BELIEF-PROPAGATION FOR WEIGHTED b-MATCHINGS ON ARBITRARY GRAPHS AND ITS RELATION TO LINEAR PROGRAMS WITH INTEGER SOLUTIONS

MOHSEN BAYATI*, CHRISTIAN BORGS†, JENNIFER CHAYES‡, AND RICCARDO ZECCHINA§

Abstract. We consider the general problem of finding the minimum weight b-matching on arbitrary graphs. We prove that, whenever the linear programming (LP) relaxation of the problem has no fractional solutions, then the belief propagation (BP) algorithm converges to the correct solution. We also show that when the LP relaxation has a fractional solution then the BP algorithm can be used to solve the LP relaxation. Our proof is based on the notion of graph covers and extends the analysis of [5, 27].

These results are notable in the following regards: (1) It is one of a very small number of proofs showing correctness of BP without any constraint on the graph structure. (2) Variants of the proof work for both synchronous and asynchronous BP; it is the first proof of convergence and correctness of an asynchronous BP algorithm for a combinatorial optimization problem.

1. Introduction. Motivated by the cavity method in statistical physics, very fast distributed heuristic algorithms have recently been developed for the solution of random constraint satisfaction problems [33], [13], [11], [17], [1]. Similar heuristic methods have been known for many years [22] in the context of coding theory. And a variety of specific examples of such algorithms have been developed in artificial intelligence, signal processing, and digital communications. Well-known examples include the Viterbi algorithm, the iterative decoding algorithm in turbo codes and in low-density parity-check codes [41], Pearl's belief propagation algorithm for Bayesian networks [38], the Kalman filter, and certain fast Fourier transform (FFT) algorithms. Very recent applications can also be found in systems biology [21], [25], [61], computer vision [48], and data clustering [19].

In some cases, the algorithms generated by the cavity method are exactly of the form of a classic belief propagation (max-product or min-sum) i.e., a message-passing algorithm for efficiently computing marginal probabilities or finding the most likely assignment (MAP assignment) of a joint probability distribution of discrete variables. The belief propagation (BP) algorithm converges to a correct solution if the associated graph is a tree, and may be also a good heuristic for some graphs with cycles. In other cases, the cavity method may lead to a more involved survey propagation (SP) algorithm [33], [12] in which some form of correlation among variables is controlled.

In this paper, we study the problem of finding the minimum weight b-matchings in *arbitrary* graphs via the min-sum version of BP algorithm[1].

Our Results.. Let $G = (V, E)$ be an undirected graph with edge weights w_{ij} for each edge $\{i, j\} \in E$ and node capacities b_i for each node $i \in V$. The iterative message-passing algorithm based on synchronous BP for solving the weighted perfect b-matching problem (see our Section 2 for the precise definition) is the following *simple* procedure: At each time, every vertex of the graph sends (real valued) messages to each of its neighbors. The message transmitted at time t from i to j is w_{ij} minus

*Stanford University, Stanford CA 94305 (bayati@stanford.edu).

†Microsoft Research, Cambridge MA 02142 (borgs@microsoft.com).

‡Microsoft Research, Cambridge MA 02142 (jchayes@microsoft.com).

§Politecnico Di Torino, Torino Italy (riccardo.zecchina@polito.it).

[1]Throughout this paper, the term BP algorithm refers to the min-sum version of the Belief Propagation algorithm.

the b_i^{th} minimum of the messages previously received by i at time $t-1$ from all of its neighbors except j. At the end of each iteration, every vertex i selects b_i of its adjacent edges that correspond to the b_i smallest received messages.

We will show the following result: For arbitrary graphs G, and all sets of weights $\{w_{ij}\}$, after $O(n)$ iterations, the set of selected edges converges to the correct solution, i.e., to the minimum weight perfect b-matching of G, provided that the LP relaxation of the problem (see Section 2 for definitions) has no fractional solutions. Additionally we introduce a new construction, a *generalized computation tree*, which allows us to analyze the more complicated case of BP with an asynchronous updating scheme, and prove convergence and correctness of it when each edge of the graph transmits at least $\theta(n)$ messages. To the best of our knowledge, this technique is new and can be applied in the analysis of asynchronous BP in other problems as well. These are extensions of the previous results of [5] and [27] which showed convergence and correctness of the above algorithm for bipartite graphs.[2] Moreover, our proof gives a better understanding of the often-noted but poorly understood connection between BP and LP through *graph covers*. We also modify our BP algorithm and its analysis to include the problem of finding the *non-perfect* weighted b-matchings. Recently and independently from our work a similar result for the scenario of using synchronous BP for non-perfect matchings was shown by Sanghavi, Malioutov and Willsky [44].

Finally, we note that a subset of our results appeared in a shorter version of this paper [8] that analyzed BP through dual LP variables. Due to space limitations several lemmas in [8] were stated without proof. For the sake of completeness we provide that proof in Section 8 of the present paper. The analysis of this paper is based on graph covers which yields simpler proofs with stronger results. For example it allows us to show that when LP has a solution that is partially fractional then a slightly modified BP can be used to solve the LP (cf. Section 7). The latter results were presented at the workshop on "Phase Transitions, Hard Combinatorial Problems and Message Passing Algorithms, Banff, CA" in June 2008.

Related Works.. The weighted b-matching problem is an important problem in combinatorial optimization. It belongs to a family of integer linear programs which have been well-studied and can be solved in strongly polynomial time [15], [16], [29], [14]. For extensive surveys see [26] and [39]. In physics, the study of the random 1-matching problem goes back to the work of Mèzard and Parisi [32] who made a celebrated conjecture for the expected optimum weight ($\pi^2/6$) that was proven to be exact a decade later by Aldous [2].

BP algorithms have been the subject of extensive study in several communities. The general BP algorithm is known to be correct on graphs with no cycles [38]. For graphs with a single cycle, convergence and correctness of BP have also been rigorously analyzed [3], [56]. For arbitrary graphs, relatively little is known about the correctness of BP, although some interesting progress has been made in [62], [58], [49], [53], [30]. Performance of the BP algorithm usually depends on the length of cycles in graphs; most analytical results require that the graphs have no short cycles (i.e., that they are large-girth graphs) [41], [4], [23]. For the case of weighted matchings and a few other problems, there were results that BP works correctly on graphs with many short cycles ([57], [42], [5], [34], [37], [36]). Very recently, [24] showed this result holds for the family of minimum-cost network flow problems. They also show that BP can be used to obtain a fully polynomial approximation scheme (FPRAS) for min-cost

[2]Both of these results were assuming that the minimum weight matching is unique. Note that if there is more than one solution, then one can construct a fractional solution to the LP relaxation.

network flow problems.

Recent works have also suggested a connection between the BP algorithm and linear programming (LP) in particular problems. A relationship between iterative decoding of channel codes and LP decoding was studied in [18], [51], [50]. In fact our proof is based on the notion of graph-cover that is used in [51] and [50] as well. Other relationships were noted in the context of BP algorithms with convex free energies [55], [54], [59], and in the case of BP algorithms for resource allocations [35]. For weighted 1-matchings, the connection was studied [6] in the context of similarities between BP equations and the primal-dual auction algorithm of Bertsekas [9]. And it was further clarified recently for non-perfect matchings in [43] and [44] where it was shown that BP does not converge to the correct solution if the LP relaxation has fractional solutions. Another recent result studies this connection for the weighted independent set problem [45]. Moreover, some recent discussions by [28] connects LP integrality to graph perfection and convergence of BP.

Finally, we note that the BP equations for solving the weighted matching problem which we use in this paper have been previously studied in [6], [27]. These equations are also very similar to equations for weighted matching problems and traveling salesman problems given in [32], [52], [2], [23], and to the equations for various other problems given in [60], [63], [40], [31].

Organization of the Paper.. The rest of the paper is organized as follows. In Section 2, we provide the setup, define the weighted b-matching problem, and describe the LP relaxation. In Section 3, we describe our algorithm for the minimum weighted perfect b-matching problem, and state our main result. The analysis of our algorithm is given in Section 4. The extension of our algorithm and results to the *non-perfect* minimum weighted b-matching problem are presented in Section 5. In Section 6, we state the asynchronous version of the BP algorithm and present its analysis. In Section 7 we show that BP can be used to solve the LP relaxation in finite number of iterations. Additionally, we show how a recent result of [24] can be combined with our graph cover construction to obtain an FPRAS for solving the LP relaxation using BP. Finally, Section 8 is dedicated to the alternative proofs for the results of Sections 3-6 using dual variables of the LP relaxation that were partially presented in [8].

2. Definitions and Problem Statement. Consider an undirected simple graph $G = (V, E)$, with vertices $V = \{1, \ldots, n\}$, and edges E. Let each edge $\{i, j\}$ have weight $w_{ij} \in \mathbb{R}$. Denote the set of neighbors of each vertex i in G by $N(i)$ and denote the number of elements in $N(i)$ by $\deg_G(i)$. Let $\mathrm{b} = (b_1, \ldots, b_n)$ be a sequence of positive integers such that $b_i \leq \deg_G(i)$. A subgraph M of G is called a b-*matching* (*perfect* b-*matching*) if the degree of each vertex i in M is at most b_i (equal to b_i). Denote the set of b-matchings (perfect b-matchings) of G by $\mathrm{M}_G(\mathrm{b})$ ($\mathrm{PM}_G(\mathrm{b})$), and assume that it is non-empty. Clearly $\mathrm{PM}_G(\mathrm{b}) \subset \mathrm{M}_G(\mathrm{b})$.

The weight of a (perfect or non-perfect) b-matching M, denoted by W_M, is defined by $W_M = \sum_{ij} w_{ij} 1_{\{i,j\} \in M}$. In the next two sections, we will restrict ourselves to the case of *perfect* b-matchings. We will extend the analysis to (possibly non-perfect) b-matchings in Section 5. The minimum weight perfect b-Matching (MWP-b-M), M^*, is defined by $M^* = \mathrm{argmin}_{M \in \mathrm{PM}_G(\mathrm{b})} W_M$. The goal of this paper is to find M^* via a min-sum belief propagation algorithm. Throughout the paper, we will assume that M^* is unique. Let ϵ be the difference between the weights of M^* and the second minimum weight b-matching; i.e.,

$$\epsilon = \min_{M \neq M^*} (W_M) - W_{M^*}.$$

due to the uniqueness of the M^*, $\epsilon > 0$. Also define w^* to be $w^* = \max_{\{i,j\}\in E}(|w_{ij}|)$

Relation to Maximum Weight Matching problem.. In some of the previous papers [5, 6, 27, 44], BP was used for finding the maximum weight matching. Note that finding MWP-b-M is equivalent to finding the maximum weight perfect b-matching. This can be easily seen by changing the signs of all weights $\{w_{ij}\}$.

