A Graphical Transformation for Belief Propagation: Maximum Weight Matchings and Odd-Sized Cycles

Jinwoo Shin* Andrew E. Gelfand† Michael Chertkov‡

May 29, 2017

Abstract

Max-product 'belief propagation' (BP) is a popular distributed heuristic for finding the Maximum A Posteriori (MAP) assignment in a joint probability distribution represented by a Graphical Model (GM). It was recently shown that BP converges to the correct MAP assignment for a class of loopy GMs with the following common feature: the Linear Programming (LP) relaxation to the MAP problem is tight (has no integrality gap). Unfortunately, tightness of the LP relaxation does not, in general, guarantee convergence and correctness of the BP algorithm. The failure of BP in such cases motivates reverse engineering a solution – namely, given a tight LP, can we design a 'good' BP algorithm.

In this paper, we design a BP algorithm for the Maximum Weight Matching (MWM) problem over general graphs. We prove that the algorithm converges to the correct optimum if the respective LP relaxation, which may include inequalities associated with non-intersecting odd-sized cycles, is tight. The most significant part of our approach is the introduction of a novel graph transformation designed to force convergence of BP. Our theoretical result suggests an efficient BP-based heuristic for the MWM problem, which consists of making sequential, "cutting plane", modifications to the underlying GM. Our experiments show that this heuristic performs as well as traditional cutting-plane algorithms using LP solvers on MWM problems.

*Mathematical Sciences Department at IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA. Email: mijirim@gmail.com

†Department of Computer Science at University of California, Irvine, CA 92697, USA. The author is also with Theoretical Division of Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: agelfand@ics.uci.edu

‡Theoretical Division & Center for Nonlinear Studies of Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: chertkov@lanl.gov

1 Introduction

Graphical Models (GMs) provide a useful representation for reasoning in a range of scientific fields [1, 2, 3, 4]. Such models use a graph structure to encode the joint probability distribution, where vertices correspond to random variables and edges (or lack of thereof) specify conditional dependencies. An important inference task in many applications involving GMs is to find the most likely assignment to the variables in a GM - the maximum a posteriori (MAP) configuration. Belief Propagation (BP) is a popular algorithm for approximately solving the MAP inference problem. BP is an iterative, message passing algorithm that is exact on tree structured GMs. However, BP often shows remarkably strong heuristic performance beyond trees, i.e. on GMs with loops. Distributed implementation, associated ease of programming and strong parallelization potential are the main reasons for the growing popularity of the BP algorithm.

The convergence and correctness of BP was recently established for a certain class of loopy GM formulations of several classic combinatorial optimization problems, including matchings [5, 6, 7], perfect matchings [8], independent sets [9] and network flows [10]. The important common feature of these instances is that BP converges to a correct MAP assignment when the Linear Programming (LP) relaxation of the MAP inference problem is tight, i.e., it shows no integrality gap. While this demonstrates that LP tightness is necessary for the convergence and correctness of BP, it is unfortunately not sufficient in general. In other words, BP may not work even when the corresponding LP relaxation to the MAP inference problem is tight. This motivates a quest for improving the BP approach so that it works, at least, when the LP is tight.

In this paper, we study if BP can be used as an iterative, message passing-based LP solver in cases when a LP (relaxation) is tight. We do so by considering a specific class of GMs corresponding to the Maximum Weight Matching (MWM) problem. It was recently shown [13] that a MWM can be found in polynomial time by solving a carefully chosen sequence of LP relaxations, where the sequence of LPs are formed by adding and removing sets of so-called "blossom" inequalities [11] to the base LP relaxation. Utilizing successive LP relaxations to solve the MWM problem is an example of the popular cutting plane method for solving combinatorial optimization problems [12]. While the approach in [13] is remarkable in that one needs only a polynomial number of "cut" inequalities, it unfortunately requires solving an emerging sequence of LPs via traditional, centralized methods (e.g., ellipsoid, interior-point or simplex) that may not be practical for large-scale problems. This motivates our search for an efficient and distributed BP-based LP solver for this class of problems.

Our work builds upon that of Sanghavi, Malioutov and Willsky [6], who studied BP for the GM formulation of the MWM problem on an arbitrary graph. The authors showed that the max-product BP converges to the correct, MAP solution if the base LP relaxation with no blossom - referred to herein as MWM-LP - is tight. Unfortunately, the tightness is not guaranteed in general, and the convergence and correctness for the max-product BP do not readily extend to a GM formulation with blossom constraints.

To resolve this issue, we propose a novel GM formulation of the MWM problem and show that the max-product BP on this new GM converges to the MWM assignment as long as the MWM-LP relaxation with blossom constraints is tight. The only restriction placed on our GM construction is that the set of blossom constraints added to the base MWM-LP be non-intersecting (in edges). Our GM construction is motivated by the so-called 'degree-two' (DT) condition, which simply means that every variable in a GM is associated to at most two (simple) factor functions. The DT condition is necessary for analysis of BP using the computational tree technique, developed and advanced in [5, 6, 10, 14, 16, 17] – the technique is one of the most powerful tools for analyzing BP algorithms. Note, that the DT condition is not satisfied by the standard MWM GM, and hence, we design a new GM satisfying the DT condition via a certain graphical transformation - collapsing odd cycles into new vertices and defining new weights on the contracted graph. Importantly, the MAP assignments of two GMs are in one-to-one correspondence. This clever graphical transformation allows us to use the computational tree techniques to prove the convergence and correctness of BP on the new GM.

Our theoretical result naturally guides a cutting-plane method suggesting a sequential and adaptive design of GMs using respective BPs. We use the output of BP to identify odd-sized cycle constraints - "cuts" - to add to the MWM-LP, construct a new GM using our graphical transformation, run BP and repeat. We evaluate this heuristic approach empirically and show that its performance is close to the traditional cutting-plane approach employing LP solvers rather than BP, i.e., BP is as powerful as an LP solver. Finally, we note that the DT condition may neither be sufficient nor necessary for BP to work. It was necessary, however, to provide theoretical guarantees for the special class of GMs considered. We believe that our success in crafting a graphical transformation will offer useful insight into the design and analysis of BP algorithms on a wider class of MAP inference problems.

Organization. In Section 2, we introduce a standard GM formulation of the MWM problem as well as the corresponding BP and LP. In Section 3, we introduce our new GM and describe performance guarantees of the respective BP algorithm. In Section 4, we describe a cutting-plane(-like) method using BP for the MWM problem and show its empirical performance for random MWM instances.

