Belief Propagation and LP Relaxation for Weighted Matching in General Graphs

Sujay Sanghavi, *Member, IEEE*, Dmitry Malioutov, and Alan Willsky, *Fellow, IEEE*

***Abstract*—Loopy belief propagation has been employed in a wide variety of applications with great empirical success, but it comes with few theoretical guarantees. In this paper, we analyze the performance of the max-product form of belief propagation for the weighted matching problem on general graphs. We show that the performance of max-product is exactly characterized by the natural linear programming (LP) relaxation of the problem. In particular, we first show that if the LP relaxation has no fractional optima then max-product always converges to the correct answer. This establishes the extension of the recent result by Bayati, Shah and Sharma, which considered bipartite graphs, to general graphs. Perhaps more interestingly, we also establish a tight *converse*, namely that the presence of any fractional LP optimum implies that max-product will fail to yield useful estimates on some of the edges. We extend our results to the weighted b-matching and r-edge-cover problems. We also demonstrate how to simplify the max-product message-update equations for weighted matching, making it easily deployable in distributed settings like wireless or sensor networks.**

***Index Terms*—Belief propagation, combinatorial optimization, graphical models, Markov random fields, matching, message passing.**

I. Introduction

LOOPY belief propagation (LBP) and its variants [1]–[3] have been shown empirically to be effective in solving many instances of hard problems in a wide range of fields. These algorithms were originally designed for exact inference (i.e., calculation of marginals/MAP estimates) in probability distributions whose associated graphical models are tree-structured. While some progress has been made in understanding their convergence and accuracy on general "loopy" graphs (see [3]–[5] and their references), it still remains an active research area.

In this paper, we study the application of the widely used max-product form of LBP (or simply max-product (MP) algorithm), to the weighted matching problem.[1] Our motivation for doing so is twofold: first, weighted matching is a classical problem with much structure, and this structure can be used to provide a much finer characterization of max-product performance than would be possible for general graphical models. Second, fast and distributed computation of weighted matchings is often required in areas as diverse as resource allocation, scheduling in communications networks [8], and machine learning [9].

Given a graph $G = (V, E)$ with nonnegative weights w_e on its edges $e \in E$, the weighted matching problem is to find the heaviest set of mutually disjoint edges (i.e., a set of edges such that no two edges share a node). Weighted matching can be naturally formulated as an integer program (IP). The technique of linear programming (LP) relaxation involves replacing the integer constraints with linear inequality constraints. In general graphs, the linear program for weighted matching can have fractional optima—i.e., those that assign fractional mass to edges. The primary contribution of this paper is an exact characterization of max-product performance for the weighted matching problem: we show that

- If the LP has no fractional optima (i.e., if the optimum of LP is unique and integral), then max-product will converge and the resulting solution will be exactly the max-weight matching (Theorem 1).
- For any edge, if there exists an optimum of LP that assigns fractional mass to that edge, then the max-product estimate for that edge will either oscillate or be ambiguous (Theorem 2). For the entire graph, this implies that if fractional optima exist then max-product will fail (Corollary 1).

Most of the existing analysis of classical loopy belief propagation either provides sufficient conditions for correctness of solutions (e.g., [10] and [4]), or provides an analysis/interpretation of fixed points (e.g., [5] and [3]). However, there are relatively few results that provide *necessary* conditions for the convergence/correctness of the *iterative procedure*. Theorem 2 is thus significant in this regard, and we believe it is more general than the weighted matching and covering problems discussed in this paper.

Many tantalizing connections between belief propagation and linear programming (in various forms) have been observed/conjectured [11]. This paper provides a precise connection between the two for the weighted matching problem. An interesting insight in this regard, obtained from our work, is the importance of the uniqueness of the LP optimum, as opposed to uniqueness of the IP optimum. In particular, it is easy to construct examples where the LP has a unique integer optimum, but also has additional spurious fractional optima, for which max-product fails to be informative. A more detailed discussion of this is presented in Section V.

We extend our analysis to establish this equivalence between max-product and LP relaxation for two related problems:

Manuscript received December 14, 2007; revised September 26, 2010; accepted October 11, 2010. Date of current version March 16, 2011. This work was supported in part by NSF grants 0954059 (CAREER) and 0964391. The material in this paper was presented at the IEEE Information Theory Workshop (ITW), Tahoe, CA, May 2007.

S. Sanghavi is with the University of Texas, Austin, TX 78750 USA (e-mail: sanghavi@mail.utexas.edu).

D. Malioutov is with DRW Inc., Chicago, IL USA (e-mail: dmm@mit.edu).

A. Willsky is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: willsky@mit.edu).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2011.2110170

[1] This publication is the journal version of earlier results reported in [6]. Also related are recent results by Bayati, Borgs, Chayes and Zecchina [7]. See Section I for a discussion.

0018-9448/$26.00 © 2011 IEEE

weighted b-matching and r-edge-cover. Given a graph with edge weights and node capacities b_i, the *weighted b-matching* problem is to pick the heaviest set of edges so that at most b_i edges touch node i, for each $i \in V$. Similarly, if the graph has node requirements r_i, the *weighted r-edge-cover* problem is to pick the lightest set of edges so that each node $i \in V$ has at least r_i edges incident on it. Theorems 3 and 4 pertain to b-matching, and Theorems 5 and 6 to r-edge-cover.

In an insightful paper, Bayati, Shah and Sharma [10] were the first to analyze max-product for weighted matching problems; they established that max-product correctly solves weighted matching in *bipartite graphs*, when the optimal matching is unique. Theorem 1 represents a generalization of this result,[2] as for bipartite graphs it is well known that the extreme points of the matching LP polytope are integral. This means that if the LP has a fractional optimum, it has to *also* have multiple integral optima, i.e., multiple optimal matchings. So, requiring unique optima in bipartite graphs is equivalent to requiring no fractional optima for the LP relaxation. In [9] the results of [10] were extended to weighted b-matchings on bipartite graphs. Theorem 3 represents the corresponding extension of our results to b-matching on general graphs.

A preliminary version [6] of this paper contained a different proof of both Theorems 1 and 2. The proofs in that paper can be adapted handle more general message update rules (as opposed to the "fully synchronous" case considered in this paper). Both [6] and this paper consider the case of "imperfect" matchings, where each node can have at most one edge in the matching, but may have none. Independently developed recent results by Bayati *et al.* [7] provide an alternative proof for one of the two theorems—Theorem 1 which shows that tightness of LP implies BP success—for the conceptually harder case of perfect matchings. Their proof also holds for arbitrary message update schedules.

The outline of the paper is as follows. In Section III we set up the weighted matching problem and its LP relaxation. We describe the max-product algorithm for weighted matching in Section IV. The main result of the paper is stated and proved in Section V. In Section VI we establish the extensions to b-matching and r-edge-cover. Finally, in Section VII we show how max-product can be radically simplified to make it very amenable for implementation.

II. Related Work

This paper proves new results on the correctness and convergence of Loopy Belief Propagation for the weighted matching problem on general graphs. Belief propagation and its variants have proven extremely popular in practice for the solution of large-scale problems in inference, constraint satisfaction etc.; here we provide a summary of the work most directly related to this paper.

