LETTER

On the exactness of the cavity method for weighted $b$-matchings on arbitrary graphs and its relation to linear programs

Mohsen Bayati$^1$, Christian Borgs$^1$, Jennifer Chayes$^1$ and Riccardo Zecchina$^2$

$^1$ Microsoft Research, One Memorial Dr., Cambridge, MA 02142, USA
$^2$ Physics Department, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
E-mail: mohsenb@microsoft.com, borgs@microsoft.com, jchayes@microsoft.com and riccardo.zecchina@polito.it

Received 15 April 2008
Accepted 24 May 2008
Published 23 June 2008

Online at stacks.iop.org/JSTAT/2008/L06001
doi:10.1088/1742-5468/2008/06/L06001

Abstract. We consider the general problem of finding the minimum weight $b$-matching on arbitrary graphs. We prove that, whenever the linear programing relaxation of the problem has no fractional solutions, then the cavity or belief propagation equations converge to the correct solution both for synchronous and asynchronous updating.

Keywords: analysis of algorithms, exact results, message-passing algorithms
Motivated by the cavity method, very fast distributed heuristic algorithms have recently been developed for the solution of random constraint satisfaction problems. In some cases, namely in the replica symmetric scheme, 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 assignment with highest probability of a joint discrete probability distribution defined on a graph. 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, e.g. when the space of solutions clusters into many subsets, the cavity method may lead to a more involved survey propagation (SP) algorithm [1]–[4], in which some form of long range correlation among variables is included in the formalism.

In this letter, we study the problem of finding the minimum weight b-matchings in arbitrary graphs via the min-sum version of the cavity/BP algorithm.

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 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 the $b_j$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. This procedure can be derived as the zero temperature limit of the cavity equations for the minimum b-matching problem. In what follows we will use as the synonymous cavity equation, the belief propagation (BP) or min-sum equations.

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 Linear Programing (LP) relaxation of the problem 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.

Our proof gives new insight of the often-noted but poorly understood connection between the cavity method (or BP) and LP through the dual of the LP relaxation. We also modify our BP algorithm and its analysis to include the problem of finding the non-perfect weighted b-matchings. Independent work on this aspect can be found in [5]. Previous exact results on bipartite graphs was obtained in [6] for $b = 1$ and then was extended to all $b$ in [7].

The weighted b-matching problem is an important problem in combinatorial optimization. For extensive surveys see [8] and [9]. In physics, the study of the random 1-matching problem by the cavity method goes back to the work of Mézard and Parisi [10] who made a celebrated conjecture for the expected optimum weight $(\pi^2/6)$ that was proven to be exact a decade later by Aldous [11]. BP algorithms have been the subject of extensive study in several communities (see [6] and [22] for a detailed survey of rigorous results about BP). Recent works have also suggested a connection between the BP algorithm and linear programing (LP) in particular problems. A relationship between iterative decoding of channel codes and LP decoding was studied in [12]–[14]. Other relationships were noted.
in the context of BP algorithms with convex free energies [15, 16], and in the case of
BP algorithms for resource allocations [17]. For weighted 1-matchings, the connection
was studied [18] in the context of similarities between BP equations and the primal–dual
auction algorithm of Bertsekas [19]. Finally, we note that the BP equations for solving
the weighted matching problem which we use in this letter have been previously studied
in [18, 7]. These equations are also similar (though not identical) to the cavity equations
for weighted matching problems and traveling salesman problems which are found in the
statistical physics literature, see, e.g., [10, 11, 20], and [21].

The main contributions of our results can be summarized as follows

(1) The properties of the BP equations for the weighted matching was first used in [6]
and its correctness and convergence was shown for bipartite graphs with a unique
optimum solution. The same technique was used in [7] to extend the result to b-
matchings in bipartite graphs. But this technique fails for graphs containing cycles
with odd length. In order to bypass this difficulty we use a completely different
tool, complementary slackness conditions of the LP relaxation and its dual, which is
independent of the graph structure.

(2) The connection between LP and the cavity method (or BP) is particularly important
given that LP is a widely used technique for optimization. We show both convergence
and correctness of the BP algorithm when LP relaxation has no fractional solutions,
thus establishing a solid link between the two methods. In some recent work, [16],
which generalizes methods of [15], the connection of the BP algorithm and LP
relaxation is studied in the converged case of the BP. The authors also study
interesting variations of the BP which have convex free energies.

