Message Passing for Maximum Weight Independent Set

Sujay Sanghavi, *Member, IEEE*, Devavrat Shah, and Alan S. Willsky, *Fellow, IEEE*

Abstract—In this paper, we investigate the use of message-passing algorithms for the problem of finding the max-weight independent set (MWIS) in a graph. First, we study the performance of the classical loopy max-product belief propagation. We show that each fixed-point estimate of max product can be mapped in a natural way to an extreme point of the linear programming (LP) polytope associated with the MWIS problem. However, this extreme point may not be the one that maximizes the value of node weights; the particular extreme point at final convergence depends on the initialization of max product. We then show that if max product is started from the natural initialization of uninformative messages, it always solves the correct LP, if it converges. This result is obtained via a direct analysis of the iterative algorithm, and cannot be obtained by looking only at fixed points. The tightness of the LP relaxation is thus necessary for max-product optimality, but it is not sufficient. Motivated by this observation, we show that a simple modification of max product becomes gradient descent on (a smoothed version of) the dual of the LP, and converges to the dual optimum. We also develop a message-passing algorithm that recovers the primal MWIS solution from the output of the descent algorithm. We show that the MWIS estimate obtained using these two algorithms in conjunction is correct when the graph is bipartite and the MWIS is unique. Finally, we show that any problem of maximum *a posteriori* (MAP) estimation for probability distributions over finite domains can be reduced to an MWIS problem. We believe this reduction will yield new insights and algorithms for MAP estimation.

Index Terms—Belief propagation, combinatorial optimization, distributed algorithms, independent set, iterative algorithms, linear programming (LP), optimization.

I. INTRODUCTION

THE max-weight independent set (MWIS) problem is the following: given a graph with positive weights on the nodes, find the heaviest set of mutually nonadjacent nodes. MWIS is a well-studied combinatorial optimization problem that naturally arises in many applications. It is known to be NP-hard, and hard to approximate [6]. In this paper, we investigate the use of message-passing algorithms, like loopy max-product belief propagation, as practical solutions for the MWIS problem. We now summarize our motivations for doing so, and then outline our contribution.

Our primary motivation comes from applications. The MWIS problem arises naturally in many scenarios involving resource allocation in the presence of interference. It is often the case that large instances of the weighted independent set problem need to be (at least approximately) solved in a distributed manner using lightweight data structures. In Section II-A, we describe one such application: scheduling channel access and transmissions in wireless networks. Message-passing algorithms provide a promising alternative to current scheduling algorithms.

Another, equally important, motivation is the potential for obtaining new insights into the performance of existing message-passing algorithms, especially on loopy graphs. Tantalizing connections have been established between such algorithms and more traditional approaches like linear programming (LP; see [1], [2], [12], and references therein). We consider MWIS problem to understand this connection as it provides a rich (it is NP-hard), yet relatively (analytically) tractable, framework to investigate such connections.

A. Related Work

The design of message-passing algorithms for LP relaxations for combinatorial optimization problems have been of interest for a while now. For example, the auction algorithm by Bertsekas [27] attempts to design message-passing algorithm for the assignment problem by means of an approximate primal-dual algorithm, which is in turn based on the dual coordinate descent algorithm. More recently, Wainwright, Jaakkola, and Willsky [8] proposed a tree reweighted (TRW) algorithm—a generalization of the max-product algorithm. They showed that fixed points of their algorithm that satisfied a further property, strong tree agreement (STA), will correspond to the optimum of a (certain) LP relaxation of the maximum *a posteriori* (MAP) estimation problem. In subsequent work, Kolmogorov [4] provided a counter example to show that the correspondence between fixed points of TRW and the solution of LP relaxation may not hold in general. However, Kolmogorov and Wainwright [20] established that for binary problems, such as the problem of interest in this paper, the correspondence will always hold; i.e., the fixed points of the TRW algorithm always correspond to solution of the LP relaxation. However, this still does not guarantee that TRW will converge to the fixed point.

S. Sanghavi is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906 USA (e-mail: sanghavi@purdue.edu).

D. Shah and A. S. Willsky are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: devavrat@mit.edu; willsky@mit.edu).

In work by Kolmogorov [4], a subsequential convergence property of TRW was established under a modified (or sequential) "scheduling of message passing." That is, the subsequential limit point of the algorithm will satisfy what is known as the weak tree agreement (WTA) condition. For binary problems, this will mean that such a subsequential limit point will correspond to solution of LP relaxation.

Specialized to the case of MWIS, a combination of these two results will imply the following: under the modified scheduling of the TRW, there exists a limit point (which may or may not be identifiable) of the algorithm that corresponds to solving the LP relaxation of the problem. Therefore, when the LP relaxation is tight and has unique integral solution, then this will yield the MWIS.

The focus of this paper is somewhat different. Unlike many of the above approaches where the algorithm is *designed* to solve the corresponding LP relaxation, we investigate whether there is any connection between the *original* max-product algorithm—which at best can be viewed as tree-based approximation dynamic programming—and LP relaxation. Along these lines, a series of recent works [1], [2], [13] lead to the conclusion that for the problem of b-matching, indeed the max product is as powerful as (certain) LP relaxation. We refer an interested reader to a recent monograph on a related topic by Wainwright and Jordan [10].

B. Our Contributions

To begin with, we formally describe the MWIS problem, formulate it as an integer program, and present its natural LP relaxation. We also describe how the MWIS problem arises in wireless network scheduling (see Section II).

Next, we describe how max product can be used (as a heuristic) for solving the MWIS problem. Specifically, we construct a probability distribution whose MAP estimate is the MWIS of the given graph. Max product, which is a heuristic for finding MAP estimates, emerges naturally from this construction (see Section III).

Now, max product is an iterative algorithm, and is typically executed until it converges to a fixed point. In Section IV, we show that fixed points always exist, and characterize their structure. Specifically, we show that there is a one-to-one map between estimates of fixed points, and extreme points of the independent set LP polytope. This polytope is defined only by the graph, and each of its extrema corresponds to the LP optimum for a different node weight function. This implies that max-product fixed points attempt to solve (the LP relaxation of) an MWIS problem on the correct graph, but with different (possibly incorrect) node weights. This stands in contrast to its performance for the weighted matching problem [1], [2], [13], for which it is known to *always* solve the LP with correct weights.

Since max product is a deterministic algorithm, the particular fixed point (if any) that is reached depends on the initialization. In Section V, we pursue an alternative line of analysis, and directly investigate the performance of the iterative algorithm itself, started from the "natural" initialization of uninformative messages. For this case, we show that max-product estimates exactly correspond to the *true* LP, at all times, not just the fixed point.

Max product bears a striking semantic similarity to dual coordinate descent on the LP. With the intention of modifying max product to make it as powerful as LP, in Section VI, we develop two iterative message-passing algorithms. The first, obtained by a minor modification of max product, approximately calculates the optimal solution to the dual of the LP relaxation of the MWIS problem. It does this via coordinate descent on a convexified version of the dual. The second algorithm uses this approximate optimal dual to produce an estimate of the MWIS. This estimate is correct when the original graph is bipartite. We believe that this algorithm should be of broader interest. We note that, to the best of our knowledge, this is the first iterative/message-passing algorithm for solving MWIS on weighted bipartite graph with provable convergence and correctness guarantees. This result stands in contrast with the fact that the modified TRW of Kolmogorov [4] along with analysis of Kolmogorov and Wainwright [20] only yields "subsequential convergence" guarantee; it is not clear if such a convergence can be indeed verified (at least not clear to the authors).

The above uses of max product for MWIS involved posing the MWIS as a MAP estimation problem. In the final Section VII, we do the reverse: we show how *any* MAP estimation problem on finite domains can be converted into an MWIS problem on a suitably constructed auxiliary graph. This implies that any algorithm for solving the independent set problem immediately yields an algorithm for MAP estimation. This reduction may prove useful from both practical and analytical perspectives.

II. MAX-WEIGHT INDEPENDENT SET AND ITS LP RELAXATION

Consider a graph $G = (V, E)$, with a set V of nodes and a set E of edges. Let $\mathcal{N}(i) = \{j \in V : (i,j) \in E\}$ be the neighbors of $i \in V$. Positive weights $w_i, i \in V$ are associated with each node. A subset of V will be represented by vector $\mathbf{x} = (x_i) \in \{0,1\}^{|V|}$, where $x_i = 1$ means i is in the subset and $x_i = 0$ means i is not in the subset. A subset $\mathbf{x}$ is called an *independent set* if no two nodes in the subset are connected by an edge: $(x_i, x_j) \neq (1,1)$ for all $(i,j) \in E$. We are interested in finding an MWIS $\mathbf{x}^*$. This can be naturally posed as an integer program, denoted below by IP. The *linear programing relaxation* of IP is obtained by replacing the integrality constraints $x_i \in \{0,1\}$ with the constraints $x_i \geq 0$. We will denote the corresponding linear program by LP. The dual of LP is denoted below by DUAL

$$
\begin{aligned}
\text{IP}: \quad &\max \quad \sum_{i=1}^{n} w_i x_i \\
\text{s.t.} \quad &x_i + x_j \leq 1, \qquad \text{for all } (i,j) \in E \\
&x_i \in \{0,1\}
\end{aligned}
$$

$$
\begin{aligned}
\text{LP}: \quad &\max \quad \sum_{i=1}^{n} w_i x_i \\
\text{s.t.} \quad &x_i + x_j \leq 1, \qquad \text{for all } (i,j) \in E \\
&x_i \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\text{DUAL}: \quad &\min \quad \sum_{(i,j)\in E} \lambda_{ij} \\
\text{s.t.} \quad &\sum_{j\in\mathcal{N}(i)} \lambda_{ij} \geq w_i, \qquad \text{for all } i \in V \\
&\lambda_{ij} \geq 0, \qquad \text{for all } (i,j) \in E.
\end{aligned}
$$

It is well known that LP can be solved efficiently, and if it has an integral optimal solution then this solution is an MWIS of G. If this is the case, we say that there is no *integrality gap* between LP and IP, or equivalently, that the LP relaxation is *tight*.

