

# Approximate Inference on Planar Graphs using Loop Calculus and Belief Propagation

**Vicenç Gómez**

**Hilbert J. Kappen**

*Department of Biophysics*

*Radboud University Nijmegen*

*6525 EZ Nijmegen, The Netherlands*

V.GOMEZ@SCIENCE.RU.NL

B.KAPPEN@SCIENCE.RU.NL

**Michael Chertkov**

*Theoretical Division and Center for Nonlinear Studies*

*Los Alamos National Laboratory*

*Los Alamos, NM 87545*

CHERTKOV@LANL.GOV

**Editor:** Tommi Jaakkola

## Abstract

We introduce novel results for approximate inference on planar graphical models using the loop calculus framework. The loop calculus (Chertkov and Chernyak, 2006a) allows to express the exact partition function of a graphical model as a finite sum of terms that can be evaluated once the belief propagation (BP) solution is known. In general, full summation over all correction terms is intractable. We develop an algorithm for the approach presented in Chertkov et al. (2008) which represents an efficient truncation scheme on planar graphs and a new representation of the series in terms of Pfaffians of matrices. We analyze the performance of the algorithm for models with binary variables and pairwise interactions on grids and other planar graphs. We study in detail both the loop series and the equivalent Pfaffian series and show that the first term of the Pfaffian series for the general, intractable planar model, can provide very accurate approximations. The algorithm outperforms previous truncation schemes of the loop series and is competitive with other state of the art methods for approximate inference.

**Keywords:** belief propagation, loop calculus, approximate inference, partition function, planar graphs, Ising model

## 1. Introduction

Graphical models are popular tools widely used in many areas which require modeling of uncertainty. They provide an effective approach through a compact representation of the joint probability distribution. The two most common types of graphical models are Bayesian networks (BN) and Markov random fields (MRFs).

The partition function of a graphical model, which plays the role of normalization constant in a MRF or probability of evidence (likelihood) in a BN is a fundamental quantity which arises in many contexts such as hypothesis testing or parameter estimation. Exact computation of this quantity is only feasible when the graph is not too complex, or equivalently, when its tree-width is small. Currently many methods are devoted to approximate this quantity.

The belief propagation (BP) algorithm (Pearl, 1988) is at the core of many of these approximate inference methods. Initially thought as an exact algorithm for tree graphs, it is widely used as an

approximation method for loopy graphs (Murphy et al., 1999; Frey and MacKay, 1998). The exact partition function is explicitly related to the BP approximation through the loop calculus framework introduced by Chertkov and Chernyak (2006a). Loop calculus allows to express the exact partition function as a finite sum of terms (loop series) that can be evaluated once the BP solution is known. Each term maps uniquely to a subgraph, also denoted as a generalized loop, where the connectivity of any node within the subgraph is *at least* degree two. Summation of the entire loop series is a hard combinatorial task since the number of generalized loops is typically exponential in the size of the graph (see also Watanabe and Fukumizu, 2009). However, different approximations can be obtained by considering different subsets of generalized loops in the graph.

It has been shown empirically (Gómez et al., 2007; Chertkov and Chernyak, 2006b) that truncating this series may provide efficient corrections to the initial BP approximation. More precisely, whenever BP performs satisfactorily, which occurs in the case of sufficiently weak interactions between variables (or short-range influence of loops), accounting for only a small number of terms is sufficient to recover the exact result. On the other hand, for those cases where BP requires many iterations to converge, many terms of the series are required to improve substantially the approximation. Nevertheless, a formal characterization of the classes of problems which are tractable via loop calculus still remains an open question.

A step toward this goal with a focus on the class of planar graphs has been done in Chertkov et al. (2008). A graph is said to be planar if it can be embedded into a plane without crossing edges. Chertkov et al. (2008) showed that under this condition summation of certain (large) subset of terms can be performed in polynomial time via mapping to the problem of counting perfect matchings. A perfect matching is a subset of edges where every vertex has exactly one attached edge in the subset. Furthermore, the full loop series can be expressed as a sum over certain Pfaffians (or determinants), where each Pfaffian may account for a large number of loops and is solvable in polynomial time as well.

The approach of Chertkov et al. (2008) builds upon classical results from Fisher (1961), Kasteleyn (1961) and Temperley and Fisher (1961) who addressed the question of counting the number of perfect matchings on a planar grid, also known as the dimer covering problem in statistical physics. These classical results are consistent with the following related statement: the partition function of a *planar graphical model defined in terms of binary variables* can be solved in polynomial time by computing an appropriate Pfaffian under the key restriction that pairwise interactions only depend on agreement or disagreement between the signs of their variables, and not on their individual values. Such a model is known in statistical physics as planar Ising model without external field and in the machine learning community as planar, binary MRF with pure interaction potentials. Notice that exact inference for a general binary graphical model on a planar graph, namely Ising model *with* external field, is intractable, in particular #P (Barahona, 1982). A gentle overview of counting perfect matchings in planar graphs can be found in Jerrum (2003).

Recently, other inference methods models based on the Fisher-Temperley-Kasteleyn (FTK) method have been introduced in the machine learning community. Globerson and Jaakkola (2007) obtained upper bounds on the partition function for non-planar graphs with binary variables by decomposition of the partition function into a weighted sum over partition functions of spanning tractable (zero field) planar models. The resulting problem is a convex optimization problem and, since exact inference can be done in each planar *sub*-model via the FTK method, the bound can be calculated in polynomial time.

Another example is the work of Schraudolph and Kamenetsky (2008) which provides a framework for exact inference on a restricted class of planar graphs using directly the FTK approach. Since a planar Ising model can be transformed to a model *without* external field at the cost of introducing additional edges, one can allow local fields for a subset  $\mathcal{B}$  of variables. If the graphical model is  $\mathcal{B}$ -outerplanar, which means that there exists a planar embedding in which the subset  $\mathcal{B}$  of the nodes lie on the same face, the FTK technique can still be applied.

Contrary to these two approaches, which rely on exact inference on a tractable planar model, the loop calculus directly leads to a framework for approximate inference on general planar graphs. Truncating the loop series according to Chertkov et al. (2008) already gives the exact result in the zero external field case. In the general planar case, however, this truncation may result into an accurate approximation that can be incrementally corrected by considering subsequent terms in the series.

In the next Section we review the main theoretical results of the loop calculus approach for planar graphs and introduce the proposed algorithm for approximating the partition function and single-variable marginals. In Section 3 we provide experimental results for regular grids and other types of planar graphs. We focus on a planar-intractable binary model with symmetric pairwise interactions but nonzero single variable potentials.<sup>1</sup> We end this manuscript with conclusions and discussion of future work in Section 4.

## 2. Belief Propagation and Loop Series for Planar Graphs

We consider the Forney graph representation, also called general vertex model (Jr., 2001; Loeliger, 2004), of a probability distribution  $p(\boldsymbol{\sigma})$  defined over a vector  $\boldsymbol{\sigma}$  of binary variables (vectors are denoted using bold symbols). Forney graphs are associated with general graphical models which subsume other factor graphs, for instance those correspondent to BNs and MRFs. In Appendix A we explain how our approach is related to the more common bipartite factor graph model.

A binary Forney graph  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$  consists of a set of nodes  $\mathcal{V}$  where each node  $a \in \mathcal{V}$  represents an interaction and each edge  $(a, b) \in \mathcal{E}$  represents a binary variable  $ab$  which take values  $\sigma_{ab} := \{\pm 1\}$ . We denote  $\bar{a}$  the set of neighbors of node  $a$ . Interactions  $f_a(\boldsymbol{\sigma}_a)$  are arbitrary functions defined over typically small subsets of variables where  $\boldsymbol{\sigma}_a$  is the vector of variables associated with node  $a$ , that is,  $\boldsymbol{\sigma}_a := (\sigma_{ab_1}, \sigma_{ab_2}, \dots)$  where  $b_i \in \bar{a}$ .

The joint probability distribution of such a model factorizes as:

$$p(\boldsymbol{\sigma}) = Z^{-1} \prod_{a \in \mathcal{V}} f_a(\boldsymbol{\sigma}_a), \quad Z = \sum_{\boldsymbol{\sigma}} \prod_{a \in \mathcal{V}} f_a(\boldsymbol{\sigma}_a), \quad (1)$$

where  $Z$  is the normalization factor, also called the partition function.

From a variational perspective, a fixed point of the BP algorithm represents a stationary point of the Bethe "free energy" approximation under proper constraints (Yedidia et al., 2000). In the Forney style notation:

$$Z^{BP} = \exp(-F^{BP}), \quad (2)$$

$$F^{BP} = \sum_a \sum_{\boldsymbol{\sigma}_a} \tau_a(\boldsymbol{\sigma}_a) \ln \left( \frac{\tau_a(\boldsymbol{\sigma}_a)}{f_a(\boldsymbol{\sigma}_a)} \right) - \sum_{b \in \bar{a}} \sum_{\sigma_{ab}} \tau_{ab}(\sigma_{ab}) \ln \tau_{ab}(\sigma_{ab}),$$

---

1. The source code used in this paper is freely available at <http://www.mbfys.ru.nl/staff/v.gomez/>.

where  $\tau_a(\sigma_a)$  and  $\tau_{ab}(\sigma_{ab})$  are the beliefs (pseudo-marginals) associated with each node  $a \in \mathcal{V}$  and edge  $(a, b) \in \mathcal{E}$ . For graphs without loops, Equation (2) coincides with the Gibbs "free energy" and therefore  $Z^{BP}$  coincides with the exact partition function  $Z$ . If the graph contains loops,  $Z^{BP}$  is just an approximation critically dependent on how strong the influence of the loops is.

We introduce now some convenient definitions related to the loop calculus framework.

**Definition 1** A *generalized loop* in a graph  $\mathcal{G} := \langle \mathcal{V}, \mathcal{E} \rangle$  is any subgraph  $C := \langle V', E' \rangle$ ,  $V' \subseteq \mathcal{V}, E' \subseteq (V' \times V') \cap \mathcal{E}$  such that each node in  $V'$  has degree two or larger.