Linear Programming Relaxation.. Assigning variables $x_{ij} \in \{0, 1\}$ to the edges in E, we can express the weighted perfect b-matching problem as the problem of finding a vector x $\in \{0, 1\}^{|E|}$ that minimizes the total weight $\sum_{ij\in E} x_{ij}w_{ij}$, subject to the constraints $\sum_{j\in N(i)} x_{ij} = b_i$ for all $i \in V$. Relaxing the constraint that x_{ij} is integer, this leads to the following well-known linear program (e.g. see [29]):

$$\begin{array}{rll} \min & \sum_{\{i,j\}\in E} x_{ij}w_{ij} & \\ \text{subject to} & & \\ & \sum_{j\in N(i)} x_{ij} = b_i & \forall \ \ i \\ & 0 \le x_{ij} \le 1 & \forall \ \{i, j\} \in E \end{array} \tag{2.1}$$

We say the LP relaxation (2.1) has *no fractional solution* if, every optimal solution x of LP satisfies $x \in \{0, 1\}^{|E|}$. Note that absence of fractional solutions implies uniqueness of integer solutions, since any convex combination of two integer solutions is a solution to the LP as well. We want to show that the BP algorithm for our problem converges to the correct solution, provided the LP relaxation (2.1) has no fractional solution.

3. Algorithm and Main Results. The following algorithm is a synchronous implementation of BP for finding the minimum weight perfect b-matching (MWP-b-M). The main intuition behind this algorithm (and, indeed, all BP algorithms) is that each vertex of the graph assumes the graph has no cycles, and makes the best (greedy) decision based on this assumption. This is shown in more detail in Section 4.1.

Before applying the BP algorithm, we remove all *trivial* vertices from the graph. A vertex i is called trivial if $\deg_G(i) = b_i$. This is because all of the edges adjacent to i should be in every perfect b-matching. Therefore the graph can be simplified by removal of all trivial vertices and their adjacent edges.

Algorithm Sync-BP.

---

(1) At times $t = 0, 1, \ldots$, each vertex sends real-valued messages to each of its neighbors. The message of i to j at time t is denoted by $m_{i\to j}(t)$.
(2) Messages are initialized[3] by $m_{i\to j}(0) = w_{ij}$ for all $\{i, j\} \in E$.
(3) For $t \ge 1$, messages in iteration t are obtained from messages in iteration $t - 1$ recursively as follows:

$$\forall \ \{i, j\} \in E : \quad m_{i\to j}(t) = w_{ij} - b_i^{th}\min_{\ell\in N(i)\setminus\{j\}}\Big[m_{\ell\to i}(t-1)\Big] \tag{3.1}$$

where k^{th}-min$[A]$ denotes the k^{th} minimum[4] of set A.

[3]We show in Section 4.4 that the messages can be initialized to any arbitrary values.

[4]Note that the $b_i^{th}\min_{\ell\in N(i)\setminus\{j\}}$ is well defined since we assumed that all trivial vertices are removed and thus there are at least $b_i + 1$ elements in the set $N(i)$ for each i.

(4) The estimated MWP-b-M at the end of iteration t is $M(t) = \cup_{i=1}^{n} E_i(t)$ where $E_i(t) = \big\{\{i, j_1\}, \ldots, \{i, j_{b_i}\}\big\}$ is such that $N(i) = \{j_1, j_2, \ldots, j_{\deg_G(i)}\}$ and $m_{j_1 \to i}(t) \leq m_{j_2 \to i}(t) \cdots \leq m_{j_{\deg_G(i)} \to i}(t)$. i.e., among all i's neighbors, choose edges to the b_i neighbors that transfer the smallest incoming messages to i.

(5) Repeat (3)-(4) until $M(t)$ converges[5].

In Corollary 4.2, we will show the main intuition behind the equation (3.1) and how it is derived. But we note that one can also use the graphical model representations of [5], [27], [43] to obtain the standard BP equations for this problem, which, after some algebraic calculations, yield the recursive equation (3.1).

The main result of the paper is rather surprising: it says that the above algorithm, which is designed for graphs with no cycle (i.e., for trees), works correctly for a much larger family of graphs including those with many short cycles.

THEOREM 3.1. *Assume that the LP relaxation (2.1) has no fractional solution. Then the algorithm Sync-BP converges to M^* after at most $\lceil \frac{2nw^*}{\epsilon} \rceil$ iterations.*

REMARK 1. *We will show in Section 7 that if BP is used properly then it can solve all cases of the LP relaxation* (2.1).

If the LP relaxation (2.1) has a fractional solution whose cost is strictly less than W_{M^*}, then [43], [44] have shown (for non-perfect matchings) that BP does not converge to M^*. It is straightforward to generalize this to perfect b-matching as well. But, when the LP has a fractional solution whose cost is equal to W_{M^*}, BP may fail.

For the case of 1-matchings with random edge weights, [47] show that for all $\tau > 0$, there exist $n(\tau)$ and $k(\tau)$ such that for $n > n(\tau)$ the BP algorithm finds correct assignment to $1 - \tau$ fraction of nodes in $k(\tau)$ iterations with probability at least $1 - \tau$.

4. Analysis of the Synchronous BP. This section contains the analysis of the synchronous BP algorithm for perfect b-matchings. First, in Section 4.1 we show one derivation of the equations for Sync-BP and its representation in term of the so-called computation tree. Next, in Section 4.2 we introduce the notion of graph-covers which connects the graph G to a bipartite graph $\tilde{G}$ that has all the information for finding the minimum weight perfect b-matching in the graph G. This connection is used in Section 8.3 to prove that, when the LP relaxation has no fractional solutions, then solutions on the computation tree are the same as the solutions on the original graph G.

4.1. Computation Tree and Derivation of Sync-BP. The main idea behind the algorithm Sync-BP is that it assumes the graph G has no cycle. In other words, it finds the MWP-b-M of a graph G' that has the same local structure as G but no cycles. In this section we rigorously define such a graph G' (computation tree) and show its connection with the Sync-BP algorithm.

Computation Tree.. For any $i \in V$, let T_i^t be the t-level computation tree corresponding to i, defined as follows: T_i^t is a weighted tree of height $t + 1$, rooted at i. All tree-nodes have labels from the set $\{1, \ldots, n\}$ according to the following recursive rules:

(a) The root has label i.

(b) The set of labels of the $\deg_G(i)$ children of the root is equal to $N(i)$.

[5] The subgraph $M(t)$ is not necessarily a perfect b-matching of G but we will show that after $O(n)$ iterations it will be the minimum weight perfect b-matching.

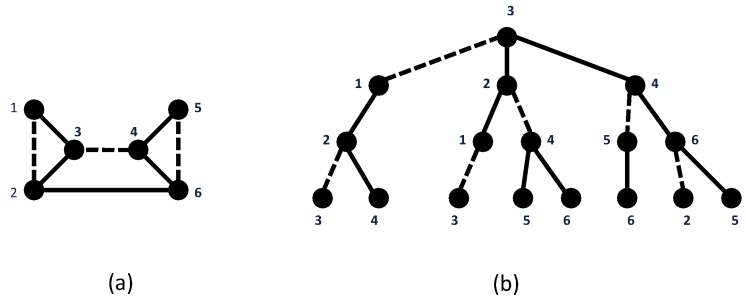

FIG. 4.1. *Part (a) shows a graph G where dashed edges represent a* 1*-matching. Part (b) shows the computation tree T_3^2 corresponding to G where the set of dashed edges form a TMWP-*1*-M.*

(c) If s is a non-leaf node whose parent has label r, then the set of labels of its children is $N(s)\backslash\{r\}$.

REMARK 2. *T_i^t is often called the* unwrapped tree *at node i. The computation tree is constructed by replicating the local connectivity of the original graph. The messages received by node i in the belief propagation algorithm after t iterations in graph G are equivalent to those that would have been received by the root i in the computation tree, if the messages were passed up along the tree from the leaves to the root. Computation trees have been used in most of the previous analysis of BP algorithms; see e.g. [22, 5, 56, 58, 57, 20].*

A subtree $\mathcal{M}$ of edges in the computation tree T_i^t is called a perfect *tree*-b-*matching* if for each *non-leaf* vertex with label i we have $\deg_{\mathcal{M}}(i) = b_i$. Now denote the tree minimum weight perfect b-matching (TMWP-b-M) of the computation tree T_i^t by $\mathcal{N}^*(T_i^t)$. We will show that Sync-BP can be seen as a dynamic programming procedure that finds the minimum weight perfect tree-b-matching over the computation tree. Figure 4.1 shows a graph G and one of its corresponding computation tree.

Sync-BP Equations.. Consider the computation tree T_i^t. Let us assume that $\deg_G(i) = k$, and that $i_1, \ldots, i_k$ are neighbors of i in G which are children of the root i as well. Let us denote the subtree of T_i^t that consists of the root edge (i, i_j) and all descendants of i_j by $T_{i_j \to i}^t$. Given this, we define the following weights and weight differences:

$$W_{i_j \to i}^+(t) = \text{Weight of TMWP-b-M in } T_{i_j \to i}^t \text{ that contains the root edge } (i, i_j).$$
$$W_{i_j \to i}^-(t) = \text{Weight of TMWP-b-M in } T_{i_j \to i}^t \text{ that } \textit{does not} \text{ contain the root edge } (i, i_j).$$
$$n_{i_j \to i}(t) = W_{i_j \to i}^+(t) - W_{i_j \to i}^-(t).$$

Clearly, for any edge $\{i, j\}$ of graph G the real number $n_{j \to i}(t)$ is well-defined; the next lemma shows its relation with the messages passed in Sync-BP.

LEMMA 4.1. *For all $1 \le i, j \le n$ such that $\{i, j\}$ is an edge of G and all $t = 0, 1, \ldots$, the following is true: $n_{j \to i}(t) = m_{j \to i}(t)$.*

Proof. We proceed by induction on t. For $t = 0$ by definition the computation tree T_i^0 has height 1. Therefore each branch $T_{i_j \to i}^0$ consists of a single root edge (i, i_j). Thus $W_{i_j \to i}^+(0) = w_{ii_j}$ and $W_{i_j \to i}^-(0) = 0$ which gives: $n_{i_j \to i}(0) = w_{ii_j}$, and by definition this is equal to $m_{i_j \to i}(0)$. Now for the general case consider the computation tree T_i^t and fix a branch $T_{i_j \to i}^t$. Denote the children of i_j in this branch by $j_1, \ldots, j_\ell$

with $\ell = \deg_G(i_j) - 1$ (by rule (c) from the construction of the computation tree described above). For simplicity of notation let $a = b_{i_j}$. Without loss of generality assume that the children $j_1, \ldots, j_\ell$ are ordered so that

$$W^+_{j_1 \to i_j}(t-1) \leq W^+_{j_2 \to i_j}(t-1) \leq \cdots \leq W^+_{j_\ell \to i_j}(t-1).$$

Now it is not hard to see that

$$W^+_{i_j \to i}(t) = w_{ii_j} + \sum_{r=1}^{a-1} W^+_{j_r \to i_j}(t-1) + \sum_{r=a}^{\ell} W^-_{j_r \to i_j}(t-1)$$

$$W^-_{i_j \to i}(t) = \sum_{r=1}^{a} W^+_{j_r \to i_j}(t-1) + \sum_{r=a+1}^{\ell} W^-_{j_r \to i_j}(t-1),$$

so that

$$\begin{aligned} n_{i_j \to i}(t) &= W^+_{i_j \to i}(t) - W^-_{i_j \to i}(t) \\ &= w_{ii_j} - W^-_{j_a \to i_j}(t-1) + W^-_{j_{a+1} \to i_j}(t-1) \\ &= w_{ii_j} - n_{j_a \to i_j}(t-1) \\ &= w_{ii_j} - a^{th}\min_{r \in N(i_j) \setminus \{i\}} \Big(n_{j_r \to i_j}(t-1) \Big). \end{aligned}$$

Therefore we have shown that variables $n_{j \to i}(t)$ satisfy the same recursive relation as variables $m_{j \to i}(t)$, equation (3.1), and satisfy the same initial conditions. Thus they are equal. □

It follows immediately from the above lemma that the set of edges $E_i(t)$ which is selected in iteration t of the algorithm Sync-BP consists of exactly the same edges which are adjacent to root i in $\mathcal{M}^*(T_i^t)$. This is formalized in the following corollary.

