

Loop Calculus and Bootstrap-Belief Propagation for Perfect Matchings on Arbitrary Graphs

Michael Chertkov ^a, Andrew Gelfand ^{a,b}, Jinwoo Shin ^c

Theoretical Division & Center for Nonlinear Studies, Los Alamos National Laboratory,
Los Alamos, NM 87545, USA

Department of Computer Science, University of California, Irvine, Irvine, CA 92697-3435, USA

Mathematical Sciences Department, IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA

E-mail: chertkov@lanl.gov

Abstract. This manuscript discusses computation of the Partition Function (PF) and the Minimum Weight Perfect Matching (MWPM) on arbitrary, non-bipartite graphs. We present two novel problem formulations - one for computing the PF of a Perfect Matching (PM) and one for finding MWPMs - that build upon the inter-related Bethe Free Energy, Belief Propagation (BP), Loop Calculus (LC), Integer Linear Programming (ILP) and Linear Programming (LP) frameworks. First, we describe an extension of the LC framework to the PM problem. The resulting formulas, coined (fractional) Bootstrap-BP, express the PF of the original model via the BFE of an alternative PM problem. We then study the zero-temperature version of this Bootstrap-BP formula for approximately solving the MWPM problem. We do so by leveraging the Bootstrap-BP formula to construct a sequence of MWPM problems, where each new problem in the sequence is formed by contracting odd-sized cycles (or blossoms) from the previous problem. This Bootstrap-and-Contract procedure converges reliably and generates an empirically tight upper bound for the MWPM. We conclude by discussing the relationship between our iterative procedure and the famous Blossom Algorithm of Edmonds '65 and demonstrate the performance of the Bootstrap-and-Contract approach on a variety of weighted PM problems.

1. Introduction

Belief Propagation (BP) is an iterative message passing algorithm with physics roots in H. Bethe [1] and R. Peierls' [2] work on “melting of lattices”, information theory roots in decoding of Gallager's Low Density Parity Check (LDPC) codes [3], and artificial intelligence roots dating to J. Pearl [4] (who actually coined the term). For decades the three fields developed BP and the homonymous Bethe-Peierls approximation independently. Only recently, under the Bethe Free Energy (BFE) framework of Yedidia, Freeman and Weiss in [5, 6], have the parallel developments been united.

BP is a naturally distributed and easy to implement algorithm for iteratively solving the system of BP equations - a set of equations that also characterize the fixed points of the BFE function [5, 6]. The so-called BP/BFE approach can be used to compute the Partition Function (PF) or find the Maximum Likelihood (ML) assignment of a Graphical Model (GM), where a GM defines a joint probability over a set of random variables via local probability measures that factorize according to some underlying graph structure. BP/BFE provides an exact answer to the PF and ML tasks if the underlying graph is a tree. BP/BFE is a heuristic on graphs with loops, but it typically provides an accurate approximation.

Two recent developments have altered the paradigm that BP/BFE is only “good” for tree or tree-like GMs. First, it was shown that BP can exactly recover the ML assignment for certain binary GMs with

dense, loopy graph structures. For example, BP can be used to find a PM on a bipartite graph [7, 8, 9]. Interestingly, all models of this special type can be formulated as Linear Programming (LP) problems that have an integral optimal solution, though this condition is not sufficient in general [9, 10, 11].

The Loop Calculus (LC) was developed at the same time and provides a way to express the PF a general binary GM as a finite (but still exponential in the size of the graph) series [12, 13]. Each term in the series is related to a generalized loop of the underlying graph (which is a subgraph where all vertices have degree two or higher) and is expressed explicitly using the solution to the BP equations. In some special cases, e.g. for PMs over bipartite graphs where the partition function is a permanent [14], the multiplicative mismatch between the PF of the original problem and the respective BFE/BP expression is itself the PF of a new GM defined on the same graph. In such cases, the LC can be used to improve upon the BP estimate for the PF (when the BP estimate is not tight) by re-summing a number of important LC terms [15, 16]. As suggested in [17, 18, 19, 20], the BP approximation to the ML problem can be improved by analyzing higher order terms of the LC (sometimes re-summing as in [20]) and using this information to modify the original GM. The hope is that when BP is run on the new GM it will be exact or, at least, improve upon its initial approximation.

In this manuscript we continue to work on improving BP, focusing on MWPM problems in general (non-bipartite) graphs. This choice is not arbitrary. MWPM is a special problem in Matching Theory which can be solved in polynomial time by Edmonds’ famous Blossom Algorithm (BA) [21] (see [22] for details of BlossomV, the state-of-the-art implementation of the BA). The most recent development in this classical sub-field of Computer Science is due to Chandrasekaran, Végh and Vempala [23] who proposed a polynomial time “cutting-plane” procedure to finding a MWPM. In contrast to Edmonds’ primal-dual algorithm, their approach carefully constructs a sequence of linear inequalities (or cut constraints) that tighten the LP relaxation, while ensuring that the added “cuts” maintain half-integrality of the LP.

While Edmonds’ BA is efficient, it is not a distributed algorithm. Several recent results motivate the use of distributed, message passing implementations of BP to solve the MWPM problem. First, standard max-product BP was shown to be convergent and correct when the optimal solution to the LP relaxation of the weighted matching problem is integral. This was shown for PMs on bipartite graphs in [7, 8, 9] and extended to general weighted matchings in [11]. And in [20], the MWPM was shown to be an example of a problem where the LP optima and iterative BP solution are equivalent (when the latter converges).

Unfortunately, the initial LP relaxation is not tight in general. The work in [20] considered situations when the optimal solution is non-integral and proposed running BP on a modified version of the weighted matching GM. In particular, higher order factors enforcing Edmonds’ blossom (cut) constraints were added to the original GM and a finite temperature version of BP (with proper annealing and damping) was used to improve convergence to the LP optima. A provably accurate alternative to [20], discussed in [24], would involve: (a) adaptively selecting the blossoms/constraints used to modify the GM; (b) maintaining half-integrality of the underlying LP at every iteration; and (c) proving convergence and correctness using the computational tree approach of [9, 11].