(3) The asynchronous BP, which includes the synchronous version as a special case,
has been a more popular version for practical purposes. But, due to its more
complicated structure, it has not been the subject of much rigorous study. Here we
provide a correctness and convergence proof of asynchronous BP for a combinatorial
optimization problem based on the construction of a generalized computation tree,
which can be used for the analysis of the both convergence and correctness of
asynchronous message-passing algorithms as the cavity method.

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 $N(i)$. Let $b = (b_1, \ldots, b_n)$ be a sequence of positive integers such that
$b_i \leq \text{deg}_G(i)$. 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 $M_G(b)$ ($PM_G(b)$), and assume that it is non-empty. Clearly
$PM_G(b) \subset M_G(b)$.

The weight of a (perfect or non-perfect) b-matching $M$, denoted by $W_M$, is defined
by $W_M = \sum_{i,j} w_{ij} 1_{\{i,j\} \in M}$. In the next two sections, we will restrict ourselves to the case of perfect b-matchings. The analysis is extended to (possibly non-perfect) b-
matchings in [22]. The minimum weight perfect b-matching (b-MWPM), $M^*$, is defined
by $M^* = \arg\min_{M \in PM_G(b)} W_M$. The goal of this letter is to find $M^*$ via a min-sum belief
propagation algorithm. Throughout the letter, we will assume that $M^*$ is unique.

**Linear prograning 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

$$
\begin{align*}
doi:10.1088/1742-5468/2008/06/L06001
\end{align*}
$$
Complementary slackness conditions. Complementary slackness for the LP and its dual statement that the variables $x_i$ and $y_{ij}$ satisfy:

$$\sum_{j \in S(i)} x_{ij} = b_i \quad \forall \; i \in V$$

Relaxing the constraint that $x_{ij}$ is integer, this leads to the following linear program and its dual:

**PrimalLP**

$$\begin{align*}
\min \quad & \sum_{(i,j) \in E} x_{ij} w_{ij} \\
\text{subject to} \quad & \sum_{j \in S(i)} x_{ij} = b_i \quad \forall \; i \in V \\
& 0 \leq x_{ij} \leq 1 \quad \forall \{i,j\} \in E
\end{align*}$$

**DualLP**

$$\begin{align*}
\max \quad & \sum_{i=1}^n b_i y_i - \sum_{(i,j) \in E} \lambda_{ij} \\
\text{subject to} \quad & w_{ij} + \lambda_{ij} \geq y_i + y_j \quad \forall \{i,j\} \in E \\
& \lambda_{ij} \geq 0 \quad \forall \{i,j\} \in E
\end{align*}$$

We say the LP relaxation (1) has no fractional solution if every optimal solution $x$ of LP satisfies $x \in \{0,1\}^{|E|}$. Note that the 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 (1) has no fractional solution.

**Algorithm and main result.** The following algorithm is a synchronous implementation of BP for finding the minimum weight perfect b-matching (b-MWPM). 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. Before applying the BP algorithm, we remove all trivial vertices from the graph. A vertex $i$ is called trivial if $\text{deg}_G(i) = b_i$. 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 $m_{i\rightarrow j}(t)$. 

doi:10.1088/1742-5468/2008/06/L06001
(2) Messages are initialized by \( m_{i \rightarrow j}(0) = w_{ij} \) for all \( \{i, j\} \in E \) (the messages can actually be initialized by any arbitrary values \[22]\).

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

\[
\forall \{i, j\} \in E: \quad m_{i \rightarrow j}(t) = w_{ij} - b^t_{ij} - \min_{t \in N(i) \setminus \{j\}} m_{i \rightarrow t}(t - 1)
\]

(3) where \( k \text{th-min}[A] \) denotes the \( k \)th minimum\(^3\) of set \( A \).

(4) The estimated \( b \)-MWPM at the end of iteration \( t \) is \( M(t) = \bigcup_{i=1}^{n} E_i(t) \) where \( E_i(t) = \{\{i, j_1\}, \ldots, \{i, j_{b_i}\}\} \) is such that \( N(i) = \{j_1, j_2, \ldots, j_{\deg_G(i)}\} \) and \( m_{j_1 \rightarrow i}(t) \leq m_{j_2 \rightarrow i}(t) \cdots \leq m_{j_{\deg(i)} \rightarrow i}(t) \). 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\(^4\).