Classical BP in graphical models has two common flavors—SumProduct, which is used for finding marginals of individual/small groups of variables, and MaxProduct, which is used for finding the global most likely assignment of variables. Both flavors are iterative message-passing algorithms, designed to be exact when the graphical model is a tree. Analysis of their performance in graphs with cycles has been of much recent interest; existing analysis falls into two methodological categories. The first category is the direct analysis of fixed points of the iterative algorithm: [3] shows that the fixed points of SumProduct on general graphs correspond to zero-gradient points of the Bethe approximation to the energy function. [12] shows that the convergence of SumProduct is related to the uniqueness of the Gibbs measure on the infinite model represented by the computation tree. [11] shows the correspondence between BP fixed points and linear programming (LP) solutions for the decoding problem. For MaxProduct on general graphs, [5] establish that the fixed point solutions are locally optimal, in a graph-theoretic sense.

The second category of analysis, also the one taken in this paper, involves direct analysis of the dynamics of the iterative procedure, to jointly establish both convergence and relation to the correct solution. This approach was first used in [10] in the context of weighted matching on bipartite graphs (i.e., those that have no odd cycles). They established that if the optimum is unique, MaxProduct always converges to it; they also precisely bound the rate of convergence. Their approach generalizes to b-matchings as well, as established in [9]. Our paper generalizes this result to all (i.e., not just bipartite) graphs, where the relevant notion is not uniqueness of the true optimum, but uniqueness of the LP relaxation. Independent work in the recent paper [7] also establishes this result. Our paper also establishes a *converse*: that MaxProduct will fail on edges where the LP has a fractional value at some optimum. Parallel work [13] establishes this converse for the more general problem of finding the maximum weight independent set.

A related but separate algorithmic approach to inference are the variational techniques developed by [14] (see [15] for a more recent tutorial survey of this and related methods). For ML estimation, these algorithms involve a variant of direct coordinate descent on the dual of the LP. The algorithm in [16] is shown to always converge to the dual optimum for binary pairwise integer problems; more generally convergence of these algorithms is not fully understood.

III. Weighted Matching and its LP Relaxation

Suppose that we are given a graph G with edge-weights w_e. A *matching* is any subset of edges such that the total number of edges incident to any node i is at most 1. The weighted matching problem is to find the matching of largest weight. Weighted matching can be formulated as the following integer program:

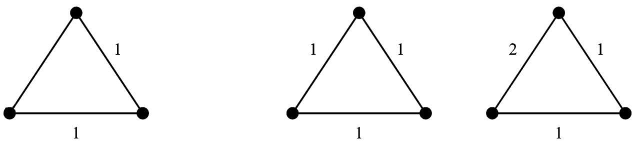
$$\begin{aligned}\mathsf{IP}:\quad & \max \sum_{e\in E} w_e x_e,\\ \text{s.t.}\quad & \sum_{e\in E_i} x_e \le 1 \text{ for all } i\in V,\\ & x_e\in\{0,1\} \text{ for all } e\in E.\end{aligned}$$

Here E_i is the set of edges incident to node i. The linear programming (LP) relaxation of the above problem is to replace the constraint $x_e \in \{0,1\}$ with the constraint $0 \le x_e \le 1$, for each $e \in E$. We denote the corresponding linear program by LP.

[2] [10] uses a graphical model which is different from ours to represent weighted matching, but this does not change the results.

In this paper, we are interested in the presence or absence of fractional optima for LP. An optimum x^* of LP is *fractional* if there exists some edge e to which it assigns fractional mass, i.e., if there is an e such that $0 < x_e^* < 1$. Note that LP will have no fractional optima if and only if LP has a unique optimum, and this optimum is integral.

Example 0 (Fractional Optima of LP*):* Consider, for example, the following three graphs.



In the cycle on the left, the LP has no fractional optima: the unique optimum (1,0,0) places mass 1 on the edge with weight 3, and 0 on the other two edges. The two cycles on the right, however, do have fractional optima. The middle cycle has $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$ as its unique optimum, while the one on the right has many optima: (1,0,0), $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$, and every convex combination of the two. Note that in the right-most cycle the LP relaxation is "tight", i.e., the optimal values of IP and LP are equal. Also, the IP has a unique optimum. However, there still exist fractional optima for the LP. ■

Note that if the graph is bipartite (i.e., it contains no odd cycles), then all the extreme points of the LP polytope are integral. As a result, in this case, fractional optima exist if and only if there are multiple integral optima of the LP. This is the reason our Theorem 1 is a generalization of [10].

We need the following lemma for the proof of Theorem 1. Its proof is obvious, and is omitted.

Lemma 1: Let $\mathcal{P}$ be the polytope of feasible solutions for LP, and let the optimum x^* be unique. Define

$$c = \inf_{x \in \mathcal{P} - x^*} \frac{w'(x^* - x)}{|x^* - x|}.$$

Then, it has to be that $c > 0$.

Remark: In the above lemma, $|x^* - x| = \sum_e |x_e^* - x_e|$ is the ℓ_1-norm of the perturbation from x^*. The fact that the LP has a unique optimum means that moving away from x^* along any direction that remains within $\mathcal{P}$ will result in a strict linear decrease in the objective function. The constant c is nothing but the smallest such rate of decrease. Uniqueness of x^* implies that c should be strictly positive.

Remark 2: While c has been defined via an infimum over all points in the polytope, it is clear that we can replace this with a minimum over all extreme points of the polytope. So, if we consider the right-most triangle graph in Example 0 above—the one with edge weights 3,1,1–then $c = 1/3$. This is because the LP optimum is $x* = (1, 0, 0)$ with weight $w'x* = 3$, and among the other extreme points (in this case all feasible points where each coordinate is 0, 1 or $\frac{1}{2}$ [17]) the one which achieves the minimum is the point $x = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$, which has weight $w'x = \frac{5}{2}$.

IV. Max-Product for Weighted Matching

The Max-product form of belief propagation is used to find the most likely state—the MAP estimate—of a probability distribution, when this distribution is known to be a product of factors, each of which depends only on a subset of the variables. Max-product operates by iteratively passing messages between variables and the factors they are a part of. In order to apply max-product, we now formulate weighted matching on G as a MAP estimation problem, by constructing a suitable probability distribution. This construction is naturally suggested by the form of the integer program IP. Associate a binary variable $x_e \in \{0, 1\}$ with each edge $e \in E$, and consider the following probability distribution:

$$p(x) \propto \prod_{i \in V} \psi_i(x_{E_i}) \prod_{e \in E} \exp(w_e x_e) \tag{1}$$

which contains a factor $\psi_i(x_{E_i})$ for each node $i \in V$, the value of which is $\psi_i(x_{E_i}) = 1$ if $\sum_{e \in E_i} x_e \leq 1$, and 0 otherwise. Note that we use i to refer *both* to the nodes of G and factors of p, and e to refer both to the edges of G and variables of p. The factor $\psi(x_{E_i})$ enforces the constraint that at most one edge incident to node i can be assigned the value "1". It is easy to see that, for any x, $p(x) \propto \exp(\sum_e w_e x_e)$ if the set of edges $\{e | x_e = 1\}$ constitute a matching in G, and $p(x) = 0$ otherwise. Thus the max-weight matching of G corresponds to the MAP estimate of p.

