Message Passing for Max-weight Independent Set

Sujay Sanghavi
LIDS, MIT
sanghavi@mit.edu

Devavrat Shah
Dept. of EECS, MIT
devavrat@mit.edu

Alan Willsky
Dept. of EECS, MIT
willsky@mit.edu

Abstract

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 loopy max-product belief propagation. We show that, if it converges, the quality of the estimate is closely related to the tightness of an LP relaxation of the MWIS problem. We use this relationship to obtain sufficient conditions for correctness of the estimate. We then develop a modification of max-product – one that converges to an optimal solution of the dual of the MWIS problem. We also develop a simple iterative algorithm for estimating the max-weight independent set from this dual solution. 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 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.

1 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 non-adjacent 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 2.1 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 (see [9] and references). The MWIS problem provides a rich, yet relatively tractable, first framework in which to investigate such connections.

1.1 Our contributions

In Section 4 we construct a probability distribution whose MAP estimate corresponds to the MWIS of a given graph, and investigate the application of the loopy Max-product algorithm to this distribuion. We demonstrate that there is an intimate relationship between the max-product fixed-points and the natural LP relaxation of the original independent set problem. We use this relationship to provide a *certificate of correctness* for the max-product fixed point in certain problem instances.

In Section 5 we develop two iterative message-passing algorithms. The first, obtained by a minor modification of max-product, calculates the optimal solution to the dual of the LP relaxation of the MWIS problem. The second algorithm uses this optimal dual to produce an estimate of the MWIS. This estimate is correct when the original graph is bipartite.

In Section 3 we show that *any* problem of MAP estimation in which all the random variables can take a finite number of values (and the probability distribution is positive over the entire domain) can be reduced to a max-weight independent set problem. This implies that any algorithm for solving the independent set problem immediately yields an algorithm for MAP estimation. We believe this reduction will prove useful from both practical and analytical perspectives.

2   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 $x_i = 0$ 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 a maximum weight independent set (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.

$$
\begin{array}{ll}
\text{IP :} \quad & \max \sum_{i=1}^{n} w_i x_i, \\[2mm]
\text{s.t.} \quad & x_i + x_j \leq 1 \quad \text{for all } (i,j) \in E, \\
& x_i \in \{0,1\}.
\end{array}
\qquad
\begin{array}{ll}
\text{DUAL :} \quad & \min \sum_{(i,j) \in E} \lambda_{ij}, \\[2mm]
\text{s.t.} \quad & \sum_{j \in \mathcal{N}(i)} \lambda_{ij} \geq w_i, \text{ for all } i \in V, \\
& \lambda_{ij} \geq 0, \text{ for all } (i,j) \in E.
\end{array}
$$

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*. usIt is well known [3] that the LP relaxation is tight for bipartite graphs. More generally, for non-bipartite graphs, tightness will depend on the node weights. We will use the performance of LP as a benchmark with which to compare the performance of our message passing algorithms.

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

Lemma 2.1 *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)* $x_i \left(\sum_{j \in \mathcal{N}(i)} \lambda_{ij} - w_i \right) = 0$ *for all* $i \in V$*, (c)* $(x_i + x_j - 1) \lambda_{ij} = 0$*, for all* $(i,j) \in E$*.*

2.1   Sample Application: Scheduling in Wireless Networks

We now briefly describe an important application that requires an efficient, distributed solution to the MWIS problem: transmision scheduling in wireless networks that lack a centralized infrastructure, and where nodes can only communicate with local neighbors (e.g. see [4]). 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 *(a)* there is no interference, and *(b)* 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.

3 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 [7]. 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 auxillary 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 $\delta(\alpha, \mathbf{y}_\alpha)$.

There is an edge in $\widetilde{G}$ between any two nodes $\delta(\alpha_1, \mathbf{y}^1_{\alpha_1})$ and $\delta(\alpha_2, \mathbf{y}^2_{\alpha_2})$ 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 $m \in \alpha_2$, and
2. the corresponding variable assignments are different, i.e. $y^1_m \neq y^2_m$.

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 3.1 *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.*

Example. 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 to the right. 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.

3
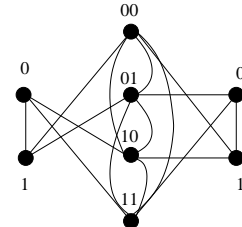
4 Max-product for MWIS

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 $\mathbf{x} \in \{0, 1\}^n$,

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

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 a maximum weight independent set 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:

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

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

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

(iii) Estimate max. wt. independent set $\mathbf{x}(b^{t+1})$ as follows: $x_i(b_i^{t+1}) = \mathbf{1}_{\{b_i^{t+1}(1) > b_i^{t+1}(0)\}}$.