A. Properties of the LP

We now briefly state some of the well-known properties of the MWIS LP, as these will be used/referred to in the paper. The polytope of the LP is the set of feasible points for the linear program. An extreme point of the polytope is one that cannot be expressed as a convex combination of other points in the polytope.

Lemma 2.1 [16, Th. 64.7]: The LP polytope has the following properties:
  1) for any graph, the MWIS LP polytope is half-integral: any extreme point will have each $x_i = 0, 1$ or $\frac{1}{2}$;
  2) for bipartite graphs, the LP polytope is integral: each extreme point will have $x_i = 0$ or 1.

Half-integrality is an intriguing property that holds for LP relaxations of a few combinatorial problems (e.g., vertex cover, matchings, etc.). Half-integrality implies that any extremum optimum of LP will have some nodes set to 1, and all their neighbors set to 0. The nodes set to $\frac{1}{2}$ will appear in clusters: each such node will have at least one other neighbor also set to $\frac{1}{2}$. We will see later that a similar structure arises in max-product fixed points.

Lemma 2.2 [22, Corollary 64.9a]: LP optima are *partially correct*: for any graph, any LP optimum x^* and any node i, if the mass x_i^* is integral then there exists an MWIS for which that node's membership is given by x_i^*.

The next lemma states the standard complimentary slackness conditions of LP, specialized for the MWIS LP, and for the case when there is no integrality gap.

Lemma 2.3: When there is no integrality gap between IP and LP, there exists a pair of optimal solutions $\mathbf{x} = (x_i)$, $\lambda = (\lambda_{ij})$ of LP and DUAL, respectively, such that: a) $\mathbf{x} \in \{0,1\}^n$, b) $x_i(\sum_{j \in \mathcal{N}(i)} \lambda_{ij} - w_i) = 0$ for all $i \in V$, and c) $(x_i + x_j - 1)\lambda_{ij} = 0$, for all $(i,j) \in E$.

B. Sample Application: Scheduling in Wireless Networks

We now briefly describe an important application that requires an efficient, distributed solution to the MWIS problem: transmission scheduling in wireless networks that lack a centralized infrastructure, and where nodes can only communicate with local neighbors (e.g., see [19]). Such networks are ubiquitous in the modern world: examples range from sensor networks that lack wired connections to the fusion center, and ad hoc networks that can be quickly deployed in areas without coverage, to the 802.11 wi-fi networks that currently represent the most widely used method for wireless data access.

Fundamentally, any two wireless nodes that transmit at the same time and over the same frequencies will interfere with each other, if they are located close by. Interference means that the intended receivers will not be able to decode the transmissions. Typically in a network only certain pairs of nodes interfere. The

scheduling problem is to decide which nodes should transmit at a given time over a given frequency, so that 1) there is no interference, and 2) nodes which have a large amount of data to send are given priority. In particular, it is well known that if each node is given a *weight* equal to the data it has to transmit, optimal network operation demands scheduling the set of nodes with highest total weight. If a " conflict graph" is made, with an edge between every pair of interfering nodes, the scheduling problem is *exactly* the problem of finding the MWIS of the conflict graph. The lack of an infrastructure, the fact that nodes often have limited capabilities, and the local nature of communication, all necessitate a lightweight distributed algorithm for solving the MWIS problem.

III. MAX-PRODUCT FOR MWIS

The classical max-product algorithm is a heuristic that can be used to find the MAP assignment of a probability distribution. Now, given an MWIS problem on $G = (V, E)$, associate a binary random variable X_i with each $i \in V$ and consider the following joint distribution: for $\mathbf{x} \in \{0,1\}^n$

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{(i,j) \in E} \mathbf{1}_{\{x_i + x_j \leq 1\}} \prod_{i \in V} \exp(w_i x_i) \qquad (1)$$

where Z is the normalization constant. In the above, $\mathbf{1}$ is the standard indicator function: $\mathbf{1}_{\text{true}} = 1$ and $\mathbf{1}_{\text{false}} = 0$. It is easy to see that $p(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_i w_i x_i\right)$ if $\mathbf{x}$ is an independent set, and $p(\mathbf{x}) = 0$ otherwise. Thus, any MAP estimate $\arg\max_{\mathbf{x}} p(\mathbf{x})$ corresponds to an MWIS of G.

The update equations for max product can be derived in a standard and straightforward fashion from the probability distribution. We now describe the max-product algorithm as derived from p. At every iteration t, each node i sends a *message* $\{m_{i \to j}^t(0), m_{i \to j}^t(1)\}$ to each neighbor $j \in \mathcal{N}(i)$. Each node also maintains a *belief* $\{b_i^t(0), b_i^t(1)\}$ vector. The message and belief updates, as well as the final output, are computed as follows.

Max-Product for MWIS

(o) Initially, $m_{i \to j}^0(0) = m_{j \to i}^0(1) = 1$ for all $(i,j) \in E$.

(i) The messages are updated as follows:

$$m_{i \to j}^{t+1}(0) = \max \left\{ \prod_{k \neq j, k \in \mathcal{N}(i)} m_{k \to i}^t(0), \right.$$
$$\left. e^{w_i} \prod_{k \neq j, k \in \mathcal{N}(i)} m_{k \to i}^t(1) \right\}$$
$$m_{i \to j}^{t+1}(1) = \prod_{k \neq j, k \in \mathcal{N}(i)} m_{k \to i}^t(0).$$

(ii) Nodes $i \in V$, compute their beliefs as follows:

$$b_i^t(0) = \prod_{k \in \mathcal{N}(i)} m_{k \to i}^t(0)$$
$$b_i^t(1) = e^{w_i} \prod_{k \in \mathcal{N}(i)} m_{k \to i}^t(1).$$

(iii) Estimate MWIS $\mathbf{x}(b^{t+1})$ as follows:

$$
\begin{array}{lll}
x_i(b_i^t) = 1 & \text{if} & b_i^t(1) > b_i^t(0) \\
x_i(b_i^t) = 0 & & b_i^t(1) < b_i^t(0) \\
x_i(b_i^t) = ? & & b_i^t(1) = b_i^t(0)
\end{array}
$$

(iv) Update $t = t + 1$; repeat from (i) until $\mathbf{x}(b^t)$ converges and output the converged estimate.

For the purpose of analysis, we find it convenient to transform the messages and their dynamics as follows. First, define

$$
\gamma_{i \to j}^t = \log\left(\frac{m_{i \to j}^t(0)}{m_{i \to j}^t(1)} \right).
$$

Here, since the algorithm starts with all messages being strictly positive, the messages will remain strictly positive over any finite number of iterations. Therefore, taking logarithm is a valid operation. With this new definition, step (i) of the max-product becomes

$$
\gamma_{i \to j}^{t+1} = \left(w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}^t \right)_+ \tag{2}
$$

where we use the notation $(x)_+ = \max\{x, 0\}$. The final estimation step (iii) of max product takes the following form:

$$
x_i(\gamma^t) = 1 \quad \text{if} \quad w_i > \sum_{k \in \mathcal{N}(i)} \gamma_{k \to i}^t \tag{3}
$$

$$
x_i(\gamma^t) = 0 \qquad w_i < \sum_{k \in \mathcal{N}(i)} \gamma_{k \to i}^t \tag{4}
$$

$$
x_i(\gamma^t) = ? \qquad w_i = \sum_{k \in \mathcal{N}(i)} \gamma_{k \to i}^t. \tag{5}
$$

This modification of max product is often known as the "min-sum" algorithm, and is just a reformulation of the max product. In the rest of this paper, we refer to this as simply the max-product algorithm.

IV. FIXED POINTS OF MAX PRODUCT

When applied to general graphs, max product may either 1) not converge, 2) converge, and yield the correct answer, or 3) converge but yield an incorrect answer. Characterizing when each of the three situations can occur is a challenging and important task. One approach to this task has been to look directly at the fixed points, if any, of the iterative procedure (see, e.g., [11]). In this section, we investigate properties of fixed points, by formally establishing a connection to the LP polytope.

