Belief Propagation for the Maximum-Weight Independent Set and Minimum Spanning Tree Problems[*]

Kamiel Cornelissen[1], Bodo Manthey[1]

[1]University of Twente, Department of Applied Mathematics
k.cornelissen@utwente.nl, b.manthey@utwente.nl

The belief propagation (BP) algorithm is a message-passing algorithm that is used for solving probabilistic inference problems. In practice, the BP algorithm performs well as a heuristic in many application fields. However, the theoretical understanding of BP is limited. To improve the theoretical understanding of BP, the BP algorithm has been applied to many well-understood combinatorial optimization problems. In this paper, we consider BP applied to the maximum-weight independent set (MWIS) and minimum spanning tree (MST) problems.

Sanghavi et al. (*IEEE Trans. Inform. Theory*, 2009) applied the BP algorithm to the MWIS problem. We denote their algorithm by BP-MWIS. They showed that if the LP relaxation of the MWIS problem has a unique integral optimal solution and BP-MWIS converges, then BP-MWIS finds the optimal solution. Also, they showed that if the LP relaxation has a non-integral optimal solution, then BP-MWIS does not converge. In this paper, we precisely characterize the graphs for which BP-MWIS is guaranteed to find the optimal solution, regardless of the node weights.

Bayati et al. (*J. Math. Phys.*, 2008) applied the BP algorithm to the MST problem. We denote their algorithm by BP-MST. They showed that if BP-MST converges, then it finds the optimal solution. In this paper, however, we provide an instance for which BP-MST does not converge. Also, since this instance is small and simple, we believe that BP-MST does not converge for most instances encountered in practice.

1 Introduction

The belief propagation (BP) algorithm is a message-passing algorithm that is used for solving probabilistic inference problems on graphical models. It was proposed by Pearl in 1988 [8]. Typical graphical models to which BP is applied are Bayesian networks, Markov random fields, and factor graphs. In this paper, we consider the max-product variant of BP (or the

[*]This research was supported by NWO grant 613.001.023.

functionally equivalent min-sum variant), which is used to compute maximum *a posteriori* probability (MAP) estimates.

Recently, BP has experienced great popularity. It has been applied in many fields, such as machine learning, image processing, computer vision, and statistics. For an introduction to BP and several applications, we refer to Yedidia et al. [17]. There are two main reasons for the popularity of BP. First, it is widely applicable and easy to implement because of its simple and iterative message-passing nature. Second, it performs well in practice in numerous applications [14,16].

If the graphical model is tree-structured, BP computes exact MAP estimates. However, if the graphical model contains cycles, the convergence and correctness of BP have been shown only for specific classes of graphical models. To improve the general understanding of BP and to gain new insights about the algorithm, it has recently been tried to rigorously analyze the performance of BP as either a heuristic or an exact algorithm for several combinatorial optimization problems. Amongst others, it has been applied to the maximum-weight matching (MWM) problem [1,3,4,9,10], the minimum spanning tree (MST) problem [2], the minimum-cost flow (MCF) problem [4,7], the maximum-weight independent set (MWIS) problem [11,12], and the 3-coloring problem [5]. BP has even been used to analyze the satisfiability threshold [6]. The reason to consider BP applied to these combinatorial optimization problems is that these optimization problems are well understood. This facilitates a rigorous analysis of BP, which is often difficult for other applications.

In this paper, we consider BP applied to the MWIS and the MST problem. Sanghavi et al. [12] introduced a variant of BP for the MWIS problem, which we denote by BP-MWIS. They showed that BP-MWIS does not converge if the LP relaxation of the problem has a non-integral optimal solution. Also, they showed that even if the LP relaxation of the problem has a unique integral optimal solution, BP-MWIS is not guaranteed to converge. In this paper we characterize precisely the graph structures for which BP-MWIS is guaranteed to work well. This means that we characterize the graph structures for which BP-MWIS is guaranteed to converge to the correct solution irrespective of the node weights (as long as the MWIS is unique). We show (Section 3) that the graphs for which BP-MWIS converges to the correct solution for all possible node weights are exactly those graphs that contain at most one even cycle and no odd cycles.

Bayati et al. [2] introduced a variant of BP for the MST problem, which we denote by BP-MST. The MST problem is easily solvable using a variety of algorithms. Still, it is interesting to analyze the performance of BP applied to the MST problem since the MST problem has a global connectivity constraint. This is in contrast to, for example, the MWM, MCF, and MWIS problems, which only have local constraints. Bayati et al. showed the following positive result for BP-MST: if BP-MST converges, then it converges to the correct solution. In this paper, we show a negative result for BP-MST. In Section 4, we show a small instance for which BP-MST does not converge. In addition, the property of this instance that ensures that BP-MST does not converge is quite general and carries over to many other instances. Therefore, we believe that BP-MST does not converge for most instances in practice.