For simplicity, we will use the term "loop", instead of "generalized loop", in the rest of this manuscript. Loop calculus allows to represent  $Z$  explicitly in terms of the BP approximation via the loop series expansion:

$$Z = Z^{BP} \cdot z, \quad z = \left( 1 + \sum_{C \in \mathcal{C}} r_C \right), \quad r_C = \prod_{a \in C} \mu_{a; \bar{a}_C}, \quad (3)$$

where  $\mathcal{C}$  is the set of all the loops within the graph and  $\bar{a}_C$  the set of neighbors of node  $a$  within the loop  $C$ . Each term  $r_C$  is a product of terms  $\mu_{a; \bar{a}_C}$  associated with every node  $a$  of the loop  $C$ :

$$\mu_{a; \bar{a}_C} = \frac{\mathbb{E}_{\tau_a} [\prod_{b \in \bar{a}_C} (\sigma_{ab} - m_{ab})]}{\sqrt{\prod_{b \in \bar{a}_C} \text{Var}_{\tau_{ab}}(\sigma_{ab})}}, \quad m_{ab} = \mathbb{E}_{\tau_{ab}} [\sigma_{ab}], \quad (4)$$

where we have used  $\mathbb{E}_{\tau}[\cdot]$  to denote expectation with respect to the pseudo-marginal distribution  $\tau$ . Equation (4) states that all terms of the expansion can be written as correlation functions between associated variables defined on the BP pseudo-marginals. In the case of  $\{\pm 1\}$  alphabet we have:

$$\mu_{a; \bar{a}_C} = \frac{\sum_{\sigma_a} (\tau_a(\sigma_a) \prod_{b \in \bar{a}_C} (\sigma_{ab} - m_{ab}))}{\prod_{b \in \bar{a}_C} \sqrt{1 - m_{ab}^2}}, \quad m_{ab} = \tau_{ab}(+1) - \tau_{ab}(-1). \quad (5)$$

In this work we consider planar graphs where nodes have degree at most three, that is  $|\bar{a}_C| \leq 3$ . We denote by *triplet* a node with degree exactly three in the graph. In Appendix A.2 we show how a graphical model stated in a bipartite factor graph representation can be converted to a Forney graph which satisfies this constraint at the cost of introducing auxiliary nodes.

**Definition 2** A *2-regular loop* is a loop in which all nodes have degree exactly two.

**Definition 3** The *2-regular partition function*  $Z_0$  is the truncated form of (3) which sums all 2-regular loops only:<sup>2</sup>

$$Z_0 = Z^{BP} \cdot z_0, \quad (6)$$

$$z_0 = 1 + \sum_{\substack{C \in \mathcal{C}, \\ |\bar{a}_C|=2, \forall a \in C}} r_C.$$

As an example, Figure 1a shows a small Forney graph and Figure 1c shows seven loops found in the corresponding 2-regular partition function.

2. Notice that this is the *single-connected partition function* in Chertkov et al. (2008). We use the term 2-regular partition function instead because loops with more than one connected component are also included in this part of the series.

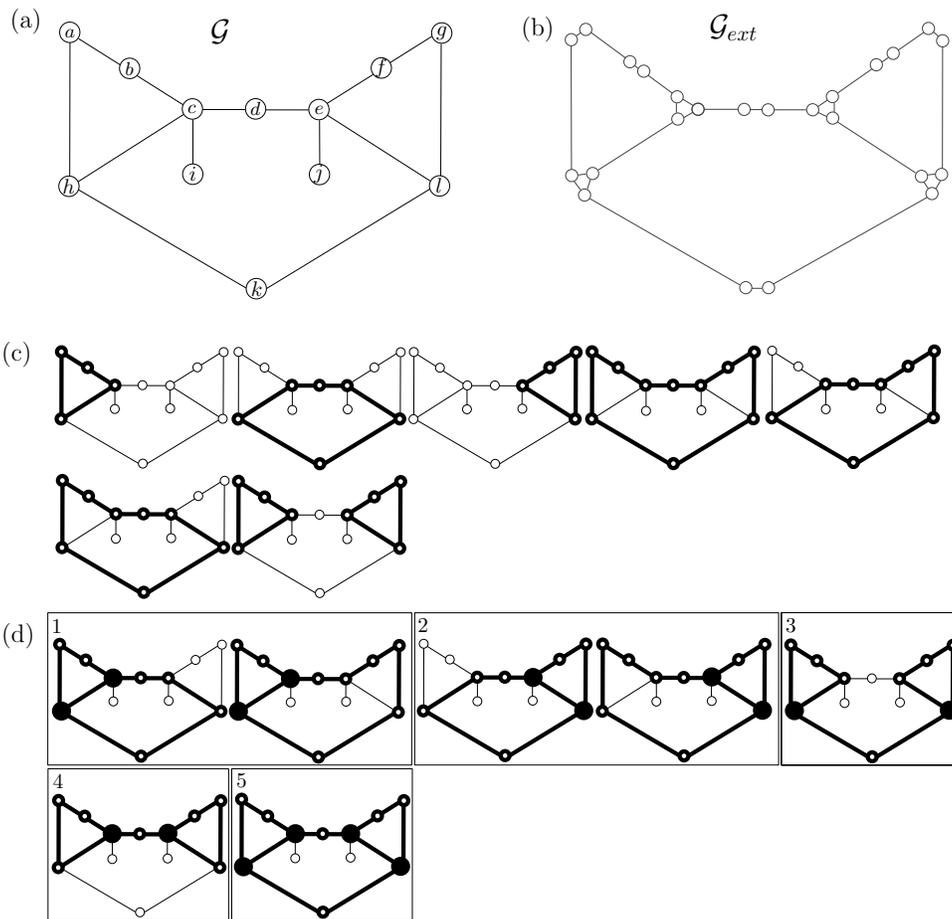


Figure 1: Example. **(a)** A Forney graph. **(b)** Corresponding extended graph. **(c)** Loops (in bold) included in the 2-regular partition function. **(d)** Loops (in bold) not included in the 2-regular partition function. Marked nodes denote triplets. Grouped in gray squares, the loops considered in different subsets  $\Psi$  of triplets: (d.1)  $\Psi = \{c, h\}$ , (d.2)  $\Psi = \{e, l\}$ , (d.3)  $\Psi = \{h, l\}$ , (d.4)  $\Psi = \{c, e\}$  and (d.5)  $\Psi = \{c, e, h, l\}$  (see Section 2.3).

**Definition 4** Consider the set  $P$  of all permutations  $\alpha$  of the set  $S = \{1, \dots, 2n\}$  in pairs:  $\alpha = (i_1, j_1), \dots, (i_n, j_n)$ ,  $i_k < j_k, \forall k = 1, \dots, n$ . The Pfaffian of a skew-symmetric matrix  $A = (A_{ij})_{1 \leq i < j \leq 2n}$  with  $(A_{ij} = -A_{ji})$  is:

$$\text{Pfaffian}(A) = \sum_{\alpha \in P} \text{sign}(\alpha) \prod_{(i,j) \in \alpha} A_{ij},$$

where the sign of a permutation  $\alpha$  is  $-1$  if the number of transpositions to get  $\alpha$  from  $S$  is odd and  $+1$  otherwise. The following identity allows to obtain the Pfaffian up to a sign by computing the determinant:

$$\text{Pfaffian}^2(A) = \text{Det}(A).$$

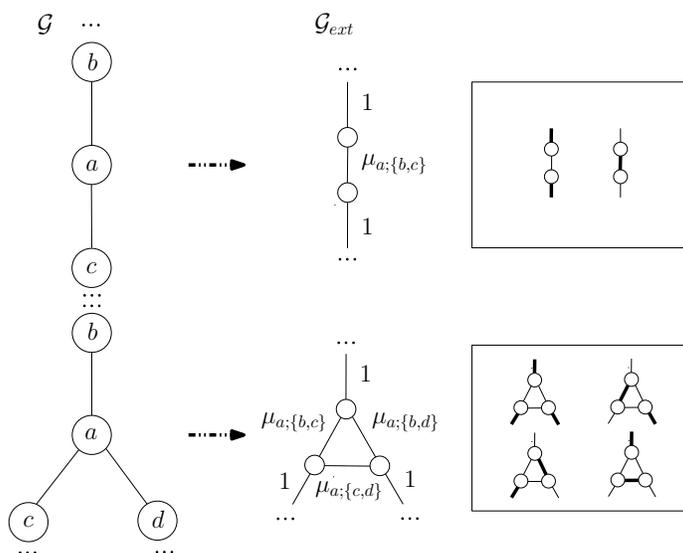


Figure 2: Fisher’s rules. **(Top)** A node  $a$  of degree two in  $\mathcal{G}$  is split in two nodes in  $\mathcal{G}_{ext}$ . **(Bottom)** A node  $a$  of degree three in  $\mathcal{G}$  is split in three nodes in  $\mathcal{G}_{ext}$ . The squares on the right indicate all possible matchings in  $\mathcal{G}_{ext}$  related to node  $a$ . Note that the rules preserve planarity.

### 2.1 Computing the 2-regular Partition Function Using Perfect Matchings

In Chertkov et al. (2008) it has been shown that computation of  $Z_0$  in a Forney graph  $\mathcal{G}$  can be mapped to the computation of the sum of all weighted perfect matchings within another extended weighted graph  $\mathcal{G}_{ext} := (\mathcal{V}_{\mathcal{G}_{ext}}, \mathcal{E}_{\mathcal{G}_{ext}})$ . A perfect matching is defined as a subset of edges such that each node neighbors exactly one edge from the subset and its weight is the product of weights of edges in it. The key idea of this mapping is that 2-regular loops in  $\mathcal{G}$  are in one-to-one correspondence to perfect matchings in  $\mathcal{G}_{ext}$  (see Figures 1b and c for an illustration). If  $\mathcal{G}_{ext}$  is planar, the sum of its weighted perfect matchings can be calculated in polynomial time via the FTK approach, evaluating the Pfaffian of an associated matrix. Here we reproduce these results with little variations and emphasis on the algorithmic aspects.

Given a Forney graph  $\mathcal{G}$  and the BP approximation, we obtain the 2-core of  $\mathcal{G}$  by removing nodes of degree 1 recursively. The 2-core excludes all nodes that do not appear in any loop. After this step,  $\mathcal{G}$  is either the null graph (and then BP is exact) or it is only composed of vertices of degree two or three.

To construct  $\mathcal{G}_{ext}$  we split each node in  $\mathcal{G}$  according to the rules introduced by Fisher (1966) and illustrated in Figure 2. The procedure results in an extended graph of  $|\mathcal{V}_{\mathcal{G}_{ext}}| \leq 3|\mathcal{V}|$  nodes and  $|\mathcal{E}_{\mathcal{G}_{ext}}| \leq 3|\mathcal{E}|$  edges. To see that each 2-regular loop in  $\mathcal{G}$  is associated with a perfect matching in  $\mathcal{G}_{ext}$  consider, for instance, the vertex of degree three in the bottom of Figure 2. Given a 2-regular loop  $C$ , vertex  $a$  can appear in four different configurations: either node  $a$  does not appear in  $C$ , or  $C$  contains one of the following three paths:  $-b-a-c-$ ,  $-b-a-d-$  or  $-c-a-d-$ . These four cases correspond to node terms in a loop with values  $1$ ,  $\mu_{a;\{b,c\}}$ ,  $\mu_{a;\{b,d\}}$  and  $\mu_{a;\{c,d\}}$  respectively, and coincide with

the matchings in  $\mathcal{G}_{ext}$  shown within the box on the bottom-right. An simpler argument applies to the vertex of degree two of the top portion of Figure 2.