This manuscript develops a new theoretical and algorithmic approach towards constructing efficient and distributed algorithms for resolving the MWPM problem. This new approach, which is complementary to both [20] and [24], is based on extending the LC paradigm of [12, 13, 14, 25] to PMs over an arbitrary graph at a finite temperature, T , and careful analysis of the ML, $T \rightarrow 0$, limit. Analysis of the $T \rightarrow 0$ limit led to the development of an algorithm for (approximately) solving the MWPM problem - the BP-Bootstrap-Contract procedure - which we evaluated empirically.

The material in the manuscript is organized as follows. We provide some background on Graphical Models (GMs) and Perfect Matching (PM) problems in Section 2; the BP/BFE framework is briefly reviewed in Section 3 and in Appendix A. The main results involving the LC are covered in Section 4. Finally, the Bootstrap BP and fractional Bootstrap BP ([25]) formulas are discussed in Section 5 and Appendix B. The ML, $T \rightarrow 0$, versions of the Bootstrap formulas are discussed and illustrated on a small example in Section 6. The Bootstrap-and-Contract algorithm is presented in Section 7 and evaluated empirically in Section 8. Finally, we summarize and discuss open questions in Section 9.

2. Graphical Model of Perfect Matching

Consider an arbitrary undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with associated edge weights $w = (\exp(-\varepsilon_{ij}/T) | (i, j) \in \mathcal{E})$, where T is a positive temperature parameter term and ε_{ij} is the energy of a Perfect Matching (PM), or dimer¹, associated with the edge (i, j) . One problem of interest is to evaluate the PF of the PM over the tuple $(\mathcal{V}, \mathcal{E}, w)$:

$$Z(w) = \sum_{\sigma \in \Sigma} \exp \left(-\frac{1}{T} \sum_{(i,j) \in \mathcal{E}} \varepsilon_{ij} \sigma_{ij} \right) = \sum_{\sigma \in \Sigma} (w_{ij})^{\sigma_{ij}} = \sum_{\sigma \in \Sigma} W(\sigma), \quad (1)$$

where $\Sigma = \left\{ \sigma | \forall (i, j) : \sigma_{ij} \in \{0, 1\}; \forall i : \sum_{j: (i,j) \in \mathcal{E}} \sigma_{ij} = 1 \right\}$ is the set of PMs in \mathcal{G} , where a PM is a subset of the edges in \mathcal{E} such that an edge is adjacent to each vertex in \mathcal{V} . Note that $\mathcal{P}(\sigma) = W(\sigma)/Z(w)$ is thus the probability of the PM σ .

The total weight, E , of the Maximum Weight Perfect Matching (MWPM) arises by taking the zero temperature limit, $T \rightarrow 0$, of the PF

$$E = - \lim_{T \rightarrow 0} \frac{\log Z}{T} = \min_{\sigma \in \Sigma} \sum_{(i,j) \in \mathcal{E}} \varepsilon_{ij} \sigma_{ij}. \quad (2)$$

The rhs of Eq. (2) is an Integer Linear Programming (ILP) problem. The LP relaxation of Eq. (2) is

$$E_0 = \min_{x \in X} \sum_{(i,j) \in \mathcal{E}} \varepsilon_{ij} x_{ij}, \quad (3)$$

where $X = \left\{ x_{ij} | \forall (i, j) : x_{ij} \in [0, 1]; \forall i : \sum_{j: (i,j) \in \mathcal{E}} x_{ij} = 1 \right\}$ is the so-called PM polytope. Since Eq. (3) constitutes a relaxation of Eq. (2), $E_0 \leq E$. As shown in [26] the optimal x in Eq. (3) is half integer.

If the graph is bipartite then (a) computing the PF in (1) is equivalent to computing the permanent of the matrix of w -weights; (b) the PM polytope, X is equal to the convex hull of Σ ; (c) $E_0 = E$, i.e. the LP-relaxation is gapless; and (d) the optimal x in Eq. (3) is integral (assuming the formulation in (2) is not degenerate).

3. Bethe Free Energy, Belief Propagation and LP-BP

For a generic Graphical Model (GM) over \mathcal{G} assigning (un-normalized) weight w to state σ , one defines the exact variational² function

$$\mathcal{F}\{b\} \equiv T \sum_{\sigma} b(\sigma) \ln \frac{b(\sigma)}{W(\sigma)}. \quad (4)$$

The belief, $b(\sigma)$ is understood as a proxy to the probability $\mathcal{P}(\sigma)$ because under the normalization condition $\sum_{\sigma \in \Sigma} b(\sigma) = 1$, the Gibbs function is convex and it achieves its only minimum at $b(\sigma) = \mathcal{P}(\sigma)$ and $\mathcal{F}\{\mathcal{P}\} = -T \ln Z$.

Belief Propagation (BP) is an iterative method for computing beliefs that is exact when the underlying GM is a tree. As shown in [6], the BP fixed point equations can be derived as a relaxation of the constrained Gibbs function (4). We briefly review the concepts of [6] with application to the PM problem. The material generalizes the description of [14], which was limited to bipartite graphs (permanents).

¹ The two terms, used respectively in Statistical Physics and Computer Science, are equivalent.

² It is referred to as the Gibbs and Kullback-Leibler function in statistical physics and statistics, respectively.