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

For the purpose of analysis, we find it convenient to transform the messages be defining[1] $\gamma_{i \to j}^t = \log\left(\frac{m_{i \to j}^t(0)}{m_{i \to j}^t(1)}\right)$. Step (i) of max-product now becomes

$$\gamma_{i \to j}^{t+1} = \max\left\{ 0, \left(w_i - \sum_{k \neq j, k \in \mathcal{N}(i)} \gamma_{k \to i}^t \right) \right\}, \tag{2}$$

where we use the notation $(x)_+ = \max\{x, 0\}$. The estimation of step (iii) of max-product becomes: $x_i(\gamma^{t+1}) = \mathbf{1}_{\{w_i - \sum_{k \in \mathcal{N}(i)} \gamma_{k \to i} > 0\}}$. 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 the paper we refer to this as simply the max-product algorithm.

---

[1]If the algorithm starts with all messages being strictly positive, the messages will remain strictly positive over any finite number of iterations. Thus taking logs is a valid operation.

4.1 Fixed Points of Max-Product

When applied to general graphs, max product may either *(a)* not converge, *(b)* converge, and yield the correct answer, or *(c)* 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 [8].

Proposition 4.1 *Let γ represent a fixed point of the algorithm, and let $\mathbf{x}(\gamma) = (x_i(\gamma))$ be the corresponding estimate for the independent set. Then, the following properties hold:*

(a) Let i be a node with estimate $x_i(\gamma) = 1$, and let $j \in \mathcal{N}(i)$ be any neighbor of i. Then, the messages on edge (i, j) satisfy $\gamma_{i \to j} > \gamma_{j \to i}$. Further, from this it can be deduced that $\mathbf{x}(\gamma)$ represents an independent set in G.

(b) Let j be a node with $x_j(\gamma) = 0$, which by definition means that $w_j - \sum_{k \in \mathcal{N}(j)} \gamma_{k \to j} \leq 0$. Suppose now there exists a neighbor $i \in \mathcal{N}(j)$ whose estimate is $x_i(\gamma) = 1$. Then it has to be that $w_j - \sum_{k \in \mathcal{N}(j)} \gamma_{k \to j} < 0$, i.e. the inequality is strict.

(c) For any edge $(j_1, j_2) \in E$, if the estimates of the endpoints are $x_{j_1}(\gamma) = x_{j_2}(\gamma) = 0$, then it has to be that $\gamma_{j_1 \to j_2} = \gamma_{j_2 \to j_1}$. In addition, if there exists a neighbor $i_1 \in \mathcal{N}(j_1)$ of j_1 whose estimate is $x_{i_1}(\gamma) = 1$, then it has to be that $\gamma_{j_1 \to j_2} = \gamma_{j_2 \to j_1} = 0$ (and similarly for a neighbor i_2 of j_2).

The properties shown in Proposition 4.1 reveal striking similarities between the messages γ of fixed points of max-product, and the optimal λ that solves the dual linear program DUAL. In particular, suppose that γ is a fixed point at which the corresponding estimate $\mathbf{x}(\gamma)$ is a *maximal* independent set: for every j whose estimate $x_j(\gamma) = 0$ there exists a neighbor $i \in \mathcal{N}(j)$ whose estimate is $x_i(\gamma) = 1$. The MWIS, for example, is also maximal (if not, one could add a node to the MWIS and obtain a higher weight). For a maximal estimate, it is easy to see that

- $(x_i(\gamma) + x_j(\gamma) - 1)\, \gamma_{i \to j} = 0$ for all edges $(i, j) \in E$.
- $x_i(\gamma) \left(\gamma_{i \to j} + \sum_{k \in \mathcal{N}(i) - j} \gamma_{k \to i} - w_i \right) = 0$ for all $i, j \in V$

At least semantically, these relations share a close resemblance to the complimentary slackness conditions of Lemma 2.1. In the following lemma we leverage this resemblance to derive a certificate of optimality of the max-product fixed point estimate for certain problems.