Therefore, if we associate to each internal edge (new edge in  $\mathcal{G}_{ext}$  not in  $\mathcal{G}$ ) of each split node  $a$  the corresponding weight  $\mu_{a;\bar{a}_C}$  of Equation (4) and to the external edges (existing edges already in  $\mathcal{G}$ ) weight 1, the sum over all weighted perfect matchings defined on  $\mathcal{G}_{ext}$  is precisely  $z_0$ :

$$z_0 = \sum \text{perfect matchings in } \mathcal{G}_{ext}.$$

The 2-regular partition function  $Z_0$  is obtained using Equation (6).

Kasteleyn (1963) provided a method to compute this sum in polynomial time for planar graphs. First, edges are properly oriented in such a way that for every face (except possibly the external face) the number of clockwise oriented edges is odd. We use the linear time algorithm in Karpinski and Rytter (1998) described here as Algorithm 1 to produce such an orientation. It receives an undirected graph  $\mathcal{G}_{ext}$  and constructs a copy  $\mathcal{G}'_{ext} := (\mathcal{V}'_{\mathcal{G}'_{ext}}, \mathcal{E}'_{\mathcal{G}'_{ext}})$  with properly oriented edges  $\mathcal{E}'_{\mathcal{G}'_{ext}}$ . It is convenient that  $\mathcal{G}_{ext}$  is bi-connected, that is, has no articulation points. If needed, we add edges with zero weight which do not alter  $Z$ .

---

**Algorithm 1** Pfaffian orientation

---

**Arguments:** undirected bi-connected extended graph  $\mathcal{G}_{ext}$ .

- 1: Construct a planar embedding  $\tilde{\mathcal{G}}_{ext}$  of  $\mathcal{G}_{ext}$ .
  - 2: Construct a spanning tree  $T$  of  $\tilde{\mathcal{G}}_{ext}$ .
  - 3: Construct a graph  $H$  having vertices corresponding to the faces of  $\tilde{\mathcal{G}}_{ext}$ :  
Connect two vertices in  $H$  if the respective face boundaries share an edge not in  $T$ .  
 $H$  is a tree. Root  $H$  to the external face.
  - 4:  $\mathcal{G}'_{ext} := T$ .
  - 5: Orient all edges in  $\mathcal{G}'_{ext}$  arbitrarily.
  - 6: **for all** face (vertex in  $H$ ) traversed in post-order **do**
  - 7:   Add to  $\mathcal{G}'_{ext}$  the unique edge not in  $\mathcal{G}'_{ext}$ .
  - 8:   Orient it such that the number of clock-wise oriented edges is odd.
  - 9: **end for**
  - 10: **RETURN** directed bi-connected extended graph  $\mathcal{G}'_{ext}$ .
- 

Second, denote  $\mu_{ij}$  the weight of the edge between nodes  $i$  and  $j$  in  $\mathcal{G}'_{ext}$  and define the following skew-symmetric matrix:

$$\hat{A}_{ij} = \begin{cases} +\mu_{ij} & \text{if } (i, j) \in \mathcal{E}'_{\mathcal{G}'_{ext}} \\ -\mu_{ij} & \text{if } (j, i) \in \mathcal{E}'_{\mathcal{G}'_{ext}} \\ 0 & \text{otherwise} \end{cases}.$$

The Pfaffian of  $\hat{A}_{ij}$  is the weighted sum of all perfect matchings. Calculation of  $z_0$  can be performed in time  $O(N_{\mathcal{G}_{ext}}^3)$ :

$$z_0 = \sqrt{\text{Det}(\hat{A})}.$$

For the tractable case, namely, Ising model without external field or pairwise MRF with pure interaction potentials, the 2-regular partition function coincides with the exact partition function  $Z =$

$Z_0 = Z^{BP} \cdot z_0$  since other terms in the loop series vanish. In this case, it can be shown that all terms  $\mu_{a;\{b,c,d\}}$  associated with triplets  $a$  (vertices with degree three in  $\mathcal{G}$ ) are zero using the following argument: first, note that a BP fixed point is attained when all variables have zero magnetizations, that is, uniform beliefs  $\tau_{ab}(\sigma_{ab}) = 1/2$ , which reduces Equation (5) to:

$$\mu_{a;\{b,c,d\}} = \sum_{\sigma_a} \tau_a(\sigma_a) \sigma_{ab} \sigma_{ac} \sigma_{ad}. \tag{7}$$

Second, since by construction pseudo-marginals on the triplets  $\tau_a(\sigma_a)$  only depend on agreement or disagreement between the variables:

$$\tau_a(\sigma_a) = \begin{cases} \tau_a(=), & \text{for } \sigma_{ab} = \sigma_{ac} = \sigma_{ad}, \\ \tau_a(\neq), & \text{otherwise,} \end{cases}$$

it suffices to see that for each term  $\tau_a(\sigma_a) \sigma_{ab} \sigma_{ac} \sigma_{ad}$  in the sum, the "symmetric" term where  $\sigma_{ab}, \sigma_{ac}$  and  $\sigma_{ad}$  are replaced by their opposite values, has same absolute value but different sign, therefore the sum (7) is zero.

### 2.2 Computing Estimates of Marginal Distributions

Given the estimate  $Z_0$ , estimates for the marginals  $P_{ab}(\sigma_{ab})$  can be obtained:

$$P_{ab}(\sigma_{ab}) = \left. \frac{\partial \log Z_0(\theta_{ab})}{\partial \theta_{ab}(\sigma_{ab})} \right|_{\theta_{ab} \rightarrow 0}, \quad \text{where} \quad Z_0(\theta_{ab}) := \sum_{\sigma} \exp(\theta_{ab} \sigma_{ab}) \prod_{a \in \mathcal{V}} f_a(\sigma_a)$$

is the partition sum of the network perturbed with respect to an additional local field potential  $\theta_{ab}$  of variable  $\sigma_{ab}$ .

Alternatively, one can compute different partition functions for different settings of the variables  $Z_0^{\sigma_{ab}=+1}$  and  $Z_0^{\sigma_{ab}=-1}$ , and derive the marginals from respective ratios:

$$P_{ab}(\sigma_{ab} = +1) = \frac{Z_0^{\sigma_{ab}=+1}}{Z_0^{\sigma_{ab}=+1} + Z_0^{\sigma_{ab}=-1}},$$

where  $Z_0^{\sigma_{ab}=+1}$  indicates the 2-regular partition function calculated from the same model conditioning on variable  $\sigma_{ab}$ , that is, with variable  $\sigma_{ab}$  fixed (clamped) to  $+1$ . Therefore, approximation errors in the computation of any marginal can be related to approximation errors in the computation of  $Z_0$ . In the following we will mainly focus on evaluating  $Z_0$ , although marginal probabilities will be discussed as well.

### 2.3 Computing the Full Loop Series Using Perfect Matchings

Chertkov et al. (2008) established that  $z_0$  is just the first term of a finite sum involving Pfaffians. We briefly reproduce their results here and provide an algorithm for computing the full loop series as a Pfaffian series.

Consider  $\mathcal{T}$  defined as the set of all possible triplets in the original graph  $\mathcal{G}$ . For each possible subset  $\Psi \in \mathcal{T}$ , including an *even* number of triplets, there exists a unique correspondence between loops in  $\mathcal{G}$  including the triplets in  $\Psi$  and perfect matchings in another extended graph  $\mathcal{G}_{ext\Psi}$  constructed after removal of the triplets  $\Psi$  in  $\mathcal{G}$ . Using this representation the full loop series can be

represented as a Pfaffian series, where each term  $Z_\Psi$  is tractable and is a product of the respective Pfaffian and the  $\mu_{a;\bar{a}}$  terms associated with each triplet of  $\Psi$ :<sup>3</sup>

$$z = \sum_{\Psi} Z_{\Psi} \qquad Z_{\Psi} = z_{\Psi} \prod_{a \in \Psi} \mu_{a;\bar{a}} \qquad (8)$$

$$z_{\Psi} = \text{sign}(\text{Pfaffian}(\hat{B}_{\Psi})) \cdot \text{Pfaffian}(\hat{A}_{\Psi}).$$

where  $\hat{B}_{\Psi}$  corresponds to the original Kasteleyn matrix with weights  $+1$  instead of  $+\mu_{ij}$  and  $-1$  instead of  $-\mu_{ij}$  and we make explicit use of the Pfaffians to correct for the sign. The correction  $\text{sign}(\text{Pfaffian}(\hat{B}_{\Psi}))$  is necessary because loop terms can be negative and even evaluating  $\text{Pfaffian}(\hat{A}_{\Psi})$  with the correct sign would only give the contribution up to a sign. In the previous Subsection, we assumed  $z_0$  positive.

The 2-regular partition function thus corresponds to  $\Psi = \emptyset$ . We refer to the remaining terms of the series ( $\Psi \neq \emptyset$ ) as higher order terms. Notice that matrices  $\hat{A}_{\Psi}$  and  $\hat{B}_{\Psi}$  depend on the removed triplets and therefore each  $z_{\Psi}$  requires different matrices and different edge orientations. In addition, after removal of vertices in  $\mathcal{G}$ , the resulting extended graph may be disconnected. As before, in these cases we add dummy edges with zero weight so that  $\mathcal{G}_{ext}$  remains bi-connected.

Figure 1d shows loops corresponding to the higher order Pfaffian terms on our illustrative example. The first and second subsets of triplets (d.1 and d.2) include summation over two loops whereas the remaining Pfaffian terms include uniquely one loop.

Exhaustive enumeration of all subsets of triplets leads to a  $2^{|\mathcal{T}|}$  time algorithm, which is prohibitive. However, many triplet combinations may lead to forbidden configurations. Experimentally, we found that a principled way to look for higher order Pfaffian terms with large contribution is to search first for pairs of triplets, then groups of four, and so on. For large graphs, this also becomes intractable. However, the key advantage of the Pfaffian representation is that  $Z_0$  is always the term that accounts for the largest number of loop terms in the series. In this work we do not derive any heuristic for searching higher order Pfaffian terms with larger contributions. Instead, in Section 3.1 we study the full Pfaffian series and subsequently we restrict our attention to analyze the accuracy of  $Z_0$ .

Algorithm 2 describes the procedure for computing all terms using Equation (8). The main loop of the algorithm can be interrupted at any time, thus leading to a sequence of algorithms producing corrections incrementally.

### 3. Experiments

In this Section we study numerically the proposed algorithm. To facilitate the evaluation and the comparison with other algorithms we focus on the pairwise binary Ising model, a particular case of the model (1) where factors only depend on the disagreement between variables and take the form:  $\log f_a(\sigma_{ab}, \sigma_{ac}) = \phi_a \sigma_{ab} \sigma_{ac}$ . We consider also nonzero local potentials in all variables parameterized by:  $\log f_a(\sigma_{ab}) = \theta_{ab} \sigma_{ab}$  so that the model becomes planar-intractable.