In lemma 1, we will show the main intuition behind equation (3) and how it is derived. But we note that one can also use the graphical model representations of \[6,7,23\] to obtain the standard BP equations for this problem, which, after some algebraic calculations, yield the recursive equation (3).

The main result of the letter 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 1.** Assume that the LP relaxation (1) has no fractional solution. Then the algorithm Sync-BP converges to \( M^* \) after at most \( \left[2nL/\epsilon\right] \) iterations.

If the LP relaxation (1) has a fractional solution whose cost is strictly less than \( W_{M^*} \), then \[23,5\] have shown for the case of 1-matching that BP does not converge to \( M^* \). It is straightforward to generalize this to perfect \( b \)-matching as well. But for the case in which the LP relaxation has a fractional solution whose cost is equal to \( W_{M^*} \), BP fails in general. This is because the \( b^t_{ij} \) minimum in equation (3) is not unique and one needs an oracle to make the right decision. If such an oracle exists, then BP converges to \( M^* \).

In what follows we first display the connection between the Sync-BP equations and the so-called computation tree. Next we discuss how the complementary slackness conditions is related to alternating paths in the graph \( G \). These results are eventually used 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 \( G \).

**Analysis of the synchronous BP via computation tree.** The main idea behind the algorithm Sync-BP is that it assumes the graph \( G \) has no cycle. In other words, it finds the \( b \)-MWPM of a graph \( G' \) that has the same local structure as \( G \) but no cycles. The precise definition of the computational tree for Sync-BP goes as follows.

**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

\(^3\) Note that the \( b^t_{ij} - \min_{t \in N(i) \setminus \{j\}} \) 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 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.
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 $\text{deg}_G(i)$ children of the root is equal to $N(i)$, and (c) if $s$ is a non-leaf node whose parent has label $r$, then the set of labels of its children is $N(s)\setminus\{r\}$. $T^i_t$ is often called the unwrapped tree at node $i$. The computation tree is a well-known technique for analyzing algorithms and 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. A subtree $M$ 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 $\text{deg}_M(i) = b_i$. Now denote the minimum weight perfect tree-b-matching (b-TMWPM) of the computation tree $T^i_t$ by $N^*(T^i_t)$. The following lemma shows that Sync-BP can be seen as a dynamic programing procedure that finds the minimum weight perfect tree-b-matching over the computation tree. Figure 1 shows a graph $G$ and one of its corresponding computation trees.

**Lemma 1.** The algorithm Sync-BP solves the b-TMWPM problem on the computation tree. In particular, for each vertex $i$ of $G$, the set of $E_i(t)$ which was chosen at the end of iteration $t$ by Sync-BP is exactly the set of $b_i$ edges which are attached to the root in b-TMWPM of $T^i_t$.

Lemma 1 characterizes the estimated b-MWPM, $M(t)$, and will be used later in the proof of the main result.

The following technical lemma is also crucial for the proof of equivalence between BP and LP. It connects the complementary slackness conditions to paths on the graph $G$ and on the computation tree. This lemma provides the connection between the absence of fractional solutions and the correctness of BP.

**Definition 1.** A path $P = (i_1, i_2, \ldots, i_k)$ in $G$ is called an alternating path if it has the following two properties: (i) there exists 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\}), and (ii) the path $P$ might intersect itself or even repeat its own edges but no edge is repeated.

Figure 1. Part (a) shows a graph $G$ where dashed and gray edges represent a 1-matching. Part (b) shows the computation tree $T^2_3$ corresponding to $G$ where the set of dashed and gray edges form a 1-TMWPM.
Proof of lemma 3. Assume that the LP relaxation (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 of theorem 1.** We will prove theorem 1, namely that if the LP relaxation (1) has no fractional solution, then for any vertex $i$ of $G$ and for any $t > 2nL/\epsilon$, the set of edges that are adjacent to root $i$ in $N^*(T^*_i)$ are exactly those edges that are connected to $i$ in $M^*$.