COROLLARY 4.2. *The algorithm Sync-BP solves the TMWP-*b*-M problem on the computation tree. In particular, for each vertex i of G, the set $E_i(t)$ is exactly the set of b_i edges which are attached to the root in TMWP-*b*-M of T_i^t.* Corollary 4.2 characterizes the estimated MWP-b-M, $M(t)$, and will be used in the proof of the main result in Subsection 4.3. In the next subsection we present a very useful connection between graph G and its double-cover graph that is crucial for the proofs of Subsection 4.3.

4.2. Graph Covers. For a simple graph $G = (V, E)$, define the *double-cover* of G as a bipartite graph $\tilde{G} = (V_1 \cup V_2, E(\tilde{G}))$ where $V_1,\ V_2$ are exact copies of V. That is for each $v \in V$ there are unique vertices $v_1 \in V_1,\ v_2 \in V_2$ and vice versa. Similarly, each edge of G has exactly two copies in $\tilde{G}$ according to the following rule: $\{u, v\} \in E$ if and only if $\{u_1, v_2\} \in E(\tilde{G}),\ \{u_2, v_1\} \in E(\tilde{G})$ such that vertices u_1, u_2 (v_1, v_2) are the two copies of u (v) in $\tilde{G}$. Note that G and $\tilde{G}$ have the same local structure that is for any vertex v in G the vertices v_1, v_2 in $\tilde{G}$ have the same neighborhood as v. For weighted graphs we assign the weight w_{ij} of the edge $\{i, j\}$ to both edges $\{i_1, j_2\},\ \{i_2, j_1\}$ of $\tilde{G}$. Figure 4.2 shows a graph G and its double-cover $\tilde{G}$. We can now write an analogues LP relaxation to (2.1) for $\tilde{G}$.

$$\begin{array}{rll} \min & \sum_{\{i,j\} \in E} (x_{i_1 j_2} + x_{i_2 j_1})\, w_{ij} & \\ \text{subject to} & & \\ & \sum_{j_2 \in N(i_1)} x_{i_1 j_2} = \sum_{j_1 \in N(i_2)} x_{i_2 j_1} = b_i & \forall\ \ i \\ & 0 \leq x_{i_r j_s} \leq 1 & \forall\ \{i_r, j_s\} \in E(\tilde{G}) \end{array} \tag{4.1}$$

Now we investigate the relationship between the LP relaxations given in (2.1) and (4.1). The following lemma characterizes this relationship.

LEMMA 4.3. *Let* $\mathrm{x}^* = (x^*_{ij})$ *and* $\tilde{\mathrm{x}}^* = (x^*_{i_r j_s})$ *be the optimum solutions for the LP relaxations* (2.1) *and* (4.1) *respectively.*

(a) $\sum_{\{i,j\}\in E} \left(x^*_{i_1 j_2} + x^*_{i_2 j_1}\right) w_{ij} = 2 \sum_{\{i,j\}\in E} x^*_{ij} w_{ij}$.

(b) If the optimum x^* *of LP* (2.1) *is integer and is unique then the optimum* $\tilde{\mathrm{x}}^*$ *of LP* (4.1) *is also integer and is unique. Moreover for all* $\{i,j\} \in E$ *the following holds* $x^*_{i_1 j_2} = x^*_{i_2 j_1} = x^*_{ij}$. *We call such optimum of LP* (4.1)*, a symmetric integer solution.*

(c) If the optimum x^* *of LP* (2.1) *is fractional then there exists an optimum* $\tilde{\mathrm{x}}^*$ *of LP* (4.1) *which is integer and satisfies the following constraint. For all* $\{i,j\} \in E$*, either* $x^*_{i_1 j_2} = x^*_{i_2 j_1} = x^*_{ij}$ *holds or* $x^*_{i_1 j_2} = 1 - x^*_{i_2 j_1}$, $x^*_{ij} = .5$ *holds. We call such optimum of LP* (4.1)*, a non-symmetric integer solution.*

Proof. Starting from an optimum x^* for the LP (2.1), it is easy to construct a feasible solution to the LP (4.1) that is denoted by $f(\mathrm{x}^*) = (y_{i_r j_s})$ and is defined by $y_{i_1 j_2} = y_{j_1 i_2} = x^*_{ij}$. Now, one side of the equality in (a) can be shown by:

$$\sum_{\{i,j\}\in E} \left(x^*_{i_1 j_2} + x^*_{i_2 j_1}\right) w_{ij} \le \sum_{\{i,j\}\in E} \left(y_{i_1 j_2} + y_{i_2 j_1}\right) w_{ij} = 2 \sum_{\{i,j\}\in E} x^*_{ij} w_{ij}.$$

On the other hand, starting from an optimum $\tilde{\mathrm{x}}^*$ of the LP (4.1), we define a feasible solution for the LP (2.1) that is denoted by $g(\tilde{\mathrm{x}}^*) = (z_{ij})$ and is defined by $z_{ij} = (x^*_{i_1 j_2} + x^*_{i_2 j_1})/2$. Now the other side of the equality in part (a) is proven by the following:

$$\sum_{\{i,j\}\in E} \left(x^*_{i_1 j_2} + x^*_{i_2 j_1}\right) w_{ij} = 2 \sum_{\{i,j\}\in E} z_{ij} w_{ij} \ge 2 \sum_{\{i,j\}\in E} x^*_{ij} w_{ij},$$

where the last inequality uses the fact that x^* is an optimum of the LP (2.1). As a corollary of this discussion one can see that the mappings f, g provide a correspondence (not necessarily one to one) between the optimums of LPs (2.1) and (4.1).

Now to prove (b), if the optimum x^* of the LP (2.1) is integer then by the above discussion $f(\mathrm{x}^*)$ is an integer optimum of the LP (4.1). And since by assumption, x^* is the unique optimum of (2.1) then $g(f(\mathrm{x}^*))$ which is an optimum of (2.1), is equal to x^*. This proves (b).

For (c) we use the well-known fact that vertices of the b-matching polytope on bipartite graphs are all integer solutions [46]. So there exist an optimum $\tilde{\mathrm{x}}^*$ for (4.1) which is integer. Since the optimum of (2.1) is unique therefore it has to be equal to $g(\tilde{\mathrm{x}}^*)$. Thus $\mathrm{x}^* = g(\tilde{\mathrm{x}}^*)$. But $\tilde{\mathrm{x}}^*$ being integer means that for all $\{i,j\} \in E$ we either have $x^*_{i_1 j_2} = x^*_{i_2 j_1} = x^*_{ij}$ or $x^*_{i_1 j_2} = 1 - x^*_{i_2 j_1}$, $x^*_{ij} = .5$ which proves (c). □ Note that the above analysis reproves a well-known fact about coordinates of the vertices of the polytope for LP (2.1). It shows that those coordinates are from the set $\{0, .5, 1\}$.

Next we will show that the algorithm Sync-BP for graphs G and $\tilde{G}$ is the same. In particular both graphs G and $\tilde{G}$ have similar local structure and therefore they have similar computation trees. The only difference is that the vertices with a fixed label i in the computation tree of G now have labels i_1 or i_2 in the computation tree of $\tilde{G}$ depending on the parity of their distance from the root. More specifically the following lemma is straightforward.

LEMMA 4.4. *There exists an isomorphism* $\phi : T^t_i \to T^t_{i_1}$ *between the computation trees of graphs* G *and* $\tilde{G}$ *that preserves the roots and for any vertex* v *with label* j

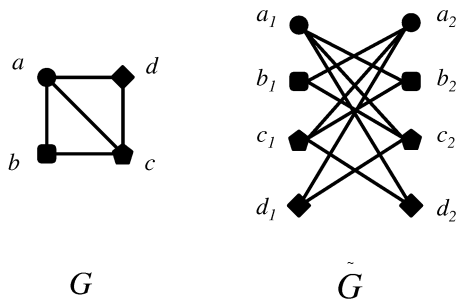

FIG. 4.2. *A graph G with four vertices and its double-cover $\tilde{G}$ on the right.*

in T_i^t, the label of $\phi(j)$ is j_1 (or j_2) if v has even (or odd) distance from the root of T_i^t. Similar isomorphism exists between T_i^t and $T_{i_2}^t$. Therefore the minimum weight perfect tree-b-matchings for T_i^t, $T_{i_1}^t$, and $T_{i_2}^t$ choose the same edges at the roots. This shows the following corollary.

COROLLARY 4.5. *The output of the algorithm Sync-BP for a vertex i of graph G is the same as the output of the algorithm Sync-BP, started with the same initial conditions, for vertices i_1 and i_2 of the double-cover $\tilde{G}$.* The notion of graph-covers and its relation with LP and BP has been studied before in the context of coding theory [51], [50] and similar notions in the combinatorial optimization context exist in [46].

4.3. Proof of Theorem 3.1. We will prove Theorem 3.1, namely that if the LP relaxation (2.1) has no fractional solution and hence M^* is unique, then Sync-BP converges to the correct MWP-b-M. We will do this by using Corollary 4.5 to reduce the problem to bipartite graphs and then use Lemma 4.3 and results of [5] and [27] to prove correctness of the BP. Alternatively, we provide a different proof of Theorem 3.1 which is also independent from the previous results [5] and [27] in Section 8. This second proof was also presented in the preliminary version of this paper [7].

Let us summarize results of [5] and [27] by the following theorem:

THEOREM 4.6. *For any weighted bipartite graph with unique minimum weight* b*-matching (and gap ϵ), the algorithm Sync-BP, converges to the correct solution within $\lceil \frac{2nw^*}{\epsilon} \rceil$.* Note that ϵ, w^* for bipartite graphs are defined the same way as in Section 3 for general graphs. Now we are ready to prove Theorem 3.1.

First note that since LP (2.1) has no fractional solution then by Lemma 4.3(a) the LP (4.1) has unique integer solution which is also symmetric. And therefore proofs given in [5] and [27] show correctness of Sync-BP in the double-cover $\tilde{G}$ after $\lceil \frac{2nw^*}{\epsilon} \rceil$ iterations. Now using Corollary 4.5, we obtain correctness of Sync-BP for the original graph G after $\lceil \frac{2nw^*}{\epsilon} \rceil$ iterations.

4.4. Independence from Initial Conditions.. We would like to point out that changing the initial condition for the messages in step (2) of Sync-BP to any arbitrary values does not change the convergence and correctness of algorithm Sync-BP. The only effect of initial condition is on the number of iterations needed for convergence. Theorem 3.1 remains true by *re-defining* w^* according to: $w^* = \max_{\{i,j\}\in E} |w_{ij}| + \max_{\{i,j\}\in E} |m_{i\to j}(0)|$. This follows because, by changing the initial condition, the algorithm Sync-BP runs over a slightly modified computation tree. The new computation tree is almost the same computation tree as T_i^t, except that the leaf edges of the tree have arbitrary weights and not w_{ij}'s from G. As it appears in proof of

Lemma 3 in [5], the only place where the weight of leaf edges appears is the inequality (16) of that paper, which will be satisfied by new definition of w^* (similarly in proof of Lemma 1.3 of [27]).