We create different inference problems by choosing different interactions  $\phi_a$  and local field parameters  $\theta_{ab}$ . To generate them we draw independent samples from a normal distribution  $\phi_a \sim \mathcal{N}(0, \beta/2)$  and  $\theta_{ab} \sim \mathcal{N}(0, \beta\Theta)$ , where  $\Theta$  and  $\beta$  determine how difficult the inference problem is. Generally, for  $\Theta = 0$  the planar problem is tractable (zero field). For  $\Theta > 0$ , small values of  $\beta$  result

---

3. We omit the loop index in the triplet term  $\mu_{a;\bar{a}}$  because nodes have at most degree three and therefore the set  $\bar{a}$  always coincide in all loops which contain that triplet.

---

**Algorithm 2** Pfaffian series

---

**Arguments:** Forney graph  $\mathcal{G}$ 

```

1:  $z := 0$ .
2: for all  $\{\Psi \in \mathcal{T}, |\Psi| \text{even}\}$  do
3:   Build extended graph  $\mathcal{G}_{ext\Psi}$  applying rules of Figure 2.
4:   Set Pfaffian orientation in  $\mathcal{G}_{ext\Psi}$  according to Algorithm 1.
5:   Build matrices  $\hat{A}$  and  $\hat{B}$ .
6:   Compute Pfaffian with sign correction  $z_\Psi$  according to Equation (8).
7:    $z := z + z_\Psi \prod_{a \in \Psi} \mu_{a;\bar{a}}$ .
8: end for
9: RETURN  $Z^{BP} \cdot z$ .

```

---

in weakly coupled variables (easy problems) and large values of  $\beta$  in strongly coupled variables (hard problems). Larger values of  $\Theta$  result in easier inference problems.

In the next Subsection we analyze the full Pfaffian series using a small example and compare it with the original representation based on the loop series. Next, we compare our algorithm with the following ones:<sup>4</sup>

**Truncated Loop-Series for BP (TLSBP)** (Gómez et al., 2007), which accounts for a certain number of loops by performing depth-first-search on the factor graph and then merging the found loops iteratively. We adapted TLSBP as an any-time algorithm (**anyTLSBP**) in a way that the length of the loop is used as the only parameter instead of the two parameters  $S$  and  $M$  (see Gómez et al., 2007, for details). This is equivalent to setting  $M = 0$  and discarding  $S$ . In this way, anyTLSBP does not compute all possible loops of a certain length (in particular, complex loops<sup>5</sup> are not included), but search can be performed faster.

**Cluster Variation Method (CVM-Loopk)** A double-loop implementation of CVM (Heskes et al., 2003). This algorithm is a special case of generalized belief propagation (Yedidia et al., 2005) with convergence guarantees. We use as outer clusters all (maximal) factors together with loops of four ( $k=4$ ) or six ( $k=6$ ) variables in the factor graph.

**Tree-Structured Expectation Propagation (TreeEP)** (Minka and Qi, 2004). This method performs exact inference on a base tree of the graphical model and approximates the other interactions. The method yields good results if the graphical model is very sparse.

When possible, we also compare with the following two variational methods which provide upper bounds on the partition function:

**Tree Reweighting (TRW)** (Wainwright et al., 2005) which decomposes the parameterization of a probabilistic graphical model as a mixture of spanning trees of the model, and then uses the convexity of the partition function to get an upper bound.

**Planar graph decomposition (PDC)** (Globerson and Jaakkola, 2007) which decomposes the parameterization of a probabilistic graphical model as a mixture of tractable planar graphs (with zero local field).

---

4. We use the libDAI library (Mooij, 2008) for algorithms **CVM-Loopk**, **TreeEP** and **TRW**.

5. A complex loop is defined as a loop which can not be expressed as the union of two or more circuits or simple loops.

To evaluate the accuracy of the approximations we consider errors in  $Z$  and, when possible, computational cost as well. As shown in Gómez et al. (2007), errors in  $Z$ , obtained from a truncated form of the loop series, are very similar to errors in single variable marginal probabilities, which can be obtained by conditioning over the variables under interest. We only consider tractable instances for which  $Z$  can be computed via the junction tree algorithm (Lauritzen and Spiegelhalter, 1988) using 8GB of memory.

The error measure for a given approximation  $Z'$  of  $Z$  is:

$$\text{error } Z' = \frac{|\log Z - \log Z'|}{\log Z},$$

and given  $P'_{ab}(\sigma_{ab})$  estimates for the exact marginals  $P_{ab}(\sigma_{ab})$ , the error is:

$$\text{error marginals} = \text{mean}_{\substack{(a,b) \in \mathcal{E} \\ \sigma_{ab} = \pm 1}} |P'_{ab}(\sigma_{ab}) - P_{ab}(\sigma_{ab})|.$$

As in Gómez et al. (2007), we use four different message updates for BP: fixed and random sequential updates, parallel (or synchronous) updates, and residual belief propagation (RBP), a method proposed by Elidan et al. (2006) which selects the next message to be updated which has maximum *residual*, a quantity defined as an upper bound on the distance of the current messages from the fixed point. We report non-convergence when none of the previous methods converged. We report convergence at iteration  $t$  when the maximum absolute value of the updates of all the messages from iteration  $t - 1$  to  $t$  is smaller than a threshold  $\vartheta = 10^{-14}$ .

### 3.1 Full Pfaffian Series

In the previous Section we have described two equivalent representations for  $Z$  in terms of the loop series and the Pfaffian series. Here we analyze numerically how these two representations differ using an example, shown in Figure 3 as a bipartite factor graph, for which all terms of both series can be computed. We analyze a single instance, parameterized using  $\Theta = 0.1$  and different pairwise interactions  $\beta \in \{0.1, 0.5, 1.5\}$ .

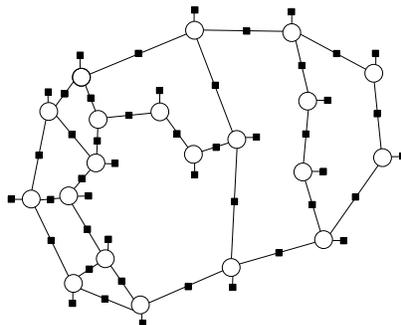


Figure 3: Planar bipartite factor graph used to compare the full Pfaffian series with the loop series. Circles and black squares denote variables and factors respectively.

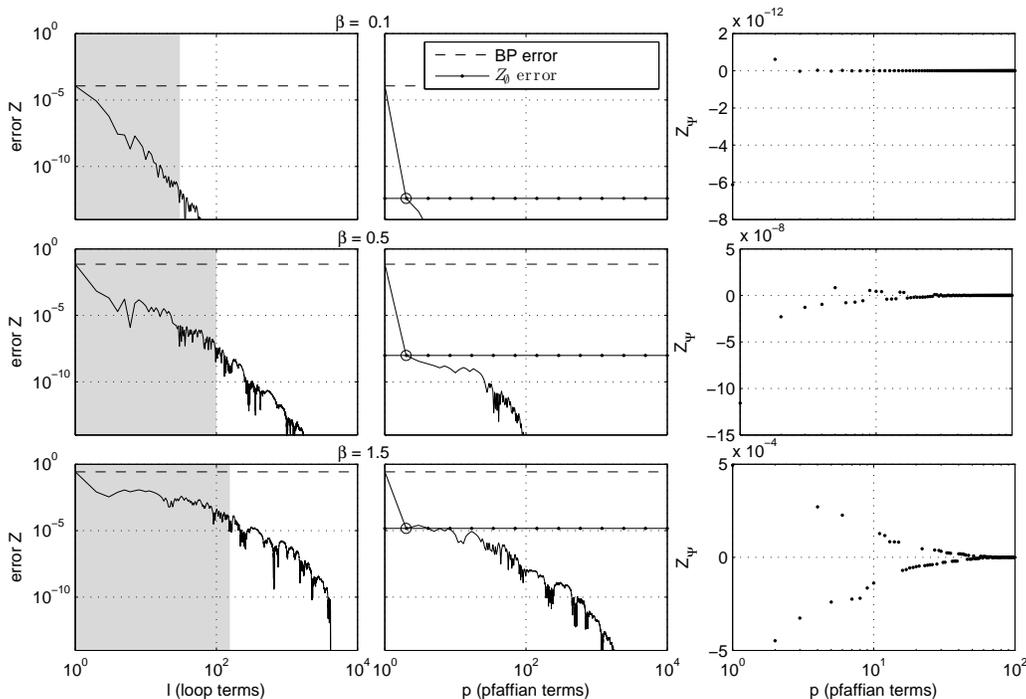


Figure 4: Comparison between the full loop series and the full Pfaffian series. Each row corresponds to a different value of the interaction strength  $\beta$ . **Left column** shows the error, considering loop terms  $Z^{TLSBP}(l)$  in log-log scale. Shaded regions include all loop terms (not necessarily 2-regular loops) required to achieve the same (or better) accuracy than the accuracy of the 2-regular partition function  $Z_0$ . **Middle column** shows the error considering Pfaffian terms  $Z^{Pf}(p)$  also in log-log scale. The first Pfaffian term corresponds to  $Z_0$ , marked by a circle. **Right column** shows the values of the first 100 Pfaffian terms sorted in descending order in  $|Z_\Psi|$  and excluding  $z_0$ .

We use TLSBP to retrieve all loops, 8123 for this example, and Algorithm 2 to compute all Pfaffian terms. To compare the two approximations we sort all contributions, either loops or Pfaffians, by their absolute values in descending order, and then analyze how the errors are corrected as more terms are included in the approximation. We define partition functions for the truncated series in the following way:

$$Z^{TLSBP}(l) = Z^{BP} \left( 1 + \sum_{i=1 \dots l} r_{C_i} \right), \quad Z^{Pf}(p) = Z^{BP} \left( \sum_{i=1 \dots p} Z_{\Psi_i} \right).$$

Then  $Z^{TLSBP}(l)$  accounts for the  $l$  most contributing loops and  $Z^{Pf}(p)$  accounts for the  $p$  most contributing Pfaffian terms. In all cases, the Pfaffian term with largest absolute value  $Z_{\Psi_1}$  corresponds to  $z_0$ .

Figure 4 shows the error  $Z^{TLSBP}$  (first column) and  $Z^{Pf}$  (second column) for both representations. For weak interactions ( $\beta = 0.1$ ) BP converges fast and provides an accurate approximation

with an error of order  $10^{-4}$ . Summation of less than 50 loop terms (top-left panel) is enough to obtain machine precision accuracy. Notice that the error is almost reduced totally with  $Z_0$  (top-middle panel). In this scenario, higher order terms of the Pfaffian series are negligible (top-right panel).