The BP approximate belief of state $\sigma \in \Sigma$ is

$$b(\sigma) \approx b_{BP}(\sigma) = \frac{\prod_i b_i(\sigma_i)}{\prod_{(i,j) \in \mathcal{E}} b_{ij}(\sigma_{ij})}, \quad (5)$$

where $b_i(\sigma_i)$ ($b_{ij}(\sigma_{ij})$) are vertex (edge) beliefs related to each other by

$$\forall (i,j) \in \mathcal{E} : \quad b_{ij}(\sigma_{ij}) = \sum_{\sigma_i \setminus \sigma_{ij}} b_i(\sigma_i) = \sum_{\sigma_j \setminus \sigma_{ij}} b_j(\sigma_j), \quad (6)$$

and where $\sigma_i = (\sigma_{ij}|j : (i,j) \in \mathcal{E})$ is a vector of edge beliefs. The beliefs should also satisfy the normalization conditions:

$$\forall (i,j) \in \mathcal{E} : \quad b_{ij}(1) + b_{ij}(0) = 1. \quad (7)$$

Substituting Eq. (5) into Eq. (4) and utilizing Eqs. (6,7) one arrives at the following Bethe Free Energy function (BFE)

$$\mathcal{F}_{BP}(\beta) \equiv E - TS, \quad E \equiv \sum_{(i,j)} b_{ij}(1) \varepsilon_{ij}, \quad (8)$$

$$S \equiv \sum_{(i,j) \in \mathcal{E}} \sum_{\sigma_{ij}} b_{ij}(\sigma_{ij}) \ln b_{ij}(\sigma_{ij}) - \sum_{i \in \mathcal{V}} \sum_{\sigma_i} b_i(\sigma_i). \quad (9)$$

Eqs. (6,7,8,9) can be greatly simplified. One can express Eqs. (6,7) solely in terms of $\beta_{ij} \equiv b_{ij}(1)$ satisfying the following Doubly Stochastic (DS) constraints

$$\forall (i,j) \in \mathcal{E} : 0 \leq \beta_{ij} \leq 1; \quad \forall i : \sum_j \beta_{ij} = 1. \quad (10)$$

We will use $\beta \in \text{DS}$ to indicate that the matrix β is DS, i.e. lies in the DS polytope defined by Eq. (10). The entropy Eq. (9) becomes

$$S(\beta) = \sum_{(i,j) \in \mathcal{E}} ((1 - \beta_{ij}) \ln(1 - \beta_{ij}) - \beta_{ij} \ln \beta_{ij}). \quad (11)$$

Therefore, the BFE/BP approximation to the partition function of the PM model over \mathcal{G} becomes

$$\begin{aligned} -\log Z_{BP} &= \frac{1}{T} \min_{\beta \in \text{DS}} \mathcal{F}_{BP}(\beta) \\ &= \min_{\beta \in \text{DS}} \sum_{(i,j) \in \mathcal{E}} \left(\beta_{ij} \left(\frac{\varepsilon_{ij}}{T} + \log \beta_{ij} \right) - (1 - \beta_{ij}) \log(1 - \beta_{ij}) \right). \end{aligned} \quad (12)$$

The MWPM can also be estimated within the BFE/BP approach as:

$$E_{BP} = \min_{\beta \in \text{DS}} \sum_{(i,j) \in \mathcal{E}} \beta_{ij} \varepsilon_{ij}. \quad (13)$$

Notice that Eq. (13) is an LP, that will be called BPLP because of its relation to the BP concept. As shown in [9], $E_{BP} = E_0$, i.e. $\text{LP} \equiv \text{BPLP}$, where LP is defined by Eq. (3). One also finds that the optimal β in Eq. (13) is half-integral.

For a most general GM the BFE function is bounded from below but non-convex. However, as shown in [27] for the PM over bi-partite graph, $\mathcal{F}_{BP}(\beta)$ is a convex function over $\beta \in \text{DS}$. As argued in Appendix A the BFE convexity proof of [27] extends straightforwardly to PM GM over arbitrary \mathcal{G} .

To analyze the minima of the constrained BFE in Eq. (12) we incorporate Lagrange multipliers μ_i enforcing the constraints in Eqs. (10). Looking for a stationary point of the respective Lagrange function over the β variables, one arrives at the following set of quadratic equations for edge variables, β_{ij}

$$\forall (i, j) \in E : \quad \beta_{ij}(1 - \beta_{ij}) = \exp\left(\frac{\mu_i + \mu_j - \varepsilon_{ij}}{T}\right). \quad (14)$$

Eqs. (14) combined with the conditions (10) constitute the so-called Belief Propagation (BP) equations.

To find a solution of BP Eqs. (10,14) one relies on an iterative procedure. For a description of a set of iterative BP algorithms convergent to a minimum of the BFE for the perfect matching problem we refer the interested reader to [28, 29, 30].

4. Loop Series

As shown in [12, 13], the exact partition function of a generic GM can be expressed in terms of a Loop Series (LS), where each term is computed explicitly using beliefs found by solving the BP equations.

Adapting this general result to the PM problem one derives

$$Z = Z_{\text{BP}} \cdot z, \quad z \equiv 1 + \sum_C r_C, \quad r_C = \left(\prod_{i \in C} \psi_{i;C} \right), \quad (15)$$

$$\psi_{i;C} \equiv \frac{\sum_{\sigma_i} b_i(\sigma_i) \prod_{j:(i,j) \in C} (m_{ij} - 2\sigma_{ij} + 1)}{\prod_j^{(i,j) \in C} \sqrt{1 - (m_{ij})^2}} = (1 - q_i) \prod_{j \in C: (i,j) \in C} \sqrt{\frac{\beta_{ij}}{1 - \beta_{ij}}}, \quad (16)$$

$$m_{ij} \equiv \sum_{\sigma_{ij}} (2\sigma_{ij} - 1) b_{ij}(\sigma_{ij}) = 2\beta_{ij} - 1, \quad q_i \equiv \sum_{j \in C}^{(i,j) \in C} 1, \quad (17)$$

where once again $\{b\}$ (or $\{\beta\}$) are solutions to the BP-equations Eqs. (10,14) and C stands for an arbitrary generalized loop, defined as a subgraph of the complete bipartite graph with all its vertices having a degree larger than 1. The q_i in Eq. (19) are the C -dependent degrees, i.e. $q_i = \sum_{j|(i,j) \in C} 1$.

Let us clarify derivation of the compact expressions on the rhs of Eqs. (16):

$$\begin{aligned} \sum_{\sigma_i} b_i(\sigma_i) \prod_{j:(i,j) \in C} (m_{ij} - 2\sigma_{ij} + 1) &= \sum_{k:(i,k) \notin C} \beta_{ik} \prod_{j:(i,j) \in C} (m_{ij} + 1) \\ &+ \sum_{k:(i,k) \in C} \beta_{ik} (m_{ik} - 1) \prod_{j \neq k}^{(i,j) \in C} (m_{ij} + 1) = 2^{q_i} (1 - q_i) \prod_{j:(i,j) \in C} \beta_{ij}, \end{aligned} \quad (18)$$

where we used relations (17) and also the double stochasticity of β .