5. Extension to Possibly Non-Perfect b-Matchings. In this section we show that the algorithm and the results of the previous sections can be easily generalized to the case of b-matchings (subgraphs H of G such that degree of each vertex i in H is *at most* b_i). Let $U(H) \subset V$ be the set of *unsaturated* vertices of G (vertices $i \in V$ such that $\deg_H(i) < b_i$). Similar to Section 2, the minimum weight b-Matching (MW-b-M), H^*, is the b-Matching such that

$$H^* = \text{argmax}_{H \in \text{M}_G(\text{b})} \, W_H.$$

Note that H^* does not include any edge with positive weight because removing such edges from H^* reduces its weight while keeping it a b-matching. Therefore in this section we assume that for all $\{i,j\} \in E: \quad w_{ij} \leq 0$. The LP relaxation is slightly different from before:

$$\begin{array}{rll} \min & \sum_{\{i,j\} \in E} x_{ij} w_{ij} & \\ \text{subject to} & & \\ & \sum_{j \in N(i)} x_{ij} \leq b_i & \forall \;\; i \\ & 0 \leq x_{ij} \leq 1 & \forall \; \{i,j\} \in E \end{array} \tag{5.1}$$

Similar to Section 2 we define $\epsilon' > 0$ to be the difference between W_{H^*} and weight of the second best b-matching.

Now we can present the modified algorithm Sync-BP for finding MW-b-M in G:

Algorithm Sync-BP(2).

---

(1) At times $t = 0, 1, \ldots$, each vertex sends real-valued messages to each of its neighbors. The message of i to j at time t is denoted by $m_{i \to j}(t)$.
(2) Messages are initialized by $m_{i \to j}(0) = w_{ij}$ for all $\{i,j\} \in E$.
(3) For $t \geq 1$, messages in iteration t are obtained from messages in iteration $t-1$ recursively as follows:

$$\forall \; \{i,j\} \in E: \qquad m_{i \to j}(t) = w_{ij} - \min\left(0, b_i^{th}\text{min}_{\ell \in N(i) \setminus \{j\}} \Big[m_{\ell \to i}(t-1) \Big]\right) \tag{5.2}$$

where k^{th}-min(A) denotes the k^{th} minimum[6] of set A.
(4) The estimated MW-b-M at the end of iteration t is $H(t) = \cup_{i=1}^n F_i(t)$ where $F_i(t) = \big\{\{i, j_1\}, \ldots, \{i, j_{c_i}\}\big\}$ is such that $m_{j_\ell \to i}(t) < 0$ for all $1 \leq \ell \leq c_i$, i.e., choose edges that transfer negative messages to i.
(5) Repeat (3)-(4) until $H(t)$ converges.

The results for b-matchings generalize as follows:

THEOREM 5.1. *Assume that the LP relaxation (5.1) has no fractional solution. Then the algorithm Sync-BP(2) converges to H^* after at most $\lceil \frac{4nw^*}{\epsilon'} \rceil$ iterations.* The proof of Theorem 5.1 is similar to the one of Section 4 with the following modifications:

[6]Here $b_i^{th}\text{min}_{\ell \in N(i) \setminus \{j\}}$ is defined to be 0 if $\deg_G(i) = b_i$.

1. The computation tree T_i^t and TMW-b-M are defined as before, while Lemma 4.1 is slightly modified. A careful analysis of W^+ and W^- for the tree-b-matchings yields equations (5.2) for finding TMW-b-M in the computation tree. This is how the new equations are obtained.
2. The LP (4.1) that is defined on the double-cover $\tilde{G}$ should be modified to address possibly non-perfect b-matching. Then Lemmas 4.3-6.3 and Corollary 4.5 hold for the modified LP.
3. The proof of Lemmas 2, 3 in [5] and Lemma 1.3 should be slightly modified. In particular, the alternating path that is constructed in those papers can be different: One can show that if the TMW-b-M $\mathcal{N}^*(T_i^t)$ and the tree-b-matching $\mathcal{H}^*$ choose different sets of edges at the root i, then an alternating path can be constructed in T_i^t (as in [5], [27]) which includes the root i. But endpoints of this alternating path are either leaves of T_i^t or vertices inside T_i^t which have labels from $U(H^*)$ (are un-saturated in G by H^*). In the case in which there is at least one leaf as an endpoint, the same argument as in [5], [27] can be used since the length of the path grows with the depth of computation tree. But in the case in which both endpoints are non-leaf vertices of the computation tree one can show by switching the edges on the path, a better TMW-b-M can be achieved. In particular using the proof of Lemma 3 in [5] we can partition the path to many simple alternating cycles and an alternating simple path. For the cycles, as in [5], switching the edges improves the weight by at least ϵ'. For the simple path, the situation is better here compared to [5]. We can show switching yields a gain of at least ϵ'. This is because both endpoints of the path belong to $U(H^*)$ which means switching the edges of the path (in graph G) yields a valid b-matching which should have larger weight than W_{H^*} by at least ϵ'.

6. Analysis of the Asynchronous BP. In this section we study the asynchronous version of the BP algorithm. The update equations are exactly analogous to the synchronous version, but at each time only a subset of the edges are updated in an arbitrary order. Consider the set $\vec{E}$ of all directed edges in the G; i.e., $\vec{E} = \{(i \to j) \;\; s.t. \;\; i \neq j \in V\}$. Let A be a sequence $\vec{E}(1), \vec{E}(2), \ldots$ of subsets of the set $\vec{E}$. Then the asynchronous BP algorithm corresponding to the sequence A can be obtained by modifying only the step (3) of the algorithm Sync-BP for the perfect b-matchings:

(3) For $t \geq 1$, messages in iteration t are obtained from messages in iteration $t-1$ recursively as follows:

$$m_{i\to j}(t) = \begin{cases} w_{ij} - b_i^{th}\min_{\ell \in N(i)\setminus\{j\}}\Big[m_{\ell\to i}(t-1)\Big] & \text{if } (i\to j)\in \vec{E}(t) \\ m_{i\to j}(t-1) & \text{otherwise} \end{cases}$$

REMARK 3. *This is the most general form of the asynchronous BP and it includes the synchronous version ($\vec{E}(t) = \vec{E}$ for all $t = 1, 2, \ldots$) as a special case. In many applications, a special case of the asynchronous BP is used for which each set $\vec{E}(t)$ consists of a single element.*

We assume that the sequence A of the updates does not have *redundancies*. That is, no edge direction $(i \to j) \in \vec{E}$ is re-updated before at least one of its incoming edge directions ($(\ell \to i)$ for $\ell \in N(i)\setminus\{j\}$) is updated. More formally, if $(i \to j) \in \vec{E}(t) \cap \vec{E}(t+s)$ and $(i \to j) \notin \cup_{r=1}^{s-1}\vec{E}(t+r)$, then at least for one $\ell \in N(i)\setminus\{j\}$, we should have $(\ell \to i) \in \cup_{r=1}^{s-1}\vec{E}(t+r)$.

Let us denote the above algorithm by Async-BP. We claim that, if each edge direction $(i \to j) \in \vec{E}$ is updated $\theta(n)$ times, then the same result as Theorem 3.1 can be proved here. That is, let $u(t)$ be the minimum number of times that an edge direction of the graph G appears in the sequence $\vec{E}(1), \ldots, \vec{E}(t)$; i.e.,

$$u(t) = \min_{(i\to j)\in\vec{E}} \Big(\Big| \big\{ \ell : \quad s.t. \;\; 1 \le \ell \le t \quad \text{and} \quad (i \to j) \in \vec{E}(\ell) \big\} \Big| \Big).$$

From the definition, $u(t)$ is a non-decreasing function of t. We claim that the following result holds:

THEOREM 6.1. *Assume that the LP relaxation (2.1) has no fractional solution. Then the algorithm Async-BP converges to M^* after at most t iterations, provided $u(t) > \frac{2nw^*}{\epsilon}$.* Before proving the above theorem let us define the notion of generalized computation tree for the asynchronous version of the BP algorithm.

6.1. Generalized Computation Tree for the Asynchronous BP. In order to define the generalized computation tree (GCT) for the asynchronous BP, we will begin with some definitions. For any $(i \to j) \in \vec{E}(t)$, define $R^t_{i\to j}$ to be the *computation branch* of i to j at time t which is a weighted rooted tree (not necessarily a balanced rooted tree) and recursively defined according to the following rules:

(a) The root has label j.

(b) The root has only one child which has label i.

(c) If $t = 0$, then the child i has no child ($R^0_{i\to j}$ is just a single edge $\{i, j\}$).

(d) For $t > 0$, if $(i \to j) \notin \vec{E}(t)$ then $R^t_{i\to j} = R^{t-1}_{i\to j}$. Otherwise the child i has $\deg_G(i) - 1$ children which have all of labels in the set $N(i)\backslash\{j\}$ and for any child r of i the subtree that consists of all descendants of r and the edge $\{r, i\}$ to i is $R^{t-1}_{r\to i}$.

The edge between nodes labeled i, j in the tree is assigned weight w_{ij} for $1 \le i < j \le n$. Now for any vertex $i \in V$ and any t, the GCT R^t_i is a weighted rooted tree with root i such that all its branches starting from the root are the computation branches $R^t_{r\to i}$ for all $r \in N(i)$. Since the GCT R^t_i is not necessarily balanced, we will define its *depth* to be the length of the shortest path from the root i to a leaf and denote it by $d(R^t_i)$.

Similarly to the Section 4.1, we can define the minimum weight perfect tree-b-matching (TMWP-b-M) for GCT R^t_i and denote it by $\mathcal{M}^*(R^t_i)$. Moreover, arguments similar to the ones in the Section 4.1 show that the algorithm Async-BP is solving the TMWP-b-M for GCTs R^t_i. In other words, the following corollary holds:

COROLLARY 6.2. *The algorithm Async-BP solves the TMWP-*b*-M problem on the GCT. In particular, for each vertex i of the G, the set $E_i(t)$ that was chosen at the end of iteration t by Async-BP is exactly the set of b_i edges that are attached to the root in TMWP-*b*-M of R^t_i.*

6.2. Technical Analysis of the Asynchronous BP. Now we can use the same analysis as in Section 8.3 to show that if the depth of the generalized computation tree (GCT) is large enough, then for any vertex i, its neighbors in M^* (MWP-b-M of G) are exactly those children that are selected in $\mathcal{N}^*(R^t_i)$ (TMWP-b-M of R^t_i). We will show this by relating the function $u(t)$ to the depth of the GCT Here is the main lemma which summarizes the above claim:

LEMMA 6.3. *If the LP relaxation (2.1) has no fractional solution, then for any vertex i of G and for any t such that $u(t) > \frac{2nw^*}{\epsilon}$, the set of edges that are adjacent to root i in $\mathcal{N}^*(R^t_i)$ are exactly those edges that are connected to i in M^*.* The proof

of Lemma 6.3 is similar to the proof of Lemma 3 from [5] or Lemma 1.3 from [27], with the following slight modifications:

(i) One can construct alternating paths P_ℓ in the same way as before for $1 \le \ell \le d(R_i^t)$.

