, C. S. (1998). Pedigree of breeding pigs. https://www.cse.huji.ac.il/~galel/Repository/Datasets/pigs/pigs.htm.

Jensen, C. S., & Kong, A. (1996). Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops. Research report R-96-2048, Department of Computer Science, Aalborg University, Denmark, Fredrik Bajers Vej 7, DK-9220 Aalborg Ø.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1998). An introduction to variational methods for graphical models. Tech. rep. UCB/CSD-98-980, EECS Department, University of California, Berkeley.

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

Komodakis, N., & Paragios, N. (2008). Beyond loose lp-relaxations: Optimizing mrfs by repairing cycles. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pp. 806–820, Berlin, Heidelberg. Springer-Verlag.

Kristensen, K. (1998). A preliminary model for barley. `https://www.cse.huji.ac.il/~galel/Repository/Datasets/barley/barley.htm`.

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

Lin, P. (2020). Triplet region construction. `https://github.com/penglin17/triplet-region-construction/`.

Liu, Q., & Ihler, A. T. (2011). Bounding the partition function using holder's inequality. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 849–856.

Marinescu, R. (2019). Merlin. `https://github.com/radum2275/merlin/blob/master/README.md`.

Masegosa, A. R., Martínez, A. M., Ramos-López, D., Cabañas, R., Salmerón, A., Nielsen, T. D., Langseth, H., & Madsen, A. L. (2016). d-VMP: Distributed variational message passing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, Vol. 52, pp. 321–332.

Masegosa, A. R., Martínez, A. M., Ramos-López, D., Cabañas, R., Salmerón, A., Nielsen, T. D., Langseth, H., & Madsen, A. L. (2017). AMIDST: a java toolbox for scalable probabilistic machine learning. *CoRR*, *abs/1704.01427*.

Mateescu, R., Kask, K., Gogate, V., & Dechter, R. (2010). Join-graph propagation algorithms. *J. Artif. Intell. Res. (JAIR)*, *37*, 279–328.

Meltzer, T., Globerson, A., & Weiss, Y. (2009). Convergent message passing algorithms - a unifying view. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pp. 393–401.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Neil, M., Chen, X., & Fenton, N. E. (2012). Optimizing the calculation of conditional probability tables in hybrid bayesian networks using binary factorization. *IEEE Trans. Knowl. Data Eng.*, *24*(7), 1306–1312.

Rollon, E., & Dechter, R. (2010). Evaluating partition strategies for mini-bucket elimination. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2010, Fort Lauderdale, Florida, USA, January 6-8, 2010*.

Sontag, D., Meltzer, T., Globerson, A., Weiss, Y., & Jaakkola, T. (2008). Tightening LP relaxations for MAP using message-passing. In *24th Conference in Uncertainty in Artificial Intelligence*, pp. 503–510. AUAI Press.

Wainwright, M. J., & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, *1*(1-2), 1–305.

Weiss, Y., Yanover, C., & Meltzer, T. (2007). MAP estimation, linear programming and belief propagation with convex free energies. In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, pp. 416–425.

Weller, A., & Jebara, T. (2013). Bethe bounds and approximating the global optimum. In *Sixteenth International Conference on Artificial Intelligence and Statistics*.

Welling, M. (2004). On the choice of regions for generalized belief propagation. In *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, pp. 585–592.

Welling, M., Minka, T. P., & Teh, Y. W. (2005). Structured region graphs: Morphing EP into GBP. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pp. 609–614.

Winn, J. (2004). *Variational Message Passing and its Applications*. Ph.D. thesis.

Winn, J. M., & Bishop, C. M. (2005). Variational message passing. *J. Mach. Learn. Res.*, *6*, 661–694.

Yedidia, J. S., Freeman, W. T., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*(7), 2282–2312.

Yuille, A. L. (2002). Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, *14*, 2002.

Yuille, A. L., & Rangarajan, A. (2003). The concave-convex procedure. *Neural Comput.*, *15*(4), 915–936.