Combining Eqs. (15,16) one derives

$$r_C = \left(\prod_{i \in C} (1 - q_i) \right) \prod_{(i,j) \in C} \frac{\beta_{ij}}{1 - \beta_{ij}}. \quad (19)$$

In Eq. (19), loops comprised of an even (odd) number of vertices give positive (negative) contributions to r_C . Eq. (19) was stated in [14] for the bipartite case. From the above derivation, we can see that the result is valid for any graph \mathcal{G} .

5. Bootstrapping BP and Fractional BP

Another important result for PMs is the following re-summation of the LS

$$\frac{Z(w)}{Z_{\text{BP}}(w)} = \frac{Z(\tilde{\beta})}{\prod_{(i,j) \in \mathcal{E}} (1 - \beta_{ij})}, \quad (20)$$

where $\tilde{\beta} = (\beta_{ij}(1 - \beta_{ij}) | (i,j) \in \mathcal{E})$. Eq. (20) was stated in [14] only for the case of permanent - PM over binary graph, even though direct verification confirms (see Appendix B) that the formula is generic, applicable to PM over an arbitrary graph.

We call Eq. (20) the Bootstrap-BP equation because the rhs of Eq. (20) provides a quantitative measure of the relative accuracy of BP.

Extending the approach of [31] from bipartite to general graphs, one arrives at the following fractional generalization of Eq. (20)

$$\frac{Z(w)}{Z_{\gamma}(w)} = Z(\tilde{\beta}_{\gamma}) \prod_{(i,j) \in \mathcal{E}} (1 - \beta_{ij})^{\gamma}, \quad (21)$$

$$-\log Z_{\gamma}(w) = \min_{\beta \in \text{DS}} \sum_{(i,j) \in \mathcal{E}} \left(\beta_{ij} \left(\frac{\varepsilon_{ij}}{T} + \log \beta_{ij} \right) + \gamma(1 - \beta_{ij}) \log(1 - \beta_{ij}) \right), \quad (22)$$

where $\tilde{\beta}_{\gamma} = (\beta_{ij} * (1 - \beta_{ij})^{-\gamma} | (i,j) \in \mathcal{E})$ and the doubly stochastic β_{ij} 's entering Eqs. (20,22) come from solving the following γ -version of Eq. (14)

$$\forall (i,j) \in \mathcal{E} : \quad \beta_{ij}(1 - \beta_{ij})^{-\gamma} = \exp \left(\frac{\mu_i + \mu_j - \varepsilon_{ij}}{T} \right), \quad (23)$$

γ can be any number in $[-1; 1]$, with $\gamma = -1$ corresponding to the BP case. Notice, that the Loop Series representation for the partition function correction only applies to the BP case. Its generalization to arbitrary γ is not known to the authors.

6. Finding MWPMs via BP: An Illustration

For an arbitrary graph, the BPLP=LP relaxation of the MWPM problem is known to be half-integral [11]. When the solution to the relaxation is integral, one has a certificate of optimality and $E = E_0$. When the solution is half integral, $E_0 < E$, simply because BPLP is a relaxation of the original MWPM problem. The Bootstrap BP recasting of the LS provides a compact expression relating E and E_0 in terms of some new partition function. The question explored in the remainder of this section is the following:

- Can the (fractional) Bootstrap-BP formula in Eq. (21) be used to close the gap between E_0 and E ?

The rhs of Eq. (21) involves the PF of a new GM defined on graph \mathcal{G} . Our hope is that the new GM is easier to deal with than the original GM. Before checking to see if this is true, we first analyze the $T \rightarrow 0$ limit of Eq. (21).

The fractional parameter γ in Eq. (21) can be chosen arbitrary from the interval $[-1; 1]$. However, in order to avoid resolving a fictitious singularity³ it is convenient to consider $\gamma = 0$. Then in the $T \rightarrow 0$ limit one derives from Eq. (21,23):

$$E = E_0 + E_1, \quad E_1 = \min_{\sigma \in \Sigma} \sum_{(i,j) \in \mathcal{E}} (\varepsilon_{ij} - \mu_i - \mu_j) \sigma_{ij} \quad (24)$$

where E and E_0 are as in Eqs. (2,3), and μ are dual variables (often called chemical potentials in statistical physics) that can be reconstructed from the optimal solution of the BPLP (3) as follows:

³ The algebraic factor on the rhs of Eq. (21) is singular in the $T \rightarrow 0$ limit at $\gamma < -1$ when some β_{ij} approaches unity. However, the singularity is fictitious, as the PF on the rhs of Eq. (21) will turn to 0 (in the limit). Since $Z(w) \sim \exp(-E/T)$ and $Z_{\gamma}(w) \sim \exp(-E_0/T)$ and $E_0 < E$ in the case of interest, we expect to get zero (in the limit) for the rhs of Eq. (21).