The proof of lemma 3 is the main technical part of this work. The high level overview of the underlying argument goes as follows. Consider the computation tree $(T^*_i)$ rooted at vertex $i$ and look at $N^*(T^*_i)$. We will assume that the claim of the lemma does not hold. That is, we assume that, at the root, $N^*(T^*_i)$ 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 [6] for the case of perfect 1-matching in bipartite graphs. For this part we will use the solutions to the dual LP (1).

**Proof of lemma 3.** Let us denote the lifting of a perfect b-matching $M^*$ to a perfect tree-b-matching on $T^*_i$ by $\mathcal{M}^*$. That is, $\mathcal{M}^*$ consists of all edges of the computation tree with endpoint labels $i, j$ such that $\{i, j\} \in M^*$ as an edge in $G$. The goal is to show that $N^*(T^*_i)$ and $\mathcal{M}^*$ have the same set of edges at the root of the computation tree. To lighten the notation, we denote the b-TMWPM of $T^*_i$ by $N^*$.

Assume the contrary, that there exist children $i_{-1}, i_1$ of root $i$ such that $\{i, i_1\} \in \mathcal{M}^* \setminus N^*$ and $\{i, i_{-1}\} \in N^* \setminus \mathcal{M}^*$. Since both $\mathcal{M}^*$, $N^*$ are perfect tree-b-matchings, they have $b_{i_1}$ edges connected to $i_1$. Therefore there exists a child $i_2$ of $i_1$ such that $\{i_1, i_2\} \in N^* \setminus \mathcal{M}^*$. Similarly there is a child $i_{-2}$ of $i_{-1}$ such that $\{i_{-1}, i_{-2}\} \in \mathcal{M}^* \setminus N^*$. Therefore we can construct a set of alternating paths $P_\ell$, $\ell \geq 0$, in the computation tree, that contain edges from $\mathcal{M}^*$ and $N^*$ alternatively defined as follows. Let $i_0 = \text{root } i$ 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: $P_{2r+1} = (i_{-(2r+1)}, P_{2r}, i_{2r+1})$, $P_{2r+2} = (i_{-(2r+2)}, P_{2r+1}, i_{2r+2})$ where $i_{-(2r+1)}$, $i_{2r+1}$ are nodes at level $2r + 1$ such that $\{i_{2r}, i_{2r+1}\} \in M^* \setminus \mathcal{M}^*$ and $\{i_{-2r}, i_{-(2r+1)}\} \in \mathcal{M}^* \setminus M^*$. Similarly $i_{-(2r+2)}$, $i_{2r+2}$ are nodes at level $2r + 2$ such that $\{i_{2r+1}, i_{2r+2}\} \in \mathcal{N}^* \setminus \mathcal{M}^*$ and $\{i_{-(2r+1)}, i_{-(2r+2)}\} \in \mathcal{M}^* \setminus \mathcal{N}^*$. Note that, by definition, such paths $P_\ell$ for $0 \leq \ell \leq t$ exist since the tree $T^*_i$ 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 $P_t$ ($\mathcal{M}^*$-edges of $P_t$). It is straightforward that this process produces a new perfect tree-b-matching $\mathcal{N}''$ in $T^t_i$.

Let us assume, for the moment, the following lemma

**Lemma 4.** The weight of the perfect tree-b-matching $\mathcal{N}''$ is strictly less than that of $\mathcal{N}^*$ on $T^t_i$.

This completes the proof of lemma 3 since lemma 4 shows that $\mathcal{N}^*$ is not the minimum weight perfect tree-b-matching on $T^t_i$, leading to a contradiction. □

**Proof of lemma 4.** 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 $i_t \in P_t$, consider the value $y_{i_t}^*$ from the optimum solution to the dual LP (1). Using the inequality $w_{ij} \leq y_{i_t}^* + y_{j_t}^*$ for edges of $\mathcal{M}^*$, we obtain: $\sum_{(i,j) \in P_t \cap \mathcal{M}^*} w_{ij} \leq (\sum_{r=\ldots} y_{i_t}^* - y_{i_{(r-t)}}^* - k_1 \epsilon$ 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 $w_{ij} \leq y_{i_t}^* + y_{j_t}^*$, with a gap of at least $\epsilon$. On the other hand, using the strict inequality $w_{ij} \geq y_{i_t}^* + y_{j_t}^*$ for edges of $\mathcal{N}^*$ we have: $\sum_{(i,j) \in P_t \cap \mathcal{N}^*} w_{ij} \geq (\sum_{r=\ldots} y_{i_t}^* - y_{i_{(r-t)}}^* + k_2 \epsilon$, where now $k_2$ is the number of $\mathcal{N}^*$-edges of $P_t$ that belong to $S$, or equivalently the number of times the inequality $w_{ij} \geq y_{i_t}^* + y_{j_t}^*$ is strict with a gap of at least $\epsilon$. One finds