Note that a set of messages γ^* is a fixed point of max product if, for all $(i, j) \in E$

$$
\gamma_{i \to j}^* = \left(w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}^* \right)_+. \tag{6}
$$

The following lemma establishes that fixed points always exist. We note that such arguments have been used in literature in the context of establishing existence of fixed points (e.g., see [7]).

Lemma 4.1: There exists at least one fixed point γ^* such that $\gamma_{i \to j}^* \in [0, w_i]$ for each $(i, j) \in E$

Proof: Let $w^* = \max_i w_i$, and suppose at time t each $\gamma_{i \to j}^t \in [0, w^*]$. From (2), it is clear that this will result in the messages γ^{t+1} at the next time also having each $\gamma_{i \to j}^{t+1} \in [0, w^*]$. Thus, the max-product update rule (2) maps a message vector $\gamma^t \in [0, w^*]^{2|E|}$ into another vector in $[0, w^*]^{2|E|}$. Also, it is easy to see that (2) is a continuous function. Therefore, by Brouwer's fixed-point theorem, there exists a fixed point $\gamma^* \in [0, w^*]^{2|E|}$. ☐

We now study properties of the fixed points in order to understand the correctness of the estimate output by max product. The following theorem characterizes the structure of estimates at fixed points. Recall that the estimate $x_i(\gamma^*)$ for node i can be $0, 1$, or $?$.

Theorem 4.1: Let γ^* be a fixed point, and let $\mathbf{x}(\gamma^*) = (x_i(\gamma^*))$ be the corresponding estimate. Then:
  1) if $x_i(\gamma^*) = 1$, then every neighbor $j \in \mathcal{N}(i)$ has $x_j(\gamma^*) = 0$;
  2) if $x_i(\gamma^*) = 0$, then at least one neighbor $j \in \mathcal{N}(i)$ has $x_j(\gamma^*) = 1$;
  3) if $x_i(\gamma^*) = ?$, then at least one neighbor $j \in \mathcal{N}(i)$ has $x_j(\gamma^*) = ?$.

Before proving Theorem 4.1, we discuss its implications. Recall from Lemma 2.1 that every extreme point of the LP polytope consists of each node having a value of $0, 1$, or $\frac{1}{2}$. If all weights are positive, the optimum of LP will have the following characteristics: every node with value 1 will be surrounded by nodes with value 0, every node with value 0 will have at least one neighbor with value 1, and every node with value $\frac{1}{2}$ will have one neighbor with value $\frac{1}{2}$. These properties bear a remarkable similarity to those in Theorem 4.1. Indeed, given a fixed point γ^* and its estimates $\mathbf{x}(\gamma^*)$, make a vector $\mathbf{y}$ by setting

$$
\begin{array}{lll}
y_i = \frac{1}{2} & \text{if estimate for } i \text{ is} & x_i(\gamma^*) = ? \\
y_i = 1 & & x_i(\gamma^*) = 1 \\
y_i = 0 & & x_i(\gamma^*) = 0.
\end{array}
$$

Then, Theorem 4.1 implies that $\mathbf{y}$ will be an extreme point of the LP polytope, and also one that maximizes *some* weight function consisting of positive node weights. Note, however, that this *may not* be the true weights w_i. In other words, given any MWIS problem with graph G and weights w, each max-product fixed point represents the optimum of the LP relaxation of some MWIS problem on the same graph G, but possibly with different weights $\hat{w}$.

The fact that max-product estimates optimize a different weight function means that both eventualities are possible: LP giving the correct answer but max product failing, and *vice versa*. We now provide simple examples for each one of these situations.

Figs. 1 and 2 present graphs and the corresponding fixed points of max product. In each graph, numbers represent node weights, and an arrow from i to j represents a message value
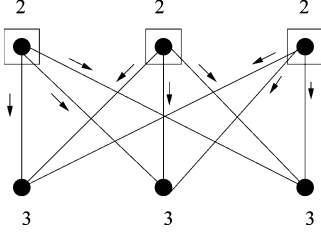
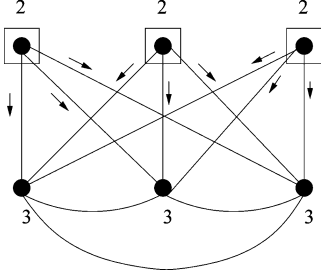Fig. 1. This example shows that max-product fixed point may result in an incorrect answer even though LP is tight.

Fig. 2. This example shows that max-product fixed point can find right MWIS even though LP relaxation is not tight.

of $\gamma^*_{i \to j} = 2$. All other messages, which do not have arrows, have value zero. The boxed nodes indicate the ones for which the estimate $x_i(\gamma^*) = 1$. It is easy to verify that both examples represent max-product fixed points.

For the graph in Fig. 1, the max-product fixed point results in an incorrect estimate. However, the graph is bipartite, and hence LP will provide the correct answer. For the graph in Fig. 2, there is an integrality gap between LP and IP: setting each $x_i = \frac{1}{2}$ yields an optimal value of 7.5 for LP, while the optimal solution to IP has value 6. Note that the estimate at the fixed point of max product is the correct MWIS. It is also worth noticing that for both of these examples, the fixed points lie in the strict interiors of a nontrivial region of attraction: starting the iterative procedure from within these regions will result in convergence to the corresponding fixed point. These examples indicate that it may not be possible to resolve the question of relative strength of the two procedures based solely on an analysis of the fixed points of max product.

The particular fixed point, if any, that max product converges to depends on the initialization of the messages; each fixed point will have its own region of convergence. In Section V, we directly analyze the iterative algorithm when started from the "natural" initialization of unbiased messages. As a byproduct of this analysis, we prove that if max product from this initialization converges, then the resulting fixed-point estimate is the optimum of LP; thus, in this case, the max-product fixed point solves the "correct" LP.

Proof of Theorem 4.1: The proof of Theorem 4.1 follows from manipulations of the fixed point (6). For ease of notation, we replace γ^* by γ. We first prove the following statements on how the estimates determine the relative ordering of the two messages (one in each direction) on any given edge:

$$x_i(\gamma) = 1 \Rightarrow \gamma_{i \to j} > \gamma_{j \to i} \qquad \forall j \in \mathcal{N}(i) \qquad (7)$$

$$x_i(\gamma) = ? \Rightarrow \gamma_{i \to j} = \gamma_{j \to i} \qquad \forall j \in \mathcal{N}(i). \qquad (8)$$

The above equations cover every case except for edges between two nodes with 0 estimates. This is covered by the following:

$$x_i(\gamma) = 0 \text{ and } x_j(\gamma) = 0 \Rightarrow \gamma_{i \to j} = \gamma_{j \to i} = 0. \qquad (9)$$

Suppose first that i is such that $x_i(\gamma^*) = 1$. By definition (6) of the fixed point

$$\gamma_{i \to j} \geq w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}.$$

However, by (3), the fact that $x_i(\gamma) = 1$ implies that

$$w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i} > \gamma_{j \to i}.$$

Putting the above two equations together proves (7). The proof of (8) is along similar lines. Suppose now i is such that $x_i(\gamma) = ?$. By (5), this implies that $w_i = \sum_{k \in \mathcal{N}(i)} \gamma_{k \to i}$, and so from (6), we have that

$$\gamma_{i \to j} = w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}.$$

Also, the fact that $x_i(\gamma) = ?$ means that

$$w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i} = \gamma_{j \to i}.$$

Putting the above two equations together proves (8). We now prove the three parts of Theorem 4.1.

Proof of Part 1): Let i have estimate $x_i(\gamma) = 1$, and suppose there exists a neighbor $j \in \mathcal{N}(i)$ such that $x_j(\gamma) = ?$ or 1. Then, from (7), it follows that $\gamma_{i \to j} > \gamma_{j \to i}$, and from (8), it further follows that $\gamma_{i \to j} \leq \gamma_{j \to i}$. However, this is a contradiction, and thus every neighbor of i has to have estimate 0.

Proof of Part 2): Let i have estimate $x_i(\gamma) = 0$. Since $w_i \geq 0$, (4) implies that there exists at least one neighbor $j \in \mathcal{N}(i)$ such that the message $\gamma_{j \to i} > 0$. From (9), this means that the estimate $x_j(\gamma)$ cannot be 0. Suppose now that $x_j(\gamma) = ?$. From (7), it follows that $\gamma_{i \to j} = \gamma_{j \to i} > 0$, and so

$$\gamma_{i \to j} = w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}.$$

However, since $\gamma_{i \to j} = \gamma_{j \to i}$, this means that

$$\gamma_{j \to i} = w_i - \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i}$$

which violates (4), and thus the assumption that $x_i(\gamma) = 0$. Thus, it has to be that $x_i(\gamma) = 1$.