Max-Product for Weighted Matching

- **(INIT)** Set $t = 0$ and initialize each message to 1.
- **(ITER)** Iteratively compute new messages until convergence as follows:
  Variable to Factor:
  $$m_{e \to i}^{t+1}[x_e] = \exp(x_e w_e) \times m_{j \to e}^t[x_e].$$
  Factor to Variable:
  $$m_{i \to e}^{t+1}[x_e] = \max_{x_{E_i \setminus e}} \left\{ \psi_i(x_{E_i}) \prod_{e' \in E_i \setminus e} m_{e' \to i}^t[x_{e'}] \right\}.$$
  Also, at each t compute beliefs
  $$n_e^t[x_e] = \exp(w_e x_e) \times m_{i \to e}^t[x_e] \times m_{j \to e}^t[x_e]$$
- **(ESTIM)** Each edge e has estimate $\hat{x}^t \in \{0, 1, ?\}$ at time t
  $$\begin{aligned} \hat{x}_e^t &= 1 \quad \text{if } n_e^t[1] > n_e^t[0], \\ \hat{x}_e^t &= 0 \quad \text{if } n_e^t[1] < n_e^t[0], \\ \hat{x}_e^t &= ? \quad \text{if } n_e^t[1] = n_e^t[0]. \end{aligned}$$

The factor-graph version of the max-product algorithm [1] passes messages between variables and the factors that contain them at each iteration t. For the p in (1), each variable is a member of exactly two factors. The output is an estimate $\hat{x}$ of the MAP of p. We now present the max-product update equations adapted for the p in (1). We use e and (i, j) to denote the same edge. Also, for two sets A and B the set difference is denoted by the notation $A \backslash B$.

Note that estimate $\hat{x}_e^t = 1$ means that, at time t, Max-product estimates that edge e is part of a max-weight matching, while $\hat{x}_e^t = 0$ means that it is not. $\hat{x}_e^t = ?$ means that Max-product cannot decide on the membership of e. In this paper, we will say that the max-product estimate for an edge is *uninformative* if its value keeps changing even after a large amount of time has passed, or if its value remains constant and equal to ?.

The message update rules are described above in a form familiar to readers already acquainted with Max-product. In Section VII we show that the update rules can be substantially simplified into a "node-to-node" protocol that is much more amenable to implementation.

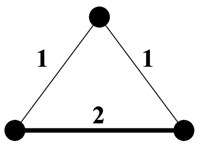
V. Main Results

We now state and prove the main results of this paper. Theorem 1 states that whenever the LP relaxation has no fractional optima, max-product is successful at finding the max-weight matching. Theorem 2, and Corollary 1, state the converse: if there exist fractional optima, then max-product will fail.

Theorem 1: Let $G = (V, E)$ be a graph with nonnegative real weights w_e on the edges $e \in E$. If the linear programming relaxation LP has no fractional optima, then the max-product estimate $\hat{x}^t$ is correct (i.e., it is the true max-weight matching) for all times $t > \frac{2w_{\max}}{c}$, where $w_{\max}$ is the maximum weight of any edge in the graph, and c is as defined in Lemma 1.

Remark 1: Note that the requirement of "no fractional optima" is equivalent to saying that the LP has a unique optimum, *and* that this optimum is integral. The time after which the estimates $\hat{x}^t$ will converge to correct values is determined by the "pointedness" of the LP polytope at the optimum, as represented by the constant c of Lemma 1.

As noted previously, the requirement of absence of fractional optima is in general strictly stronger than tightness of the LP relaxation. It is illustrative at this point to consider the performance of max-product on the right-most graph in Example 0: the three-cycle with weights 2,1,1. For this there are infinitely many optimal solutions to LP: (1,0,0), $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$, and all convex combinations of the two. Thus, even though the LP relaxation is tight, there exist fractional optima. For this graph, it can be easily verified (e.g., using the computation tree interpretation below) that the estimates as a function of time will oscillate as shown in the table below.



$t =$	1	2	3	4	5	6	$\cdots$
if we = 1, estimate $\hat{x}_e^t =$	1	0	?	0	?	0	$\cdots$
if we = 2, estimate $\hat{x}_e^t =$	1	?	1	?	1	?	$\cdots$

We see that the edges with weights 1 will have estimates that oscillate between 0 and ?, while the edge with weight 2 will oscillate between 1 and ?. The oscillatory behavior of this example is not just a particular case, it holds in general—as stated in the following theorem. We first state the most general form of the theorem, followed by corollaries and discussion.

Theorem 2: Let $G = (V, E)$ be a graph with nonnegative real weights w_e on the edges $e \in E$. The corresponding LP may, in general, have multiple optima. Then, for any edge e in G,

1) If there exists *any* optimum x^* of LP for which the mass assigned to edge e satisfies $x_e^* > 0$, then the max-product estimate $\hat{x}_e^t$ is 1 or ? for all *odd* times t.
2) If there exists *any* optimum x^* of LP for which the mass assigned to edge e satisfies $x_e^* < 1$, then the max-product estimate $\hat{x}_e^t$ is 0 or ? for all *even* times t.

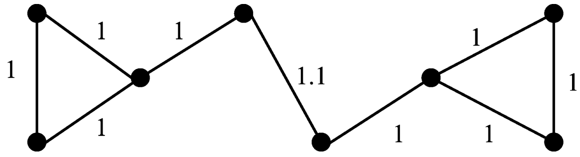
Remark: In light of this theorem, it is easy to see that max-product yields useful estimates for all edges if and only if each x_e^* has an integral value that is consistent at all optima x^* of LP. This means that LP has to have a unique optimum, and this optimum has to be integral. Hence, Theorem 1 is tight: any deviation from the sufficient condition therein will result in useless estimates for some edges.

Corollary 1: Suppose the LP has at least one fractional optimum. Then, Theorem 2 implies that max-product estimates will be uninformative for all edges that are assigned nonintegral mass at any LP optimum.

In the case of nonunique optima, note that in Theorem 2 the choice of LP optimum x^* is allowed to depend on e, the edge of interest. Thus, if there are optima $\overline{x}$ and $\widetilde{x}$ of LP such that $\overline{x}_e < 1$ and $\widetilde{x}_e > 0$, then the estimate $\hat{x}_e^t$ will either keep changing at every iteration, or will remain fixed at $\hat{x}_e^t = ?$, an uninformative estimate. It is thus easy to see that Theorem 2 covers both the case when the LP relaxation is loose (has no integral optima), and the case when the LP relaxation is tight, but multiple optima exist.

In general, when fractional optima exist, max-product may converge to useful estimates for some edges and oscillate or be uninformative for others. It follows from theorem 2 that

- The useful estimates are exactly as predicted by the LP relaxation: if $\widehat{x}_e = 1$ for some $e \in G$, then $x_e^* = 1$ for all optima x^* of LP, and correspondingly if $\widehat{x}_e = 0$ then $x_e^* = 0$.
- Any edge with fractional mass $0 < x_e^* < 1$ will not have useful estimates. However, the converse is *not* true: there may exist edges that are assigned the same integral mass in every max-weight matching, but for which max-product is un-informative. Thus, in a sense Max-product is weaker than LP relaxation for the matching problem. Consider the example below.



The unique LP optimum puts mass $\frac{1}{2}$ on all six edges in the two triangles, mass 1 on the middle edge of weight 1.1, and mass 0 on the other two edges in the path. Max-product estimates oscillate between 0 and 1 on all edges.