2 Preliminaries

2.1 Graphical Model for Maximum Weight Matchings

A joint distribution of n (discrete) random variables $Z = [Z_i] \in \Omega^n$ is called a Graphical Model (GM) if it factorizes as follows: for $z = [z_i] \in \Omega^n$,

$$\Pr[Z = z] \;\propto\; \prod_{\alpha \in F} \psi_\alpha(z_\alpha), \tag{1}$$

where F is a collection of subsets of Ω, $z_\alpha = [z_i : i \in \alpha \subset \Omega]$ is a subset of variables, and ψ_α is some (given) non-negative function. The function ψ_α is called a factor (variable) function if $|\alpha| \geq 2$ ($|\alpha| = 1$). For variable functions ψ_α with $\alpha = \{i\}$, we simply write $\psi_\alpha = \psi_i$. One calls z a valid assignment if $\Pr[Z = z] > 0$. The MAP assignment z^* is defined as

$$z^* \;=\; \arg\max_{z \in \Omega^n} \Pr[Z = z].$$

Let us introduce the Maximum Weight Matching (MWM) problem and its related GM. Suppose we are given an undirected graph $G = (V, E)$ with weights $\{w_e : e \in E\}$ assigned to its edges. A *matching* is a set of edges without common vertices. The weight of a matching is the sum of corresponding edge weights. The MWM problem consists of finding a matching of maximum weight. Associate a binary random variable with each edge $X = [X_e] \in \{0,1\}^{|E|}$ and consider the probability distribution: for $x = [x_e] \in \{0,1\}^{|E|}$,

$$\Pr[X = x] \;\propto\; \prod_{e \in E} e^{w_e x_e} \prod_{i \in V} \psi_i(x) \prod_{C \in \mathcal{C}} \psi_C(x), \tag{2}$$

where

$$\psi_i(x) = \begin{cases} 1 & \text{if } \sum_{e\in\delta(i)} x_e \le 1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \psi_C(x) = \begin{cases} 1 & \text{if } \sum_{e\in E(C)} x_e \le \frac{|C|-1}{2} \\ 0 & \text{otherwise} \end{cases}.$$

Here $\mathcal{C}$ is a set of odd-sized cycles $\mathcal{C} \subset 2^V$, $\delta(i) = \{(i,j) \in E\}$ and $E(C) = \{(i,j) \in E : i,j \in C\}$. Throughout the manuscript, we assume that cycles are non-intersecting in edges, i.e., $E(C_1) \cap E(C_2) = \emptyset$ for all $C_1, C_2 \in \mathcal{C}$. It is easy to see that a MAP assignment x^* for the GM (2) induces a MWM in G. We also assume that the MAP assignment is unique.

2.2 Belief Propagation and Linear Programming for Maximum Weight Matchings

In this section, we introduce max-product Belief Propagation (BP) and the Linear Programming (LP) relaxation to computing the MAP assignment in (2). We first describe the BP algorithm for the general GM (1), then tailor the algorithm to the MWM GM (2). The BP algorithm updates the set of $2|\Omega|$ messages $\{m^t_{\alpha\to i}(z_i), m^t_{i\to\alpha}(z_i) : z_i \in \Omega\}$ between every variable i and its associated factors $\alpha \in F_i = \{\alpha \in F : i \in \alpha, |\alpha| \ge 2\}$ using the following update rules:

$$m^{t+1}_{\alpha\to i}(z_i) = \sum_{z':z'_i=z_i} \psi_\alpha(z') \prod_{j\in\alpha\setminus i} m^t_{j\to\alpha}(z'_j) \qquad \text{and} \qquad m^{t+1}_{i\to\alpha}(z_i) = \psi_i(z_i) \prod_{\alpha'\in F_i\setminus\alpha} m^t_{\alpha'\to i}(z_i).$$

Here t denotes time and initially $m^0_{\alpha\to i}(\cdot) = m^0_{i\to\alpha}(\cdot) = 1$. Given a set of messages $\{m_{i\to\alpha}(\cdot), m_{\alpha\to i}(\cdot))\}$, the BP (max-marginal) beliefs $\{n_i(z_i)\}$ are defined as follows:

$$n_i(z_i) = \psi_i(z_i) \prod_{\alpha\in F_i} m_{\alpha\to i}(z_i).$$

For the GM (2), we let $n^t_e(\cdot)$ to denote the BP belief on edge $e \in E$ at time t. The algorithm outputs the MAP estimate at time t, $x^{\text{BP}}(t) = [x^{\text{BP}}_e(t)] \in [0,?,1]^{|E|}$, using the using the beliefs and the rule:

$$x^{\text{BP}}_e(t) = \begin{cases} 1 & \text{if } n^t_e(0) < n^t_e(1) \\ ? & \text{if } n^t_{ij}(0) = n^t_e(1) \\ 0 & \text{if } n^t_e(0) > n^t_e(1) \end{cases}.$$

The LP relaxation to the MAP problem for the GM (2) is:

$$\begin{aligned} \text{C-LP}: \qquad & \max \sum_{e\in E} w_e x_e \\ \text{s.t.} \qquad & \sum_{e\in\delta(i)} x_e \le 1, \quad \forall i \in V, \qquad \sum_{e\in E(C)} x_e \le \frac{|C|-1}{2}, \quad \forall C \in \mathcal{C}, \qquad x_e \in [0,1]. \end{aligned}$$

Observe that if the solution $x^{\text{C-LP}}$ to C-LP is integral, i.e., $x^{\text{C-LP}} \in \{0,1\}^{|E|}$, then it is a MAP assignment, i.e., $x^{\text{C-LP}} = x^*$. Sanghavi, Malioutov and Willsky [6] proved the following theorem connecting the performance of BP and C-LP in a special case:

Theorem 1 *If $\mathcal{C} = \emptyset$ and the solution of C-LP is integral and unique, then $x^{BP}(t)$ under the GM (2) converges to the MWM assignment x^*.*

Adding small random component to every weight guarantees the uniqueness condition required by Theorem 1. A natural hope is that the Theorem 1 extends to a non-empty $\mathcal{C}$ since adding more cycles can help to reduce the integrality gap of C-LP. However, the theorem does not hold when $\mathcal{C} \neq \emptyset$. For example, BP does not converge for a triangle graph with edge weights $\{2, 1, 1\}$ and $\mathcal{C}$ consisting of the only cycle. This is true even though the solution of the corresponding C-LP is unique and integral.