The rest of this paper is organized as follows. First we introduce the MWIS (Section 1.1) and MST (Section 1.2) problems. In Section 2, we introduce the BP algorithm and the variants for the MWIS problem by Sanghavi et al. [12] and the MST problem by Bayati et al. [2]. In Section 3 we state our results for BP-MWIS. Finally, in Section 4 we state our results for BP-MST.

To conclude this section, we introduce some notation and assumptions. We denote the weight of a node v by $w(v)$. Also, we denote the weight of a set of nodes V by $w(V)$. That is,

$$w(V) = \sum_{v \in V} w(v).$$

For a graph $G = (V, E)$ we define the *neighborhood* $N(v)$ of a node v as

$$N(v) = \{u \mid (u, v) \in E\}.$$

In this paper, we assume that all graphs are connected. For the MST problem we do this since no spanning tree exists for a disconnected graph. For the MWIS problem we do this since maximum-weight independent sets on disconnected graphs can be computed by separately computing maximum-weight independent sets on the individual components and then taking the union of those sets. Finally, as is commonly done, we assume that the optimal solutions for the MST and MWIS problems are unique, since it is well-known that BP does not converge for instances that have multiple optimal solutions for these problems [1–3, 10].

1.1 Maximum-Weight Independent Set Problem

Let $G = (V, E)$ be an undirected weighted graph. An independent set S is a subset $S \subset V$ of nodes such that for every edge $(u, v) \in E$ at most one of u and v is in S. The MWIS problem consists of finding an independent set of maximum weight. A subset of nodes $S^* \subset V$ is an *MWIS* of G if and only if

$$S^* \in \text{argmax}\{w(S) \mid S \text{ is an independent set of } G\}.$$

It is straightforward to formulate the MWIS problem as an integer program by identifying with each node $u \in V$ a binary variable $x_u \in \{0, 1\}$. Here $x_u = 0$ can be interpreted as x not being part of the independent set S, while $x_u = 1$ can be interpreted as x being part of S. The integer program contains constraints that prevent two neighboring nodes from both being included in S. The integer program (IP-MWIS) is as follows

$$\begin{aligned} \max & \sum_{u \in V} w(u) x_u \\ \text{s.t.} \quad & x_u + x_v \leq 1 \qquad \text{for all } (u, v) \in E, \\ & x_u \in \{0, 1\}. \end{aligned}$$

We obtain the LP relaxation of IP-MWIS by relaxing the constraint that x_u should take an integer value. We denote this LP relaxation by LP-MWIS.

$$\begin{aligned} \max & \sum_{u \in V} w(u) x_u \\ \text{s.t.} \quad & x_u + x_v \leq 1 \qquad \text{for all } (u, v) \in E, \\ & 0 \leq x_u \leq 1. \end{aligned}$$

The independent set polytope is given by all feasible solutions of LP-MWIS. Every extreme point of the independent set polytope has $x_u \in \{0, \frac{1}{2}, 1\}$ for all $u \in V$ [13].

1.2 Minimum Spanning Tree Problem

Let $G = (V, E)$ be an undirected graph. A spanning tree T of G is a connected subgraph $T = (V, F)$ of G, such that each node in V is incident to at least one of the edges in F and T is a tree. That is, T does not contain any cycles. The MST problem consists of finding a spanning tree of G of minimum total weight. A tree T^* is a *MST* of G if and only if

$$T^* \in \operatorname{argmin}\{w(T) \mid T \text{ is a spanning tree of } G\}.$$

2 Belief Propagation

In this section we give a brief introduction to the belief propagation (BP) algorithm. We only introduce the aspects that are relevant for our analysis in Sections 3 and 4. For a more elaborate introduction, we refer to Yedidia et al. [17]. Suppose we are given a graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$ and for each $u \in V$ an associated random variable X_u that takes values in a finite set $\mathcal{X}_u$. We define $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$. Consider the probability distribution

$$\hat{P}(\mathbf{x}) = \frac{1}{Z} \prod_{u \in V} \phi_u(x_u) \prod_{(u,v) \in E} \phi_{uv}(x_u, x_v), \quad \mathbf{x} = (x_v)_{v \in V} \in \mathcal{X}. \tag{1}$$

In the above, the ϕ_u and ϕ_{uv} are non-negative functions and Z is a normalization constant. The graph G and the probability distribution $\hat{P}(\mathbf{x})$ together form a graphical model, in particular a pairwise Markov random field (MRF). Since both the MWIS problem and the MST problem can be modeled as pairwise MRFs, we restrict ourselves to pairwise MRFs in this paper.