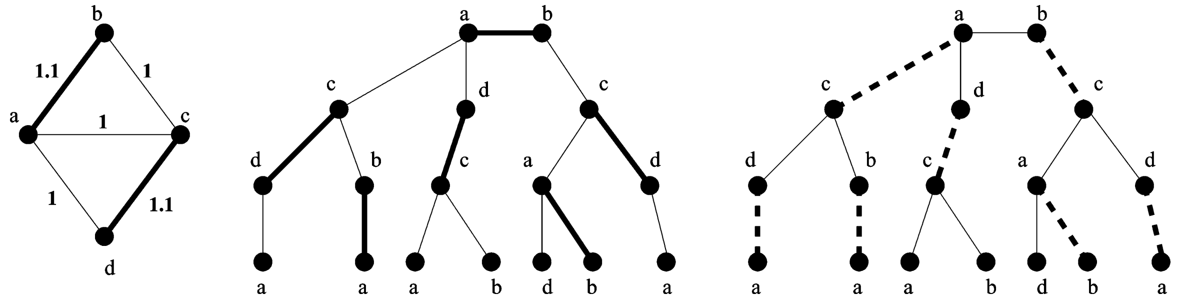


Fig. 1. Computation tree figure for example 1.

We now proceed to prove the two theorems above. Both proofs rely on the well-known computation tree interpretation of Max-product beliefs [5], [12], which we describe first. The proofs follow immediately after.

A. The Computation Tree for Weighted Matching

Recall the variables of the distribution p in (1) correspond to edges in G, and nodes in G correspond to factors. For any edge e, the *computation tree* at time t rooted at e, which we denote by $T_e(t)$, is defined recursively as follows: $T_e(1)$ is just the edge e, the *root* of the tree. The two endpoints of the root (nodes of G) are the leaves of $T_e(1)$. The tree $T_e(t)$ at time t is generated from $T_e(t-1)$ by adding to each leaf of $T_e(t-1)$ a copy of each of its neighbor edges in G, except for the neighbor edge that is already present in $T_e(t-1)$. Each edge in T_e is a copy of an edge in G, and the weights of the edges in T_e are the same as the corresponding edges in G.

For any edge e and time t, the max-product estimate accurately represents the membership of the root e in max-weight matchings on the computation tree $T_e(t)$, as opposed to the original graph G. This is the computation tree interpretation, and is stated formally in the following lemma (for a proof, see e.g., [5]).

Lemma 2: For any edge e at time t,

- $\hat{x}_e^t = 1$ if and only if the root of $T_e(t)$ is a member of *every* max-weight matching on $T_e(t)$.
- $\hat{x}_e^t = 0$ if and only if the root of $T_e(t)$ is not a member of *any* max-weight matching on $T_e(t)$.
- $\hat{x}_e^t = ?$ else.

Remarks: The beliefs $n_e^t[x_e]$ are the max-marginals at the root of the computation tree $T_e(t)$. If $n_e^t[1] > n_e^t[0]$ then any matching in $T_e(t)$ which excludes the root has a suboptimal weight. Similarly, if $n_e^t[1] < n_e^t[0]$, then any matching in $T_e(t)$ including the root is suboptimal. However, when $n_e^t[1] = n_e^t[0]$, then there exists an optimal matching with $x_e^t = 0$, and another optimal matching with $x_e^t = 1$.

Note that max-product estimates correspond to max-weight matchings on the computation trees $T_e(t)$, as opposed to on the original graph G. Suppose M is a matching on the original graph G, and T_e is a computation tree. Then, the *image* of M in T_e is the set of edges in T_e whose corresponding copy in G is a member of M. We now illustrate the ideas of this section with a simple example.

Example 1 (Concepts Related to Computation Trees): Consider Fig. 1. G appears on the left, the numbers are the edge weights and the letters are node labels. The max-weight matching on G is $M^* = \{(a,b),(c,d)\}$, depicted in bold on G. In the center plot we show $T_{(a,b)}(4)$, the computation tree at time $t=4$ rooted at edge (a,b). Each node is labeled in accordance to its copy in G. The bold edges in the middle tree depict M_T^*, the matching which is the image of M^* onto $T_{(a,b)}(4)$. The weight of this matching is 6.6, and it is easy to see that *any* matching on $T_{(a,b)}(4)$ that includes the root edge will have weight at most 6.6. In the right-most tree, the dotted edges represent M, the max-weight matching on the tree $T_{(a,b)}(4)$. M has weight 7.3. In this example we see that even though (a,b) is in the unique optimal matching in G, it turns out that root (a,b) is not a member of any max-weight matching on $T_{(a,b)}(4)$, and hence we have that $\hat{x}_{(a,b)}^4 = 0$. Note also that the dotted edges are *not* an image of any matching in the original graph G. This example thus illustrates how "spurious" matchings in the computation tree can lead to incorrect beliefs, and estimates. In the example above the reason why Max-product disagrees with LP relaxation is that Max-product has not yet converged. ■

B. Proof of Theorem 1

We now prove that the uniqueness and tightness of the LP relaxation ensures that each estimate $\widehat{x}_e$ is 0 or 1, and also that the estimate corresponds to the optimal matching. As mentioned in the Introduction, this is a generalization of the bipartite graph result in [10]—since it is well known [17] that in the bipartite case all vertices of the LP polytope are integral.[3] Let M^* be the optimal matching, and x^* the corresponding 0–1 vector that is the unique optimum of LP.

To prove the theorem, we need to show that, for a large enough time t, the estimates satisfy

$$\hat{x}_e^t = 0 \text{ for all edges } e \notin M^*$$
$$\hat{x}_e^t = 1 \text{ for all edges } e \in M^*.$$

Consider now any time $t > \frac{2w_{\max}}{c}$, where $w_{\max} = \max_e w_e$ is the weight of the heaviest edge, and c is as in Lemma 1 above. Suppose that there exists an edge $e \in M^*$ for which the estimate

[3] Our proof below is along similar lines to the one in [10], namely that both proofs proceed via contradiction by constructing a new optimum. In [10], this new optimum is actually an alternate matching on the computation tree; in ours it is a new LP optimum.

at time t is not correct: $\hat{x}_e^t \neq 1$ (i.e., $\hat{x}_e^t \in \{0, ?\}$). We now show that this leads to a contradiction.

We start with a **brief outline** of the proof. Let $T_e(t)$ be the computation tree at time t for that edge e. From Lemma 2, the fact that $\hat{x}_e^t \neq 1$ means that there exists a max-weight matching M on $T_e(t)$ that does not contain the root e. Due to the uniqueness of the LP optimum we can use M^* to modify M and obtain a matching M' on $T_e(t)$ which has *strictly* larger weight than M. This contradicts the optimality of M on $T_e(t)$, and proves that $\hat{x}_e^t$ has to be equal to 1.

We now give the details in full. Let M_T^* be the image of M^* onto $T_e(t)$. By assumption, $e \in M^*$ in original graph G, and hence the root $e \in M_T^*$. Recall that, from Lemma 2, $\hat{x}_e^t \neq 1$ implies there exists some max-weight matching M of $T_e(t)$ that does not contain the root, i.e., root $e \notin M$. Thus the root $e \in M_T^* - M$. From root e, build an alternating path P on $T_e(t)$ by successively adding edges as follows: first add e, then add all edges adjacent to e that are in $M - M_T^*$, then all *their* adjacent edges that are in $M_T^* - M$, and so forth until no more edges can be added. This will occur either because no edges are available that maintain the alternating structure, or because a leaf of $T_e(t)$ has been reached. Note also that P will be a path, because M and M_T^* are matchings and so any node in $T_e(t)$ can have at most one adjacent edge in each of the two matchings.