Proof of Part 3): Let i have estimate $x_i(\gamma) = ?$. Since $w_i \geq 0$, (5) implies that there exists at least one neighbor $j \in \mathcal{N}(i)$ such that the message $\gamma_{j \to i} > 0$. From (8), it follows that

$$\gamma_{i \to j} = \gamma_{j \to i} = w_j - \sum_{l \neq i} \gamma_{l \to j}.$$

Thus, $w_j = \sum_l \gamma_{l \to j}$, which by (5) means that $x_j(\gamma) = ?$. Thus, i has at least one neighbor j with estimate $x_j(\gamma) = ?$. $\square$

We end this section with a brief discussion about the half-interality property of the MWIS problem, as summarized by Lemma 2.1. For us, this property enabled a natural interpretation of the "?" estimates at a max-product fixed point: we simply set those nodes to $\frac{1}{2}$. It would be interesting to see if such an interpretation also holds for other problems with known half-integrality properties. That is, given a max-product fixed point for one of these problems, does interpreting "?" estimates as an LP mass of $\frac{1}{2}$ yield an extreme point of the LP polytope? A general answer to this question would be interesting.

V. DIRECT ANALYSIS OF THE ITERATIVE ALGORITHM

In the last section, we saw that fixed points of max product may correspond to optima "wrong" linear programs: ones that operate on the same feasible set as LP, but optimize a different linear function. However, there will also be fixed points that correspond to optimizing the correct function. Max product is a deterministic algorithm, and so which of these fixed points (if any) are reached is determined by the initialization. In this section, we directly analyze the iterative algorithm itself, as started from the "natural" initialization $\gamma = 0$, which corresponds to uninformative messages

We show that the resulting estimates are characterized by optima of the *true* LP, at every instant (not just at fixed points). This implies that, if a fixed point is reached, it will exactly reflect an optimum of LP. Our main theorem in this section is stated below.

Theorem 5.1: Given any MWIS problem on weighted graph G, suppose max product is started from the initial condition $\gamma = 0$. Then, for any node $i \in G$:
1) if there exists *any* optimum x^* of LP for which the $x_i^* < 1$, then the max-product estimate $x_i(\gamma^t)$ is 0 or ? for all *even* times t;
2) if there exists *any* optimum x^* of LP for which the $x_i^* > 0$, then the max-product estimate $x_i(\gamma^t)$ is 1 or ? for all *odd* times t.

We make note of two important implications of the Theorem 5.1.
1) *If LP has a nonintegral solution, then the max-product estimates will not converge to the correct answer.* This is because, if $x_i^* \in (0, 1)$, then by above theorem, the estimate of i will either keep varying every alternate time slot, or will converge to ?. Either way, max product will fail to provide a useful estimate for node i.
2) *A stopping condition: stop when estimate is the same (and 0/1) for two consecutive time slots.* This is because, by statement of theorem it follows that if the estimates under max product at two consecutive times are $0, 0$ (or $1, 1$), then the solution of all the LP optima must be such that $x_i^* = 0$ (or $x^* = 1$).

The proof of this theorem relies on the computation tree interpretation of max-product estimates. We now specify this interpretation for our problem, and then prove Theorem 5.1.

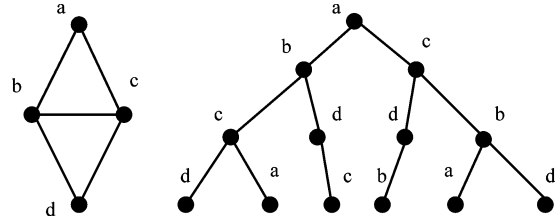### A. Computation Tree for MWIS

The proof of Theorem 5.1 relies on the computation tree interpretation [23], [26] of the loopy max-product estimates. In this section, we briefly outline this interpretation. For any node i, the *computation tree* at time t, denoted by $T_i(t)$, is defined recursively as follows: $T_i(1)$ is just the node i. This is the *root* of the tree, and in this case, it is also its only leaf. The tree $T_i(t)$ at time t is generated from $T_i(t-1)$ by adding to each leaf of $T_i(t-1)$ a copy of each of its neighbors in G, *except for the one neighbor that is already present in $T_i(t-1)$*. Each node in T_i is a copy of a node in G, and the weights of the nodes in T_i are the same as the corresponding nodes in G. The computation tree interpretation is stated in the following lemma.

Lemma 5.1: For any node i at time t:
- $x_i(\gamma^t) = 1$ if and only if the root of $T_i(t)$ is a member of *every* MWIS on $T_i(t)$;
- $x_i(\gamma^t) = 0$ if and only if the root of $T_i(t)$ is not a member of *any* MWIS on $T_i(t)$;
- $x_i(\gamma^t) = ?$ else.

Thus, the max-product estimates correspond to MWISs on the computation trees $T_i(t)$, as opposed to on the original graph G.

Example: Consider the following figure:

On the left is the original loopy graph G. On the right is $T_a(4)$, the computation tree for node a at time 4.

Proof of Theorem 5.1: We now prove Theorem 5.1. For brevity, in this proof, we will use the notation $\hat{x}_i^t = x_i(\gamma^t)$ for the estimates. Suppose now that part 1 of the theorem is not true, i.e., there exists node i, an optimum x^* of LP with $x_i^* > 0$, and an odd time t at which the estimate is $\hat{x}_i^t = 0$. Let $T_i(t)$ be the corresponding computation tree. Using Lemma 5.1, this means that the root i is *not* a member of any MWIS of $T_i(t)$. Let I be some MWIS on $T_i(t)$. We now define the following set of nodes:

$$I^* = \left\{ j \in T_i(t) : j \notin I, \text{ and copy of } j \text{ in } G \text{ has } x_j^* > 0 \right\}.$$

In other words, I^* is the set of nodes in $T_i(t)$, which are not in I, and whose copies in G are assigned strictly positive mass by the LP optimum x^*.

Note that by assumption the root $i \in I^*$ and $i \notin I$. Now, from the root, recursively build a *maximal alternating subtree* S as follows: first add root i, which is in $I^* - I$. Then, add all neighbors of i that are in $I - I^*$. Then, add all *their* neighbors in $I^* - I$, and so on. The building of S stops either when it hits the bottom level of the tree, or when no more nodes can be added while still maintaining the alternating structure. Note the following properties of S.
- S is the disjoint union of $(S \cap I)$ and $(S \cap I^*)$.

- For every $j \in S \cap I$, all its neighbors in I^* are included in $S \cap I^*$. Similarly, for every $j \in S \cap I^*$, all its neighbors in I are included in $S \cap I$.
- Any edge (j, k) in $T_i(t)$ has at most one endpoint in $(S \cap I)$, and at most one in $(S \cap I^*)$.

We now state a lemma, which we will prove later. The proof uses the fact that t is odd.

Lemma 5.2: The weights satisfy $w(S \cap I) \leq w(S \cap I^*)$.

We now use this lemma to prove the theorem. Consider the set I', which changes I by flipping S

$$I' = I - (S \cap I) + (S \cap I^*).$$

We first show that I' is also an independent set on $T_i(t)$. This means that we need to show that every edge (j, k) in $T_i(t)$ touches at most one node in I'. There are thus three possible scenarios for edge (j, k).

- $j, k \notin S$. In this case, membership of j, k in I' is the same as in I, which is an independent set. So (j, k) has at most one node touching I'.
- One node $j \in S \cap I$. In this case, $j \notin I'$, and hence again at most one of j, k belongs to I'.
- One node $k \in S \cap I^*$ but other node $j \notin S \cap I$. This means that $j \notin I$, because every neighbor of k in I should be included in $S \cap I$. This means that $j \notin I'$, and hence only node $k \in I'$ for edge (j, k).

Thus, I' is an independent set on $T_i(t)$. Also, by Lemma 5.2, we have that

$$w(I') \geq w(I).$$

However, I is an MWIS, and hence it follows that I' is also an MWIS of $T_i(t)$. However, by construction, root $i \in I'$, which violates the fact that $\hat{x}_i(t) = 0$. The contradiction is thus established, and part 1 of the theorem is proved. Part 2 is proved in a similar fashion. $\qquad\square$

Proof of Lemma 5.2: The proof of this lemma involves a perturbation argument on the LP. For each node $j \in G$, let m_j denote the number of times j appears in $S \cap I$ and n_j the number of times it appears in $S \cap I^*$. Define

$$x = x^* + \epsilon(m - n). \tag{10}$$

We now show state a lemma that is proved immediately following this one.

Lemma 5.3: x is a feasible point for LP, for small enough ϵ.

We now use this lemma to finish the proof of Lemma 5.2. Since x^* is an optimum of LP, it follows that $w'x \leq w'x^*$, and so $w'm \leq w'n$. However, by definition, $w'm = w(S \cap I)$ and $w'n = w(S \cap I^*)$. This finishes the proof. $\qquad\square$

Proof of Lemma 5.3: We now show that this x as defined in (10) is a feasible point for LP, for small enough ϵ. To do so we have to check node constraints $x_j \geq 0$ and edge constraints $x_j + x_k \leq 1$ for every edge $(j, k) \in G$. Consider first the node constraints. Clearly, we only need to check them for any j which has a copy $j \in I^* \cap S$. If this is so, then by the definition (V) of I^*, $x_j^* > 0$. Thus, for any m_j and n_j, making ϵ small enough can ensure that $x_j^* + \epsilon(m_j - n_j) \geq 0$.

Before we proceed to checking the edge constraints, we make two observations. Note that for any node j in the tree, $j \in S \cap I$, then we have the following.