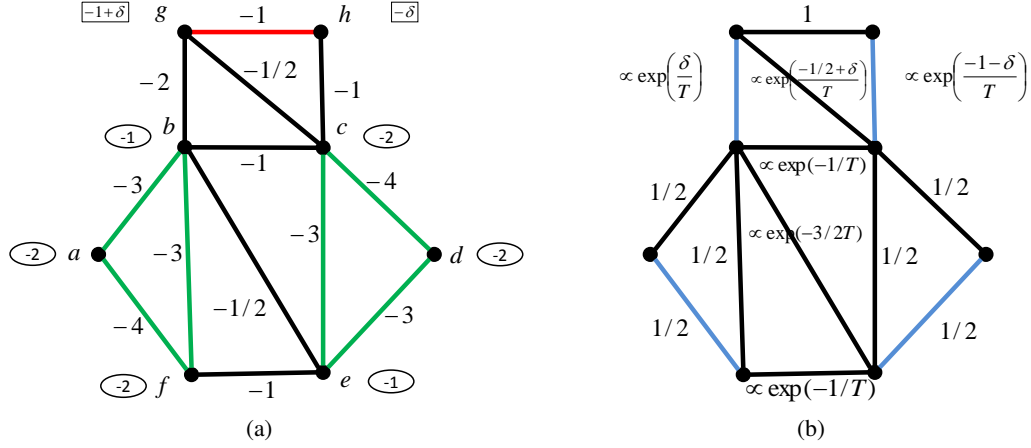


Figure 1. Illustrative 8-node example analysed with the $\gamma = 0$ version of the bootstrap-BP approach. (a) Numbers next to edges are ε_{ij} . Edges shown in green and red output $\beta_{ij} = 1/2$ and $\beta_{ij} = 1$ respectively in BP-LP. MWPM is $(a, f), (e, d), (g, b), (h, c)$. Numbers shown in ovals and squares correspond to the leading asymptotic estimates for the Lagrangian multipliers, μ_i . Freedom in choosing the parameter δ , entering μ_i adjusted to $\beta_{ij} = 1$ does not affect the $E - E_0$ correction we are looking for in here. (b) Expressions next to the edges are respective β_{ij} . Edges marked blue give the leading order PM contribution to $Z(\tilde{\beta})$.

- One first identifies any non-intersecting half-integral cycles C of \mathcal{G} , i.e. odd-length cycles where $\beta_{ij} = 1/2$ for each edge $(i, j) \in C$ and $\{\beta_{ij}\}$ is the solution of BPLP⁴.
- One computes Lagrangian multipliers, μ_i , along each half-integral cycle, resolving, separately for every cycle the set of linear equations, $\mu_i + \mu_j = \varepsilon_{ij}$.
- For every edge, (i, j) with $\beta_{ij} = 1$ one computes the Lagrangian multipliers associated with the vertices i and j , $\mu_i \leftarrow \varepsilon_{ij}/2 + \delta_{ij}$ and $\mu_j \leftarrow \varepsilon_{ij}/2 - \delta_{ij}$, where δ_{ij} can be chosen arbitrarily, for example $\delta_{ij} \leftarrow \varepsilon_{ij}/2$.⁵

To demonstrate utility of the bootstrap BPLP formulas (3,24) let us consider the 8-node example shown in Fig. (1). In this example, the MWPM is $(a, f), (e, d), (g, b), (h, c)$ and, therefore, $E = -4 - 3 - 2 - 1 = -10$. The solution to the BPLP is fractional. All edges on the two cycles, (a, b, f) and (c, d, e) have $\beta = 1/2$, edge (g, h) has $\beta_{gh} = 1$ and all other edges in the graph have $\beta = 0$. Therefore, $E_0 = (-4 - 3 - 3)/2 + (-4 - 3 - 3)/2 - 1 = -11$, which is clearly less than E . All dual variables, μ_i , associated with the half-integral cycles (shown in green in Fig. (1a)) are determined unambiguously. The dual variables for edge (g, h) must satisfy $\mu_g + \mu_h + 1 = 0$ and we choose $\mu_g - \mu_h = 2\delta$. In principle, the freedom can be fixed taking into account the DB vertex constraints on all the β -corrections. However, this is not necessary because the δ 's simply cancel when computing the correction $E_1 = \delta + 1 - \delta = 1 = E_0 - E$ for the PM contribution to $Z(\beta)$ marked in blue in Fig. (1b).

A number of things are worth noting from this simple example. First, the Bootstrap formula (20) allowed us to reduce the original MWPM problem to an auxiliary MWPM problem on the same graph. At first, the new MWPM problem does not appear to be easier to solve than the original MWPM problem. However, this example suggests a direction to further simplify the auxiliary problem by transforming it to a MWPM problem on a strictly smaller graph. Indeed, for the example considered one can *contract* each of the “green, half-integer” cycles to individual nodes. This suggestion is inspired by the following

⁴ If the BPLP is fractional, then at least two such cycles exist. This is guaranteed by half-integrality of BPLP and because we are solving a PM problem.

⁵ The choice of δ does not affect the optimal σ defining E_1 in Eq. (24).

observations [21, 22, 23]:

- Define a *blossom* to be an odd-sized cycle and choose our blossoms to be the odd-sized cycles with $\beta_{ij} = 1/2$ – marked green in Fig. (1a). Such blossoms contribute zero weight to the energy correction E_1 .
- Edmond’s “outer” constraint associated with any of the “half-integer” (green) blossom states that *at least one edge* from the cut (outer) edges should be in the optimal MWPM solution of the original problem. Note that for any of the “half-integer” blossoms in the example of Fig. (1) exactly one cut edge (and not more than one!) contributes to the optimal MWPM of the original problem.

In the following section we describe an algorithm that repeatedly applies this contraction concept to identify a sequence of corrections to the MWPM energy E .

7. Algorithm for finding MWPM sequentially

Algorithm: BP-Bootstrap-Contract-Repeat

Initialization: The master system is defined as a non-degenerate MWPM over the $(\mathcal{V}, \mathcal{E}, w)$ tuple. (Add small random corrections to w to ensure non-degeneracy.). Set $E = 0$.

Iterate:

(1) Solve the LP=BP-LP, Eqs. (3,13) over $(\mathcal{V}, \mathcal{E}, w)$. (For efficient distributed implementation use the algorithm of [9, 11].) β and E_0 are the outputs. Set, $E+ = E_0$.

(2) If solution of the LP=BP-LP is integer: **Stop**.