For illustration, consider Example 1 of Section IV. e in this case is the edge (a, b), and M^* is denoted by the bold edges in the left-most figure G. The computation tree $T_e(4)$ at time 4 is shown in the center, with the image M_T^* marked in bold. Note that the root $e \in M_T^*$. In the right-most figure is depicted M, a max-weight matching of $T_e(t)$. The alternating path P, as defined above, would in this example be the path *adcabcda* that goes from the left-most leaf to the right-most leaf. It is easy to see that this path alternates between edges in $M - M_T^*$ and $M_T^* - M$. We now use the following lemma to complete the proof of Theorem 1.

Lemma 3: Suppose LP has no fractional optima. Let M be a matching in $T_e(t)$ which disagrees with M_T^* on the root, i.e., root $e \in \{M - M_T^*\} \cup \{M_T^* - M\}$. Let P be the maximal alternating path containing the root. Then $w(P \cap M_T^*) > w(P \cap M)$, provided $t > \frac{2w_{max}}{c}$.

Lemma 3 is proved in the appendix, using a perturbation argument: if lemma is false, then it is possible to perturb x^* to obtain a new feasible point $x \in \mathcal{P}$ such that $w'x \geq w'x^*$, thus violating the optimality and uniqueness of x^* for the LP on G.

Now consider the matching M, and change it by "flipping" the edges in P. Specifically, let $M' = M - (P \cap M) + (P \cap M_T^*)$ be the matching containing all edges in M except the ones in P, which are replaced by the edges in $P \cap M_T^*$. It is easy to see that M' is a matching in $T_e(t)$. Also, from Lemma 3(a) it follows that $w(M') > w(M)$. This however, violates the assumption that M is an optimal matching in $T_e(t)$. We have arrived at a contradiction, and thus it has to be the case that $\hat{x}_e^t = 1$ for all $e \in M^*$.

A similar argument can be used to establish that $\hat{x}_e^t = 0$ for all $e \notin M^*$. In particular, suppose that $\hat{x}_e^t \neq 0$ for some $e \notin M^*$. This means there exists a max-weight matching M in $T_e(t)$ that contains the root e. Again, let M_T^* be the image of M^* onto $T_e(t)$. Note that the root $e \in M - M_T^*$. Let P be a maximal alternating path that the root e. Using Lemma 3, it follows that $w(P \cap M_T^*) > w(P \cap M)$. Now, as before, define $M' = M - (P \cap M) + (P \cap M_T^*)$. It follows that $w(M') > w(M)$, violating the assumption that M is an optimal matching in $T_e(t)$. Thus the root e has to have $\hat{x}_e^t = 0$. This proves the theorem.

C. Proof of Theorem 2

We now prove Theorem 2. Suppose part 1 is not true, i.e., there exists edge e, an optimum x^* of LP with $x_e^* > 0$, and an odd time t at which the estimate is $\hat{x}_e^t = 0$. Let $T_e(t)$ be the corresponding computation tree. Using Lemma 2 this means that the root e is *not* a member of any max-weight matching of $T_e(t)$. Let M be some max-weight matching on $T_e(t)$. We now define the following set of edges

$$E_1^* = \{e' \in T_e(t) : \\ e' \notin M, \text{ and copy of } e' \text{ in G has } x^* > 0\}.$$

In words, E_1^* is the set of edges in $T_e(t)$ which are not in M, and whose copies in G are assigned strictly positive mass by the LP optimum x^*.

Note that by assumption the root $e \in E_1^*$ and hence $e \notin M$. Now, as done in the proof of Theorem 1, build a maximal alternating path P which includes the root e, and alternates between edges in M and edges in E_1^*. By maximal, we mean that it should not be possible to add edges to P and still maintain its alternating structure. Note that in contrast to Theorem 1, we may have multiple edges in E_1^* touching a node. In such a case, we pick an arbitrary one of them and add to P. We use the following lemma.

Lemma 4: The weights satisfy $w(P \cap M) \leq w(P \cap E_1^*)$.

The proof is included in the appendix and is similar in principle to that of Lemma 3: if the weights are not as specified, then it is possible to perturb x^* to obtain a feasible solution of LP with *strictly* higher value than x^*, thus violating the assumption that x^* is an optimum of LP. The fact that t is odd is used to ensure that the perturbation results in a feasible point.

We now use Lemma 4 to finish the proof of part 1 of Theorem 2. Consider $M' = M - (M \cap P) + (E_1^* \cap P)$, which is a new matching of $T_e(t)$. Lemma 4 implies that $w(M') \geq w(M)$, i.e., M' is also a max-weight matching of $T_e(t)$. However, note that the root $e \in M'$, and so this contradicts the fact that root e should not be in any max-weight matching of $T_e(t)$. This proves part 1 of the theorem.

Part 2 is proved in a similar fashion, with the perturbation argument now requiring that t be odd. Specifically, suppose part 2 is not true, then there exists an edge e, an optimum x^* of LP with $x_e^* < 1$, and an even time t at which the estimate is $\hat{x}_e^t = 1$. This implies that root e is a member of *every* max-weight matching of $T_e(t)$. Let M be any such max-weight matching in $T_e(t)$, and define the following set of edges

$$E_2^* = \{e' \in T_e(t) : \\ e' \notin M, \text{ and copy of } e' \text{ in } G \text{ has } x_{e'}^* > 0\}.$$

In words, E_2^* is the set of edges in $T_e(t)$ which are not in M, and whose copies in G are assigned strictly positive mass by the

LP optimum x^*. Note that the root $e \in M$ and hence $e \notin E_2^*$. Let P be a maximal alternating path which includes the root e, and alternates between edges in M and edges in E_2^*.

Lemma 5: The weights satisfy $w(P \cap M) \leq w\,(P \cap E_2^*)$.

The proof of this lemma is similar to that of Lemma 4, and is given in the appendix. It uses the fact that t is even. Now, as before, consider $M' = M - (M \cap P) + (E_2^* \cap P)$, which is a new matching of $T_e(t)$. Lemma 5 implies that $w(M') \geq w(M)$, i.e., M' is also a max-weight matching of $T_e(t)$. However, note that the root $e \notin M'$, and so this contradicts the fact that root e should be in every max-weight matching of $T_e(t)$. This proves part 2 of the theorem.

VI. Extensions

We now establish the extensions of Theorems 1 and 2 to the weighted b-matching and r-edge-cover problems. The main ideas remain unchanged, and thus the proofs are outlines, with just the important differences from the corresponding proofs for the simple matching highlighted.

A. Weighted b-Matching

The weighted b-matching problem is given by the following integer program: given numbers $b_i \geq 0$ for each node i

$$\begin{aligned} b\mathsf{IP}: \quad & \max \sum_{e \in E} w_e x_e, \\ \text{s.t.} \quad & \sum_{e \in E_i} x_e \leq b_i \text{ for all } i \in V, \\ & x_e \in \{0,1\} \text{ for all } e \in E. \end{aligned}$$

The LP relaxation of this integer program is obtained by replacing the constrains $x_e \in \{0,1\}$ by the constraints $x_e \in [0,1]$ for each $e \in E$. We will denote the resulting linear program by $b\mathsf{LP}$.

To apply Max-product, first consider a probability distribution as in (1), but with $\psi_i(x_{E_i})$ now defined to be 1 if $\sum_{e \in E_i} x_e \leq b_i$, and 0 otherwise. The max-product updates remain as specified in Section IV. The following two theorems are the respective generalizations of Theorems 1 and 2.