$$\sum_{(i,j) \in P_t \cap \mathcal{N}^*} w_{ij} - \sum_{(i,j) \in P_t \cap \mathcal{M}^*} w_{ij} = y_{i_{(t-1)t}}^* - y_{i_{(t-1)t+1}}^* + (k_1 + k_2) \epsilon \geq (k_1 + k_2) \epsilon - 2L \geq 0.$$

Here $(a)$ uses the definition of $L$ from equation (2) and $(b)$ uses the fact that for all $i, j : \lambda_{ij}^* \geq 0$. The main step is $(c)$, which uses lemma 2 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 2. This means the path has at least one edge from the set $S$. Thus $(k_1 + k_2) \geq 2t/2n > 2L/\epsilon$. This completes the proof of lemma 4. □

**Extension to possibly non-perfect b-matchings.** Here we note 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 the 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 $deg_H(i) < b_i$). The minimum weight b-matching (b-MWM), $H^*$, is the b-matching such that $H^* = \arg\max_{H \in \mathcal{M}(b)} W_H$. 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.

**Asynchronous BP.** In the remaining 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

\[ \text{doi:10.1088/1742-5468/2008/06/L06001} \]
set $\vec{E}$ of all directed edges in the $G$; i.e. $\vec{E} = \{ (i \rightarrow j) \text{ s.t. } i \neq j \in V \}$. 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 update rule in step (3) of the algorithm Sync-BP as follows: 

$$m_{i \rightarrow j}(t) = w_{ij} - \ell_{i}^{th} \cdot \min \{ \ell \in N(i) \cup \{j\} \} \{ m_{\ell \rightarrow i}(t-1) \}$$

if $(i \rightarrow j) \in \vec{E}(t)$ and if $(i \rightarrow j) \notin \vec{E}(t)$ then the message will not be updated, i.e. it remains equal to $m_{i \rightarrow j}(t-1)$.

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 $(i \rightarrow j) \in \vec{E}$ is re-updated before at least one of its incoming edge directions $((\ell \rightarrow i) \text{ for } \ell \in N(i) \cup \{j\})$ is updated. More formally, if $(i \rightarrow j) \in \vec{E}(t) \cap \vec{E}(t+s)$ and $(i \rightarrow j) \notin \cup_{r=1}^{s-1} \vec{E}(t + r)$, then at least for one $\ell \in N(i) \cup \{j\}$, we should have $(\ell \rightarrow i) \in \cup_{r=1}^{s-1} \vec{E}(t + r)$.

Let us denote the above algorithm by Async-BP. We claim that, if each edge direction $(i \rightarrow j) \in \vec{E}$ is updated $O(n)$ times, then the same result as theorem 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., $u(t) = \min_{(u \rightarrow j) \in \vec{E}}(\{ \ell: \text{ s.t. } 1 \leq \ell \leq t \text{ and } (i \rightarrow j) \in \vec{E}(\ell) \})$. From the definition, $u(t)$ is a non-decreasing function of $t$. We claim that the following result holds

**Theorem 2.** Assume that the LP relaxation (1) has no fractional solution. Then the algorithm Async-BP converges to $M^*$ after at most $t$ iterations, provided $u(t) > 2nL/\epsilon$.

Proof of the above theorem relies on the notion of generalized computation tree for the asynchronous version of the BP algorithm which will be given in the longer version of this letter [22].

Finally we note that the same algorithm as Async-BP and the same result as theorem 2 can be stated and proved for the (possibly non-perfect) b-matchings as well.