- $x_j^* < 1$, i.e., the mass x_j^* put on j by the LP optimum x^* is *strictly* less than 1. This is because of the alternating way in which the tree is constructed: a node j in the tree is included in $S \cap I$ *only if* the parent p of j is in $S \cap I^*$ (note that the root $i \in S \cap I^*$ by assumption). However, from the definition of I^*, this means that $x_p^* > 0$, i.e., the parent has positive mass at the LP optimum x^*. This means that $x_j^* < 1$, as having $x_j^* = 1$ would mean that the edge constraint $x_p^* + x_j^* \leq 1$ is violated.
- j is not a leaf of the tree. This is because S alternates between I and I^*, and starts with I^* at the root in level 1 (which is odd). Hence, $S \cap I$ will occupy even levels of the tree, but the tree has odd depth (by assumption t is odd).

Now consider the edge constraints. For any edge (j, k), if the LP optimum x^* is such that the constraint is loose, i.e., if $x_j^* + x_k^* < 1$, then making ϵ small enough will ensure that $x_j + x_k \leq 1$. So we only need to check the edge constraints which are tight at x^*.

For edges with $x_j^* + x_k^* = 1$, every time any copy of one of the nodes j or k is included in $S \cap I$, the other node is included in $S \cap I^*$. This is because of the following: if j is included in $S \cap I$, and k is its parent, we are done since this means $k \in S \cap I^*$. So suppose k is not the parent of j. From the above it follows that j is not a leaf of the tree, and hence k will be one of its children. Also, from above, the mass on j satisfies $x_j^* < 1$. However, by assumption, $x_j^* + x_k^* = 1$, and hence, the mass on k is $x_k^* > 0$. This means that the child k has to be included in $S \cap I^*$.

It is now easy to see that the edge constraints are satisfied: for every edge constraint which is tight at x^*, every time the mass on one of the endpoints is increased by ϵ (because of that node appearing in $S \cap I$), the mass on the other endpoint is decreased by ϵ (because it appears $S \cap I^*$). $\qquad\square$

VI. A CONVERGENT MESSAGE-PASSING ALGORITHM

In Section V, we saw that max product started from the natural initial condition solves the correct LP at the fixed point, *if it converges*. However, convergence is not guaranteed, indeed it is quite easy to construct examples where it will not converge. For example, consider a three-node complete graph (a triangle graph) with each node having exactly the same node weight $W > 0$. Let all initial messages be 0 along all edges. Then, messages will oscillate between 0 and W at even and odd times.

In this section, we present a convergent message-passing algorithm for finding the MWIS of a graph. It is based on modifying max product by drawing upon a dual coordinate descent and the barrier method. The algorithm retains the iterative and distributed nature of max product. The algorithm leads to an optimal solution of DUAL for any weighted graph G. Now when G is bipartite, the LP relaxation is tight. Therefore, in principle, one can hope to obtain solution of MWIS by solving LP. Now, the solutions of DUAL and LP (primal) do satisfy complimentary slackness conditions. But this, in general, does not guarantee recovery of primal or LP solution from DUAL. Here, we develop a novel *primal recovery* algorithm based on the optimal solution of DUAL when the MWIS and LP have unique solution for bipartite graph. The algorithm is simple, iterative, and stops with $O(n)$ iterations. In our opinion, this should be of interest in its own right.

Now, we provide an overview of our algorithm. The algorithm operates in two steps, as described below.

$\underline{\mathsf{ALGO}(\varepsilon, \delta, \delta_1)}$

(o) Given an MWIS problem, and (small enough) positive parameters ε, δ, run subroutine $\mathsf{DESCENT}(\varepsilon, \delta)$ to obtain an output $\lambda^{\varepsilon,\delta} = (\lambda_{ij}^{\varepsilon,\delta})_{(i,j)\in E}$ $\lambda^{\varepsilon,\delta}$ that is an approximate dual of the MWIS problem.

(i) Next, using (small enough) $\delta_1 > 0$, use $\mathsf{EST}(\lambda^{\varepsilon,\delta}, \delta_1)$, to produce an estimate for the MWIS as an output of the algorithm.

Next, we describe $\mathsf{DESCENT}$ and EST, state their properties, and then combine them to produce the following result about the convergence, correctness and bound on convergence time for the overall algorithm.

A. DESCENT: Algorithm

Here, we describe the $\mathsf{DESCENT}$ algorithm. It is influenced by the max product and dual coordinate descent algorithm for DUAL. First, consider the standard coordinate descent algorithm for DUAL. It operates with variables $\{\lambda_{ij}, (i,j) \in E\}$ (with notation $\lambda_{ij} = \lambda_{ji}$). It is an iterative procedure; in each iteration t one edge $(i,j) \in E$ is picked[1] and updated

$$\lambda_{ij}^{t+1} = \max\left\{0, \left(w_i - \sum_{k\in\mathcal{N}(i),k\neq j}\lambda_{ik}^t\right), \left(w_j - \sum_{k\in\mathcal{N}(j),k\neq i}\lambda_{jk}^t\right)\right\}. \quad (11)$$

The λ on all the other edges remain unchanged from t to $t+1$. Notice the similarity (at least syntactic) between standard dual coordinate descent (11) and max product (2). In essence, the dual coordinate descent can be thought of as a *sequential bidirectional* version of the max-product algorithm.

Since the dual coordinate descent algorithm is designed so that at each iteration, the cost of the DUAL is nonincreasing, it always converges in terms of the cost. However, the converged solution may not be optimum because DUAL contains the "nonbox" constraints $\sum_{j\in\mathcal{N}(i)}\lambda_{ij} \geq w_i$. Therefore, a direct usage of dual coordinate descent is not sufficient. In order to make the algorithm convergent with minimal modification while retaining its iterative message-passing nature, we use barrier (penalty) function-based approach. With an appropriate choice of barrier and using result of Luo and Tseng [3], we will find the new algorithm to be convergent.

To this end, consider the following convex optimization problem obtained from DUAL by adding a logarithmic barrier for constraint violations with $\varepsilon \geq 0$ controlling penalty due to violation. Define

$$g(\varepsilon, \lambda) = \left(\sum_{(i,j)\in E}\lambda_{ij}\right) - \varepsilon\left(\sum_{i\in V}\log\left[\sum_{j\in\mathcal{N}(i)}\lambda_{ij} - w_i\right]\right).$$

[1]Edges can be picked either in round-robin fashion, or uniformly at random.

Then, the modified DUAL optimization problem becomes

$$\mathsf{CP}(\varepsilon): \quad \min\ g(\varepsilon, \lambda)$$
$$\text{subject to} \quad \lambda_{ij} \geq 0, \qquad \text{for all } (i,j) \in E.$$

The algorithm $\mathsf{DESCENT}(\varepsilon, \delta)$ is coordinate descent on $\mathsf{CP}(\varepsilon)$, to within tolerance δ, implemented via passing messages between nodes. We describe it in detail as follows.

$\underline{\mathsf{DESCENT}(\varepsilon, \delta)}$

(o) The parameters are variables λ_{ij}, one for each edge $(i,j) \in E$. We will use notation that $\lambda_{ij}^t = \lambda_{ji}^t$. The vector λ is iteratively updated, with t denoting the iteration number.
- Initially, set $t = 0$ and $\lambda_{ij}^0 = \max\{w_i, w_j\}$ for all $(i,j) \in E$.

(i) In iteration $t+1$, update parameters are as follows.
- Pick an edge $(i,j) \in E$. The edge selection is done in a round-robin manner over all edges.
- For all $(i',j') \in E, (i',j') \neq (i,j)$ do nothing, i.e., $\lambda_{i'j'}^{t+1} = \lambda_{i'j'}^t$.
- For edge (i,j), nodes i and j exchange messages as follows:

$$\gamma_{i\xrightarrow{} j}^{t+1} = \left(w_i - \sum_{k\neq j,k\in\mathcal{N}(i)}\lambda_{ki}^t\right)_+$$
$$\gamma_{j\xrightarrow{} i}^{t+1} = \left(w_j - \sum_{k'\neq i,k'\in\mathcal{N}(j)}\lambda_{k'j}^t\right)_+.$$

- Update λ_{ij}^{t+1} as follows: with $a = \gamma_{i\xrightarrow{} j}^{t+1}$ and $b = \gamma_{j\xrightarrow{} i}^{t+1}$

$$\lambda_{ij}^{t+1} = \left(\frac{a+b+2\varepsilon+\sqrt{(a-b)^2+4\varepsilon^2}}{2}\right)_+. \quad (12)$$

(ii) Update $t = t + 1$ and repeat until algorithm converges within δ for each component.

(iii) Output the vector λ, denoted by $\lambda^{\varepsilon,\delta}$, when the algorithm stops.