3 Graphical Transformation for Convergent and Correct Belief Propagation

The loss of convergence and correctness of BP when the MWM LP is tight (and unique) but $\mathcal{C} \neq \emptyset$ motivates the work in this section. We resolve the issue by designing a new GM, equivalent to the original GM, such that when BP is run on this new GM it converges to the MAP/MWM assignment whenever the LP relaxation is tight and unique - even if $\mathcal{C} \neq \emptyset$. The new GM is defined on an auxiliary graph $G' = (V', E')$ with new weights $\{w'_e : e \in E'\}$, as follows

$$V' = V \cup \{i_C : C \in \mathcal{C}\}, \qquad E' = E \cup \{(i_C, j) : j \in V(C), C \in \mathcal{C}\} \setminus \{e : e \in \cup_{C \in \mathcal{C}} E(C)\}$$
$$w'_e = \begin{cases} \frac{1}{2} \sum_{e' \in E(C)} (-1)^{d_C(j,e')} w_{e'} & \text{if } e = (i_C, j) \quad \text{for some } C \in \mathcal{C} \\ w_e & \text{otherwise} \end{cases}.$$

Here $d_C(j, e)$ is the graph distance of j and e in cycle $C = (j_1, j_2, \ldots, j_k)$, e.g., if $e = (j_2, j_3)$, then $d_C(j_1, e) = 1$.

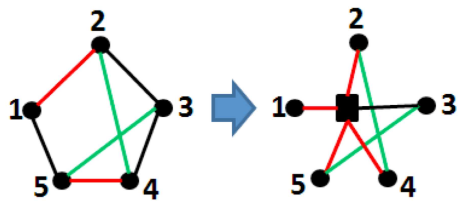

Figure 1: Example of original graph G (left) and new graph G' (right).

Associate a binary variable with each new edge and consider the new probability distribution on $y = [y_e : e \in E'] \in \{0, 1\}^{|E'|}$:

$$\Pr[Y = y] \propto \prod_{e \in E'} e^{w'_e y_e} \prod_{i \in V} \psi_i(y) \prod_{C \in \mathcal{C}} \psi_C(y), \tag{3}$$

where

$$\psi_i(y) = \begin{cases} 1 & \text{if } \sum\limits_{e \in \delta(i)} y_e \leq 1 \\ 0 & \text{otherwise} \end{cases} \qquad \psi_C(y) = \begin{cases} 0 & \text{if } \sum\limits_{e \in \delta(i_C)} y_e > |C| - 1 \\ 0 & \text{if } \sum\limits_{j \in V(C)} (-1)^{d_C(j,e)} y_{i_C, j} \notin \{0, 2\} \text{ for some } e \in E(C) \\ 1 & \text{otherwise} \end{cases}.$$

It is not hard to check that the number of operations required to update messages at each round of BP under the above GM is $O(|V||E|)$, since the number of non-intersecting cycles is at most $|E|$ and messages updates involving the factor ψ_C can be efficiently done using the dynamic programming. We are now ready to state the main result of this paper.

Theorem 2 *If the solution of C-LP is integral and unique, then the BP-MAP estimate $y^{BP}(t)$ under the GM (3) converges to the corresponding MAP assignment y^*. Furthermore, the MWM assignment x^* is reconstructible from y^* as:*

$$x_e^* = \begin{cases} \frac{1}{2}\sum_{j\in V(C)}(-1)^{d_C(j,e)}y^*_{i_C,j} & \text{if } \ e \in \bigcup_{C\in\mathcal{C}} E(C) \\ y_e^* & \text{otherwise} \end{cases}. \tag{4}$$

The proof of Theorem 2 is provided in the following sections. We also establish the convergence time of the BP algorithm under the GM (3) (see Lemma 3). We stress that the new GM (3) is designed so that each variable is associated to at most two factor nodes. We call this condition, which did not hold for the original GM (2), the 'degree-two' (DT) condition. The DT condition will play a critical role in the proof of Theorem 2. We further remark that even under the DT condition and given tightness/uniqueness of the LP, proving correctness and convergence of BP is still highly non trivial. In our case, it requires careful study of the computation tree induced by BP with appropriate truncations at its leaves.

3.1 Main Lemma for Proof of Theorem 2

Let us introduce the following auxiliary LP over the new graph and weights.

$$\begin{aligned} \text{C-LP}': \quad & \max \sum_{e\in E'} w'_e y_e \\ \text{s.t.} \quad & \sum_{e\in\delta(i)} y_e \le 1, \quad \forall i \in V, \quad y_e \in [0,1], \quad \forall e \in E', \qquad (5)\\ & \sum_{j\in V(C)} (-1)^{d_C(j,e)} y_{i_C,j} \in [0,2], \quad \forall e \in E(C), \quad \sum_{e\in\delta(i_C)} y_e \le |C|-1, \quad \forall C\in\mathcal{C}. \qquad (6) \end{aligned}$$

Consider the following one-to-one linear mapping between $x = [x_e : e \in E]$ and $y = [y_e : e \in E']$:

$$y_e = \begin{cases} \sum_{e'\in E(C)\cap\delta(i)} x_{e'} & \text{if } e = (i, i_C) \\ x_e & \text{otherwise} \end{cases} \qquad x_e = \begin{cases} \frac{1}{2}\sum_{j\in V(C)}(-1)^{d_C(j,e)}y_{i_C,j} & \text{if } e \in \bigcup_{C\in\mathcal{C}} E(C) \\ y_e & \text{otherwise} \end{cases}.$$

Under the mapping, one can check that C-LP = C-LP$'$ and if the solution $x^{\text{C-LP}}$ of C-LP is unique and integral, the solution $y^{\text{C-LP}'}$ of C-LP$'$ is as well, i.e., $y^{\text{C-LP}'} = y^*$. Hence, (4) in Theorem 2 follows. Furthermore, since the solution $y^* = [y_e^*]$ to C-LP$'$ is unique and integral, there exists $c > 0$ such that

$$c = \inf_{y\neq y^*: y \text{ is feasible to C-LP}'} \frac{w'\cdot(y^*-y)}{|y^*-y|},$$

where $w' = [w'_e]$. Using this notation, we establish the following lemma characterizing performance of the max-product BP over the new GM (3). Theorem 2 follows from this lemma directly.