(3) Otherwise, modify the master system following the procedure:

- (a) Identify non-intersecting half-integral cycles of \mathcal{G} , i.e. cycles along which all edges are half-integral, $\beta_{ij} = 1/2$, in the solution of the LP=BP-LP. (Existence of at least two such cycles is guaranteed.)
- (b) Compute the Lagrangian multipliers, μ_i , along the half-integer cycles, resolving, separately for every cycle the set of linear equations, $\mu_i + \mu_j = \varepsilon_{ij}$.
- (c) For every edge, (i, j) , which is in PM according to the LP=BP-LP, i.e. $\beta_{ij} = 1$, compute the Lagrangian multipliers associated with the vertices i and j , $\mu_i \leftarrow \varepsilon_{ij}/2 + \delta_{ij}$ and $\mu_j \leftarrow \varepsilon_{ij}/2 - \delta_{ij}$, where δ_{ij} can be chosen arbitrarily, for example $\delta_{ij} \leftarrow \varepsilon_{ij}/2$.
- (d) Update weights, $\forall (i, j) : \varepsilon_{ij} \leftarrow \varepsilon_{ij} - \mu_i - \mu_j$.
- (e) Update the $(\mathcal{V}, \mathcal{E}, w)$ tuple according to the following rules: Contract all vertices correspondent to a half-integer cycle into one new vertex. If multiple edges have appeared in the result of the contraction, combine them in one edge choosing the new weight equal to be the largest weight of the combined edges. Repeat it for all the half-integer cycles.
- (f) Add small random correction to w to remove possible (future) degeneracy.

As already noted, the BP-Bootstrap procedure outputs an energy E which is an upper bound on the value of the MWPM (proof of this statement will be presented in a future work). Once the upper bound is derived, one can use the following heuristics to find the PM corresponding to the output. Assume that all the ε weights in the original PM formulation are integer and solve the following auxiliary LP,

$$\min \sum_{(i,j) \in \mathcal{E}} \varepsilon_{ij} x_{ij}, \text{ s.t. } \sum_{(i,j) \in \mathcal{E}} \varepsilon_{ij} x_{ij} \leq E, \forall i \in \mathcal{V} : \sum_{j: (i,j) \in \mathcal{E}} x_{ij} = 1, \forall (i,j) \in \mathcal{E} x_{ij} \geq 0.$$

50 % sparse				90 % sparse			
N / M	# Cor.	# LPs	# Bos.	N / M	# Cor.	# LPs	# Bos.
100 / 1961	100	1.4[1,4]	0.9[0,6]	100 / 475	100	1.4[1,4]	0.8[0,8]
250 / 12284	100	1.6[1,5]	1.2[0,8]	250 / 2973	100	1.6[1,3]	1.2[0,4]
500 / 49159	100	1.8[1,6]	1.7[0,12]	500 / 11898	100	1.7[1,4]	1.5[0,6]
1000 / 196696	100	1.9[1,5]	1.9[0,8]	1000 / 47576	99	1.8[1,4]	1.8[0,8]

Table 1. Evaluation of BP-Bootstrap-Contract on random complete graph instances for two levels of sparsity. N and M indicate the number of nodes and average number of edges in each of the 100 instances. $\# \text{Cor.}$ is the number of instances (out of 100) in which BP-Bootstrap-Contract found the weight of the optimal matching. $\# \text{LPs}$ indicates the average number of LPs solved by BP-Bootstrap. The bracketed numbers indicate the minimum and maximum number of LPs solved across the 100 instances. $\# \text{Blos.}$ is the average number of blossoms collapsed in each MWPM problem.

N	M	# Cor.	# LPs	# Bos.
100	285	95	5.7[2,13]	16.0[4,32]
250	733	83	10.6[4,28]	45.8[18,82]
500	1481	75	16.4[5,36]	92.1[44,132]
1000	2979	60	28.8[9,70]	200.3[140,294]
5000	14975	5	78.5[24,202]	996.0[796,1212]

Table 2. Evaluation of BP-Bootstrap-Contract on random Triangulation instances.

Singly-Connected				3-Connected			
K	# Cor.	# LPs	# Bos.	K	# Cor.	# LPs	# Bos.
100	100	10.0[3,17]	59.2[36,72]	100	100	19.6[4,39]	60.3[18,102]
200	100	16.4[5,29]	128.1[86,156]	200	100	34.3[5,71]	130.2[42,192]
500	100	34.4[11,68]	336.6[262,390]	500	100	63.2[19,143]	355.8[200,466]
1000	100	58.8[19,111]	704.8[582,802]	1000	100	119.7[20,223]	793.8[490,956]

Table 3. Evaluation of BP-Bootstrap on a chain of triangles connected by a single edge or 3 edges.

If the LP outputs an integral solution, we stop. Otherwise, we run the same LP with $E - 1$ instead of E , and repeat the procedure till an integral solution is found.

8. Experiments

We conducted a set of experiments to evaluate the performance of the BP-Bootstrap-Contract procedure. The BP-Bootstrap algorithm solves a series of LP problems, $LP^{(0)}, \dots, LP^{(S)}$, where $LP^{(S)}$ is the first LP with an integral solution. $LP^{(t+1)}$, the LP in iteration $t+1$, is constructed from $LP^{(t)}$ by formulating a new weighted PM problem.

Since the LP relaxation to the PM problem is half-integral [26], we are assured that the set of edges for which $\beta_{ij}^{(t)} = 1/2$ will form non-intersecting cycles of odd length (note that $\beta_{ij}^{(t)}$ denotes the belief on edge (i, j) in the solution to $LP^{(t)}$). In addition, each odd-sized half-integral cycle identifies a constraint that can be used to tighten the relaxation of $LP^{(t)}$. Such constraints are referred to as *blossom* constraints in the matching literature [21] and we will continue to adopt that name herein. Contracting all vertices correspondent to an odd-sized half-integral cycle has the effect of adding the blossom constraint to the LP. Thus, by contracting blossoms in each iteration, we hope to construct a series $LP^{(0)}, \dots, LP^{(S)}$ of

increasingly tight LP relaxations to the PM problem.

Unfortunately, the procedure of greedily collapsing blossoms is not guaranteed to produce a tight LP for the MWPM problem. This means that the BP-Bootstrap procedure will not necessarily recover the weight E of the optimal perfect matching. We therefore rely on experimentation to gauge the efficacy of the proposed BP-Bootstrap heuristic.