Remark: The updates in $\mathsf{DESCENT}$ above are obtained by small, but important, perturbation of standard dual coordinate descent (11). To see this, consider the iterative step in (12). First, note that

$$\frac{a+b+2\varepsilon+\sqrt{(a-b)^2+4\varepsilon^2}}{2}$$
$$> \frac{a+b+2\varepsilon+\sqrt{(a-b)^2}}{2}$$
$$= \frac{a+b+|a-b|+2\varepsilon}{2}$$
$$= \max(a,b)+\varepsilon.$$

Similarly

$$\frac{a+b+2\varepsilon+\sqrt{(a-b)^2+4\varepsilon^2}}{2}$$
$$\leq \frac{a+b+2\varepsilon+\sqrt{(a-b)^2+4\varepsilon(a-b)+4\varepsilon^2}}{2}$$
$$= \frac{a+b+|a-b|+4\varepsilon}{2}$$
$$= \max(a,b)+2\varepsilon.$$

Therefore, we conclude that (12) can be rewritten as

$$\lambda_{ij}^{t+1} = \beta\varepsilon + \max\left\{-\beta\varepsilon, \left(w_i - \sum_{k\in\mathcal{N}(i)\setminus j}\lambda_{ik}^t\right),\right.$$
$$\left.\left(w_j - \sum_{k\in\mathcal{N}(j)\setminus i}\lambda_{kj}^t\right)\right\}$$

where for some $\beta \in (1,2]$ with its precise value dependent on $\gamma_{i\to j}^{t+1}, \gamma_{j\to i}^{t+1}$. This small perturbation takes λ close to the true dual optimum. In practice, we believe that instead of calculating exact value of β, use of some arbitrary $\beta \in (1,2]$ should be sufficient.

B. DESCENT: *Properties*

The DESCENT algorithm finds a good approximation to an optimum of DUAL, for small enough ε,δ. Furthermore, it always converges, and does so quickly. The following lemma specifies the convergence and correctness guarantees of DESCENT.

Lemma 6.1: For given $\varepsilon,\delta > 0$, let λ^t be the parameter value at the end of iteration $t \geq 1$ under DESCENT(ε,δ). Then, there exists a unique limit point $\lambda^{\varepsilon,\delta}$ such that

$$\|\lambda^t - \lambda^{\varepsilon,\delta}\| \leq A\exp(-Bt) \qquad (13)$$

for some positive constant A, B (which may depend on problem parameters ε and δ). Let λ^ε be the solution of CP(ε). Then

$$\lim_{\delta\to 0}\lambda^{\varepsilon,\delta} = \lambda^\varepsilon.$$

Further, by taking $\varepsilon \to 0$, λ^ε goes to λ^*, an optimal solution to the DUAL.

We first discuss the proofs of two facts in Lemma 6.1: (a) $\lim_{\delta\to 0}\lambda^{\varepsilon,\delta} = \lambda^\varepsilon$ is a direct consequence of the fact that if we ran DESCENT algorithm with $\delta = 0$, it converges; (b) the fact that as $\varepsilon \to 0$, λ^ε goes to a dual optimal solution λ^* follows from [17, Prop. 4.1.1]. Now, it remains to establish the convergence of the DESCENT (ε,δ) algorithm. This will follow as a corollary of result by Luo and Tseng [3]. In order to state the result in [3], some notation needs to be introduced as follows.

Consider a real valued function $\phi : \mathbb{R}^n \to \mathbb{R}$ defined as

$$\phi(\mathbf{z}) = \psi(E\mathbf{z}) + \sum_{i=1}^n w_i x_i$$

where $E \in \mathbb{R}^{m\times n}$ is an $m \times n$ matrix with no zero column (i.e., all coordinates of $\mathbf{z}$ are useful), $\mathbf{w} = (w_i) \in \mathbb{R}^n$ is a given fixed

vector, and $\psi : \mathbb{R}^m \to \mathbb{R}$ is a strongly convex function on its domain

$$D_\psi = \{\mathbf{y} \in \mathbb{R}^m : \psi(\mathbf{y}) \in (-\infty,\infty)\}.$$

We have D_ψ being open and let ∂D_ψ denote its boundary. We also have that, along any sequence $\mathbf{y}_k$ such that $\mathbf{y}_k \to \partial D_\psi$ (i.e., approaches boundary of D_ψ), $\psi(\mathbf{y}^k) \to \infty$. The goal is to solve the optimization problem

$$\text{minimize} \quad \phi(\mathbf{z}) \quad \text{over} \quad \mathbf{z} \in \mathcal{X}. \qquad (14)$$

In the above, we assume that $\mathcal{X}$ is box type, i.e.,

$$\mathcal{X} = \prod_{i=1}^n [\ell_i, u_i], \qquad \ell_i, u_i \in \mathbb{R}.$$

Let $\mathcal{X}^*$ be the set of all optimal solutions of the problem (14). The "round-robin" or "cyclic" coordinate descent algorithm (the one used in DESCENT) for this problem has the following convergence property, as proved in Theorem 6.2 [3].

Lemma 6.2: There exist constants α' and β' which may depend on the problem parameters in terms of $g, E, \underline{w}$ such that starting from the initial value $\mathbf{z}^0$, we have in iteration t of the algorithm

$$d(\mathbf{z}^t, \mathcal{X}^*) \leq \alpha' \exp(-\beta't)\, d(\mathbf{z}^0, \mathcal{X}^*).$$

Here, $d(\cdot, \mathcal{X}^*)$ denotes distance to the optimal set $\mathcal{X}^*$.

Proof of Lemma 6.1: It suffices to check that the conditions assumed in the statement of Lemma 6.2 apply in our set up of Lemma 6.1 in order to complete the proof.

Note first that the constraints $\lambda_{ij} \geq 0$ in CP(ε) are of "box type," as required by Lemma 6.2. Now, we need to show that $g(\cdot)$ satisfies the conditions that $\phi(\cdot)$ satisfied in (14). By observation, we see that the linear part in $g(\cdot)$ is $\sum_{ij}\lambda_{ij}$ corresponds to the linear part in ϕ. Now, the other part in $g(\cdot)$, which corresponds to $h(\varepsilon,\lambda)$ where define

$$h(\varepsilon,\lambda) = -\varepsilon\sum_i \log\left(\sum_{j\in\mathcal{N}(i)}\lambda_{ij} - w_i\right).$$

By definition, the $h(\cdot)$ is strictly convex on its domain which is an open set as for any i, if

$$\sum_{j\in\mathcal{N}(i)}\lambda_{ij} \downarrow w_i$$

then $h(\cdot) \uparrow \infty$. Note that for $h(\cdot) \to \infty$ towards boundary corresponding to $\|\lambda\| \to \infty$ can be adjusted by redefining $h(\cdot)$ to include some parts of the linear term in $g(\cdot)$. Finally, the condition corresponding to E not having any zero column in (14) follows for any connected graph, which is of our interest here. Thus, we have verified conditions of Lemma 6.2, and hence established the proof of (13). This completes the proof of Lemma 6.1. $\square$

C. EST: *Algorithm*

The algorithm DESCENT yields a good approximation of the optimal solution to DUAL, for small values of ε and δ. However,

our interest is in the (integral) optimum of LP, when it exists. There is no general procedure to recover an optimum of a linear program from an optimum of its dual. However, we show that such a recovery *is* possible through our algorithm, called EST and presented below, for the MWIS problem when G is bipartite with a unique MWIS. This procedure is likely to extend for general G when LP relaxation is tight and LP has a unique solution. In the following, δ_1 is chosen to be an appropriately small number, and λ is expected to be (close to) a dual optimum.

EST(λ, δ_1)

(o) The algorithm iteratively estimates $\mathbf{x} = (x_i)$ given λ (expected to be a dual optimum).

(i) Initially, color a node i *gray* and set $x_i = 0$ if $\sum_{j \in \mathcal{N}(i)} \lambda_{ij} > w_i + \delta_1$. Color all other nodes with *green* and leave their values unspecified.

(ii) Repeat the following steps (in any order) until no more changes can happen:
  • if i is *green* and there exists a *gray* node $j \in \mathcal{N}(i)$ with $\lambda_{ij} > \delta_1$, then set $x_i = 1$ and color it *orange*;
  • if i is *green* and some *orange* node $j \in \mathcal{N}(i)$, then set $x_i = 0$ and color it *gray*.

(iii) If any node is *green*, say i, set $x_i = 1$ and color it *red*.

(iv) Produce the output $\mathbf{x}$ as an estimation.

D. EST*: Properties*

Lemma 6.3: Let λ^* be an optimal solution of DUAL. If G is a bipartite graph with unique MWIS, then the output produced by EST$(\lambda^*, 0)$ is the maximum weight independent set of G.

Proof: Let x be output of EST$(\lambda^*, 0)$, and $\mathbf{x}^*$ the unique optimal MWIS. To establish $x = \mathbf{x}^*$, it is sufficient to establish that x and λ^* together satisfy the complimentary slackness conditions stated in Lemma 2.3, namely:

  (x1) $x_i(\sum_{j \in \mathcal{N}(i)} \lambda_{ij}^* - w_i) = 0$ for all $i \in V$;
  (x2) $(x_i + x_j - 1)\lambda_{ij}^* = 0$ for all $(i,j) \in E$;
  (x3) x is a feasible solution for the IP.

From the way the color *gray* is assigned initially, it follows that either $x_i = 0$ or $\sum_j \lambda_{ij} - w_i = 0$ for all nodes i. Thus, (x1) is satisfied.