(ii) The depth of the GCT R_i^t is related to $u(t)$ according to the following lemma:

LEMMA 6.4. *For any vertex $i \in V$ and any t, the depth of any computation branch at time t is at least $u(t)$; i.e., $d(R^t_{i\to j}) \ge u(t)$.* This tells us $d(R_i^t) \ge u(t) > \frac{2nw^*}{\epsilon}$. Applying this to the path $P_{d(R_i^t)}$, analogously to the use of Lemma 3 in [5], gives us the proof of Lemma 6.3. Therefore all that is needed is a proof of Lemma 6.4.

Proof. [Proof of Lemma 6.4] The proof follows easily by looking at the construction of the computation branch. Each computation branch $R^t_{i\to j}$ grows at time t if $(i \to j) \in \vec{E}(t)$. And if this is the case, the depth increases by at least one due to the "no redundancy condition" on the updating sequence. So if each edge is updated at least $u(t)$ times then the depth of its computation branch grows by at least $u(t)$. □

Finally we note that the same algorithm as Async-BP and the same result as Theorem 6.1 can be stated and proved for the (possibly non-perfect) b-matchings as well.

7. Solving LP with BP. In previous sections, we have shown correctness of the BP algorithm provided that the LP relaxation has no fractional solution. In this section we show that with a slight perturbation of the weights, one can solve the LP relaxation with BP. We will do this for all possibilities for the LP solutions (i.e., when LP has fractional solutions or when the integer optimum is non-unique).

We can assume that the LP (4.1) of the double cover $\tilde{G}$ has a unique optimum which is a corner of its polytope. Otherwise, there exist a small enough $\lambda > 0$ such that replacing each $w_{i_a j_b}$ for $(a,b) \in \{(1,2),\ (2,1)\}$ with $w_{i_a j_b} + \varepsilon_{i_a j_b}$ where $\{\varepsilon_{i_a j_b}\}_{(i_a,j_b)\in E(\tilde{G})}$ is a set of iid random variables $\sim U[0,\lambda]$, the updated LP (4.1) has a unique optimum y^* with probability one that is also an optimum to (4.1) with the original weights. Denote such a perturbation of $\tilde{G}$ by $\tilde{G}_\lambda$. Note that for any λ the optimum of $\tilde{G}_\lambda$ is unique with probability one. However, the new optimum might not be an optimum of $\tilde{G}$. Choosing λ to be smaller than $\tilde{\epsilon}/n$ where $\tilde{\epsilon} > 0$ is the gap between the optimum cost in $\tilde{G}$ (might be achieved by several solutions) and the second optimum cost in $\tilde{G}$ (that is strictly less than the optimum cost) guarantees that y^* is also an optimum solution for $\tilde{G}$.

Next, we can use the algorithm Sync-BP (or Async-BP) on such $\tilde{G}_\lambda$ to find y^*. Hence, with probability one, we find an optimum solution x^* to the desired LP relaxation (2.1) defined by $x^*_{ij} = (y^*_{i_1 j_2} + y^*_{i_2 j_1})/2$ for all $\{i,j\} \in E$.

To summarize, we have just shown the following result.

THEOREM 7.1. *There exist a small $\lambda > 0$ such that by running the algorithm Sync-BP (Async-BP) on the double cover $\tilde{G}_\lambda$, with probability one, an optimum solution to LP* (2.1) *can be found in finite number of iterations.*

REMARK 4. *The algorithm proposed above, converges in expected $\lfloor 2n(w^* + \lambda)/\epsilon(\lambda) \rfloor$ number of iterations where $\epsilon(\lambda)$ is the gap between the optimum cost and second optimum cost in $\tilde{G}_\lambda$. However $\epsilon(\lambda)$ is a difficult quantity to calculate. An alternative is to aim for a fully polynomial approximate solution (FPRAS) with a more quantifiable bound. In particular, using the result of [24] (algorithm AS($\mathcal{MCF}, \delta$) in Section 7 of [24]) one can use BP to find an FPRAS for $\tilde{G}$ in expected $O(|V|^8|E|^8 \log(|E|)/\delta^3)$ number of operations. Denoting the output of this FPRAS by $\bar{\mathrm{y}}$, we obtain an FPRAS with BP for the LP relaxation* (2.1) *by defining x^* to be $x^*_{ij} = (\bar{y}_{i_1 j_2} + \bar{y}_{i_2 j_1})/2$ for all $\{i,j\} \in E$.*

REMARK 5. *Running the BP algorithm on $\tilde{G}_\lambda$ is the same as running the BP on the original graph G but for each edge $(i,j) \in E(G)$ its weight in odd (even) iterations of the algorithm is equal to $w_{ij} + \varepsilon_{i_1,j_2}$ ($w_{ij} + \varepsilon_{i_2,j_1}$) or vice versa depending on the initialization.* Theorem 7.1 and provides some intuition for an often observed behavior of the BP algorithm in practice: BP converges to the correct values for the integer edges and oscillates for the fractional edges of the LP solution. This can be loosely explained (using Corollary 4.5 and Theorem 7.1) as follows: When x^*_{ij} is integer, since $y^*_{i_1 j_2} = y^*_{i_2 j_1}$ then BP (when is used on G) converges for the edge $\{i,j\}$. This is because BP (on $\tilde{G}$) selects both $\{i_1, j_2\}$ and $\{i_2, j_1\}$. Similarly, when x^*_{ij} is fractional, since $y^*_{i_1 j_2} = 1 - y^*_{i_2 j_1}$ then BP (when is used on G) oscillates between selecting and not selecting the edge $\{i,j\}$. This is because BP (on $\tilde{G}$) selects exactly one of $\{i_1, j_2\}$ and $\{i_2, j_1\}$.

8. Alternative analysis of the algorithms in Sections 3-6. For the proofs of this section first we define the dual of LP relaxation (2.1).

$$\begin{array}{ccc|ccc} \min & \sum_{\{i,j\}\in E} x_{ij} w_{ij} & | & \max & \sum_{i=1}^{n} b_i y_i - \sum_{\{i,j\}\in E} \lambda_{ij} & \\ & & | & & & \\ \text{s.t.} & & | & \text{s.t.} & & \\ \forall\ i & \sum_{j\in N(i)} x_{ij} = b_i & | & \forall\ \{i,j\} \in E & w_{ij} + \lambda_{ij} \geq y_i + y_j & (8.1) \\ \forall\ \{i,j\} \in E & 0 \leq x_{ij} \leq 1 & | & \forall\ \{i,j\} \in E & \lambda_{ij} \geq 0 & \\ & & | & & & \\ & & | & & & \\ & \text{Primal LP} & | & & \text{Dual LP} & \end{array}$$

Next, we need to define the complementary slackness conditions.

8.1. Complementary Slackness Conditions.. Complementary slackness conditions for the LP and its dual state that the variables $\mathrm{x}^* = (x^*_{ij})$ and $\mathrm{y}^* = (y^*_i)$, $\lambda^* = (\lambda^*_{ij})$ are optimum solutions to the LP relaxation and its dual (8.1), respectively, if and only if

(CS-i) For all edges $\{i,j\}$ of G: $x^*_{ij}(w_{ij} + \lambda^*_{ij} - y^*_i - y^*_j) = 0$.
(CS-ii) For all edges $\{i,j\}$ of G: $(x^*_{ij} - 1)\lambda^*_{ij} = 0$.

See [10], [46] for more information about LP, dual LP and complementary slackness conditions.

Using the fact that the LP has no fractional solution, one can deduce the following modified complementary slackness conditions: For all $\{i,j\} \in M^*$; $w_{ij} + \lambda^*_{ij} = y^*_i + y^*_j$ and for all $\{i,j\} \notin M^*$; $\lambda^*_{ij} = 0$.

By these conditions and the fact that $\lambda^*_{ij} \geq 0$, we have that $w_{ij} \leq y^*_i + y^*_j$ for all $\{i,j\} \in M^*$, and $w_{ij} \geq y^*_i + y^*_j$ for all $\{i,j\} \notin M^*$. However, it is in general not true that these inequalities are strict even when the LP has no fractional solution. Let S be the set of those edges in G for which $|w_{ij} - y^*_i - y^*_j| > 0$. We will assume the minimum gap is ϵ'', i.e., $\epsilon'' = \min_{\{i,j\}\in S} |w_{ij} - y^*_i - y^*_j| > 0$. Throughout this paper we assume that there exist an edge in G for which the strict inequality $|w_{ij} - y^*_i - y^*_j| > 0$ holds and therefore $\epsilon'' > 0$ is well defined. The other cases, where for each $\{i,j\} \in E$ the equality $w_{ij} = y^*_i + y^*_j$ holds, happen only for special cases and are discussed in Section 8.4. Let also $L = \max_{1\leq i\leq n} |y^*_i|$.

Now we are ready to state the analogous version of Theorem 3.1 and its proof.

THEOREM 8.1. *Assume that the LP relaxation (2.1) has no fractional solution. Then the algorithm Sync-BP converges to M^* after at most $\lceil \frac{2nL}{\epsilon''} \rceil$ iterations.*

A crucial part of all the proofs in 8 is the following lemma.

8.2. Main Technical Lemma. In this section we state our main technical lemma which connects the complementary slackness conditions from Section 8.1 to paths on the graph G and on the computation tree. This lemma is a key step in our proof. Its proof is quite delicate, and provides the connection between the absence of fractional solutions and the correctness of BP.

DEFINITION 8.2. *A path $P = (i_1, i_2, \ldots, i_k)$ in G is called an* alternating path *if:*

(a) There exist a partition of edges of P into two sets A, B such that either ($A \subset M^*$, $B \cap M^* = \emptyset$) or ($A \cap M^* = \emptyset$, $B \subset M^*$). Moreover A (B) consists of all odd (even) edges; i.e., $A = \{(i_1, i_2), (i_3, i_4), \ldots\}$ ($B = \{(i_2, i_3), (i_4, i_5), \ldots\}$).

(b) The path P might intersect itself or even repeat its own edges but no edge is repeated immediately. That is, for any $1 \leq r \leq k-2$: $i_r \neq i_{r+1}$ and $i_r \neq i_{r+2}$.

P is called an alternating cycle *if $i_1 = i_k$.*

LEMMA 8.3. *Assume that the LP relaxation (2.1) has no fractional solution. Then for any alternating path P of length at least $2n$, there exists an edge $\{i, j\} \in P$ such that the inequality $|w_{ij} - y_i^* - y_j^*| > 0$ holds. That is, $P \cap S \neq \emptyset$.*

Proof. Before diving into a detailed proof, lets give a high-level perspective. Assume the contrary, that is for all edges in P the equality $w_{ij} - y_i^* - y_j^* = 0$ holds. Then, we will show that if there is an even-length cycle in P, then one can construct a different b-matching than M^* which has the same weight which would contradict the uniqueness of M^*. And, if there is no even-length cycle in P, then there exist a bicycle in P (disjoint union of two odd-length alternating cycles that are joined with an odd-length alternating path). In this case, we construct a fractional optimum to the LP (2.1) that is equal to x^* outside of the bicycle, is equal to 0.5 on the edges of the two alternating odd cycles, and is equal to $1 - x^*$ on the edges of the odd path in the bicycle. This would contradict the assumption on the LP (2.1).