We synthetically generated several types of weighted graph problems and compared the weight of the matching found by BP-Bootstrap to the weight of the optimal matching. The weight of the optimal perfect matching was found using Kolmogorov’s BlossomV algorithm - the state-of-the-art algorithm for solving weighted matching problems [22].

We conducted experiments on three problem types: 1) Sparse complete graph instances; 2) Triangulation instances; and 3) Chained triangle instance. The sparse complete graph instances were generated by taking a complete graph on N nodes and eliminating edges with probability $p = \{0.5, 0.9\}$.⁶ Edge weights were assigned an integral weight drawn from a discrete uniform distribution in $[1, 2^{20}]$. The triangulation instances were generated by randomly placing points in the $2^{20} \times 2^{20}$ square and computing a Delaunay triangulation on this set of points. The edge weights were set to the rounded Euclidean distance between two points. The chained triangle instances were used in the First DIMACS Implementation Challenge [32]. Based on the algorithms t.f and tt.f by N. Ritchey and B. Mattingly, we generate a sequence of K triangles. The chain of triangles is formed by connecting neighboring triangles via a single edge or by three edges. While certainly not exhaustive, this set of problems encompass an interesting range of graph types.

The results from running BP-Bootstrap on sparse complete graph instances are shown in Table 1. For each number of nodes $N \in \{100, 250, 500, 1000\}$, we generated 100 random instances. We report the average number of edges, M , for each setting of N and # *Cor.*, the number of instances (out of 100) for which BP-Bootstrap found the optimal weight matching. In addition, we report the average and [minimum, maximum] number of LP s solved by BP-Bootstrap as well as the number of blossoms contracted by the BP-Bootstrap procedure. From this table it seems clear that BP-Bootstrap procedure is accurate on dense complete graphs. A reason for the success of the method on this class of graphs is that $LP^{(0)}$, the initial LP relaxation, is tight a large percentage of the time. This is indicated by the fact that on average less than two LP s need to be solved by BP-Bootstrap.

In contrast, the Bootstrap method performs rather poorly on the random Triangulation instances shown in Table 2. The greedy blossom contracting approach does not appear to be successful on this class of problems. In fact, when we examined the operation of Kolmogorov’s BlossomV algorithm on these instances we found that BlossomV frequently needed to backtrack and expand blossoms that had been collapsed - an operation not supported by our greedy heuristic.

Finally, the results from running BP-Bootstrap on the chained triangle instances are shown in Table 3. The BP-Bootstrap method performs well on these problems. Interestingly, we see that the greedy heuristic is effective even though these instances require solving many LP s and collapsing many blossoms.

9. Questions yet to be addressed in this project

We conclude by describing what we believe are the two most important challenges needing to be resolved in our “Matching” using BP project.

- As proven in [33] for the bipartite (permanent) case: $Z \geq Z_{BP}$. As noticed in [31], the proof of [33] is in fact a direct corollary of Eq. (20) combined with a combinatorial result of Schrijver from [34]. Indeed, it was proven in [34] (see Corollary 1c) that $\text{Perm}(\beta \cdot (1 - \beta)) \geq \prod_{(i,j) \in \mathcal{E}} (1 - \beta_{ij})$, which is essentially the statement that the rhs of Eq. (20) in the bipartite case is equal or larger than unity. The proof of Schrijver is indirect and the author mentions in passing “... we have tried to find a direct proof of it, based on continuity and differentiability, and did not succeed”. Is it possible to

⁶ In doing so, we ensure that the graph contains a perfect matching by ensuring that the graph remains edge-connected

find such a direct proof based on the combination of the Loop Calculus (15,19) and the Bootstrap-BP (20) formulas in the bipartite case? Can one generalize the $Z \geq Z_{BP}$ statement and Schrijver inequality to the case of a general, non-bipartite, graph?

- In general, and as explained above, our Bootstrap-and-Contract algorithm outputs an upper bound on the MWPM. In our empirical study, we observed that when our algorithm fails to produce an optimal matching, the BlossomV algorithm needs to backtrack and expand one or more contracted blossoms. A remaining question is how to incorporate the undo/expand step into our procedure. In particular, can one do so in a distributed manner, via a message passing (or perhaps just greedy) procedure?

Appendix A. Convexity of the Bethe Free Energy

This Appendix generalizes the main result of [27], where convexity of the BFE was proven for the case of PM GM over the bi-partite graph, to the case of PM GM over a general graph. Indeed, according to Theorem 20 of [27], $\sum_{j:(i,j) \in \mathcal{E}} \log \beta_{ij} - (1 - \beta_{ij}) \log(1 - \beta_{ij})$ is convex within the polytope defined by, $\sum_{j:(i,j) \in \mathcal{E}} \beta_{ij} = 1$ and $\beta_{ij} \geq 0$, for all $(i, j) \in \mathcal{E}$ and regardless of the global structure of the graph \mathcal{G} . Therefore, \mathcal{F}_{BP} (12) is convex within the domain DS simply because it is represented as a sum of functions which are all convex within the domain.

Appendix B. Derivation of the bootstrap formulas

In this Appendix we derive Eqs. (20,21) following the logic of [14], see also Appendixes B.1, D.1 of [31]. For the purpose of generality we will consider arbitrary (fractional) γ and arbitrary graph \mathcal{G} .