Before proceeding we note that all nodes initially colored gray are correct, i.e., $x_i = x_i^* = 0$; this is because the optimal $\mathbf{x}^*$ satisfies (x1). Now consider any node j that is colored orange due to there being a neighbor i that is one of the initial grays, and $\lambda_{ij} > 0$. For this node, we have that $x_j = x_j^* = 1$, because $\mathbf{x}^*$ satisfies (x2). Proceeding in this fashion, it is easy to establish that all nodes colored gray or orange are assigned values consistent with the actual MWIS $\mathbf{x}^*$.

Now to prove (x2), consider a particular edge (i,j). For this, if $\lambda_{ij}^* = 0$, then (x2) is satisfied. Suppose $\lambda_{ij}^* > 0$, but $x_i + x_j \neq 1$. This will happen if both $x_i = x_j = 0$, or both are equal to 1. Now, both are equal to 0 only if they are both colored gray, in which case we know that the actual optima $x_i^* = x_j^* = 1$ as well. But this means that (x2) is violated by the true optimum $\mathbf{x}^*$, which is a contradiction. Thus, it has to be that $x_i = x_j = 1$ for violation to occur. However, this is also a violation of (x3),

namely, the feasibility of x for the IP. Thus, all that remains to be done is to establish (x3).

Assume now that (x3) is violated, i.e., there exists a subset E' of the edges whose both endpoints are set to 1. Let $S_1 \subset V_1, S_2 \subset V_2$ be these endpoints. Note that, by assumption, $S_1 \neq \emptyset, S_2 \neq \emptyset$. We now use S_1 and S_2 to construct two distinct optima of IP, which will be a violation of our assumption of uniqueness of the MWIS. The two optima, denoted $\hat{x}$ and $\tilde{x}$, are obtained as follows: in x, modify $x_i = 0$ for all $i \in S_1$ to obtain $\hat{x}$; in x, modify $x_i = 0$ for all $i \in S_2$ to obtain $\tilde{x}$. We now show that both $\hat{x}$ and $\tilde{x}$ satisfy all three conditions (x1), (x2), and (x3).

Recall that the nodes in S_1 and S_2 must have been colored *red* by the algorithm EST. Now, we establish optimality of $\hat{x}$ and $\tilde{x}$. By construction, both $\hat{x}$ and $\tilde{x}$ satisfy (x1) since we have only changed assignment of *red* nodes which were not *binding* for constraint (x1).

Now, we turn our attention towards (x2) and (x3) for $\hat{x}$ and $\tilde{x}$. Again, both solutions satisfy (x2) and (x3) along edges $(i,j) \in E$ such that $i \in S_1, j \in S_2$ or else they would not have been colored *red*. By construction, they satisfy (x3) along all other edges as well. Now we show that $\hat{x}, \tilde{x}$ satisfy (x2) along edges $(i,j) \in E$, such that $i \in S_1, j \notin S_2$ or $i \notin S_1, j \in S_2$. For this, we claim that all such edges must have $\lambda_{ij}^* = 0$: if not, that is $\lambda_{ij}^* > 0$, then either i or j must have been colored *orange* and an *orange* node cannot be part of S_1 or S_2. Thus, we have established that both $\hat{x}$ and $\tilde{x}$ along with λ^* satisfy (x1), (x2), and (x3). The contradiction is thus established.

Thus, we have established that x along with λ^* satisfies (x1), (x2), and (x3). Therefore, x is the optimal solution of LP, and hence of the IP. This completes the proof. □

Now, consider a version of EST where we check for updating nodes in a round-robin manner. That is, in an iteration, we perform $O(n)$ operations. Now, we state a simple bound on running time of EST.

Lemma 6.4: The algorithm EST stops after at most $O(n)$ iterations.

Proof: The algorithm stops after the iteration in which no more node's status is updated. Since each node can be updated at most once, with the above stopping condition, an algorithm can run for at most $O(n)$ iterations. This completes the proof of Lemma 6.4. □

E. *Overall Algorithm: Convergence and Correctness*

Before stating convergence, correctness, and bound on convergence time of the ALGO$(\varepsilon, \delta, \delta_1)$ algorithm, a few remarks are in order. We first note that both DESCENT and EST are iterative message-passing procedures. Second, when the MWIS is unique, DESCENT need not produce an exact dual optimum for EST to obtain the correct answer. Finally, it is important to note that the above algorithm always converges quickly, but may not produce good estimate when LP relaxation is not tight. Next, we state the precise statement of this result.

Theorem 6.1: (Convergence and Correctness): The algorithm ALGO$(\varepsilon, \delta, \delta_1)$ converges for any choice of $\varepsilon, \delta > 0$ and for any G. The solution obtained by it is correct if G is bipartite, LP has unique solution, and $\varepsilon, \delta > 0, \delta_1$ are small enough.

Proof: The claim that algorithm ALGO$(\varepsilon, \delta, \delta_1)$ converges for all values of $\varepsilon, \delta, \delta_1$ and for any G follows immediately from

Lemmas 6.1, 6.3, and 6.4. Next, we worry about the correctness property.

Lemma 6.1 implies that for $\delta \to 0$, the output of DESCENT(ε, δ), $\lambda^{\varepsilon,\delta} \to \lambda^{\varepsilon}$, where λ^{ε} is the solution of CP(ε). Again, as noted in Lemma 6.1, $\lambda^{\varepsilon} \to \lambda^{*}$ as $\varepsilon \to 0$, where λ^{*} is an optimal solution[2] of the DUAL. Therefore, given $\delta > 0$, for small enough $\varepsilon > 0$, we have

$$\left| \lambda_{ij}^{*,\varepsilon} - \lambda_{ij}^{*} \right| \leq \frac{\delta}{3n}, \qquad \text{for all } (i,j) \in E.$$

We will suppose that the ε is chosen such. As noted earlier, the algorithm converges for all choices of ε. Therefore, by Lemma 6.1, there exists large enough T such that for $t \geq T$, we have

$$\left| \lambda_{ij}^{*,\varepsilon} - \lambda_{ij}^{t} \right| \leq \frac{\delta}{3n}, \qquad \text{for all } (i,j) \in E.$$

Thus, for $t \geq T$, we have

$$\left| \lambda_{ij}^{*} - \lambda_{ij}^{t} \right| \leq \frac{2\delta}{3n}, \qquad \text{for all } (i,j) \in E. \tag{15}$$

Now, recall Lemma 6.3. It established that the EST$(\lambda^{*}, 0)$ produces the correct MWIS as its output under hypothesis of Theorem 6.1. Also recall that the algorithm EST$(\lambda^{*}, 0)$ checks two conditions: 1) whether $\lambda_{ij}^{*} > 0$ for $(i,j) \in E$; and 2) whether $\sum_{j \in \mathcal{N}(i)} \lambda_{ij}^{*} > w_i$. Given that the number of nodes and edges are finite, there exists a δ such that 1) and 2) are robust to noise of δ/n. Therefore, by selection of small δ_1 for such choice of δ, we find that the output of EST(λ^{t}, δ_1) algorithm will be the same as that of EST$(\lambda^{*}, 0)$. This completes the proof. $\qquad \square$

VII. MAP ESTIMATION AS AN MWIS PROBLEM

In this section, we show that any MAP estimation problem is equivalent to an MWIS problem on a suitably constructed graph with node weights. This construction is related to the "overcomplete basis" representation [9]. Consider the following canonical MAP estimation problem: suppose we are given a distribution $q(\mathbf{y})$ over vectors $\mathbf{y} = (y_1, \ldots, y_M)$ of variables y_m, each of which can take a finite value. Suppose also that q factors into a product of strictly positive functions, which we find convenient to denote in exponential form

$$q(\mathbf{y}) = \frac{1}{Z} \prod_{\alpha \in A} \exp\left(\phi_\alpha(\mathbf{y}_\alpha) \right)$$
$$= \frac{1}{Z} \exp\left(\sum_{\alpha \in A} \phi_\alpha(\mathbf{y}_\alpha) \right).$$

Here α specifies the domain of the function ϕ_α, and $\mathbf{y}_\alpha$ is the vector of those variables that are in the domain of ϕ_α. The α's also serve as an index for the functions. A is the set of functions. The MAP estimation problem is to find a maximizing assignment $\mathbf{y}^{*} \in \arg\max_{\mathbf{y}} q(\mathbf{y})$.

We now build an auxiliary graph $\widetilde{G}$, and assign weights to its nodes, such that the MAP estimation problem above is equivalent to finding the MWIS of $\widetilde{G}$. There is one node in $\widetilde{G}$ for each

pair $(\alpha, \mathbf{y}_\alpha)$, where $\mathbf{y}_\alpha$ is an *assignment* (i.e., a set of values for the variables) of domain α. We will denote this node of $\widetilde{G}$ by $\delta(\alpha, \mathbf{y}_\alpha)$. Note that such a graph $\widetilde{G}$ can have much larger size than G with increase in size governed by the size of each α.

There is an edge in $\widetilde{G}$ between any two nodes $\delta(\alpha_1, \mathbf{y}_{\alpha_1}^1)$ and $\delta(\alpha_2, \mathbf{y}_{\alpha_2}^2)$ if and only if there exists a variable index m such that:

1) m is in both domains, i.e., $m \in \alpha_1$ and $m \in \alpha_2$;
2) the corresponding variable assignments are different, i.e., $y_m^1 \neq y_m^2$.