Theorem 3: If $b\mathsf{LP}$ has no fractional optima, then the max-product estimate $\hat{x}^t$ is correct (i.e., it is the true max-weight b-matching) for all times $t > \frac{2w_{\max}}{c}$, where $w_{\max}$ is the maximum weight of any edge in the graph, and c is as defined in Lemma 1 (but with $\mathcal{P}$ being the b-matching polytope)

Theorem 4: For any edge e in G,

1) If there exists *any* optimum x^* of $b\mathsf{LP}$ for which the mass assigned to edge e satisfies $x_e^* > 0$, then the max-product estimate $\hat{x}_e^t$ is 1 or ? for all *odd* times t.
2) If there exists *any* optimum x^* of $b\mathsf{LP}$ for which the mass assigned to edge e satisfies $x_e^* < 1$, then the max-product estimate $\hat{x}_e^t$ is 0 or ? for all *even* times t.

The proofs of both theorems are similar to those of Theorems 1 and 2, respectively. In particular, note that there will be an alternating path between any two b-matchings on the computation tree. All the alternating path and perturbation arguments remain as before.

B. Weighted r-Edge-Cover

The min-weight r-edge-cover problem is given by the following integer program: given numbers $r_i \leq d_i$ for each node i, where d_i is the degree of node i

$$\begin{aligned} r\mathsf{IP}: \quad & \min \sum_{e \in E} w_e x_e, \\ \text{s.t.} \quad & \sum_{e \in E_i} x_e \geq r_i \text{ for all } i \in V, \\ & x_e \in \{0,1\} \text{ for all } e \in E. \end{aligned}$$

The LP relaxation of $r\mathsf{IP}$ is obtained by replacing the constrains $x_e \in \{0,1\}$ by the constraints $x_e \in [0,1]$ for each $e \in E$. We will denote the resulting linear program by $r\mathsf{LP}$. To apply max-product, consider the following probability distribution

$$q(x) \propto \prod_{i \in V} \psi_i(x_{E_i}) \prod_{e \in E} \exp(-w_e x_e) \tag{2}$$

Here the factor $\psi_i(x_{E_i})$ for node i takes value 1 if and only if $\sum_{e \in E_i} x_e \geq r_i$, and 0 otherwise. It is easy to see that any maximum of q corresponds to a min-weight r-edge-cover of the graph. The max-product updates remain as specified in Section IV, except that w_e should be replaced by $-w_e$. The two theorems are now stated here.

Theorem 5: If r-LP has no fractional optima, then the max-product estimate $\hat{x}^t$ is correct (i.e., it is the true min-cost r-edge-cover) for all times $t > \frac{2w_{\max}}{c}$, where $w_{\max}$ is the maximum weight of any edge in the graph, and c is as defined below ($\mathcal{P}$ is the feasible polytope of $r\mathsf{LP}$)

$$c = \inf_{x \in \mathcal{P} - x^*} \frac{w'x - w'x^*}{|x - x^*|}$$

Theorem 6: For any edge e in G,

1) If there exists *any* optimum x^* of $r\mathsf{LP}$ for which the mass assigned to edge e satisfies $x_e^* > 0$, then the max-product estimate $\hat{x}_e^t$ is 1 or ? for all *odd* times t.
2) If there exists *any* optimum x^* of $r\mathsf{LP}$ for which the mass assigned to edge e satisfies $x_e^* < 1$, then the max-product estimate $\hat{x}_e^t$ is 0 or ? for all *even* times t.

Theorems 5 and 6 are most easily obtained by mapping the max-product updates for the r-edge-cover problem to those of the b-matching problem. In particular, if d_i is the degree of node i, set

$$b_i = d_i - r_i.$$

Then, any edge e will be included in the min-weight r-edge-cover if and only if it is *not* included in the max-weight b-matching. The following lemma shows that there is an exact relationship between the max-product updates for the r-edge-cover problem and the corresponding b-matching problem. It can easily be proved by induction, we include the proof in the appendix.

Lemma 6: Given a weighted r-edge-cover problem, let m denote the max-product messages and n the beliefs. Consider now the weighted b-matching problem where edge weights remain the same and each $b_i = d_i - r_i$. Let $\tilde{m}$ and $\tilde{n}$ denote the messages and beliefs for this b-matching problem. Then, we have that for time t, node i and edge $e \in E_i$,

$$\frac{m^t_{i\to e}[0]}{m^t_{i\to e}[1]} = \frac{\tilde{m}^t_{i\to e}[1]}{\tilde{m}^t_{i\to e}[0]},$$
$$\frac{m^t_{e\to i}[0]}{m^t_{e\to i}[1]} = \frac{\tilde{m}^t_{e\to i}[1]}{\tilde{m}^t_{e\to i}[0]}$$
and
$$\frac{n^t_{e}[0]}{n_{e}[1]} = \frac{\tilde{n}^t_{e}[1]}{\tilde{n}_{e}[0]}.$$

Note now that the estimate $\hat{x}^t_e$ depends only on the ratio $\frac{n_e[0]}{n_e[1]}$. In particular, $\hat{x}^t_e = 0, 1, ?$ if and only if $\frac{n_e[0]}{n_e[1]}$ is, respectively, $>, <, =$ to 1. Thus, Lemma 6 implies that the r-edge cover max-product estimate for edge e will be 1 if and only if the corresponding b-matching max-product estimate is 0. Similarly, 0 maps to 1, and ? to ? Thus, Theorems 5 and 6 follow from Theorems 3 and 4, respectively.

VII. Protocol Simplification

In this section, we show that max-product for the weighted matching problem can be simplified for implementation purposes. Similar simplifications have also been performed in [9] and [10]. Recall that in the specification given in Section IV, messages are passed between edges and nodes. However, it would be more desirable to just have an implementation where messages are passed only between nodes. Toward this end, for every pair of neighbors i and j, let $e = (i, j)$ be the edge connecting the two, and define

$$a^t_{i\to j} = \log\left(\frac{m^t_{i\to e}[0]}{m^t_{i\to e}[1]}\right).$$

The protocol with the a-messages is specified below.

Simplified Max-Product for Weighted Matching

- **(INIT)** Set $t = 0$ and initialize each $a^0_{i\to j} = 0$
- **(ITER)** Iteratively compute new messages until convergence as follows: $(y_+ = \max(0, y))$

$$a^{t+1}_{i\to j} = \max_{k\in\mathcal{N}(i)-j}\left(w_{ik} - a^t_{k\to i}\right)_+$$

- **(ESTIM)** Upon convergence, output estimate $\hat{x}$: for each edge set $\hat{x}_{(i,j)} = 0, 1$ or ? if $(a_{i\to j} + a_{j\to i})$ is, respectively, $>, <$, or $= w_{ij}$.

The update equations for b-matching and r-edge-cover can also be simplified by defining a's as above.

Proof of Lemma 3: The outline of the proof is as follows: we will use P to define a new feasible point x of the LP by modifying x^*, the unique optimum of the LP. We obtain x by subtracting ϵ from $x^*_{e'}$ for every edge in $P \cap M^*_T$ and adding ϵ for every edge in $P \cap M$, counting repeated occurrences. The fact that the weight $w'x$ is strictly less than $w'x^*$ will prove the lemma.

Formally, We define two length-$|E|$ vectors α and β as follows: for every e' in the original graph,

$\alpha_{e'}$ = number of (copies of) e' that appear in $P \cap M^*_T$. Note that $\alpha_{e'} > 0$ only for edges $e' \in M^*$, and $\alpha_{e'} = 0$ for other edges $e' \notin M^*$.