As we increase  $\beta$ , the quality of the BP approximation decreases. The number of loop corrections required to correct the BP error then increases. In this example, for intermediate interactions ( $\beta = 0.5$ ) the first Pfaffian term  $z_0$  improves considerably, more than five orders of magnitude, on the BP error, whereas approximately 100 loop terms are required to achieve a similar correction (gray region of middle-left panel).

For strong interactions ( $\beta = 1.5$ ) BP converges after many iterations and gives a poor approximation. In this scenario also a larger proportion of loop terms (bottom-left panel) is necessary to correct the BP result up to machine precision. Looking at the bottom-left panel we find that approximately 200 loop terms are required to achieve the same correction as the one obtained by  $Z_0$ , which is quite accurate (bottom-middle panel).

As the right panels show, higher order Pfaffian contributions change progressively from a flat sequence of small terms to an alternating sequence of positive and negative terms which grow in absolute value as  $\beta$  increases. These oscillations are also present in the loop series expansion.

In general, we conclude that the  $Z_0$  correction to the BP approximation can give a significant improvement even in hard problems for which BP converges after many iterations. Notice again that calculating  $Z_0$ , the most contributing term in the Pfaffian series, does not require explicit search for loop nor Pfaffian terms.

### 3.2 Grids

After analyzing the full Pfaffian series on a small random example we now restrict our attention to the first Pfaffian correction using grids (nearest neighbor connectivity). First, we compare this approximation, for both  $Z_0$  and single-variable marginals, with other inference methods for different types of interactions (attractive or mixed) and then study the scalability of the method with the size of the graphs.

#### 3.2.1 ATTRACTIVE INTERACTIONS

We first focus on binary models with "ferromagnetic" tendency, which favors alignment of neighboring variables,  $\phi_a > 0$ . If local fields are also positive  $\theta_{ab} > 0$ , Sudderth et al. (2007) showed that, under some additional condition, the BP approximation gives a *lower-bound* for the exact partition function and all loops (and therefore Pfaffian terms too) have the same sign.<sup>6</sup> Although this is not formally proved for general models with attractive interactions regardless of the sign of the local fields, numerical results suggest that this property holds as well for this more general type of models.

We generate grids with positive interactions and local fields, that is  $|\phi_a| \sim \mathcal{N}(0, \beta/2)$  and  $|\theta_{ab}| \sim \mathcal{N}(0, \beta\Theta)$ , and study performance of the algorithms for various values of  $\beta$ , as well as for strong  $\Theta = 1$  and weak  $\Theta = 0.1$  local fields.

Figure 5 shows the average error over 50 instances reported by different methods. For this setup, BP converged in all instances using random sequential updates of the messages. The error curves of all methods show an initial growth and a subsequent decrease, a fact explained by the phase

---

6. The condition is that all single variable beliefs at the BP fixed point must satisfy  $m_{ab} = \tau_{ab}(+1) - \tau_{ab}(-1) > 0, \forall (a,b) \in \mathcal{E}$ .

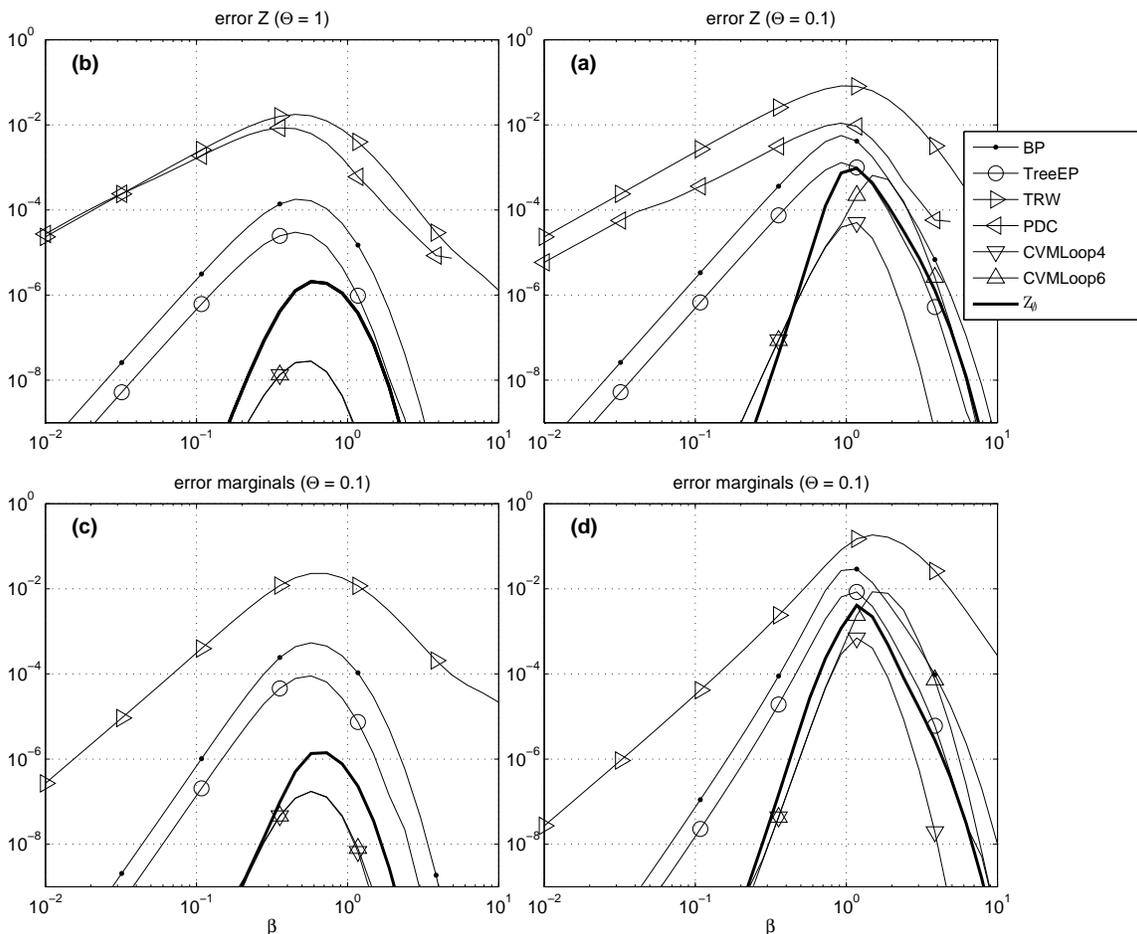


Figure 5:  $7 \times 7$  grid attractive interactions and positive local fields. BP converged always. Errors are averages over 50 random instances for fixed  $\beta$  and  $\Theta$ . Error in partition function  $Z$  for (a) strong local fields  $\Theta = 1$  and (b) weak local fields  $\Theta = 0.1$ . Error in marginals for (c) strong local fields  $\Theta = 1$  and (d) weak local fields  $\Theta = 0.1$ .

transition for  $\Theta = 0$  and  $\beta \approx 1$  (Mooij and Kappen, 2005). Figures suggest that errors are larger as  $\Theta$  approaches zero. Notice that  $Z_0 = Z$  for  $\Theta = 0$ .

We observe that for all the instances  $Z_0$  is always an improvement over the BP approximation. Corrections are most significant for weak interactions  $\beta < 1$  and strong local fields. For strong interactions  $\beta > 1$  and weak local fields, the improvement is less significant.

It appears that the  $Z_0$  approximation performs better than TreeEP in all cases except for very strong couplings, where the two algorithms show very similar results. For weak local fields  $Z_0$  performs similar to CVM-Loop4 which is known to be a very accurate approximation for this type of model, see Yedidia et al. (2000) for instance. Selecting larger outer-clusters such as loops of

length 6 does not necessarily leads to improvements although it leads to a dramatic increase in the computational cost.

The methods which provide upper bounds on  $Z$  (PDC and TRW) report the largest average error. PDC performs slightly better than TRW, as was shown in Globerson and Jaakkola (2007) for the case of mixed interactions. We note that the worse performance of PDC for strong couplings might be attributed to implementation artifacts, since for  $\beta > 4$  the algorithm suffers from numerical precision errors. In general, both empirical upper bounds are significantly less tight than the lower bounds provided by BP and  $Z_0$ .

Finally, bottom plots show that errors in marginals behave very similar to errors in  $Z$ .

### 3.2.2 MIXED INTERACTIONS

We now analyze a more general Ising model where interactions and local fields can have mixed signs. In that case,  $Z^{BP}$  and  $Z_0$  are no longer lower bounds on  $Z$  and loop terms can be positive or negative. Figure 6 shows results for this setup.

For strong local fields (subplots a,c,e), we observe that  $Z_0$  improvements over BP results become less significant as  $\beta$  increases. It is important to note that  $Z_0$  always improves the BP result, even when the couplings are very strong ( $\beta = 10$ ) and BP fails to converge for a small percentage of instances.  $Z_0$  performs very similar to CVM-Loop4 and substantially better than TreeEP for small and intermediate  $\beta$ . As in the case of attractive interactions, the best results are attained using CVM-loop4. CVM-loop6 gives worse estimates for  $\beta > 1$ .

For the case of weak local fields (subplots b,d,f),  $Z_0$  is the best approximation in the weak coupling regime. BP fails to converge near the transition to the spin-glass phase. For  $\beta = 10$ , BP converges only in less than 25% of the instances. In the most difficult domain,  $\beta > 22$ , all methods under consideration give similar results (all comparable to BP). Moreover, it may happen that  $Z_0$  degrades the  $Z^{BP}$  approximation because loops of alternating signs have major influence in the series. This result was also reported in Gómez et al. (2007) when loop terms, instead of Pfaffian terms, were considered.

Finally, as in the case of attractive interactions, errors in marginals behave similar to errors in  $Z$ .

### 3.2.3 SCALING WITH GRAPH SIZE

We now study how the accuracy of the  $Z_0$  approximation changes as we increase the size of the grid. We generate random grids with mixed couplings for  $\sqrt{N} = \{4, \dots, 18\}$  and focus on a regime of very weak local fields  $\Theta = 0.01$  and strong couplings  $\beta = 1$ , a difficult configuration according to the previous results. We compare  $Z_0$  also with anyTLSBP, a variant of our previous algorithm for truncating the loop series. We run anyTLSBP by selecting loops shorter than a given length, and the length is increased progressively. To provide a fair comparison between both methods, we run anyTLSBP for the same amount of CPU time as the one required to obtain  $Z_0$ .

Figure 7a shows a comparison of the errors reported by the different algorithms. Since variability in the errors is larger than before, we take the median for comparison. In order of increasing accuracy we get BP, TreeEP, anyTLSBP, CVM-Loop6, CVM-Loop4 and  $Z_0$ . We note again that using larger clusters in CVM does not necessarily result in better performance.