In other words, we put an edge between all pairs of nodes that correspond to *inconsistent* assignments. Given this graph $\widetilde{G}$, we now assign weights to the nodes. Let $c > 0$ be any number such that $c + \phi_\alpha(\mathbf{y}_\alpha) > 0$ for all α and $\mathbf{y}_\alpha$. The existence of such a c follows from the fact that the set of assignments and domains is finite. Assign to each node $\delta(\alpha, \mathbf{y}_\alpha)$ a weight of $c + \phi_\alpha(\mathbf{y}_\alpha)$.

Lemma 7.1: Suppose q and $\widetilde{G}$ are as above. a) If $\mathbf{y}^{*}$ is a MAP estimate of q, let $\delta^{*} = \{ \delta(\alpha, \mathbf{y}_\alpha^{*}) \, | \, \alpha \in A \}$ be the set of nodes in $\widetilde{G}$ that correspond to each domain being consistent with $\mathbf{y}^{*}$. Then, δ^{*} is an MWIS of $\widetilde{G}$. b) Conversely, suppose δ^{*} is an MWIS of $\widetilde{G}$. Then, for every domain α, there is exactly one node $\delta(\alpha, \mathbf{y}_\alpha^{*})$ included in δ^{*}. Further, the corresponding domain assignments $\{ \mathbf{y}_\alpha^{*} \, | \, \alpha \in A \}$ are consistent, and the resulting overall vector $\mathbf{y}^{*}$ is a MAP estimate of q.

Proof: A *maximal* independent set is one in which every node is either in the set, or is adjacent to another node that is in the set. Since weights are positive, any MWIS has to be maximal. For $\widetilde{G}$ and q as constructed, the following is clear.

1) If $\mathbf{y}$ is an assignment of variables, consider the corresponding set of nodes $\{ \delta(\alpha, \mathbf{y}_\alpha) \, | \, \alpha \in A \}$. Each domain α has exactly one node in this set. Also, this set is an independent set in $\widetilde{G}$, because the partial assignments $\mathbf{y}_\alpha$ for all the nodes are consistent with $\mathbf{y}$, and hence with each other. This means that there will not be an edge in $\widetilde{G}$ between any two nodes in the set.
2) Conversely, if Δ is a maximal independent set in $\widetilde{G}$, then all the sets of partial assignments corresponding to each node in Δ are all consistent with each other, and with a global assignment $\mathbf{y}$.

There is thus a one-to-one correspondence between maximal independent sets in $\widetilde{G}$ and assignments $\mathbf{y}$. The lemma follows from this observation. $\qquad \square$

Example 7.1: Let y_1 and y_2 be binary variables with joint distribution

$$q(y_1, y_2) = \frac{1}{Z} \exp(\theta_1 y_1 + \theta_2 y_2 + \theta_{12} y_1 y_2)$$

where the θ are any real numbers. The corresponding $\widetilde{G}$ is shown in Fig. 3. Let c be any number such that $c + \theta_1$, $c + \theta_2$, and $c + \theta_{12}$ are all greater than 0. The weights on the nodes in $\widetilde{G}$ are: $\theta_1 + c$ on node "1" on the left, $\theta_2 + c$ for node "1" on the right, $\theta_{12} + c$ for the node "11," and c for all the other nodes.

VIII. DISCUSSION

We believe this paper opens several interesting directions for investigation. In general, the exact relationship between max

---

[2]There may be multiple dual optima, and in this case, λ^{ε} may not have a unique limit. However, every limit point will be a dual optimum. In that case, the same proof still holds; we skip it here to keep arguments simple.
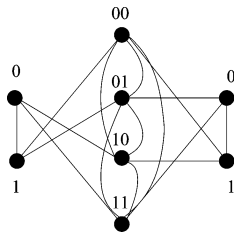
Fig. 3. Example of reduction from MAP problem to MWIS problem.

product and LP is not well understood. Their close similarity for the MWIS problem, along with the reduction of MAP estimation to an MWIS problem, suggests that the MWIS problem may provide a good first step in an investigation of this relationship. Indeed, obtaining such an understanding in the context of LP decoding and max product would be an interesting pursuit (e.g., see work by Vontobel and Koetter [14]).

Our novel message-passing algorithm and the reduction of MAP estimation to an MWIS problem immediately yields a new message-passing algorithm for general MAP estimation problem. It would be interesting to investigate the power of this algorithm on more general discrete estimation problems.

ACKNOWLEDGMENT

REFERENCES

[1] M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness and LP duality," *IEEE Trans. Inf. Theory*, 2009, accepted for publication.

[2] S. Sanghavi, D. Malioutov, and A. Willsky, "Linear programming analysis of Loopy Belief Propagation for weighted matching," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 1273–1280.

[3] Z.-Q. Luo and P. Tseng, "On the linear convergence of descent methods for convex essentially smooth minimization," *SIAM J. Control Optim.*, vol. 30, no. 2, pp. 408–425, 1992.

[4] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.

[5] C. Moallemi and B. Van Roy, "Convergence of the min-sum message passing algorithm for quadratic optimization," 2006 [Online]. Available: arXiv:cs/0603058

[6] L. Trevisan, "Inapproximability of combinatorial optimization problems," Electron. Colloq. Comput. Complex., Tech. Rep. TR04–065, 2004.

[7] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," *Statist. Comput.*, vol. 14, pp. 143–166, Apr. 2004.

[8] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP estimation via agreement on (hyper) trees: Message-passing and linear programming approaches," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3697–3717, Nov. 2005.

[9] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," Dept. Statist., Univ. California Berkeley, Berkeley, CA, Tech. Rep. 649, 2003.

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

[11] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," Mitsubishi Elect. Res. Lab., Cambridge, MA, TR-2000–26, 2000.

[12] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming and belief propagation with convex free energies," in *Proc. Conf. Uncertainty Artif. Intell.*, 2007.

[13] 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/0709.1190

[14] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Proc. Int. Symp. Inf. Theory Appl.*, Parma, Italy, Oct. 10–13, 2004, pp. 991–996.

[15] S. Sanghavi, D. Shah, and A. Willsky, "Message-passing for max-weight independent set," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 1281–1288.

[16] A. Schrijver, *Combinatorial Optimization. Polyhedra and Efficiency*. Berlin, Germany: Springer-Verlag, 2003.

[17] D. Bertsekas, *Non Linear Programming*. Belmont, MA: Athena Scientific, 1995.

[18] Grotschel, L. Lovasz, and Schrijver, "Polynomial algorithms for perfect graphs," *Ann. Discrete Math.*, vol. 21, pp. 325–356, 1984.

[19] K. Jung and D. Shah, "Low delay scheduling in wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory*, 2007, pp. 1396–1400.

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

[21] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming and belief propagation with convex free energies," in *Proc. Conf. Uncertainty Artif. Intell.*, 2007.

[22] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 553–560.

[23] 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.

[24] S. M. Aji, G. B. Horn, and R. J. McEliece, "On the convergence of iterative decoding on graphs with a single cycle," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, Aug. 1998, pp. 276–276.

[25] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[26] S. Tatikonda and M. I. Jordan, "Loopy belief propagation and Gibbs measures," in *Proc. Conf. Uncertainty Artif. Intell.*, 2002.

[27] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Comput. Optim. Appl.*, vol. 1, pp. 7–66, 1992.

Sujay Sanghavi (M'08) received the Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, in 2006.

Currently, he is an Assistant Professor at the Electrical and Computer Engineering Department, Purdue University, West Lafayette, IN, which he joined in 2008. His research interests lie in probability, optimization and algorithms, and their applications to networks, communication, and statistical inference and learning.

Devavrat Shah received the B.Tech. degree in computer science and engineering from Indian Institute of Technology (IIT), Bombay, India, in 1999 with the honor of the President of India Gold Medal and the Ph.D. degree from the Computer Science Department, Stanford University, Stanford, CA, in October 2004.

He was a postdoc in the Statistics Department, Stanford University, in 2004–2005. Currently, he is a Jamieson Career Development Assistant Professor at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge. His research focus is on theory of large complex networks which includes network algorithms, stochastic networks, network information theory, and large scale statistical inference.

Mr. Shah was coawarded the IEEE INFOCOM best paper award in 2004 and ACM SIGMETRICS/Performance best paper award in 2006. He received 2005 George B. Dantzig best dissertation award from the INFORMS and an NSF CAREER award in 2006. He is the recipient of the first ACM SIGMETRICS Rising Star Award 2008 for his work on network scheduling algorithms.

Alan S. Willsky (S'70–M'73–SM'82–F'86) received the Ph.D. degree from Massachusetts Institute of Technology (MIT), Cambridge, in 1972.

He joined MIT in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering and Co-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 to 2002, he served on the U.S. Air Force Scientific Advisory Board. He has delivered numerous keynote addresses and is coauthor of the text *Signals and Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1997). 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 several 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, and Doctorat Honoris Causa from Université de Rennes in 2005.