Our algebraic derivation consists of comparing two formulas. First, considering vectors of weights corresponding to the two sides of Eq. (23) and equating the respective partition functions one derives

$$Z(\tilde{\beta}_\gamma) = \sum_{\sigma \in \Sigma} \exp \left(\sum_{(i,j) \in \mathcal{E}} \sigma_{ij} \frac{\mu_i + \mu_j - \varepsilon_{ij}}{T} \right) = Z(w) \exp \left(\frac{\sum_{i \in \mathcal{V}} \mu_i}{T} \right), \quad (\text{B.1})$$

where we have used, $\sum_{(i,j) \in \mathcal{E}} \sigma_{ij} (\mu_i + \mu_j) = \sum_{i \in \mathcal{V}} \mu_i \sum_{j:(i,j) \in \mathcal{E}} \sigma_{ij} = \sum_{i \in \mathcal{V}} \mu_i$. Second, let us apply log to both sides of Eqs. (23) and then sum the results over all the edges of \mathcal{G} weighted with β_{ij}

$$\sum_{(i,j) \in \mathcal{E}} \beta_{ij} (\log \beta_{ij} - \gamma \log(1 - \beta_{ij})) = \frac{1}{T} \left(\sum_{i \in \mathcal{V}} \mu_i - \sum_{(i,j) \in \mathcal{E}} \sigma_{ij} \varepsilon_{ij} \right), \quad (\text{B.2})$$

where we have used the double stochasticity of the optimal β through, $\sum_{(i,j) \in \mathcal{E}} \beta_{ij} (\mu_i + \mu_j) = \sum_{i \in \mathcal{V}} \mu_i \sum_{j:(i,j) \in \mathcal{E}} \beta_{ij} = \sum_{i \in \mathcal{V}} \mu_i$. Then, combining Eqs. (B.1,B.2) with Eq. (22) we arrive at Eq. (21). \blacksquare

References

- [1] Bethe H 1935 *Proceedings of Royal Society of London A* **150** 552
- [2] Peierls H 1936 *Proceedings of Cambridge Philosophical Society* **32** 477–481
- [3] Gallager R 1963 *Low density parity check codes* (MIT Press, Cambridge, MA)
- [4] Pearl J 1988 *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (San Francisco: Morgan Kaufmann Publishers, Inc.)
- [5] Yedidia J, Freeman W and Weiss Y 2001 *Generalized belief propagation* vol 13 (Cambridge, MA, MIT Press) pp 689–695
- [6] Yedidia J S, Freeman W T and Weiss Y 2005 *Information Theory, IEEE Transactions on Information Theory* **51** 2282–2312
- [7] Bayati M, Shah D and Sharma M 2005 Maximum weight matching via max-product belief propagation *Info. Theory, 2005. ISIT 2005. Proceedings. International Symposium on* pp 1763–1767
- [8] Bayati M, Shah D and Sharma M 2006 A simpler max-product maximum weight matching algorithm and the auction algorithm *Info. Theory, 2006 IEEE International Symposium on* pp 557–561

- [9] Bayati M, Shah D and Sharma M 2008 *IEEE Transactions on Information Theory* **54** 1241–1251 proc. IEEE Int. Symp. Information Theory, 2006
- [10] Chertkov M 2008 *Journal of Statistical Mechanics: Theory and Experiment* **10** 16–+ (Preprint 0801.0341) URL http://www.iop.org/EJ/article/1742-5468/2008/10/P10016/jstat8_10_p10016.html
- [11] Sanghavi S, Malioutov D and Willsky A 2011 *Info. Theory, IEEE Trans. on* **57** 2203–2212 ISSN 0018-9448
- [12] Chertkov M and Chernyak V 2006 *Physical Review E* **73** 065102(R)
- [13] Chertkov M and Chernyak V 2006 *Journal of Statistical Mechanics* P06009
- [14] Watanabe Y and Chertkov M 2010 *Journal of Physics A: Mathematical and Theoretical* **43** 242002 URL <http://stacks.iop.org/1751-8121/43/i=24/a=242002>
- [15] V G, JM M and HJ K 2007 *Journal for Machine Learning Research (JMLR)* **8** 1987–2016
- [16] Gomez V, Kappen H J and Chertkov M 2010 *Journal of Machine Learning Research* **11** 1273–1296 URL <http://arxiv.org/abs/0901.0786>
- [17] Chertkov M and Chernyak V Y 2006 *arxiv:cs/0609154*
- [18] Kudekar S, Johnson J K and Chertkov M 2011 *CoRR* **abs/1102.5386**
- [19] Kudekar S, Johnson J K and Chertkov M 2011 *CoRR* **abs/1105.4665**
- [20] Gelfand A, Chertkov M and Shin J 2013 *arxiv:1305.4130* URL <http://arxiv.org/abs/1305.4130>
- [21] Edmonds J 1965 *Canadian Journal of mathematics* **17** 449–467 URL <http://www.cs.princeton.edu/introcs/papers/edmonds.pdf>
- [22] Kolmogorov V 2009 *Mathematical Programming Computation* **1** 43–67 URL <http://www.springerlink.com/index/E251L2WWW0258265.pdf>
- [23] Chandrasekaran K, Végh L A and Vempala S 2012 *CoRR* **abs/1207.5813**
- [24] Chertkov M, Gelfand A and Shin J 2013 *unpublished*
- [25] Yedidia A B and Chertkov M 2011 *CoRR* **abs/1108.0065**
- [26] Schrijver A 2003 *Combinatorial Optimization* (Springer-Verlag)
- [27] Vontobel P 2013 *Information Theory, IEEE Transactions on* **59** 1866–1901 ISSN 0018-9448
- [28] Chertkov M, Kroc L and Vergassola M 2008 Belief propagation and beyond for particle tracking, *arxiv:0806.1199* (Preprint *arxiv:0806.1199*) URL <http://arxiv.org/abs/0806.1199>
- [29] Chertkov M, Kroc L, Krzakala F, Vergassola M and Zdeborova L 2010 *Proceedings of National Academy of Sciences* **107**(17) 7663–7668 (Preprint *arxiv:0909.4256*) URL <http://arxiv.org/abs/0909.4256>
- [30] Huang B and Jebara T 2009 Approximating the permanent with belief propagation, *arxiv:0908.1769* (Preprint *arxiv:0908.1769*) URL <http://arxiv.org/abs/0908.1769>
- [31] Chertkov M and Yedidia A B 2011 *CoRR* **abs/1108.0065**
- [32] Johnson D S and McGeoch C C 1993 *Network Flows and Matching: First DIMACS Implementation Challenge, Generators available at ftp://dimacs.rutgers.edu/pub/netflow/generators/matching* (American Mathematical Society, Providence, RI)
- [33] Gurvits L 2011 *ArXiv e-prints* (Preprint 1106.2844)
- [34] Schrijver A 1998 *Journal of combinatorial theory. Series B* **72** 122–135