Overall, we can see that results are roughly independent of the network size  $N$  in almost all methods that we compare. The error of anyTLSBP starts being the smallest but soon increases because the proportion of loops captured decreases very fast. For  $N > 64$ , anyTLSBP performs

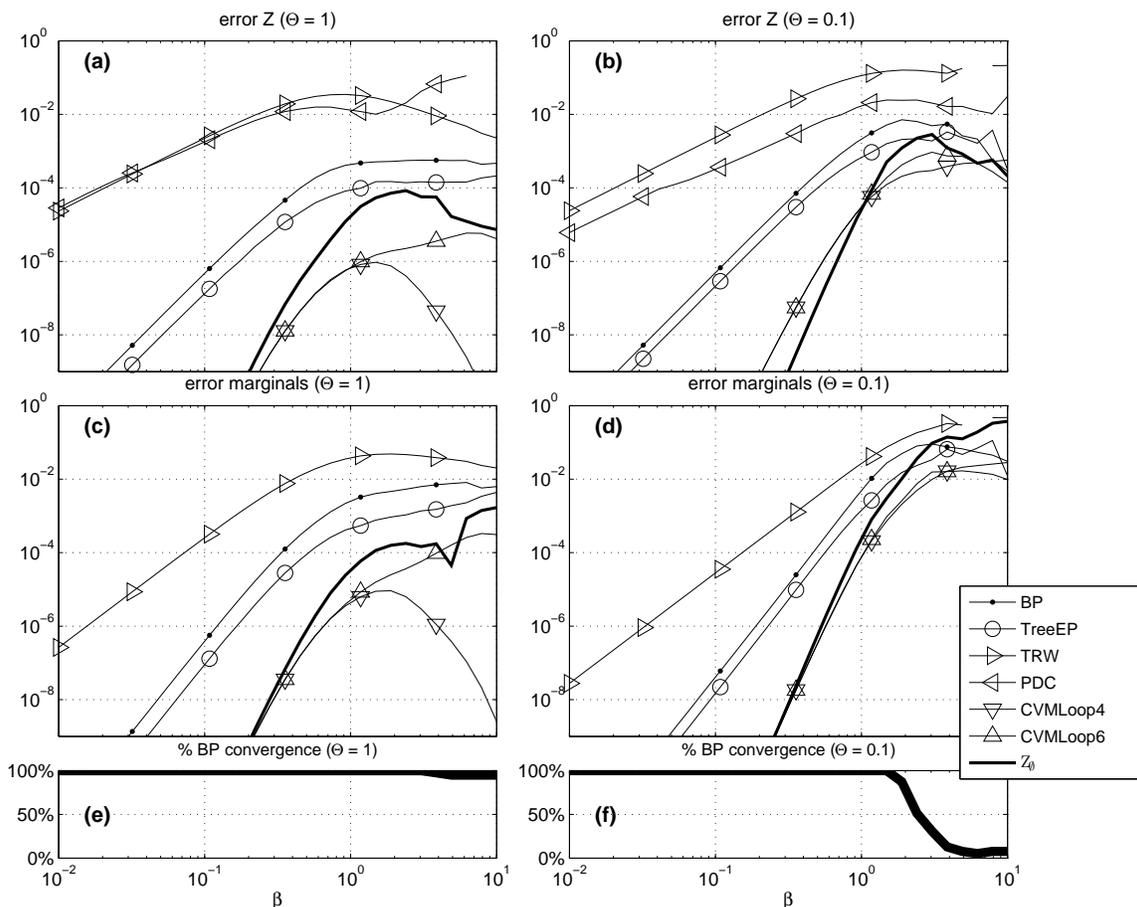


Figure 6:  $7 \times 7$  grid *mixed* interactions and positive local fields. Errors are averages over 50 random instances for fixed  $\beta$  and  $\Theta$ . Error in partition function  $Z$  for (a) strong local fields  $\Theta = 1$  and (b) weak local fields  $\Theta = 0.1$ . Error in marginals for (c) strong local fields  $\Theta = 1$  and (d) weak local fields  $\Theta = 0.1$ . Bottom panels show percentage of cases when BP converges using at least one of the methods described above for (e) strong local fields and (f) weak local fields.

worse than CVM. The  $Z_0$  correction, on the other hand, stays flat and we can conclude that it scales reasonably well. Interestingly, although  $Z_0$  and anyTLSBP use different ways to truncate the loop series, both methods show similar scaling behaviour for large graphs.

Figure 7b shows the CPU time for all the tested approaches averaged over all the cases. The CPU time of the junction tree method quickly increases with the tree-width of the graphs. For large enough  $N$ , exact solution via the junction tree method is no longer feasible because of the memory requirements. In contrast, for all the approximate inference methods, memory demands do not represent a limitation.

In order of increasing cost we have BP,  $Z_0$  with anyTLSBP, TreeEP, CVM-Loop4 and CVM-Loop6. The  $Z_0$  therefore is a very efficient correction to  $Z^{BP}$ .

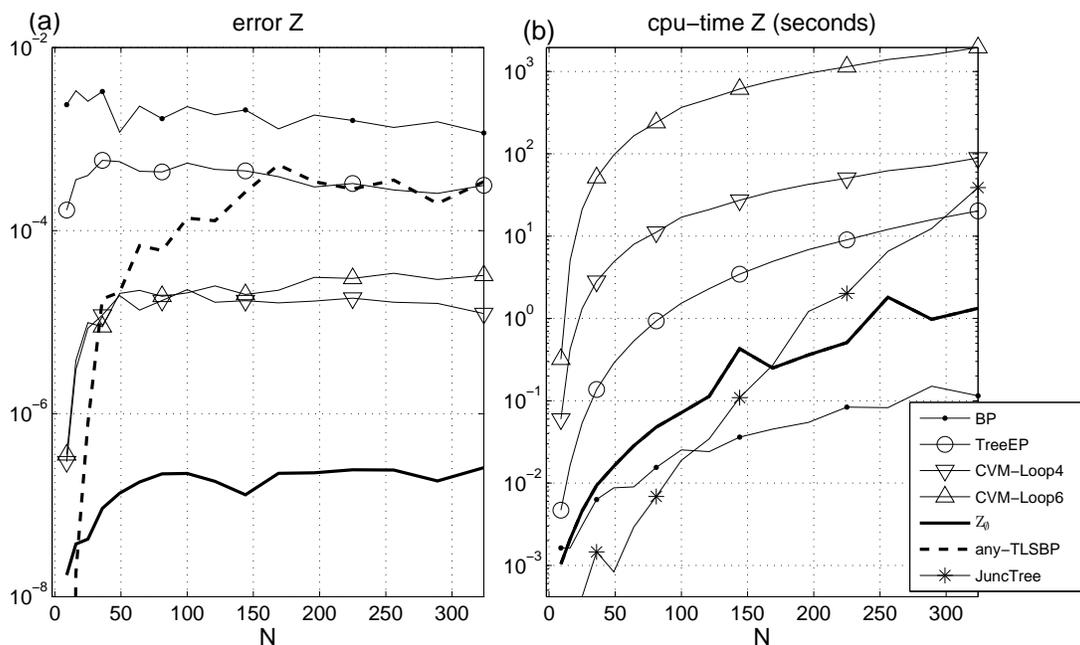


Figure 7: Results on regular grids: scaling with grid size for strong interactions  $\beta = 1$  and very weak local fields  $\Theta = 0.01$ . Error medians over 50 random instances. BP converged always. **(a)** Partition function  $Z$  error. **(b)** CPU time to compute  $Z$ .

### 3.3 Radial Grid Graphs

In the previous Subsection we analyzed the quality of the  $Z_0$  correction for graphs with a regular grid structure. Here, we carry over the analysis of the  $Z_0$  correction using planar graphs which consist of concentric polygons with a variable number of sides. Figure 8 illustrates these spider-web graphs. We generate them as factor graphs with pairwise interactions which we subsequently convert to an equivalent Forney graph using the procedure described in Appendix A.2. Again, local field potentials are parameterized using  $\Theta = 0.01$  and interactions using  $\beta = 1$ . We analyze the error in  $Z$  as a function of the degree  $d$  of the central node.

Figure 9a shows the median of errors in  $Z$  for 50 random instances. First, we see that the variability of all the methods, in particular any-TLSBP, is larger than in the regular grid scenario. CVM-Loop6 does not converge for instances with  $d > 4$  before  $10^4$  seconds and it is thus not included in the analysis. We can say that all approaches scale reasonably well, and as  $d$  grows, the errors become independent of  $d$ .

The  $Z_0$  approximation is the best method compared to the other tested approaches. The improvements of  $Z_0$  over CVM-Loop4 (the second best method) can be more than two orders of magnitude and more than three orders of magnitude compared to BP.

Computational costs are shown in 9b. Again, for larger graphs, exact solution via the junction tree is not feasible due to the large tree-width and  $Z_0$  represent the most efficient correction which improves BP of all approximate methods we compared.

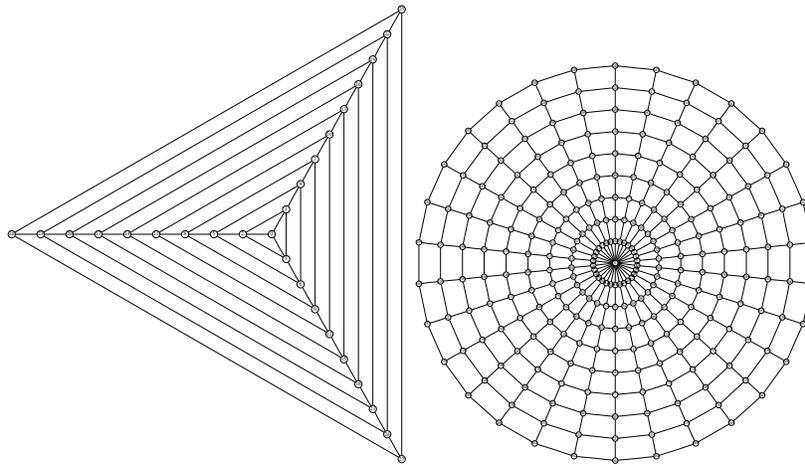


Figure 8: Two examples of planar graphs used for comparison between methods. We fix the number of concentric polygons to 9 and change the degree  $d$  of the central node within the range  $[3, \dots, 25]$ . **(left)** Graph for  $d = 3$ . **(right)** Graph for  $d = 25$ . Here nodes represent variables and edges pairwise interactions. We also add external fields which depend on the state of each node (not drawn).