Lemma 4.1 *Let γ be a fixed point of max-product and $\mathbf{x}(\gamma)$ the corresponding estimate of the independent set. Define $G' = (V, E')$ where $E' = E \backslash \{(i, j) \in E : \gamma_{i \to j} = \gamma_{j \to i} = 0\}$ is the set of edges with at least one non-zero message. Then, if G' is acyclic, we have that : (a) $\mathbf{x}(\gamma)$ is a solution to the MWIS for G, and (b) there is no integrality gap between LP and IP, i.e. $\mathbf{x}(\gamma)$ is an optimal solution to LP. Thus the lack of cycles in G' provides a certificate of optimality for the estimate $\mathbf{x}(\gamma)$.*

Max-product vs. LP relaxation. The following general question has been of great recent interest: which of the two, max-product and LP relaxation, is more powerful ? We now briefly investigate this question for MWIS. As presented below, we find that there are examples where one technique is better than the other. That is, neither technique clearly dominates the other.

To understand whether correctness of max-product (e.g. Lemma 4.1) provides information about LP relaxation, we consider the simplest loopy graph: a cycle. For bipartite graph, we know that LP relaxation is tight, i.e. provides answer to MWIS. Hence, we consider odd cycle. The following result suggests that if max-product works then it must be that LP relaxation is tight (i.e. LP is no weaker than max-product for cycles).

Corollary 4.1 *Let G be an odd cycle, and γ a fixed point of Max-product. Then, if there exists at least one node i whose estimate $x_i(\gamma) = 1$, then there is no integrality gap between LP and IP.*

Next, we present two examples which help us conclude that neither max-product nor LP relaxation dominate the other. The following figures 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 of $\gamma_{i \to j} = 2$. All other messages have γ are equal to 0. The boxed nodes indicate the ones for which the estimate $x_i(\gamma) = 1$. It is easy to verify that both represent max-product fixed points.

For the graph on the left, the max-product fixed point results in an incorrect estimate. However, the graph is bipartite, and hence LP will get the correct answer. In the graph on the right, there is an integrality gap between LP and IP: setting each $x_i = \frac{1}{2}$ yields an optimal value of 7.5, while the optimal solution to IP has value 6. However, the estimate at the fixed point of max-product is the correct MWIS. In both of these examples, the fixed points lie in the strict interiors of non-trivial regions of attraction: starting the iterative procedure from within these regions will result in convergence to the 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.

5 A Convergent Message-passing Algorithm

In this section we present our algorithm for finding the MWIS of a graph. It is based on modifying max-product by drawing upon a dual co-ordinate descent and barrier method. Specifically, the algorithm is as follows: (1) For small enough parameters ε, δ, run subroutine DESCENT(ϵ, δ) (close to) convergence. This will produce output $\lambda^{\varepsilon,\delta} = (\lambda_{ij}^{\varepsilon,\delta})_{(i,j) \in E}$. (2) For small enough parameter δ_1, use subroutine EST($\lambda^{\varepsilon,\delta}, \delta_1$), to produce an estimate for the MWIS as the output of algorithm.

Both of the subroutines, DESCENT, EST are iterative message-passing procedures. Before going into details of the subroutines, we state the main result about correctness and convergence of this algorithm.

Theorem 5.1 *The following properties hold for arbitrary graph G and weights: (a) For any choice of $\varepsilon, \delta, \delta_1 > 0$, the algorithm always converges. (b) As $\varepsilon, \delta \to 0$, $\lambda^{\varepsilon,\delta} \to \lambda^*$ where λ^* is an optimal solution of DUAL. Further, if G is bipartite and the MWIS is unique, then the following holds: (c) For small enough $\varepsilon, \delta, \delta_1$, the algorithm produces the MWIS as output.*

5.1 Subroutine: DESCENT

Consider the standard coordinate descent algorithm for DUAL: the variables are $\{\lambda_{ij}, (i,j) \in E\}$(with notation $\lambda_{ij} = \lambda_{ji}$) and at each iteration t one edge $(i,j) \in E$ is picked[2] and update

$$\lambda_{ij}^{t+1} = \max \left\{ 0, \left(w_i - \sum_{k \in \mathcal{N}(i), k \neq j} \lambda_{ik}^t \right), \left(w_j - \sum_{k \in \mathcal{N}(j), k \neq i} \lambda_{jk}^t \right) \right\} \qquad (3)$$

The λ on all the other edges remain unchanged from t to $t+1$. Notice the similarity (at least syntactic) between (3) and update of max-product (min-sum) (2): essentially, the dual coordinate descent is a *sequential bidirectional* version of the max-product algorithm !