Lemma 3 *If the solution $y^{\text{C-LP}'}$ of C-LP$'$ is integral and unique, i.e., $y^{\text{C-LP}'} = y^*$, then*

- *If* $y_e^* = 1,\ n_e^t[1] > n_e^t[0]$ *for all* $t > \frac{6w'_{\max}}{c} + 6,$
- *If* $y_e^* = 0,\ n_e^t[1] < n_e^t[0]$ *for all* $t > \frac{6w'_{\max}}{c} + 6,$

where $n_e^t[\cdot]$ *denotes the BP belief of edge* e *at time* t *under the GM* (3) *and* $w'_{\max} = \max_{e \in E'} |w'_e|$.

3.2 Proof of Lemma 3

This section provides the complete proof of Lemma 3. We focus here on the case of $y_e^* = 1$, while translation of the result to the opposite case of $y_e^* = 0$ is straightforward. To derive a contradiction, assume that $n_e^t[1] \le n_e^t[0]$ and construct a tree-structured GM $T_e(t)$ of depth $t+1$, also known as the computational tree, using the following scheme

1. Add a copy of $Y_e \in \{0,1\}$ as the (root) variable (with variable function $e^{w'_e Y_e}$).
2. Repeat the following t times for each leaf variable Y_e on the current tree-structured GM.
   - 2-1. For each $i \in V$ such that $e \in \delta(i)$ and ψ_i is not associated to Y_e of the current model, add ψ_i as a factor (function) with copies of $\{Y_{e'} \in \{0,1\} : e' \in \delta(i) \setminus e\}$ as child variables (with corresponding variable functions, i.e., $\{e^{w'_{e'} Y_{e'}}\}$).
   - 2-2. For each $C \in \mathcal{C}$ such that $e \in \delta(i_C)$ and ψ_C is not associated to Y_e of the current model, add ψ_C as a factor (function) with copies of $\{Y_{e'} \in \{0,1\} : e' \in \delta(i_C) \setminus e\}$ as child variables (with corresponding variable functions, i.e., $\{e^{w'_{e'} Y_{e'}}\}$).

It is known from [15] that there exists a MAP configuration y^{TMAP} on $T_e(t)$ with $y_e^{\text{TMAP}} = 0$ at the root variable. Now we construct a new assignment y^{NEW} on the computational tree $T_e(t)$ as follows.

1. Initially, set $y^{\text{NEW}} \leftarrow y^{\text{TMAP}}$ and e is the root of the tree.
2. $y^{\text{NEW}} \leftarrow \texttt{FLIP}_e(y^{\text{NEW}})$.
3. For each child factor ψ, which is equal to ψ_i (i.e., $e \in \delta(i)$) or ψ_C (i.e., $e \in \delta(i_C)$), associated with e,
   - (a) If ψ is satisfied by y^{NEW} and $\texttt{FLIP}_e(y^*)$ (i.e., $\psi(y^{\text{NEW}}) = \psi(\texttt{FLIP}_e(y^*)) = 1$), then do nothing.
   - (b) Else if there exists a e's child e' through factor ψ such that $y_{e'}^{\text{NEW}} \neq y_{e'}^*$ and ψ is satisfied by $\texttt{FLIP}_{e'}(y^{\text{NEW}})$ and $\texttt{FLIP}_{e'}(\texttt{FLIP}_e(y^*))$, then go to the step 2 with $e \leftarrow e'$.
   - (c) Otherwise, report ERROR.

In the construction, $\texttt{FLIP}_e(y)$ is the 0-1 vector made by flipping (i.e., changing from 0 to 1 or 1 to 0) the e's position in y. We note that there exists exactly one child factor ψ in step 3 and we only choose one child e' in step (b) (even though there are many possible candidates). Due to this reason, flip operations induce a path structure P in tree $T_e(t)$.[1] Now we state the following key lemma for the above construction of y^{NEW}.

Lemma 4 ERROR *is never reported in the construction described above.*

[1] P may not have an alternating structure since both y_e^{NEW} and its child $y_{e'}^{\text{NEW}}$ can be flipped in a same way.

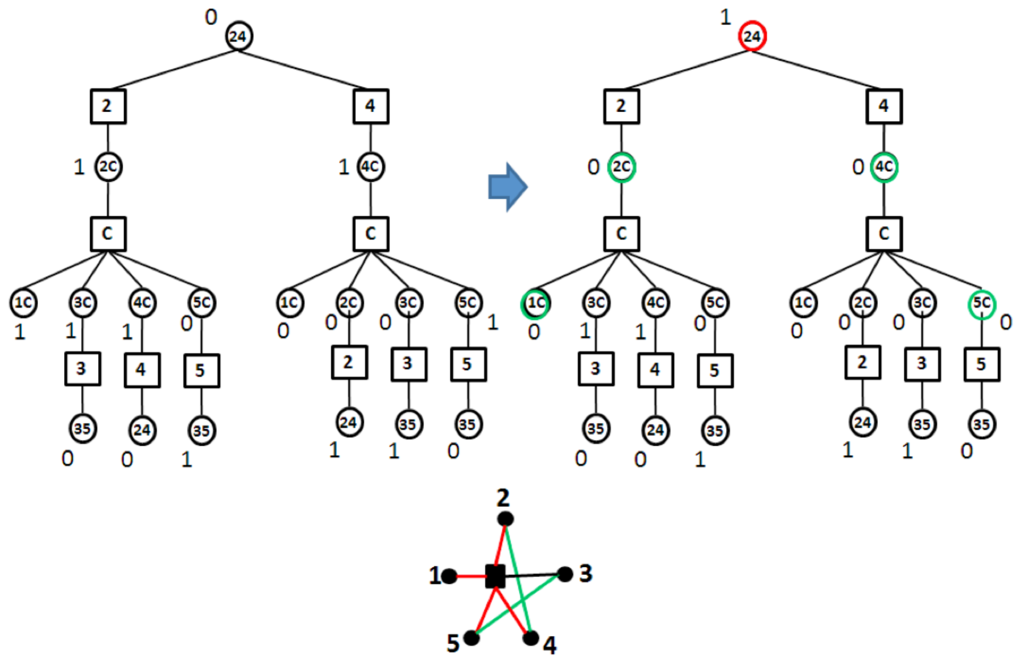

Figure 2: Example of y^{TMAP} (left) and y^{NEW} (right), where $y^*_{1,i_C}=0$, $y^*_{2,i_C}=0$, $y^*_{3,i_C}=0$, $y^*_{4,i_C}=0$, $y^*_{5,i_C}=0$, $y^*_{3,5}=1$ and $y^*_{2,4}=1$.