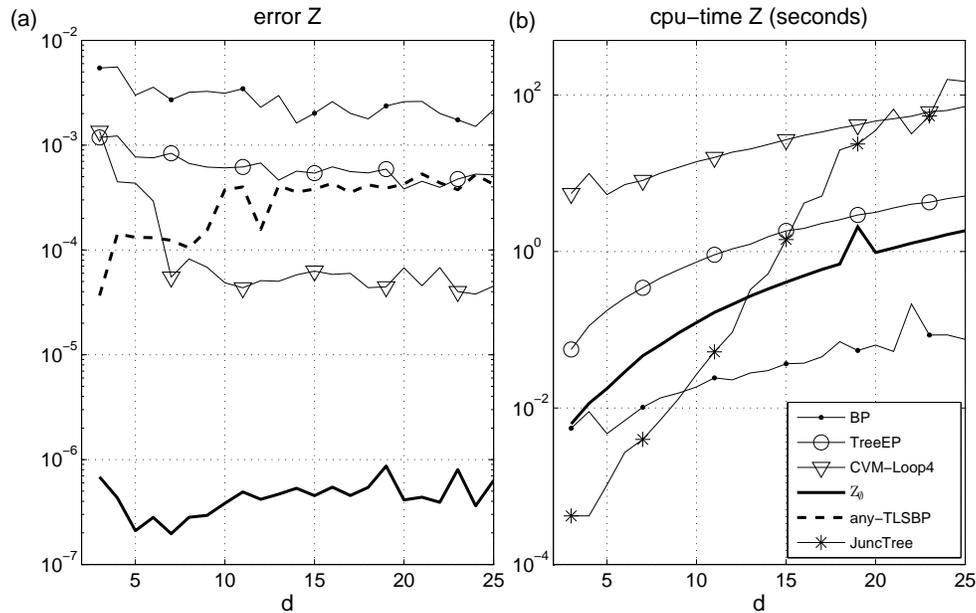


Figure 9: Results on spider-web graphs: scaling with the degree  $d$  of the central node for  $\beta = 1$  and  $\Theta = 0.01$ . BP converged always. **(a)** Median of the error in the partition function  $Z$  over 50 random instances. **(b)** CPU time required to compute  $Z$ .

#### 4. Discussion

We have presented an approximate algorithm based on the exact loop calculus framework for inference on planar graphical models defined in terms of binary variables. The proposed approach improves the estimate for the partition function provided by BP without an explicit search for loops.

The algorithm is illustrated on the example of ordered and disordered Ising model on a planar graph. Performance of the method is analyzed in terms of its dependence on the system size. The complexity of the partition function computation is exponential in the general case, unless the local fields are zero, when it becomes polynomial. We tested our algorithm on regular grids and planar graphs with different structures. Our experiments on regular grids show that significant improvements over BP are always obtained if single variable potentials (local magnetic fields) are sufficiently large. The quality of this correction degrades with decrease in the amplitude of external field, to become exact at zero external fields. This suggests that the difficulty of the inference task changes from very easy, with no local fields, to very hard, with small local fields, and then decays again as external fields become larger.

The  $Z_0$  correction turns out to be competitive with other state of the art methods for approximate inference of the partition function. First of all, we showed that  $Z_0$  is much more accurate than upper bounds based methods such as TRW or PDC, illustrating that such methods come at the cost of accuracy. We have also shown that for the case of grids with attractive symmetric interactions and positive local fields, the lower bound provided by  $Z_0$  is the most accurate.

We also found that  $Z_0$  performs much better than treeEP for weak and intermediate interactions and similar for strong interactions. Comparing with CVM, we have found that  $Z_0$  presented better results for very small local fields. Using larger outer clusters in CVM does not necessarily lead to better approximations.

Finally, we have presented a comparison of  $Z_0$  with TLSBP, which is another algorithm based on the loop series expansion for BP that uses the loop length as truncation parameter. On one hand, the calculation of  $Z_0$  involves a *re-summation* of many loop terms which in the case of TLSBP are summed individually. This consideration favors the  $Z_0$  approach. On the other hand,  $Z_0$  is restricted to the class of 2-regular loops whereas TLSBP also accounts for terms corresponding to more complex loop structures in which nodes can have degree larger than two. Overall, for planar graphs, we have shown evidence that the  $Z_0$  approach is better than TLSBP when the size of the graphs is not very small. We emphasize, however, that TLSBP can be applied to non-planar binary graphical models too.

Although planarity is a severe restriction, we emphasize that planar graphs appear in many contexts such as computer vision and image processing, magnetic and optical recording, or network routing and logistics. We have focused on inference problems defined on planar graphs with symmetric pairwise interactions and, to make the problems difficult, we have introduced local field potentials. Notice however, that the algorithm can also be used to solve models with more complex interactions, that is, more than pairwise typical from the Ising model (see Chertkov et al., 2008, for a discussion of possible generalizations). This makes our approach more applicable than other approaches, namely, (Globerson and Jaakkola, 2007; Schraudolph and Kamenetsky, 2008), designed specifically for the pairwise interaction case.

Summarizing, among the compared methods and models, the introduced approach based on  $Z_0$  represents the best estimate to the partition function as long as the given graph does not deviate too much from perfect planarity, that is, we are in the small local field regime. It also represents

an efficient correction to the BP estimate of the partition function, being much more efficient than CVM, the second best method we have analyzed. Finally, we have shown that estimates of single-variable marginals calculated using  $Z_0$  are comparable to state of the art inference methods.

### Acknowledgments

We acknowledge J. M. Mooij, A. Windsor and A. Globerson for providing their software, V. Y. Chernyak, J. K. Johnson and N. Schraudolph for interesting discussions and anonymous reviewers for valuable suggestions. This research is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024. The work at LANL was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at LANL under Contract No. DE-AC52-06NA25396.

### Appendix A. Bipartite Factor Graph Representation

The Forney graph representation is convenient because each loop decomposes naturally in terms associated with nodes which have the same analytical form of Equation (4). However, probabilistic models are more frequently represented as bipartite factor graphs. In this Appendix we first show how the presented approach differs when it is directly applied to a bipartite factor graph and second, how to convert a bipartite factor graph to a Forney graph.

We consider binary variables  $i$  which take values  $x_i \in \{\pm 1\}$  and factor functions  $\psi_a(\mathbf{x}_a)$  defined over subsets  $a$  of variables which take values  $\mathbf{x}_a := \{x_i | i \in a\}$ . On a bipartite factor graph  $\mathcal{G}_{fg} := (\mathcal{V}_{fg}, \mathcal{E}_{fg})$ , the set  $\mathcal{V}_{fg}$  consists of variable nodes  $i \in I$  and factor nodes  $a \in \mathcal{A}$ , with an edge between  $i$  and  $a$  iff  $i \in a$ , that is  $i$  appears in the factor function  $\psi_a$ .

The joint probability distribution of vector  $\mathbf{x} := \{x_i | i \in \mathcal{V}\}$  is specified as:

$$p(\mathbf{x}) = \frac{1}{Z(\psi)} \prod_{a \in \mathcal{A}} \psi_a(\mathbf{x}_a), \quad Z(\psi) = \sum_{\mathbf{x}} \prod_{a \in \mathcal{A}} \psi_a(\mathbf{x}_a),$$

where we have stressed the dependency of  $Z$  on the potential functions  $\psi$ .

#### A.1 Loop Calculus for Standard Factor-Graph Model

Following Sudderth et al. (2007), one can use the reparameterized model in terms of the factor pseudo-marginals  $\tau_a(\mathbf{x}_a)$  and the variable pseudo-marginals  $\tau_i(x_i)$  associated with the minimum of the Bethe free energy:

$$p(\mathbf{x}) = \frac{1}{Z(\tau)} \prod_{i \in I} \tau_i(x_i) \prod_{a \in \mathcal{A}} \frac{\tau_a(\mathbf{x}_a)}{\prod_{j \in a} \tau_j(x_j)},$$

and express the relation between the exact  $Z(\psi)$  and the Bethe partition functions  $Z^{BP}(\psi; \tau)$  as  $Z(\psi)/Z(\tau) = Z^{BP}(\psi; \tau)$ . The loop series correction becomes:

$$Z(\tau) = 1 + \sum_{C \in \mathcal{C}} s_C, \quad s_C = \prod_{a \in C} \mu_a(C) \prod_{i \in C} \mu_i(C),$$

where each term  $s_C$  is associated with a loop and can be specified as a product between terms  $\mu_a(C)$  and  $\mu_i(C)$  corresponding to factor nodes  $a$  and variable nodes  $i$  in  $C$  respectively:

$$\mu_a(C) = \frac{\mathbb{E}_{\tau_a} \left[ \prod_{i \in a, i \in C} (x_i - m_i) \right]}{\prod_{i \in a, i \in C} \text{Var}_{\tau_i}(x_i)} \quad \mu_i(C) = \mathbb{E}_{\tau_i} \left[ (x_i - m_i)^{d_i(C)} \right],$$

where  $d_i(C)$  is the degree of variable node  $i$  in the loop  $C$  and  $m_i$  is again  $\mathbb{E}_{\tau_i} [x_i]$ . For the  $x_i = \{\pm 1\}$  alphabet, we have  $m_i = \tau_i(+1) - \tau_i(-1)$  and the formulas become:

$$\begin{aligned} \mu_a(C) &= \frac{\sum_{\mathbf{x}_a} \tau_a(\mathbf{x}_a) \prod_{i \in a, i \in C} (x_i - m_i)}{\prod_{i \in a, i \in C} \tau_i(+1)(1 - m_i)^2 + \tau_i(-1)(-1 - m_i)^2} \\ &= \frac{\sum_{\mathbf{x}_a} \tau_a(\mathbf{x}_a) \prod_{i \in a, i \in C} (x_i - m_i)}{\prod_{i \in a, i \in C} \frac{1+m_i}{2}(1 - m_i)^2 + \frac{1-m_i}{2}(1 + m_i)^2} \\ &= \frac{\sum_{\mathbf{x}_a} \tau_a(\mathbf{x}_a) \prod_{i \in a, i \in C} (x_i - m_i)}{\prod_{i \in a, i \in C} (1 - m_i^2)}, \end{aligned} \quad (9)$$

$$\begin{aligned} \mu_i(C) &= \tau_i(+1)(1 - m_i)^{d_i(C)} + \tau_i(-1)(-1 - m_i)^{d_i(C)} \\ &= \frac{1 + m_i}{2}(1 - m_i)(1 - m_i)^{d_i(C)-1} + \frac{1 - m_i}{2}(1 + m_i)(-1)^{d_i(C)}(1 + m_i)^{d_i(C)-1} \\ &= \frac{1 - m_i^2}{2} \left( (1 - m_i)^{d_i(C)-1} + (-1)^{d_i(C)}(1 + m_i)^{d_i(C)-1} \right). \end{aligned} \quad (10)$$

One can recover the original formulation of Chertkov and Chernyak (2006a, Pag.5) reallocating the denominator of (9) to (10) and simplifying.