$\beta_{e'}$ = number of (copies of) e' that appear in $P \cap M$, *excluding* copies that touch a leaf of $T_e(t)$. Note that $\beta_{e'} > 0$ only for $e' \notin M^*$, and $\beta_{e'} = 0$ for $e' \in M^*$.

In the above, the leaves of tree $T_e(t)$ are nodes at the last level of $T_e(t)$, i.e., furthest away from the root. The path P has two endpoints, and hence it can have at most two leaf edges in $P \cap M$. Let w_1 and w_2 be equal to the weights of these two edges, if they exist, and $w_i = 0$ if the corresponding edge does not exist. Then, we have that

$$w'\alpha = w\left(P \cap M^*_T\right) \tag{3}$$
$$w'\beta = w(P \cap M) - w_1 - w_2. \tag{4}$$

For an illustration of these definitions, look at the footnote.[4] We are now ready to define the perturbation: let $\epsilon > 0$ be a small positive number, and

$$x = x^* + \epsilon(\beta - \alpha). \tag{5}$$

We now need the following auxiliary lemma, which is proved later in the Appendix.

Lemma 7: The vector x as defined in (5) is a feasible point of LP, for a small enough choice of ϵ.

We now find it convenient to separately consider two possible scenarios for the path P and weights w_1, w_2.

Case 1: $w_1 = w_2 = 0$

Suppose now that the statement of Lemma 3 is not true, i.e., suppose that $w\left(P \cap M^*_T\right) \leq w(P \cap M)$. From (3) and (4), and the assumption $w_1 = w_2 = 0$, it then follows that $w'\alpha \leq w'\beta$. From (5) it then follows that $w'x \geq w'x^*$. Note also that $x \neq x^*$ because $\beta - \alpha \neq 0$. We have thus obtained a feasible point x of the LP with weight at least as large as the unique optimum x^*. This is a contradiction, and hence for this case it has to be that $w\left(P \cap M^*_T\right) > w(P \cap M)$.

Case 2: At least one of w_1 or w_2 is nonzero.

For w_1 or w_2 to be non-zero, at least one endpoint of P has to be a leaf of $T_e(t)$. The tree has depth t, and P contains the root and a leaf, so the path length $|P| \geq t$. Now, for each edge $e' \in M^*$, $|x_{e'} - x^*_{e'}| = \epsilon\alpha_{e'}$, and for each $e' \notin M^*$, $|x_{e'} - x^*_{e'}| = \epsilon\beta_{e'}$. Thus we have that

$$|x - x^*| = \epsilon\left(\sum_{e'\in G} \alpha_{e'} + \beta_{e'}\right) = \epsilon\,|P|.$$

[4] For illustration of these definitions, we refer back to example 1 of Section V. The computation tree in the center shows the projection M^*_T, and the tree on the right shows a max-weight matching M on $T_{(a,b)}(4)$. Suppose now P is the path starting from the left-most leaf of $T_{(a,b)}(4)$ and ending at the right-most leaf. It alternates between M and M^*_T. For this P, we have that the vectors are: $\alpha_{(a,b)} = 1$, $\alpha_{(c,d)} = 2$, and $\alpha_{e'} = 0$ for all other edges e'. $\beta_{(a,c)} = 1$, $\beta_{(b,c)} = 1$, and $\beta_{e'} = 0$ for all other edges e'. The weights $w_1 = w_2 =$ weight of edge (a, d).

Thus we have that the ℓ_1-norm satisfies $|x - x^*| \geq \epsilon t$. Now, by the definition of c in Lemma 1

$$w'x^* - w'x \geq c|x - x^*| \geq c\epsilon t$$

and thus, $w'(\alpha - \beta) \geq ct$. Also, $w_1 + w_2 \leq 2w_{\max}$. Thus we have that

$$w\left(P \cap M_T^*\right) - w(P \cap M) \geq ct - 2w_{\max}.$$

However, by assumption $t > \frac{2w_{max}}{c}$, and hence it has to be that $w\left(P \cap M_T^*\right) > w(P \cap M)$. This finishes the proof. ■.

Proof of Lemma 4: The proof of this lemma is also a perturbation argument. For each edge e', let $m_{e'}$ denote the number of times e' appears in $P \cap M$ and $n_{e'}$ the number of times it appears in $P \cap E_1^*$. Define

$$x = x^* + \epsilon(m - n).$$

We now show that this x is a feasible point for LP, for small enough ϵ. To do so we have to check edge constraints $0 \leq x_{e'} \leq 1$ and node constraints $\sum_{e' \in E_i} x_{e'} \leq 1$. Consider first the edge constraints. For any $e' \in E_1^* \cap P$, by definition, $x_{e'}^* > 0$. Thus, for any $m_{e'}$ and $n_{e'}$, making ϵ small enough can ensure that $x_{e'}^* + \epsilon(m_{e'} - n_{e'}) \geq 0$. On the other hand, for any $e' \in M \cap P$, $x_{e'}^* < 1$, because a neighboring edge that belongs to E_1^* has positive weight. Making ϵ small enough ensures that $x_{e'}^* + \epsilon(m_{e'} - n_{e'}) \leq 1$.

Consider now the node constraints for a node v. For every copy of v that appears in the interior of P, the mass on one edge is increased by ϵ, and on another is decreased by ϵ. Thus the only nodes where there is a potential for constraint violation are the endpoints of P for which the corresponding last edge is in $P \cap M$. Suppose that v is one such endpoint, and assume for now that v is not a leaf node of $T_e(t)$. Note now that, by construction, every edge in $e' \in P \cap M$ has $x_{e'}^* < 1$. So, the fact that P could not be extended beyond v means that $\sum_{e' \in E_v} x_{e'}^* = x_{uv}^* < 1$, where uv is the edge in P (and M) touching v. This means that the constraint at v is inactive for x^*, and so for small ϵ the new x will be feasible.

The only remaining case to check is if the endpoint v of P is a leaf node of $T_e(t)$. If the last edge in P touching v is in $P \cap E_1^*$, the node constraint at v will not be violated since the perturbation decreases the total mass at v. Note that, since t is odd, this includes the case where v is a leaf node at the lowest level. So, consider the final case that v is a leaf node that is not at the lowest level in the tree, such that P ends in v with an edge in $P \cap M$. This edge has mass strictly less than 1. The fact that v is not at the lowest level means that v is a leaf in the original graph as well, and ha no other edges touching it. Thus it has to be that the constraint at node v is not tight at the LP optimum x^*. This means that a small finite ϵ will ensure feasibility.

Thus x is a feasible point of LP. Note that the weights satisfy

$$w'x - w'x^* = w(P \cap M) - w\left(P \cap E_1^*\right).$$

Thus, if $w(P \cap M) > w\left(P \cap E_1^*\right)$, then we would have that $w'x > w'x^*$, which violates the assumption that x^* is an optimum of LP. So it has to be that $w(P \cap M) \leq w\left(P \cap E_1^*\right)$. This proves the lemma. ■

Proof of Lemma 5: Let m, n and x be defined exactly as in the proof of Lemma 4 above, with E_1^* replaced by E_2^*. By reasoning exactly as above, it follows that all edge constraints $0 \leq x_{e'} \leq 1$ are satisfied, and also all node constraints are satisfied except possibly for nodes v that are endpoints of P which are leafs of $T_e(t)$ and also the last edge e' is in $P \cap M$. However, the fact that the root e is in M, and that t is even, means that last edge $e' \in P \cap E_2^*$ and not in $P \cap M$. Thus x is a feasible point of LP.