Proof. The case when $\psi=\psi_i$ at the step 3 is easy, and we only provide the proof for the case when $\psi=\psi_C$. We also assume that y^{NEW}_e is flipped as $1\to 0$ (i.e., $y^*_e=0$), where the proof for the case $0\to 1$ follows in a similar manner. First, one can observe that y satisfies ψ_C if and only if y is the 0-1 indicator vector of a union of disjoint even paths in the cycle C. Since y^{NEW}_e is flipped as $1\to 0$, the even path including e is broken into an even (possibly, empty) path and an odd (always, non-empty) path. We consider two cases: (a) there exists e' within the odd path (i.e., $y^{\text{NEW}}_{e'}=1$) such that $y^*_{e'}=0$ and flipping $y^{\text{NEW}}_{e'}$ as $1\to 0$ broke the odd path into two even (disjoint) paths; (b) there exists no such e' within the odd path.

For the first case (a), it is easy to see that we can maintain the structure of disjoint even paths in y^{NEW} after flipping $y^{\text{NEW}}_{e'}$ as $1\to 0$, i.e., ψ is satisfied by $\texttt{FLIP}_{e'}(y^{\text{NEW}})$. For the second case (b), we choose e' as a neighbor of the farthest end point (from e) in the odd path, i.e., $y^{\text{NEW}}_{e'}=0$ (before flipping). Then, $y^*_{e'}=1$ since y^* satisfies factor ψ_C and induces a union of disjoint even paths in the cycle C. Therefore, if we flip $y^{\text{NEW}}_{e'}$ as $0\to 1$, then we can still maintain the structure of disjoint even paths in y^{NEW}, ψ is satisfied by $\texttt{FLIP}_{e'}(y^{\text{NEW}})$. The proof for the case of the ψ satisfied by $\texttt{FLIP}_{e'}(\texttt{FLIP}_e(y^*))$ is similar. This completes the proof of Lemma 4.□

Due to how it is constructed y^{NEW} is a valid configuration, i.e., it satisfies all the factor functions in $T_e(t)$. Hence, it suffices to prove that $w'(y^{\text{NEW}})>w'(y^{\text{TMAP}})$, which contradicts to the assumption that y^{MAP} is a MAP configuration on $T_e(t)$. To this end, for $(i,j)\in E'$, let $n^{0\to 1}_{ij}$ and $n^{1\to 0}_{ij}$ be the number of flip operations $0\to 1$ and $1\to 0$ for copies of (i,j) in the step 2 of the construction of $T_e(t)$. Then, one derives

$$w'(y^{\text{NEW}})=w'(y^{\text{TMAP}})+w'\cdot n^{0\to 1}-w'\cdot n^{1\to 0},$$

where $n^{0\to1} = [n_{ij}^{0\to1}]$ and $n^{1\to0} = [n_{ij}^{1\to0}]$. We consider two cases: (i) the path P does not arrive at a leave variable of $T_e(t)$, and (ii) otherwise. Note that the case (i) is possible only when the condition in the step (a) holds during the construction of y^{NEW}.

Case (i). In this case, we define $y_{ij}^{\dagger} := y_{ij}^* + \varepsilon(n_{ij}^{1\to0} - n_{ij}^{0\to1})$, and establish the following lemma.

Lemma 5 *$y^\dagger$ is feasible to C-LP′ for small enough $\varepsilon > 0$.*

Proof. We have to show that $y^\dagger$ satisfies (5) and (6). Here, we prove that $y^\dagger$ satisfies (6) for small enough $\varepsilon > 0$, and the proof for (5) can be argued in a similar manner. To this end, for given $C \in \mathcal{C}$, we consider the following polytope $\mathcal{P}_C$:

$$\sum_{j\in V(C)} y_{i_C,j} \le |C| - 1, \quad y_{i_C,j} \in [0,1], \quad \forall j \in C, \qquad \sum_{j\in V(C)} (-1)^{d_C(j,e)} y_{i_C,j} \in [0,2], \quad \forall e \in E(C).$$

We have to show that $y_C^\dagger = [y_e : e \in \delta(i_C)]$ is within the polytope. It is easy to see that the condition of the step (a) never holds if $\psi = \psi_C$ in the step 3. For the i-th copy of ψ_C in $P \cap T_e(t)$, we set $y_C^*(i) = \texttt{FLIP}_{e'}(\texttt{FLIP}_e(y_C^*))$ in the step (b), where $y_C^*(i) \in \mathcal{P}_C$. Since the path P does not hit a leave variable of $T_e(t)$, we have

$$\frac{1}{N}\sum_{i=1}^{N} y_C^*(i) = y_C^* + \frac{1}{N}\left(n_C^{1\to0} - n_C^{0\to1}\right),$$

where N is the number of copies of ψ_C in $P \cap T_e(t)$. Furthermore, $\frac{1}{N}\sum_{i=1}^{N} y_C^*(i) \in \mathcal{P}_C$ due to $y_C^*(i) \in \mathcal{P}_C$. Therefore, $y_C^\dagger \in \mathcal{P}_C$ if $\varepsilon \le 1/N$. This completes the proof of Lemma 5. □

The above lemma with $w'(y^*) > w'(y^\dagger)$ (due to the uniqueness of y^*) implies that $w' \cdot n^{0\to1} > w' \cdot n^{1\to0}$, which leads to $w'(y^{\text{NEW}}) > w'(y^{\text{TMAP}})$.

Case (ii). We consider the case when only one end of P hits a leave variable Y_e of $T_e(t)$, where the proof of the other case follows in a similar manner. In this case, we define $y_{ij}^{\ddagger} := y_{ij}^* + \varepsilon(m_{ij}^{1\to0} - m_{ij}^{0\to1})$, where $m^{1\to0} = [m_{ij}^{1\to0}]$ and $m^{0\to1} = [m_{ij}^{0\to1}]$ is constructed as follows:

1. Initially, set $m^{1\to0}, m^{0\to1}$ by $n^{1\to0}, n^{0\to1}$.
2. If y_e^{NEW} is flipped as $1 \to 0$ and it is associated to a cycle parent factor ψ_C for some $C \in \mathcal{C}$, then decrease $m_e^{1\to0}$ by 1 and
   - 2-1. If the parent $y_{e'}^{\text{NEW}}$ is flipped from $1 \to 0$, then decrease $m_{e'}^{1\to0}$ by 1.
   - 2-2. Else if there exists a 'brother' edge $e'' \in \delta(i_C)$ of e such that $y_{e''}^* = 1$ and ψ_C is satisfied by $\texttt{FLIP}_{e''}(\texttt{FLIP}_{e'}(y^*))$, then increase $m_{e''}^{0\to1}$ by 1.
   - 2-3. Otherwise, report ERROR.
3. If y_e^{NEW} is flipped as $1 \to 0$ and it is associated to a vertex parent factor ψ_i for some $i \in V$, then decrease $m_e^{1\to0}$ by 1.
4. If y_e^{NEW} is flipped as $0 \to 1$ and it is associated to a vertex parent factor ψ_i for some $i \in V$, then decrease $m_e^{0\to1}, m_{e'}^{1\to0}$ by 1, where $e' \in \delta(i)$ is the 'parent' edge of e, and

4-1. If the parent $y_{e'}^{\text{NEW}}$ is associated to a cycle parent factor ψ_C,

4-1-1. If the grad-parent $y_{e''}^{\text{NEW}}$ is flipped from $1 \to 0$, then decrease $m_{e''}^{1\to 0}$ by 1.