We would like to thank László Lovász, Andrea Montanari, Elchanan Mossel and Amin Saberi for useful discussions. This work was done while RZ was a Visiting Researcher in the Theory Group at Microsoft Research, and was supported by the Microsoft Technical Computing Initiative.

**References**

[1] Mezard M and Zecchina R, *Random K-satisfiability: from an analytic solution to a new efficient algorithm*, 2002 Phys. Rev. E 66 056126

[2] Braunstein A, Mezard M and Zecchina R, *Survey propagation: an algorithm for satisfiability*, 2005 Random Struct. Algorithms 27 201

[3] Braunstein A, Mulet P, Pagnani A, Weigt M and Zecchina R, 2003 Phys. Rev. E 68 036702

[4] Braunstein A and Zecchina R, 2004 J. Stat. Mech. P06007

[5] Sanghavi S, Malioutov D and Willsky A, *Linear programming analysis of loopy belief propagation for weighted matching*, 2007 Proc. NIPS Conf. (Vancouver)

[6] Bayati M, Shah D and Sharma M, *Max-product for maximum weight matching: convergence, correctness, and LP duality*, 2008 IEEE Trans. Inf. Theor. 54 1241

Bayati M, Shah D and Sharma M, *Maximum weight matching via max-product belief propagation*, 2005 Proc. Int. Symp. on Information Theory p 1763 (preliminary version)

doi:10.1088/1742-5468/2008/06/L06001
On the exactness of the cavity method

[7] Huang B and Jebara T, *Loopy belief propagation for bipartite maximum weight b-matching*, 2007 Proc. Artif. Intell. Stat. (AISTATS) Conf.

[8] Gerards A, *Matching*, 1995 Hand book of Operation Research and Management Science vol 7 (Amsterdam: North-Holland) chapter 3, p 135

[9] Pulleyblank W, *Matchings and extensions*, 1995 Handbook of Combinatorics vol 1 (Amsterdam: North-Holland) chapter 3, p 179

[10] Mezard M and Parisi G, *Mean-field equations for the matching and travelling salesman problems*, 1986 Europhys. Lett. 2 913

[11] Aldous D, *The zeta (2) limit in the random assignment problem*, 2001 Random Struct. Algorithms 18 381

[12] Feldman J, Wainwright M and Karger D, *Using linear programming to decode binary linear codes*, 2005 IEEE Trans. Inf. Theory 51 954

[13] Vontobel P O and Koetter R, *On the relationship between linear programming decoding and min-sum algorithm decoding*, 2004 Proc. ISITA (Parma, Oct. 2004) p 991

[14] Vontobel P O and Koetter R, *Graph covers and iterative decoding of finite-length codes*, 2003 Proc. 3rd International Symposium on Turbo Codes and Related Topics (Brest) pp 75–82 [cs.IT/0512078]

[15] Wainwright M, Jaakkola T and Willsky A, *MAP estimation via agreement on trees: message-passing and linear programming*, 2005 IEEE Trans. Inf. Theory 51 3697

[16] Weiss Y, Yanover C and Meltzer T, *MAP estimation, linear programming and belief propagation with convex free energies*, 2007 UAI

[17] Moallemi C and Van Roy B, *A message-passing paradigm for resource allocation*, 2007 Preprint available at: http://www.moallemi.com/ciamac/papers/ra-2007.pdf

[18] Bayati M, Shah D and Sharma M, *A simpler max-product maximum weight matching algorithm and the auction algorithm*, 2006 Proc. IEEE Int. Symp. on Information Theory pp 557–61

[19] Bertsekas D P, *The auction algorithm: a distributed relaxation method for the assignment problem*, 1988 Ann. Oper. Res. 14 105

[20] Gamarnik D, Nowicki T and Swirsch G, *Maximum weight independent sets and matchings in sparse random graphs. Exact results using the local weak convergence method*, 2005 Random Struct. Algorithms 28 76

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

[22] Bayati M, Borgs C, Chayes J and Zecchina R, *Belief-propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions*, 2007 Preprint 0709.1190v2

[23] Sanghavi S, *Equivalence of LP relaxation and max-product for weighted matching in general graphs*, 2007 IEEE Information Theory Workshop (Sept. 2007) Preprint [0705.0760] (preliminary version)

doi:10.1088/1742-5468/2008/06/L06001 10