Now consider Figure 2 where the construction of the extended graph  $\mathcal{G}_{ext}$  is described. Clearly, the one-to-one correspondence from 2-regular loops in  $\mathcal{G}$  to perfect matchings in  $\mathcal{G}_{ext}$  is independent of whether  $\mathcal{G}$  is expressed as a bipartite factor graph or a Forney graph. For a bipartite factor graph, the terms of Equation (5) are replaced with terms given by Equation (10) if the expanded node is a variable or by Equation (9) if the expanded node is a factor. As previously stated, the Forney style allows to express in the same form the weights of the extended graph  $\mathcal{G}_{ext}$ .

## A.2 Converting a Bipartite Factor Graph to a Forney-Style Factor Graph.

We show here how to convert a bipartite factor graph defined in terms of binary variables to a more general Forney graph representation, for which the presented algorithm can be directly applied to.

We label variable and factor nodes using numbers and capital letters respectively. Thus  $i \in I, i := \{1, 2, \dots\}$  represents a variable and  $a = \{A, B, \dots\}, a \in \mathcal{A}$  a factor. Given  $\mathcal{G}_{fg} := (\mathcal{V}_{fg}, \mathcal{E}_{fg})$ , a direct way to obtain an equivalent Forney graph  $\mathcal{G} := (\mathcal{V}, \mathcal{E})$  is: first, to create a node  $\delta_i \in \mathcal{V}$  for each variable node  $i \in \mathcal{V}_{fg}$ , and second, to associate a new binary variable  $\delta_{ia}$  with values  $\sigma_{\delta_{ia}} = \{\pm 1\}$  to edges  $(\delta_i, a) \in \mathcal{E}$ . Nodes  $\delta_i \in \mathcal{V}$  are *equivalent factor nodes* denoting the characteristic function:  $f_{\delta_i}(\sigma_{\delta_i}) = 1$  if  $\sigma_{\delta_{ia}} = \sigma_{\delta_{ib}}, \forall a, b \in \tilde{\delta}_i$  and zero otherwise. Finally, factor nodes  $c \in \mathcal{V}_{fg}$  with associated functions  $\psi_c(\mathbf{x}_c)$  correspond to the same factor nodes  $c$  in  $\mathcal{V}$  with same associated functions  $f_c(\sigma_c)$  and defined in terms of the new variables  $\delta_{ic}, \forall i \in \bar{c}$ . Figure 10 shows an example of this transformation. Notice that, although we impose a direction in the edge labels, they remain undirected:  $(\delta_i, a) = (a, \delta_i), \forall \delta_i, a \in \mathcal{V}$ . For variables  $i \in \mathcal{V}_{fg}$  which only appear in two factors, such as variable 3 in Figure 10a, the corresponding node  $\delta_3$  is redundant and can be removed. The joint

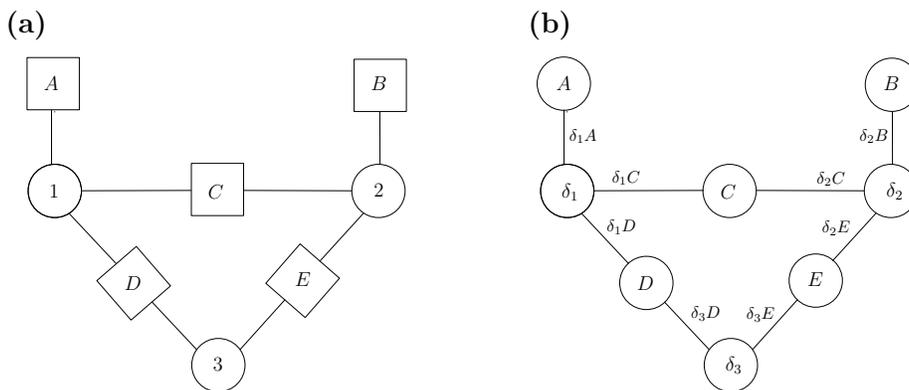


Figure 10: **(a)** Bipartite factor graph  $\mathcal{G}_{fg}$  where squares represent factors and circles represent variables. **(b)** Equivalent Forney-style factor graph  $\mathcal{G}$  where factors reside in the nodes and variables in the edges.

distribution of  $\mathcal{G}_{fg}$  is related to the joint distribution of  $\mathcal{G}$  by:

$$\begin{aligned} & \frac{1}{Z} \Psi_A(x_1) \Psi_B(x_2) \Psi_C(x_1, x_2) \Psi_D(x_1, x_3) \Psi_E(x_2, x_3) \\ & \equiv \frac{1}{Z} f_A(\sigma_{\delta_{1A}}) f_B(\sigma_{\delta_{2B}}) f_C(\sigma_{\delta_{1C}}, \sigma_{\delta_{2C}}) f_D(\sigma_{\delta_{1D}}, \sigma_{\delta_{3D}}) f_E(\sigma_{\delta_{2E}}, \sigma_{\delta_{3E}}) \\ & \quad f_{\delta_1}(\sigma_{\delta_{1A}}, \sigma_{\delta_{1C}}, \sigma_{\delta_{1D}}) f_{\delta_2}(\sigma_{\delta_{2B}}, \sigma_{\delta_{2C}}, \sigma_{\delta_{2E}}) f_{\delta_3}(\sigma_{\delta_{3D}}, \sigma_{\delta_{3E}}). \end{aligned}$$

Once  $\mathcal{G}$  has been generated following the previous procedure it may be the case that the nodes  $\delta_i \in \mathcal{V}$  have degree three or larger. This happens if a variable  $i$  appears in more than three factors on  $\mathcal{G}_{fg}$ . It is easy to convert  $\mathcal{G}$  to a graph were all  $\delta_i$  nodes have maximum degree three by introducing new auxiliary variables  $\delta_{i_1}, \delta_{i_2}, \dots$  and new equivalent nodes. For instance, if variable  $i \in \mathcal{V}_{fg}$  appears in four factors  $A, B, C, D$ :

$$f_{\delta_i}(\sigma_{\delta_{iA}}, \sigma_{\delta_{iB}}, \sigma_{\delta_{iC}}, \sigma_{\delta_{iD}}) \equiv f_{\delta_{i_1}}(\sigma_{\delta_{iA}}, \sigma_{\delta_{iB}}, \sigma_{\delta_{i_1}}) f_{\delta_{i_2}}(\sigma_{\delta_{i_1}}, \sigma_{\delta_{iC}}, \sigma_{\delta_{iD}}).$$

Notice that although the models are equivalent, the number of loops in  $\mathcal{G}$  may be larger than in  $\mathcal{G}_{fg}$ . In the case that a factor in  $\mathcal{G}_{fg}$  involves more than three variables, as sketched in Chertkov et al. (2008), one could split the node of degree  $N$  into auxiliary nodes of degree  $N - 1$  and compute  $Z_0$  on the transformed model. Alternatively, one can reduce the number of variables that enter a factor conditioning over the variables.

## References

- F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, 1982. URL <http://stacks.iop.org/0305-4470/15/3241>.
- M. Chertkov and V. Y. Chernyak. Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009, 2006a.

- M. Chertkov and V. Y. Chernyak. Loop calculus helps to improve belief propagation and linear programming decodings of LDPC codes. In *invited talk at 44th Allerton Conference*, September 2006b.
- M. Chertkov, V. Y. Chernyak, and R. Teodorescu. Belief propagation and loop series on planar graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(05):P05003 (19pp), 2008. URL <http://stacks.iop.org/1742-5468/2008/P05003>.
- G. Elidan, I. McGraw, and D. Koller. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Boston, Massachusetts, 2006. AUAI Press.
- M. E. Fisher. On the dimer solution of the planar Ising model. *Journal of Mathematical Physics*, 7(10):1776–1781, 1966.
- M. E. Fisher. Statistical mechanics of dimers on a plane lattice. *Physical Review*, 124:1664–1672, 1961.
- B. J. Frey and D. J. C. MacKay. A revolution: belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems 10*, pages 479–486. MIT Press, Cambridge, MA, 1998.
- A. Globerson and T. S. Jaakkola. Approximate inference using planar graph decomposition. In *Advances in Neural Information Processing Systems 19*, pages 473–480. MIT Press, Cambridge, MA, 2007.
- V. Gómez, J. M. Mooij, and H. J. Kappen. Truncating the loop series expansion for belief propagation. *Journal of Machine Learning Research*, 8:1987–2016, 2007. ISSN 1533-7928.
- T. Heskes, K. Albers, and H. J. Kappen. Approximate inference and constrained optimization. In *Proceedings of the 19th Annual conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 313–320, San Francisco, CA, 2003. Morgan Kaufmann Publishers.
- M. Jerrum. *Counting, Sampling and Integrating : Algorithms and Complexity*. Lectures in Mathematics - ETH Zrich. Birkhuser, Basel, 2003.
- G.D. Forney Jr. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*, 47(2):520–548, Feb 2001. ISSN 0018-9448. doi: 10.1109/18.910573.
- M. Karpinski and W. Rytter. Fast parallel algorithms for graph matching problems. In *Oxford Lecture Series in Mathematics & Its Applications Number 9*, pages 164–170. Oxford University Press, USA, 1998.
- P. W. Kasteleyn. The statistics of dimers on a lattice : I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27:1209–1225, 1961.
- P. W. Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4(2):287–293, 1963. URL <http://link.aip.org/link/?JMP/4/287/1>.
- S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical society. Series B-Methodological*, 50(2):154–227, 1988.

- H.-A. Loeliger. An introduction to factor graphs. *Signal Processing Magazine, IEEE*, 21(1):28–41, 2004. ISSN 1053-5888. doi: 10.1109/MSP.2004.1267047.
- T. Minka and Y. Qi. Tree-structured approximations by expectation propagation. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
- J. M. Mooij. libDAI: A free/open source C++ library for discrete approximate inference methods, 2008. <http://mloss.org/software/view/77/>.
- J. M. Mooij and H. J. Kappen. On the properties of the Bethe approximation and loopy belief propagation on binary networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2005 (11):P11012, 2005.
- K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy Belief Propagation for approximate inference: An empirical study. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 467–475, San Francisco, CA, 1999. Morgan Kaufmann Publishers.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Francisco, CA, 1988. ISBN 1558604790.
- N. Schraudolph and D. Kamenetsky. Efficient exact inference in planar Ising models. In *Advances in Neural Information Processing Systems 22*. MIT Press, Cambridge, MA, 2008.
- E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Loop series and Bethe variational bounds in attractive graphical models. In *Advances in Neural Information Processing Systems 20*, pages 1425–1432. MIT Press, Cambridge, MA, 2007.
- H. N. V. Temperley and M. E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.
- M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51(7):2313–2335, 2005.
- Y. Watanabe and K. Fukumizu. Loop series expansion with propagation diagrams. *Journal of Physics A: Mathematical and Theoretical*, 42(4):045001 (18pp), 2009.
- J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems 13*, pages 689–695, 2000.
- J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.