Now, as before, we have that $w'x = w'x^* + w(P \cap M) - w\left(P \cap E_2^*\right)$. Thus, if the lemma is not true, it follows that $w'x > w'x^*$, violating the optimality of x^*. The lemma is thus proved. ■

Proof of Lemma 7: We now show that x as defined in (5) is a feasible point of LP, for small enough ϵ. For this we have to show that it satisfies the edge constraints $0 \leq x_{e'} \leq 1$ for all edges $e' \in G$ and the node constraints $\sum_{e' \in E_i} x_{e'} \leq 1$ for all nodes $i \in G$ (here E_i is the set of all edges touching node i)

First the edge constraints. If $e' \in M^*$, then the assumption that x^* is integral means that $x_{e'}^* = 1$, and hence $x_{e'} = 1 - \epsilon\alpha_{e'}$. Thus for small enough ϵ, it will be the case that $0 \leq x_{e'} \leq 1$. On the other hand, if $e' \notin M^*$ then $x_{e'}^* = 0$ and $x_{e'} = \epsilon\beta_{e'}$. Thus, again, a small enough ϵ will ensure $0 \leq x_{e'} \leq 1$.

We now turn to the node constraints. Note that

$$\sum_{e' \in E_i} x_{e'} = \sum_{e' \in E_i} x_{e'}^* + \epsilon\left(\sum_{e' \in E_i} \beta_{e'} - \sum_{e' \in E_i} \alpha_{e'}\right).$$

The term $\sum_{e' \in E_i} \alpha_{e'}$ counts the number of times edges in $P \cap M_T^*$ touch (copies of) node i in the computation tree. Similarly, $\sum_{e' \in E_i} \beta_{e'}$ counts the number of times edges in $P \cap M$ touch i. Suppose first that i is not an endpoint of P, so that every time P touches i it will do so with one edge in M_T^* and one in M. This means that $\sum_{e' \in E_i} \alpha_{e'} = \sum_{e' \in E_i} \beta_{e'}$ and hence that $\sum_{e' \in E_i} x_{e'} = \sum_{e' \in E_i} x_{e'}^*$. Thus the node constraint at i is not violated.

Suppose now that i appears as an endpoint of P, and (i, j) is the corresponding last edge of P. If $(i, j) \in P \cap M_T^*$, this means that $\sum_{e' \in E_i} \beta_{e'} \leq \sum_{e' \in E_i} \alpha_{e'}$, and hence $\sum_{e' \in E_i} x_{e'} \leq \sum_{e \in E_i} x_{e'}^*$—so the constraint at node i is not violated.[5] If last edge $(i, j) \in M$ and it touches a leaf-node then it is not counted in $\beta_{e'}$ (see how β is defined). If $(i, j) \in M$ and it ends in the interior of $T_e(t)$, then the fact that P could not be extended beyond i means that there are no edges of M_T^* touching i in the tree $T_e(t)$. Since M_T^* is the image of M^*, this means there are no edges in M^* touching node i in original graph G. Thus $\sum_{e' \in E_i} x_{e'}^* = 0$. So, for small enough ϵ we can ensure that $\epsilon \sum_{e' \in E_i} (\beta_{e'} - \alpha_{e'}) \leq 1$, ensuring that the constraint at node i is not violated. ■

References

[1] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[2] J. Pearl, *Probabilistic Inference in Intelligent Systems*. Boston, MA: Morgan Kaufmann, 1988.

[5] Note that equality occurs only if i is also the *other* endpoint of P, and the corresponding last edge there is in $P \cap M$.

[3] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring AI in the New Millennium*, pp. 239–269, 2003.

[4] D. Malioutov, J. Johnson, and A. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, Oct. 2006.

[5] Y. Weiss and W. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 736–744, Feb. 2001.

[6] S. Sanghavi, "Equivalence of LP relaxation and max-product for weighted matching in general graphs," in *Proc. Inf. Theory Workshop*, Sep. 2007.

[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 [Online]. Available: http://arxiv.org/abs/0807.3159

[8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, Dec. 1992.

[9] B. Huang and T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching," in *Proc. Artif. Intell. Statist. (AISTATS)*, Mar. 2007.

[10] M. Bayati, D. Shah, and M. Sharma, "Maximum weight matching via max-product belief propagation," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Sep. 2005, pp. 1763–1767.

[11] P. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Proc. ISITA 2004*, Parma, Italy.

[12] S. Tatikonda and M. Jordan, "Loopy belief propagation and Gibbs measures," in *Proc. Loopy belief propagation and Gibbs measures*, 2002, vol. 18, pp. 493–500.

[13] S. Sanghavi, D. Shah, and A. Willsky, "Message passing for maximum weight independent set," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2007.

[14] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches," *IEEE Trans. Inf. Theory*, vol. 51, pp. 3697–3717, Nov. 2005.

[15] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends in Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, Dec. 2008.

[16] V. N. Kolmogorov and M. J. Wainwright, "On optimality of tree-reweighted max-product message-passing," in *Proc. Uncert. Artif. Intell.*, Edinburgh, Scotland, Jul. 2005.

[17] A. Schrijver, *Combinatorial Optimization*. Berlin-Heidelberg, Germany: Springer-Verlag, 2003.

Sujay Sanghavi (S'02–M'06) received the M.S. degree in both mathematics and electrical and computer engineering, from the University of Illinois, Urbana-Champaign, and the Ph.D. degree in electrical and computer engineering, in 2006.

He is an Assistant Professor of electrical and computer engineering with the University of Texas, Austin, since July 2009. He spent 2006–2008, as postdoctoral scholar with the Laboratory for Information and Decision Systems (LIDS), MIT, and 2008–2009 as an Assistant Professor of electrical and computer engineering at Purdue University, West Lafayette, IN. His interests lie at the intersection of networks, networking, and statistical machine learning.

Dr. Sanghavi received the NSF CAREER award in 2010.

Dmitry Malioutov received the B.S. degree in electrical and computer engineering from Northeastern University, Boston, MA, in 2001, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2003 and 2008, respectively.

After a postdoctoral position with the Machine Learning and Perception Group, Microsoft Research, Cambridge, U.K., he is currently a Researcher in Algorithmic Training at DRW, Chicago, IL. His research interests include statistical signal and image processing, machine learning, and convex optimization with emphasis on graphical models, message passing algorithms, and sparse signal representation.

Alan Willsky (S'70–M'73–SM'82–F'86) joined M.I.T. in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering and Director of the Laboratory for Information and Decision Systems. He was a founder of Alphatech, Inc., and Chief Scientific Consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998–2002, he served on the U.S. Air Force Scientific Advisory Board. He has delivered numerous keynote addresses and is co-author of the text *Signals and Systems*. His research interests are in the development and application of advanced methods of estimation, machine learning, and statistical signal and image processing.

Dr. Willsky has received a number of awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, Doctorat Honoris Causa from Université de Rennes in 2005, and the 2009 Technical Achievement Award from the IEEE Signal Processing Society. In 2010, he was elected to the National Academy of Engineering. He and his students have also received a variety of Best Paper Awards at various conferences and for papers in journals, including the 2001 IEEE Conference on Computer Vision and Pattern Recognition, the 2003 Spring Meeting of the American Geophysical Union, the 2004 Neural Information Processing Symposium, Fusion 2005, and the 2008 award from the journal Signal Processing for the outstanding paper in the year 2007.