Here is the detailed version. Consider two cases:

Case I) Existence of an even simple cycle in P.

Consider the subgraph of G that is generated by edges and vertices of P. If this subgraph contains an alternating cycle C that does not intersect itself (simple cycle) and has even length, then we will show that $C \cap S \neq \emptyset$. Let $C = (j_1, \ldots, j_{2\ell}, j_1)$. Without loss of generality assume that odd edges belong to M^* and even edges do not. That is, for all $1 \leq r \leq \ell$:

$$\{i_{2r-1}, i_{2r}\} \in M^* \quad , \quad \{i_{2r}, i_{2r+1}\} \notin M^*$$

where $j_{2\ell+1} = j_1$. To prove $C \cap S \neq \emptyset$, assume the contrary; that is, assume for all edges $\{i, j\}$ of C : $w_{ij} = y_i^* + y_j^*$. The weight of M^*-edges of C is equal to weight of their complement in C, due to the fact that

$$\sum_{r=1}^{\ell} w_{j_{2r} j_{2r+1}} = \sum_{s=1}^{2\ell} y_r^* = \sum_{r=1}^{\ell} w_{j_{2r-1} j_{2r}}.$$

Now one can obtain a perfect b-matching M' in G which is different from M^* and has the same weight as M^*. This can be done by defining $M' = M^*$ outside cycle C, and $M' = C \backslash M^*$ on cycle C. However, this contradicts the uniqueness assumption for MWP-b-M in G which holds due to the fact that the LP relaxation has no fractional solution. Hence we are done.

Case II) There is no even simple cycle in P.

Let $P = \{i_1, i_2, \ldots, i_k\}$. Since P has length at least $2n$, it must repeat a vertex. We also add a natural direction to each edge $\{i_j, i_{j+1}\}$ that is from i_j to i_{j+1}. Consider the first vertex that is revisited by starting from i_1 and walking along P. That is, consider the smallest numbers r, s such that $1 \leq r < s \leq n+1$ and $i_r = i_s$. Now we break P into three connected pieces as follows:

(i) Simple path $P_0 = (i_1, i_2, \ldots, i_r)$ (this part will be ignored).
(ii) Simple cycle $C_1 = (i_r, i_{r+1}, \ldots, i_s)$.
(iii) Path $P_1 = (i_{s+1}, i_{s+1}, \ldots, i_k)$.

From now on we are going to assume that path P_0 does not even exist. Basically we will show that there is one edge from S which is in $C_1 \cup P_1$. Since we assumed that P has no even simple cycle, it follows that C_1 has odd length ($s - r$ is odd). Since the length of P is at least $2n$, it follows that P has to intersect itself at least twice and there must be another vertex that is revisited after i_r. Consider the smallest numbers r', s' such that $r \leq r' < s' \leq k$ and $i_{r'} = i_{s'}$. Denote this new simple cycle by C_2; i.e., $C_2 = (i_{r'}, i_{r'+1}, \ldots, i_{s'})$. Again since C_2 is an alternating path, it has to have odd length ($s' - r'$ is odd).

Now we claim that $s \leq r'$. Again assume the contrary, that $r < r' < s$. We obtain a contradiction by finding an even simple cycle in P. Break path C_1 in two simple paths $Q_1 = (i_r, i_{r+1}, \ldots, i_{r'})$ and $Q_2 = (i_{r'}, i_{r'+1}, \ldots, i_s)$, and define the simple path $Q_3 = (i_s, i_{s+1}, \ldots, i_{s'})$. Now consider the simple cycle $C_3 = Q_1 \cup Q_3$. The length of C_3 is equal to $r' - r + s' - s$, which has the same parity as $s - r + s' - r'$, which is even. Therefore C_3 is an even cycle. Moreover, the fact that the parities of r' and s' are different guarantees the alternation of adjacent edges $\{i_{r'-1}, i_{r'}\}$ and $\{i_{s'-1}, i_{s'}\}$ in cycle C_3. Similarly the difference in parity between r and s implies alternation of adjacent edges $\{i_r, i_{r+1}\}$ and $\{i_s, i_{s+1}\}$ in cycle C_3. Thus C_3 is an even length alternating simple cycle, which is a contradiction. So the claim $s \leq r'$ is proved.

Now we are left with a final possibility which uses the integrality of the LP optimum solution. Consider the following three pieces of path P:

(i) Simple odd cycle C_1.
(ii) Simple path $P_2 = (i_{s+1}, i_{s+1}, \ldots, i_{r'})$ (could be only a point).
(iii) Simple odd cycle C_2.

If $(C_1 \cup P_2 \cup C_2) \cap S = \emptyset$, this means that for all edges $\{i, j\} \in C_1 \cup P_2 \cup C_2$, the equality $w_{ij} = y^*_i + y^*_j$ holds. We will reach a contradiction by showing the existence of an optimum fractional solution for LP relaxation (2.1). This is done by defining x' as follows:

$$\forall\ \{i,j\} \in E: \qquad x'_{ij} = \begin{cases} x^*_{ij} & \text{if } \{i,j\} \notin C_1 \cup P_2 \cup C_2 \\ 1 - x^*_{ij} & \text{if } \{i,j\} \in P_2 \\ 0.5 & \text{if } \{i,j\} \in C_1 \cap C_2. \end{cases}$$

First we need to show that x' is a feasible solutions for the LP. For this, all we need to show is that x' satisfies the same local constraints as x^* on vertices of $C_1 \cup P_2 \cup C_2$. Since all $C_1 \cup P_2 \cup C_2$ is a connected alternating path, then for all vertices $i_\ell \in C_1 \cup P_2 \cup C_2$ ($\ell \notin \{r, s, r', s'\}$) it is clear that $x'_{(\ell-1)\ell} + x'_{\ell(\ell+1)} = x^*_{(\ell-1)\ell} + x^*_{\ell(\ell+1)} = 1$. For $\ell = r$, using the fact that length of C_1 is odd and path $C_1 \cup P_2$ is an alternating sub-path of P, either $x^*_{r(r+1)} = x^*_{(s-1)s} = 1,\ x^*_{s(s+1)} = 0$ or $x^*_{r(r+1)} = x^*_{(s-1)s} = 0,\ x^*_{s(s+1)} = 1$, which leads to $x'_{r(r+1)} = x'_{(s-1)s} = 0.5,\ x'_{s(s+1)} = 1$ or $x'_{r(r+1)} = x'_{(s-1)s} = 0.5,\ x'_{s(s+1)} = 0$, respectively. In both cases, x' satisfies same local constraint as x^* at i_r. A similar argument holds at $i_{r'}$.

Next we show that x′ has the same cost as x*. This is done by applying the equality $w_{ij} = y_i^* + y_j^*$ to all edges of $C_1 \cup P_2 \cup C_2$ as follows:

$$\begin{aligned}\sum_{\{i,j\}\in C_1\cup P_2\cup C_2} w_{ij}x^*_{ij} &= \sum_{i\in C_1\cup P_2\cup C_2} y_i^* + x^*_{i_r i_{r+1}} y^*_{i_r} + x^*_{i_{r'} i_{r'+1}} y^*_{i_{r'}} \\ &= \sum_{i\in C_1\cup C_2} y_i^* + \sum_{\{i,j\}\in P_2} w_{ij}x'_{ij} \\ &= \sum_{\{i,j\}\in C_1\cup C_2\cup P_2} w_{ij}x'_{ij}.\end{aligned}$$

This completes the proof of Lemma 8.3. □

8.3. Proof of Theorem 8.1. We will prove Theorem 8.1, using a similar argument to the proof in Section 4.3. In other words we show that if the LP relaxation (2.1) has no fractional solution and hence M^* is unique, then Sync-BP converges to the correct MWP-b-M. We will do this by showing that if the depth of computation tree is large enough, then for any vertex i, its neighbors in M^* (MWP-b-M of G) are exactly those children that are selected in $\mathcal{N}^*(T_i^t)$ (TMWP-b-M of T_i^t). Here is the main lemma that summarizes the above claim:

LEMMA 8.4. *If the LP relaxation (2.1) has no fractional solution, then for any vertex i of G and for any $t > \frac{2nL}{\epsilon''}$, the set of edges that are adjacent to root i in $\mathcal{N}^*(T_i^t)$ are exactly those edges that are connected to i in M^*.* Before entering into the details of the proof here is a high level overview of the underlying argument. Consider the computation tree (T_i^t) rooted at vertex i and look at $\mathcal{N}^*(T_i^t)$. We will assume that the claim of the lemma does not hold. That is, we assume that at the root, $\mathcal{N}^*(T_i^t)$ does not choose the same edges as M^*-edges adjacent to i. Then we use the property of perfect tree-b-matchings, namely that each non-leaf vertex j is connected to exactly b_j of its neighbors, to construct a new perfect tree-b-matching on the computation tree. This new perfect tree-b-matching is going to have less total weight if the depth of the computation tree is large enough. This last step uses an alternating path argument which is a highly non-trivial generalization of the technique of [5] for the case of perfect 1-matching in bipartite graphs. For this part we will use the solutions to the dual LP (8.1).

Proof. [Proof of Lemma 8.4] Let us denote the lifting of a perfect b-matching M^* to a perfect tree-b-matching on T_i^t by $\mathcal{M}^*$. That is, $\mathcal{M}^*$ consists of all edges of the computation tree with endpoint labels i or j such that $\{i,j\} \in M^*$ as an edge in G. The goal is to show that $\mathcal{N}^*(T_i^t)$ and $\mathcal{M}^*$ have the same set of edges at the root of the computation tree. To lighten the notation, we denote the TMWP-b-M of T_i^t by $\mathcal{N}^*$.

Assume the contrary, that there exist children i_{-1}, i_1 of root i such that $\{i, i_1\} \in \mathcal{M}^* \backslash \mathcal{N}^*$ and $\{i, i_{-1}\} \in \mathcal{N}^* \backslash \mathcal{M}^*$. Since both $\mathcal{M}^*$, $\mathcal{N}^*$ are perfect tree-b-matchings, they have b_{i_1} edges connected to i_1. Therefore there exist a child i_2 of i_1 such that $\{i_1, i_2\} \in \mathcal{N}^* \backslash \mathcal{M}^*$. Similarly there is a child i_{-2} of i_{-1} such that $\{i_{-1}, i_{-2}\} \in \mathcal{M}^* \backslash \mathcal{N}^*$. Therefore we can construct a set of alternating paths P_ℓ, $\ell \geq 0$, in the computation tree, that contain edges from $\mathcal{M}^*$ and $\mathcal{N}^*$ alternatively defined as follows. Let $i_0 =$ root i and $P_0 = (i_0)$ be a single vertex path. Let $P_1 = (i_{-1}, i_0, i_1)$, $P_2 = (i_{-2}, i_{-1}, i_0, i_1, i_2)$ and similarly for $r \geq 1$, define P_{2r+1} and P_{2r+2} recursively as follows:

$$P_{2r+1} = (i_{-(2r+1)}, P_{2r}, i_{2r+1}) \quad , \quad P_{2r+2} = (i_{-(2r+2)}, P_{2r+1}, i_{2r+2})$$

where $i_{-(2r+1)},\ i_{2r+1}$ are nodes at level $2r+1$ such that $\{i_{2r}, i_{2r+1}\} \in \mathcal{M}^* \backslash \mathcal{N}^*$ and $\{i_{-2r}, i_{-(2r+1)}\} \in \mathcal{N}^* \backslash \mathcal{M}^*$. Similarly $i_{-(2r+2)},\ i_{2r+2}$ are nodes at level $2r+2$ such that $\{i_{2r+1}, i_{2r+2}\} \in \mathcal{N}^* \backslash \mathcal{M}^*$ and $\{i_{-(2r+1)}, i_{-(2r+2)}\} \in \mathcal{M}^* \backslash \mathcal{N}^*$. Note that, by definition, such paths P_ℓ for $0 \le \ell \le t$ exist since the tree T_i^t has $t+1$ levels and can support a path of length at most $2t$ as defined above. Now consider the path P_t of length $2t$. It is an alternating path on the computation tree with edges from $\mathcal{M}^*$ and $\mathcal{N}^*$. Let us refer to the edges of $\mathcal{M}^*$ ($\mathcal{N}^*$) as the $\mathcal{M}^*$-edges ($\mathcal{N}^*$-edges) of P_t.