4-1-2. Else if there exists a 'brother' edge $e''' \in \delta(i_C)$ of e' such that $y^*_{e'''} = 1$ and ψ_C is satisfied by $\texttt{FLIP}_{e'''}(\texttt{FLIP}_{e''}(y^*))$, then increase $m_{e'''}^{0\to 1}$ by 1.

4-1-3. Otherwise, report ERROR.

4-2. Otherwise, do nothing.

We establish the following lemmas.

Lemma 6 ERROR *is never reported in the above construction.*

Lemma 7 $y^\ddagger$ *is feasible to C-LP′ for small enough* $\varepsilon > 0$.

Proofs of Lemma 6 and Lemma 7 are analogous to those of Lemma 4 and Lemma 5, respectively. From Lemma 7, we have

$$c \;\le\; \frac{w' \cdot (y^* - y^\ddagger)}{|y^* - y^\ddagger|} \;\le\; \frac{\varepsilon\left(w'(m^{0\to 1} - m^{1\to 0})\right)}{\varepsilon(t-3)} \;\le\; \frac{\varepsilon\left(w'(n^{0\to 1} - n^{1\to 0}) + 3w'_{\max}\right)}{\varepsilon(t-3)},$$

where $|y^* - y^\ddagger| \ge \varepsilon(t-3)$ follows from the fact that P hits a leave variable of $T_e(t)$ and there are at most three increases or decreases in $m^{0\to 1}$ and $m^{1\to 0}$ in the above construction. Hence,

$$w'(n^{0\to 1} - n^{1\to 0}) \ge c(t-3) - 3w'_{\max} > 0 \qquad \text{if} \quad t > \frac{3w'_{\max}}{c} + 3,$$

which implies $w'(y^{\text{NEW}}) > w'(y^{\text{TMAP}})$. If both ends of P hit leave variables of $T_e(t)$, we need $t > \frac{6w'_{\max}}{c} + 6$. This completes the proof of Lemma 3.

4 Cutting-plane Algorithm using Belief Propagation

In the previous section we established that BP on a carefully designed GM using appropriate odd cycles solves the MWM problem as long as the corresponding MWM-LP relaxation is tight. However, finding a collection of odd-sized cycles to ensure tightness of the MWM-LP is a challenging task. In this section, we provide a heuristic algorithm which we call CP-BP (cutting-plane method using BP) for this task. It consists of making sequential, "cutting plane", modifications to the underlying GM using the output of the BP algorithm in the previous step. CP-BP is defined as follows:

1. Initialize $\mathcal{C} = \emptyset$.

2. Run BP on the GM (3) for T iterations

3. For each edge $e \in E$, set $y_e = \begin{cases} 1 & \text{if } n_e^T[1] > n_e^T[0] \text{ and } n_e^{T-1}[1] > n_e^{T-1}[0] \\ 0 & \text{if } n_e^T[1] < n_e^T[0] \text{ and } n_e^{T-1}[1] < n_e^{T-1}[0] \\ 1/2 & \text{otherwise} \end{cases}$.

4. Compute $x = [x_e]$ using $y = [y_e]$ as per (4), and terminate if $x \notin \{0, 1/2, 1\}^{|E|}$.

5. If there is no edge e with $x_e = 1/2$, return x. Otherwise, add a non-intersecting odd cycle (if exists) of edges $\{e : x_e = 1/2\}$ to $\mathcal{C}$ and go to step 2.

In the above procedure, BP can be replaced by an LP solver to directly obtain x in step 4. This results in a traditional cutting-plane LP (CP-LP) method for the MWM problem [18]. The primary reason why we design CP-BP to terminate when $x \notin \{0, 1/2, 1\}^{|E|}$ is because the solution x of C-LP is always half integral [2]. Note that $x \notin \{0, 1/2, 1\}^{|E|}$ occurs in CP-BP when BP does not find the solution of C-LP.

We compare CP-BP and CP-LP in order to gauge the effectiveness of BP as an LP solver for MWM problems - i.e., to test if BP is as powerful as an LP solver on this class of problems. We consider a set of random sparse graph instances. We construct a set of 100 complete graphs on $|V| = \{50, 100\}$ nodes and "sparsify" each graph instance by eliminating edges with probability $p = \{0.5, 0.9\}$. We assign integral weights, drawn uniformly in $[1, 2^{20}]$, to the remaining edges.

The results are summarized in Table 1 and show that: 1) CP-BP is almost as good as CP-LP for solving the MWM problem; and 2) our graphical transformation allows BP to solve significantly more MWM problems than are solvable by BP run on the 'bare' LP without odd-sized cycles.

50 % sparse graphs				90 % sparse graphs			
$\lvert V\rvert$ / $\lvert E\rvert$	# CP-BP	# Tight LPs	# CP-LP	$\lvert V\rvert$ / $\lvert E\rvert$	# CP-BP	# Tight LPs	# CP-LP
50 / 490	94 %	65 %	98 %	50 / 121	90 %	59 %	91 %
100 / 1963	92 %	48 %	95 %	100 / 476	63 %	50 %	63 %

Table 1: Evaluation of CP-BP and CP-LP on random MWM instances. Columns *# CP-BP* and *# CP-LP* indicate the percentage of instances in which the cutting plane methods found a MWM. The column *# Tight LPs* indicates the percentage for which the initial MWM-LP is tight (i.e. $\mathcal{C} = \emptyset$).

References

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

[2] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory.* Cambridge University Press, 2008.

[3] M. Mezard and A. Montanari, *Information, physics, and computation*, ser. Oxford Graduate Texts. Oxford: Oxford Univ. Press, 2009.