It is well known that the coordinate descent always coverges, in terms of cost for linear programs. Further, it converges to an optimal solution if the constraints are of the *product set* type (see [2] for details). However, due to constraints of type $\sum_{j \in \mathcal{N}(i)} \lambda_{ij} \geq w_i$ in DUAL, the algorithm may not

---

[2]A good policy for picking edges is round-robin or uniformly at random

converge to an optimal of DUAL. Therefore, a direct adaptation of max-product to mimic dual co-ordinate descent is not good enough. We use barrier (penalty) function based approach to overcome this difficulty. 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.

$$\mathsf{CP}(\varepsilon): \quad \min \left(\sum_{(i,j) \in E} \lambda_{ij} \right) - \varepsilon \left(\sum_{i \in V} \log \left[\sum_{j \in \mathcal{N}(i)} \lambda_{ij} - w_i \right] \right)$$

$$\text{subject to} \quad \lambda_{ij} \geq 0, \text{ for all } (i,j) \in E.$$

The following is coordinate descent algorithm for $\mathsf{CP}(\varepsilon)$.

DESCENT(ε, δ)

(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 as follows:

  ○ Pick an edge $(i,j) \in E$. This edge selection is done so that each edge is chosen infinitely often as $t \to \infty$ (for example, at each t choose an edge uniformly at random.)
  ○ 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:

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

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

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

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

(iii) Output λ, the vector of paramters at convergence,

Remark. The iterative step (4) can be rewritten as follows: for some $\beta \in [1, 2]$,

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

where β depends on values of $\gamma_{i \to j}^{t+1}, \gamma_{j \to i}^{t+1}$. Thus the updates in DESCENT are obtained by small but important perturbation of dual coordinate descent for DUAL, and making it convergent. The output of DESCENT(ε, δ), say $\lambda^{\varepsilon, \delta} \to \lambda^*$ as $\varepsilon, \delta \to 0$ where λ^* is an optimal solution of DUAL.

5.2   Subroutine: EST

DESCENT yields a good estimate of the optimal solution to DUAL, for small values of ϵ and δ. However, we are interested in the (integral) optimum of LP. In general, it is *not* possible to recover the solution of a linear program from a dual optimal solution. However, we show that such a recovery *is* possible through EST algorithm described below for the MWIS problem when G is bipartite with unique MWIS. This procedure is likely to extend for general G when LP relaxation is tight and LP has unique solution.

EST(λ, δ_1).

(o) The algorithm iteratively estimates $\mathbf{x} = (x_i)$ given λ.

(i) Initially, color a node i *gray* and set $x_i = 0$ if $\sum_{j \in \mathcal{N}(i)} \lambda_{ij} > w_i$. Color all other nodes with *green* and leave their values unspecified. The condition $\sum_{j \in \mathcal{N}(i)} \lambda_{ij} > w_i$ is checked as whether $\sum_{j \in \mathcal{N}(i)} \lambda_{ij} \geq w_i + \delta_1$ or not.

(ii) Repeat the following steps (in any order) till no more changes can happen:

  ○ if i is *green* and there exists a *gray* node $j \in \mathcal{N}(i)$ with $\lambda_{ij} > 0$, then set $x_i = 1$ and color it *orange*. The condition $\lambda_{ij} > 0$ is checked as whether $\lambda_{ij} \geq \delta_1$ or not.
  ○ 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.

---

6   Discussion

We believe this paper opens several interesting directions for investigation. In general, the exact relationship between max-product and linear programming 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.

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

References

[1] M. Bayati, D. Shah and M. Sharma, "Max Weight Matching via Max Product Belief Propagation," *IEEE ISIT*, 2005.

[2] D. Bertsekas, "Non Linear Programming", *Athena Scientific*.

[3] M. Grtschel, L. Lovsz, and A. Schrijver, "Polynomial algorithms for perfect graphs," in C. Berge and V. Chvatal (eds.) *Topics on Perfect Graphs* Ann. Disc. Math. 21, North-Holland, Amsterdam(1984) 325-356.

[4] K. Jung and D. Shah, "Low Delay Scheduling in Wireless Networks," *IEEE ISIT*, 2007.

[5] C. Moallemi and B. Van Roy, "Convergence of the Min-Sum Message Passing Algorithm for Quadratic Optimization," Preprint, 2006 available at arXiv:cs/0603058

[6] Luca Trevisan, "Inapproximability of combinatorial optimization problems," Technical Report TR04-065, Electronic Colloquium on Computational Complexity, 2004.

[7] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," UC Berkeley, Dept. of Statistics, Technical Report 649. September, 2003.

[8] J. Yedidia, W. Freeman and Y. Weiss, "Generalized Belief Propagation," *Mitsubishi Elect. Res. Lab.*, TR-2000-26, 2000.

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