We will now modify the perfect tree-b-matching $\mathcal{N}^*$ by replacing all $\mathcal{N}^*$-edges of P_t with their complement in P_t (i.e., $\mathcal{M}^*$-edges of P_t). It is straightforward that this process produces a new perfect tree-b-matching $\mathcal{N}'$ in T_i^t.

Consider the following lemma (proof follows bellow):

LEMMA 8.5. *The weight of the perfect tree-*b*-matching $\mathcal{N}'$ is strictly less than that of $\mathcal{N}^*$ on T_i^t.* This completes the proof of Lemma 8.4 since Lemma 8.5 shows that $\mathcal{N}^*$ is not the minimum weight perfect tree-b-matching on T_i^t, leading to a contradiction. □ Now, we provide the proof of Lemma 8.5.

Proof. [Proof of Lemma 8.5] It suffices to show that the total weight of the $\mathcal{N}^*$-edges of P_t is more than the total weight of $\mathcal{M}^*$-edges of P_t. For each vertex $i_r \in P_t$ consider the value $y^*_{i_r}$ from the optimum solution to the dual LP (8.1). Using the inequality $w_{ij} \le y_i^* + y_j^*$ for edges of $\mathcal{M}^*$, we obtain:

$$\sum_{\{i,j\} \in P_t \cap \mathcal{M}^*} w_{ij} \le \left(\sum_{r=-t}^{t} y^*_{i_r} \right) - y^*_{i_{(-1)^t t}} - k_1 \epsilon'' \tag{8.2}$$

where k_1 is the number of $\mathcal{M}^*$-edges of P_t that belong to S, i.e., the number of $\mathcal{M}^*$-edges of P_t endowed with the strict inequality $w_{ij} \le y_i^* + y_j^*$, with a gap of at least ϵ''. On the other hand, using the inequality $w_{ij} \ge y_i^* + y_j^*$ for edges of $\mathcal{N}^*$ we have:

$$\sum_{\{i,j\} \in P_t \cap \mathcal{N}^*} w_{ij} \ge \left(\sum_{r=-t}^{t} y^*_{i_r} \right) - y^*_{i_{(-1)^{t+1} t}} + k_2 \epsilon'' \tag{8.3}$$

where now k_2 is number of $\mathcal{N}^*$-edges of P_t that belong to S, or equivalently the number of times the inequality $w_{ij} \ge y_i^* + y_j^*$ is strict with a gap of at least ϵ''. One finds

$$\begin{aligned} \sum_{\{i,j\} \in P_t \cap \mathcal{N}^*} w_{ij} - \sum_{\{i,j\} \in P_t \cap \mathcal{M}^*} w_{ij} \ &\ge\ y^*_{i_{(-1)^t t}} - y^*_{i_{(-1)^{t+1} t}} + (k_1 + k_2)\epsilon'' \\ &\overset{(a)}{\ge} (k_1 + k_2)\epsilon'' - 2L \overset{(b)}{\ge} (k_1 + k_2)\epsilon'' - 2L \overset{(c)}{>} 0 \end{aligned} \tag{8.4}$$

where (a) uses definition of L from Section 8.1 and (b) uses the fact that for all $i, j:\ \lambda_{ij}^* \ge 0$. The main step is (c), which uses Lemma 8.3 as follows. Path P_t has length $2t$, and each continuous piece of it with length $2n$ has a projection to the graph G which satisfies the conditions of Lemma 8.3. This means the path has at least one edge from the set S. Thus $(k_1 + k_2) \ge \frac{2t}{2n} > \frac{2L}{\epsilon''}$. This completes the proof of Lemma 8.5. □

REMARK 6. *The proof of Section 4.4 on the independence of Sync-BP's convergence and correctness from the initial conditions carries over here as well. This is done by* re-defining L *according to:* $L = \max_{1 \le i \le n} |y_i^*| + \max_{\{i,j\} \in E} |m_{i \to j}(0)|$ *and noting that, in the proof of Lemma 8.4, the only place where the weight of leaf edges appears is the inequality* (a) *in equation (8.4), which will be satisfied by new definition of* L.

8.4. Sync-BP is Correct When ϵ'' is Not Well-Defined. Recall from the discussion given above about the complementary slackness conditions that, if for all edges $\{i,j\} \in E$ the equality $w_{ij} = y_i^* + y_j^*$ holds, then ϵ'' is not well defined. In this section we show that these rare cases do not cause any trouble. We will show that the condition $t > \frac{2nL}{\epsilon''}$ in the main theorem can be replaced by $t > n$. This is shown by proving the following lemma instead of Lemma 8.4.

LEMMA 8.6. *If the LP relaxation (2.1) has no fractional solution, then, for any vertex i of G and for any $t > n$, the set of edges that are adjacent to root i in $\mathcal{N}^*(T_i^t)$ are exactly those edges that are connected to i in M^*.*

Proof. The proof is similar to the proof of Lemma 8.4. If after iteration t, the claim of the Lemma 8.6 does not hold, then the alternating path P_t can be constructed as before. Now since the length of P_t is greater than $2n$, one can use the technical Lemma 8.3 for the projection of the path P_t onto G to show that the strict inequality $|w_{ij} - y_i^* - y_j^*| > 0$ happens for at least one edge. This contradicts the above assumption at the beginning of the Section. Therefore Lemma 8.6 is true. □

8.5. Modifications for the possibly non-perfect matchings. Similar to the perfect matching case the LP relaxation and its dual are:

$$\begin{array}{ccc|ccc} \min & \sum_{\{i,j\}\in E} x_{ij} w_{ij} & & \max & -\sum_{i=1}^{n} b_i y_i - \sum_{\{i,j\}\in E} \lambda_{ij} \\ & & & & \\ \text{s.t.} & & & \text{s.t.} & \\ \forall\ i & \sum_{j\in N(i)} x_{ij} \le b_i & & \forall\ \{i,j\} \in E & w_{ij} + \lambda_{ij} \ge -y_i - y_j \\ \forall\ \{i,j\} \in E & 0 \le x_{ij} \le 1 & & \forall\ \{i,j\} \in E & \lambda_{ij} \ge 0 \\ & & & \forall\ \{i\} \in V & y_i \ge 0 \\ & & & & \\ & \text{Primal LP} & & & \text{Dual LP} \end{array} \tag{8.5}$$

Complementary slackness now reads, for all $\{i,j\} \in E$: $x_{ij}^*(w_{ij} + \lambda_{ij}^* + y_i^* + y_j^*) = 0$, $(x_{ij}^* - 1)\lambda_{ij}^* = 0$ and for all $i \in V$: $(\sum_{j\in N(i)} x_{ij} - b_i) y_i^* = 0$.

Similarly to the perfect matching case, we can write the following modified complementary slackness condition using the fact that the LP relaxation has no fractional solution:

(CS'-i) For all $\{i,j\} \in H^*$; $\quad w_{ij} + \lambda_{ij}^* + y_i^* + y_j^* = 0$.
(CS'-ii) For all $\{i,j\} \notin H^*$; $\quad \lambda_{ij}^* = 0$.
(CS'-iii) For all $i \in U(H^*)$; $\quad y_i^* = 0$.

Let S' be set of those edges in G for which $|w_{ij} + y_i^* + y_j^*| > 0$. We will assume the minimum gap is ϵ'''. That is

$$0 < \epsilon''' = \min_{\{i,j\}\in S} |w_{ij} + y_i^* + y_j^*|.$$

The quantity L' is defined similarly to L by $L' = \max_{1\le i\le n} y_i^*$.

The algorithm Sync-BP(2) will remain unchanged and the modified theorem for its convergence and correctness is:

THEOREM 8.7. *Assume that the LP relaxation (8.5) has no fractional solution. Then the algorithm Sync-BP(2) converges to H^* after at most $\lceil \frac{4nL'}{\epsilon'''} \rceil$ iterations.* The proof of Theorem 8.7 is similar to the one of Section 8.3, with the following modifications:

1. The computation tree T_i^t and TMW-b-M are defined as before, while Lemma 4.1 is slightly modified. A careful analysis of W^+ and W^- for the tree-b-matchings yields equations (5.2) for finding TMW-b-M in the computation tree. This is how the new equations are obtained.
2. The technical lemma from Section 8.2 is still true and its proof does not change because the definition of alternating paths is preserved and because all cycles involved in the proof turn out to be adjacent to exactly one edge of H^*.
3. The proof of Lemmas 8.4 and 8.5 should be slightly modified. In particular, the alternating path P_t can be different: One can show that if the TMW-b-M $\mathcal{N}^*(T_i^t)$ and the tree-b-matching $\mathcal{H}^*$ choose different sets of edges at the root i, then an alternating path can be constructed as before in T_i^t which includes the root i. But endpoints of this alternating path P_t are either leaves of T_i^t or vertices inside T_i^t which have labels from $U(H^*)$ (are un-saturated in G by H^*). In the case in which there is at least one leaf as an endpoint of P_t, the same argument as equation (8.4) in Section 8.3 can be used since the length of P_t is at least t. This shows $(k_1 + k_2) \geq \frac{t}{2n} > \frac{2L'}{\epsilon'''}$. But, in the case in which both endpoints of P_t are non-leaf vertices of the computation tree, using condition (CS'-iii), the analogous version of equation (8.4) is as follows:

$$\sum_{\{i,j\} \in P_t \cap \mathcal{N}^*} w_{ij} - \sum_{\{i,j\} \in P_t \cap \mathcal{H}^*} w_{ij} = (k_1 + k_2)\epsilon'''. \tag{8.6}$$

Now all that is needed is to show $k_1 + k_2 > 0$. We will show this by the following extension of technical Lemma 8.3.