[4] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1, pp. 1–305, 2008.

[5] M. Bayati, D. Shah, and M. Sharma, "Max-product for maximum weight matching: Convergence, correctness, and lp duality," *IEEE Transactions on Information Theory*, vol. 54, no. 3, pp. 1241 –1251, 2008.

[2] A proof of $\frac{1}{2}$-integrality, which we did not find in the literature, is presented in the appendix.

[6] S. Sanghavi, D. Malioutov, and A. Willsky, "Linear Programming Analysis of Loopy Belief Propagation for Weighted Matching," in *Neural Information Processing Systems (NIPS)*, 2007

[7] B. Huang, and T. Jebara, "Loopy belief propagation for bipartite maximum weight b-matching," in *Artificial Intelligence and Statistics (AISTATS)*, 2007.

[8] M. Bayati, C. Borgs, J. Chayes, R. Zecchina, "Belief-Propagation for Weighted b-Matchings on Arbitrary Graphs and its Relation to Linear Programs with Integer Solutions," *SIAM Journal in Discrete Math*, vol. 25, pp. 989–1011, 2011.

[9] S. Sanghavi, D. Shah, and A. Willsky, "Message-passing for max-weight independent set," in *Neural Information Processing Systems (NIPS)*, 2007.

[10] D. Gamarnik, D. Shah, and Y. Wei, "Belief propagation for min-cost network flow: convergence & correctness," in *SODA*, pp. 279–292, 2010.

[11] J. Edmonds, "Paths, trees, and flowers", *Canadian Journal of Mathematics*, vol. 3, pp. 449–467, 1965.

[12] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, no. 4, pp. 393–410, 1954.

[13] K. Chandrasekaran, L. A. Vegh, and S. Vempala. "The cutting plane method is polynomial for perfect matchings," in *Foundations of Computer Science (FOCS)*, 2012

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

[15] Y. Weiss, "Belief propagation and revision in networks with loops," *MIT AI Laboratory*, Technical Report 1616, 1997.

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

[17] Y. Weiss, and W. T. Freeman, "On the Optimality of Solutions of the MaxProduct Belief-Propagation Algorithm in Arbitrary Graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 736–744. 2001.

[18] M. Grotschel, and O. Holland, "Solving matching problems with linear programming," *Mathematical Programming*, vol. 33, no. 3, pp. 243–259. 1985.

A Proof for Half Integrality of C-LP

In this section, we show that there always exists a half-integral solution to C-LP. To this end, it suffices to show that every vertex in the constraint polyope of C-LP is half integral. Let x be a feasible point in the constraint polyope of C-LP, and define the following notation.

$$\begin{aligned} E_x &= \{e \in E : x_e \in (0,1)\} \\ V_x &= \{v \in V : \sum_{e\in\delta(v)} x_e = 1\} \\ \mathcal{C}_x &= \left\{ C \in \mathcal{C} : \sum_{e \in E(C)} x_e = \frac{|C|-1}{2} \right\} \end{aligned}$$

Our goal is to show that x is either a non-vertex or a half-integral feasible point. To this end, one can assume $E_x \neq \emptyset$ (otherwise, we are done since x is integral) and and E_x does not contain a cycle consisting of half integral edges $\{e : x_e = 1/2\}$ (otherwise, one can redefine a new x by making $x_e \to 0$ for every edge e in the cycle to argue that the original x is either a non-vertex or a half-integral feasible point). Under these assumptions, we will show that x is not a vertex.

First it is easy to check that each $C \in \mathcal{C}_x$ is one of the following types:

T1. $E(C) \subset E_x$ and there exist at least two vertices $v_1, v_2 \in V(C)$ such that for $i = 1, 2$,

$$\sum_{e \in \delta(v_i) \in E(C)} x_e < 1. \tag{7}$$

T2. $E(C) \cap E_x$ is a disjoint union of $\delta(v_i) \cap E(C)$ for some $v_1, \dots, v_k \in V(C)$, and

$$\sum_{e \in \delta(v_i) \in E(C)} x_e = 1. \tag{8}$$

Now pick an arbitrary edge $e = (u, v) \in E_x$. We build a path in E_x starting from e, expanding in both directions under the following rule:

R1. When we have multiple choices of edges in the expansion procedure, we always prefer edges not in $\bigcup_{C \in \mathcal{C}_x} E(C)$.

In the first step of the expansion procedure, one can check one of the following cases occur:

C1. There exists an edge $e' = (v, w) \neq e \in E_x$;

C2. $v \notin V_x$ and $e \notin \cup_{C \in C_x} E(C)$.

Therefore, the expansion procedure terminates in one of the following cases:

(a) Case C2 occurs at both ends of the current path, i.e, impossible to expand further.

(b) A cycle $\mathcal{T}$ in E_x is found, i.e., the path self-intersects.

Case (a). In this case, we will show that x is not a vertex. Suppose the expansion procedure ends in a path $\mathcal{P}$, and consider a walk along the path. We observe that

- Once the walk enters an edge of cycle of Type T1, it goes out before traversing all edges of the cycle due to Rule R1 and (7).
- Once the walk enters an edge of cycle of Type T2, it goes out after traversing even number of edges of the cycle due to (8).

Hence, when the walk enters and goes out a cycle of Type T1, we can possibly remove traversed edges from $\mathcal{P}$ and add remaining non-traversed edges to $\mathcal{P}$ so that the walk in the modified $\mathcal{P}$ always traverses (before going out) even number of edges of the cycle once it enters. By modifying $\mathcal{P}$ in this way, we can make path $\mathcal{P} = e_1 \to e_2 \cdots \to e_k$ traverse even number of edges of cycles of Type T1 and T2 once it enters. Note that $\mathcal{P}$ may contain a same edge more than once. Finally, we construct two different points $y = [y_e], z = [z_e]$ with $x = (y+z)/2$ by starting from $x = [x_e]$ and alternatively adding/substracting some constant $\varepsilon > 0$ following path $\mathcal{P}$: $y_e = z_e = x_e$ for $e \notin \mathcal{P}$ and

$$\begin{aligned} y_{e_1} &\leftarrow x_{e_1} + \varepsilon, \quad & y_{e_2} &\leftarrow x_{e_2} - \varepsilon, \quad & y_{e_3} &\leftarrow x_{e_3} + \varepsilon, \cdots \\ z_{e_1} &\leftarrow x_{e_1} - \varepsilon, \quad & z_{e_2} &\leftarrow x_{e_2} + \varepsilon, \quad & z_{e_3} &\leftarrow x_{e_3} - \varepsilon, \cdots \end{aligned}$$

We provide an example of $\{x, y, z\}$ in Figure 3. For small enough $\varepsilon > 0$, one can show that y, z are feasible points to the constraint polytope of C-LP using the following facts:

- $\mathcal{P}$ always traverses even number of edges of cycles of Type T1 and T2 once it enters;
- Both ends of $\mathcal{P}$ belong to Case C2.

This implies that x is not a vertex.

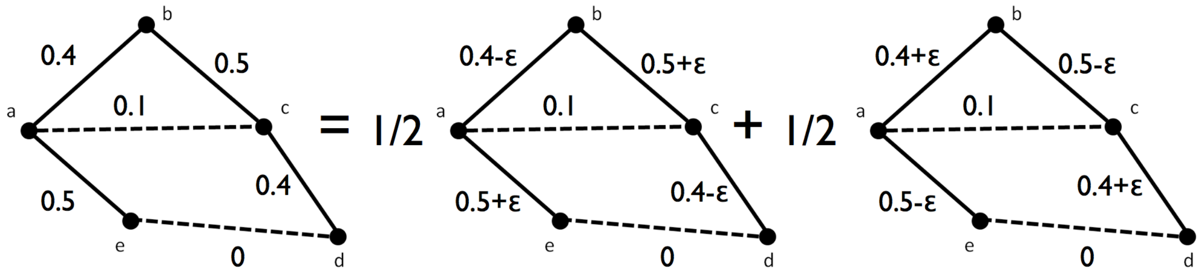

Figure 3: Example of x (left), y (middle) and z (right) for Case (a), where $\mathcal{P} = e \to a \to b \to c \to d$ and $\{a, b, c\}$ forms a cycle of Type T1. Once $\mathcal{P}$ enters the cycle, it goes out after traversing even number of edges, (a, b) and (b, c).

Case (b). Similarly as we did for Case (a), we can modify/make cycle $\mathcal{T}$ so that it always traverse even number of edges of cycles of Type T1 and T2 once it enters.[3] If the length of $\mathcal{T}$ is even, one can construct two different feasible points $y = [y_e], z = [z_e]$ with $x = (y+z)/2$ by starting from $x = [x_e]$ and alternatively adding/substracting some small constant $\varepsilon > 0$ following path $\mathcal{T}$, similarly as we did for Case (a). Hence, we assume that $\mathcal{T}$ is of odd length.

Now consider the cases:

- There exists a vertex $v \in \mathcal{T}$ such that $v \notin V_x$.
- Else, $V(\mathcal{T}) \subset V_x$ (every vertex $v \in \mathcal{T}$ is in V_x) and $V(\mathcal{T})$ forms a disjoint component in E_x.

[3]We note again that the modified cycle $\mathcal{T}$ may contain a same edge more than once.

- Else, $V(\mathcal{T}) \subset V_x$ and there exists $e \in E_x$ connecting between $V(\mathcal{T})$ and $V \setminus V(\mathcal{T})$.

For the first case, one can construct again two different feasible points $y = [y_e], z = [z_e]$ with $x = (y+z)/2$ by alternatively adding/substracting some small constant $\varepsilon > 0$ to x following path $\mathcal{T}$ with the staring point v. Hence, x is not a vertex in the first case. For the second case, one can show that every x_e for $e \in \mathcal{T}$ is half integral, which contradicts to our assumption that E_x does not contain a cycle consisting of half integral edges $\{e : x_e = 1/2\}$. In the third case, one can construct a new path $\mathcal{P}'$ starting from e until it finds a cycle $\mathcal{T}'$ in E_x as we did previously. But, now we expand a path in one direction, while we did in both directions previously. Of course, the expansion procedure may terminate before finding a cycle in E_x, which belongs to Case (a) and hence we are done, i.e., x is not a vertex. By using the previous arguments for $\mathcal{T}'$ in place of $\mathcal{T}$, we can also assume that

- $\mathcal{T}'$ always traverses even number of edges of cycles of Type T1 and T2 once it enters.
- The length of $\mathcal{T}'$ is odd.

Now the union of $\mathcal{P}'$ and $\mathcal{T}$ form two cycles $\mathcal{T}, \mathcal{T}'$ connected by (possibly, empty) bridge edges $\mathcal{B}$, where we can assume that $\mathcal{B}$ also traverses even number of edges of cycles of Type T1 and T2 once it enters. Finally, using these properties, one can construct two different feasible points $y = [y_e], z = [z_e]$ with $x = (y+z)/2$ by starting from $x = [x_e]$ and

- Alternatively adding/substracting ε following cycles $\mathcal{T}$ and $\mathcal{T}'$;
- Alternatively adding/substracting 2ε following the bridge edges $\mathcal{B}$,

where $\varepsilon > 0$ is small enough. We provide an example of $\{x, y, z\}$ in Figure 4. This implies that

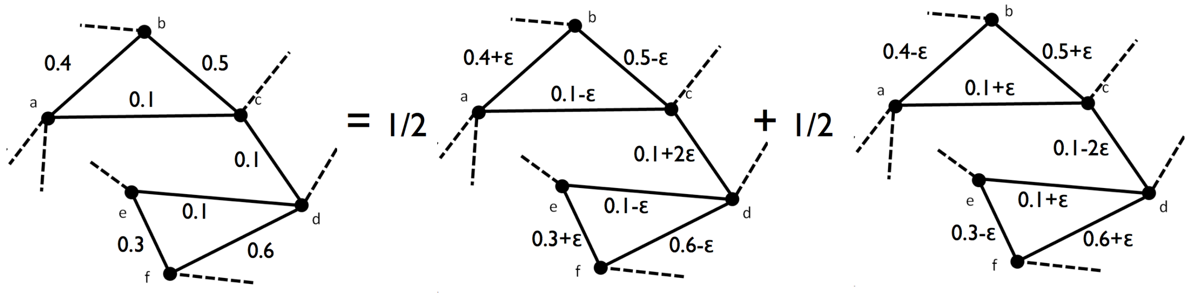

Figure 4: Example of x (left), y (middle) and z (right) for Case (b), where $\mathcal{T} = a \to b \to c \to a$, $\mathcal{B} = c \to d$ and $\mathcal{T}' = e \to d \to f \to e$.

x is not a vertex in the third case, and complete the proof of the half integrality of C-LP.