LEMMA 8.8. *Assume that the LP relaxation (8.5) has no fractional solution. Then for any alternating path P with endpoints from the set $U(H^*)$, there exists an edge $\{i, j\} \in P$ such that the inequality $|w_{ij} + y_i^* + y_j^*| > 0$ holds. That is, $P \cap S' \neq \emptyset$.*

Proof. For paths P with length at least $2n$, we can use Lemma 8.3, so there is nothing to do. If a subgraph generated by P includes at least two cycles, then the same argument as in the proof of Lemma 8.3 can be used. Therefore we can assume P intersects itself at most once. So P can be written as a union $C \cup P_1$ where C is an odd simple alternating cycle and P_1 is a simple alternating path (either C or P_1 can be empty, but not at the same time). Next, one can define a different solution x' to the LP (8.5) which has the same cost as x^* by defining $\mathrm{x}' = 1 - \mathrm{x}^*$ on path P_1 and setting x' equal to 0.5 on C. x' will still be a feasible solution since the endpoints of P_1 are elements of $U(H^*)$ and the edge adjacent to them in path P_1 is not in H^*. This contradicts the no fractional solution assumption on the LP. □

8.6. Modifications for the Asynchronous BP. The algorithm Async-BP and its analysis will be exactly similar to Section 6 except that the main theorem is slightly modified to:

THEOREM 8.9. *Assume that the LP relaxation (2.1) has no fractional solution. Then the algorithm Async-BP converges to M^* after at most t iterations, provided $u(t) > \frac{2nL}{\epsilon''}$.* Similarly, all the references to Lemma 3 of [27] should be replaced to Lemma 8.4 of this paper.

9. Acknowledgements. We would like to thank László Lovász, Andrea Montanari, Elchannan Mossel and Amin Saberi for useful discussions. We also thank the

anonymous referees for helpful suggestions. This work was done while Bayati and Zecchina were in Microsoft Research, and were supported by the Microsoft Technical Computing Initiative.

REFERENCES

[1] D. Achlioptas, F. Ricci-Tersenghi, "On the solution-space geometry of random constraint satisfaction problems," STOC 2006.

[2] D. Aldous, "The zeta (2) Limit in the Random Assignment Problem," *Random Structures and Algorithms*, Vol. 18, pp. 381-418, 2001.

[3] S. M. Aji, G. B. Horn and R. J. McEliece, "On the Convergence of Iterative Decoding on Graphs with a Single Cycle," in *Proc. IEEE Int. Symp. Information Theory*, 1998, p. 276.

[4] M. Bayati and C. Nair, "A rigorous proof of the cavity method for counting matchings", *Allerton conference on communication, control and computing*, 2006.

[5] M. Bayati, D. Shah, and M. Sharma, "Maximum weight matching via max-product belief propagation," in *IEEE Int. Symp. Information Theory*, 2005.

[6] M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: convergence, correctness and LP duality," in *IEEE Int. Symp. Information Theory*, 2006.

[7] M. Bayati, C. Borgs, J. Chayes and R. Zecchina, "Belief-Propagation for Weighted b-Matchings on Arbitrary Graphs and its Relation to Linear Programs with Integer Solutions", *ArXiv: 0709.1190*, September 2007.

[8] M. Bayati, C. Borgs, J. Chayes and R. Zecchina, "On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs", *J. Stat. Mech.*, (2008) L06001.

[9] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of Operations Research*, vol. 14, 1988.

[10] S. Boyd and L. Vandenberghe, "Convex Optimization", *Cambridge University Press*, 2004.

[11] A. Braunstein and R. Zecchina, "Survey Propagation as local equilibrium equations", *J Stat Mech Theory Experiment (JSTAT)*, p06007, 2004.

[12] A. Braunstein, M. Mezard, and R. Zecchina, "Survey propagation: an algorithm for satisfiability," *Random Structures and Algorithms*, vol. 27, pp. 201–226, 2005.

[13] A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, "Polynomial iterative algorithms for coloring and analyzing random graphs," *Phys. Rev. E* 68, 036702, 2003.

[14] V. Chvatal, "On certain polytopes associated with graphs", *Journal of Combinatorial Theory*, Series B, 13, 138154, 1975.

[15] J. Edmonds and E. Johnson, "Matching: A well-solved class of integer lienar programms," *Combinatorial Structures and their Applications, Calgary International Conference, Gordon and Breach*, 89-92, 1970.

[16] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of ACM*, vol. 18, pp. 264–284, 1972.

[17] U. Feige, E. Mossel and D. Vilenchik, "Complete convergence of message passing algorithms for some satisfiability problems", *In Proceedings of Random 2006*, LNCS 4110 Springer, 339–350, 2006.

[18] J. Feldman, M. Wainwright and D. Karger, "Using linear programming to decode binary linear codes", *IEEE Transactions on Information Theory*, vol. 51, pp. 954-972, 2005.

[19] B. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points", *Science* 315, 972, 2007.

[20] B.J. Frey, R. Koetter, "Exact inference using the attenuated max-product algorithm", *Advanced Mean Field Methods: Theory and Practice*, ed. Manfred Opper and David Saad, MIT Press, 2000.

[21] N. Friedman, "Inferring Cellular Networks Using Probabilistic Graphical Models," *Science*; 303(5659):799-805, Feb 6 2004.

[22] R. G. Gallager, "Low Density Parity Check Codes," Cambridge, MA: MIT Press, 1963.

[23] D. Gamarnik, T. Nowicki and G. Swirscsz, "Maximum Weight Independent Sets and Matchings in Sparse Random Graphs. Exact Results using the Local Weak Convergence Method", *Random Structures and Algorithm*, Vol.28, No. 1, pp. 76-106, 2005.

[24] D. Gamarnik, D. Shah and W. Wei, "Beleif Propagation for Min-cost Network Flow: Convergence & Correctness", *ACM SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[25] I. Gat-Viks, A. Tanay, D. Raijman and R. Shamir, "Factor graph network models for biological systems," *Proc. of RECOMB 2005*, pp. 31-47, Lecture Notes in Bioinformatics 3500, Springer, Berlin, 2005.

[26] A. Gerards, "Matching. Volume 7 of ," *Hand book of Operation Research and Management Science*, Chapter 3, pp. 135-224. North-Holland, 1995.
[27] B. Huang, T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching", *Artificial Intelligence and Statistics (AISTATS)*, March, 2007.
[28] T. Jebara, "MAP Estimation, Message Passing, and Perfect Graphs" *Uncertainty in Artificial Intelligence*, June 2009.
[29] L. Lovasz, "Normal hypergraphs and the weak perfect graph conjecture", *Discrete Math.*, 2, 253267, 1972.
[30] E. Maneva, E. Mossel and M. J. Wainwright., "A New Look at Survey Propagation and its Generalizations," *SODA*, 2005.
[31] E. Marinari, G. Semerjian and V. Van Kerrebroeck, "Finding long cycles in graphs", *Phys. Rev.*, 75, 066708, 2007.
[32] M. Mezard and G. Parisi "Mean-field equations for the matching and travelling Salesman problems," *Eurhophysics letters*, Vol. 2, pp. 913-918, 1986.
[33] M. Mezard and R. Zecchina "Random K-satisfiability: from an analytic solution to a new efficient algorithm," *Phys.Rev. E E*, 66, 056126, 2002.
[34] C. Moallemi and B. Van Roy, "Consensus Propagation," *IEEE Transactions on Information Theory*, Vol. 52, No. 11, pp. 4753-4766, 2006.
[35] C. Moallemi and B. Van Roy, "A Message-Passing Paradigm for Resource Allocation," preprint June 2007.
[36] C. C. Moallemi and B. Van Roy. "Convergence of min-sum message passing for convex optimization", *IEEE Transactions on Information Theory*, 56(4):20412050, 2010.
[37] A. Montanari, B. Prabhakar, and D. Tse., "Belief Propagation Based Multi–User Detection", *Allerton Conference on Communication, Control, and Computing*, 2006.
[38] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.
[39] W. Pulleyblank, "Matchings and extensions." Volume 1 of *Handbook of Combinatorics*, Chapter 3, pp. 179-232, North Holland, 1995.
[40] M. Pretti and M. Weigt, "Sudden emergence of q-regular subgraphs in random graphs", *Europhys. Lett.*, 75 8, 2006.
[41] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity Check Codes under Message-Passing Decoding," *IEEE Trans. Info. Theory*, Vol. 47, pp 599-618, 2001.
[42] P. Rusmevichientong and B. Van Roy, " An Analysis of Belief Propagation on the Turbo Decoding Graph with Gaussian Densities," *IEEE Transactions on Information Theory*, Vol. 47, No. 2, pp. 745-765, 2001.
[43] S. Sanghavi, "Equivalence of LP Relaxation and Max-Product for Weighted Matching in General Graphs", *IEEE Information Theory Workshop*, September 2007.
[44] S. Sanghavi, D. Malioutov, A. Willsky "Linear programming analysis of loopy belief propagation for weighted matching", to appear in *NIPS*, 2007.
[45] S. Sanghavi, D. Shah and A. Willsky, "Message Passing for Max-weight Independent Set", to appear in *NIPS* 2007.
[46] A. Schrijver, "Combinatorial Optimization", Springer-Verlag, 2003.
[47] J. Salez and D. Shah, "Optimality of belief propagation for random assignment problem", *SODA*, 2009.
[48] M. Tappen and W. Freemand, "Graph cuts and belief propagation for stereo, using identical MRF parameters", *ICCV*, 2003.
[49] S. Tatikonda and M. I. Jordan, "Loopy belief propagation and Gibbs measures," *In D. Koller and A. Darwiche (Eds)., Uncertainty in Artificial Intelligence (UAI), Proceedings of the Eighteenth Conference*, 2002.
[50] P.O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," to apprear in *IEEE Trans. Inform. Theory*, http://www.arxiv.org/abs/cs.IT/0512078.
[51] P.O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," *Proc. ISITA* 2004, Parma, Italy, pp. 991–996, Oct. 10-13, 2004.
[52] J. Wastlund, "The traveling salesman problem in the stochastic mean field model", *http://www.mai.liu.se/∼jowas*, 2006.
[53] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree Consistency and Bounds on the Performance of the Max–Product Algorithm and its Generalizations", *Statistics and Computing*, 14, 2004.
[54] M. Wainwright, T. Jaakkola, and A. Willsky, "A new class of upper bounds on the log partition function", *IEEE Transactions on Information Theory*, 51(7):2313-2335, 2005.
[55] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP estimation via agreement on trees:

message-passing and linear programming", *IEEE Transactions on Information Theory*, 51(11):3697:3711, 2005.

[56] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Comput.*, Vol. 12, pp. 1-42, 2000.

[57] Y. Weiss and W. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, Vol. 13, Issue 10, pp 2173-2200, 2001.

[58] Y. Weiss and W. T. Freeman, "On the Optimality of Solutions of the Max–Product Belief–Propagation Algorithm in Arbitrary Graphs", *IEEE Trans. Info. Theory*, 47: 2, 2001.

[59] Y. Weiss, C. Yanover and T. Meltzer "MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies," *UAI*, 2007.

[60] N. Wiberg, "Codes and Decoding on General Graphs", *Ph.D. thesis*, Linköping University, Sweden, 1996.

[61] C. Yanover and Y. Weiss, "Approximate inference and protein folding," *Advances in Neural Processing Systems*, 2002.

[62] J. Yedidia, W. Freeman and Y. Weiss, "Understanding Belief Propagation and its Generalizations," Mitsubishi Elect. Res. Lab., TR-2001-22, 2000.

[63] L. Zdeborová and M. Mézard, "The number of matchings in random graphs", *J. Stat. Mech.*, 2006.