A *maximum a posteriori probability* (MAP) estimate of a probability distibution $P(X)$ is a most likely realization of the random variables. That is, the MAP estimate $\hat{\mathbf{x}}$ of $P(X)$ is defined as

$$\hat{\mathbf{x}} \in \operatorname{argmax} P(\mathbf{x}).$$

In the following we assume that the MAP estimate is unique. We call the value $\hat{x}_u$ that x_u takes in the MAP estimate the *MAP assignment* of u.

Computing the MAP estimate for general probability distributions is NP-hard. The BP algorithm is a heuristic for computing the MAP estimate. For the probability distribution $\hat{P}(\mathbf{x})$ (see Equation (1)), BP computes the MAP estimate exactly when the graph G is a tree. If G contains cycles, BP is not guaranteed to compute the correct MAP estimate, but the BP algorithm is still well-defined and in practice often gives a good approximation of the MAP estimate.

In short, the BP algorithm works as follows. In each iteration k, each node u sends a message vector

$$M^k_{u \to v} = \left(m^k_{u \to v}(x_v)\right)_{x_v \in X_v}$$

to each of its neighbors $v \in N(u)$ containing a message for each possible value for X_v. A message $m^k_{u \to v}(x_v)$ can be interpreted as how "likely" the sending node u thinks it is that the random variable X_v associated with the receiving node v should take value x_v in the MAP estimate. The greater the value of the message $m^k_{u \to v}(x_v)$, the more likely it is according to node u in iteration k that X_v should take value x_v in the MAP estimate. The messages are initialized neutrally, that is, in iteration 0 the messages are

$$M^0_{u \to v} = (1, 1, \ldots, 1), \text{ for all } u, v \in N(u).$$

In iterations $k \geq 1$ the messages are computed from the messages in the previous iteration as follows:

$$m^k_{u \to v}(x_v) = \max_{x_u \in X_u} \left\{ \phi_u(x_u) \cdot \phi_{uv}(x_u, x_v) \cdot \prod_{w \in N(u) \setminus \{v\}} m^{k-1}_{w \to u}(x_u) \right\}.$$

All these messages are sent simultaneously.

The belief b^k_u of node u in iteration k is defined as

$$b^k_u(x_u) = \phi_u(x_u) \cdot \prod_{v \in N(u)} m^{k-1}_{v \to u}(x_u).$$

These beliefs can be interpreted as the "likelihood" that X_u should take value x_u in the MAP estimate. The greater the value of $b^k_u(x_u)$, the more likely that X_u should take value x_u in the MAP estimate. We denote the best estimate (breaking ties arbitrarily) for the value of X_u in the MAP estimate during iteration k by x^k_u, that is,

$$x^k_u = \text{argmax}\{b^k_u(x_u) \mid x_u \in X_u\}.$$

The vector $(x^k_u)_{u \in V}$ gives an estimate of the MAP estimate during iteration k. If, for some K, we have

$$(x^{k_1}_u)_{u \in V} = (x^{k_2}_u)_{u \in V}, \quad \text{for all } k_1, k_2 \geq K,$$

then BP has converged after K iterations. In general there are three possibilities: BP converges to the MAP estimate, BP converges to an incorrect solution, or BP does not converge at all.

2.1 Computation Tree

To show our results, we need the notion of a *computation tree*. Computation trees have been used frequently to analyze the BP algorithm, for example, in the context of the Maximum-Weight Independent Set problem [12] and the Maximum Weight Matching problem [1].

Let $G = (V, E)$ be an arbitrary undirected graph. We denote the *level-k computation tree* with the root labeled $u \in V$ by $T^k(u)$. In the following we call the root of a computation tree the CT-root, to distinguish it from the root of a directed spanning tree, which we introduce later. The tree $T^k(u)$ is a labeled rooted tree of height $k + 1$. Like Bayati et al. [2] we denote by $[x, u]$ a node x in the computation tree with label u. In the rest of the paper we will also use the term u-labeled to denote that a node in the computation tree is labeled with node $u \in V$ and the term S-labeled to denote that a node in the computation tree is labeled with a node of the subset $S \subset V$.

The CT-root in $T^0(u)$ has label u, its degree is the degree of u in G, and its children are labeled with the adjacent nodes of u in G. The tree $T^{k+1}(u)$ is obtained recursively from $T^k(u)$ by attaching nodes to every leaf node in $T^k(u)$. To each leaf node $[y, v]$ in $T^k(u)$, a number of nodes equal to the degree of v in G minus 1 is attached. These nodes are labeled with the neighbors of v in G except for the label of the parent of y in $T^k(u)$. If the nodes or edges of G are weighted, these weights are copied to the computation tree. This means that a node with label u in the computation tree has weight $w(u)$ and an edge between two nodes labeled u and v in the computation tree has weight $w(u, v)$. Figure 1 shows an example of an edge-weighted graph and computation tree.

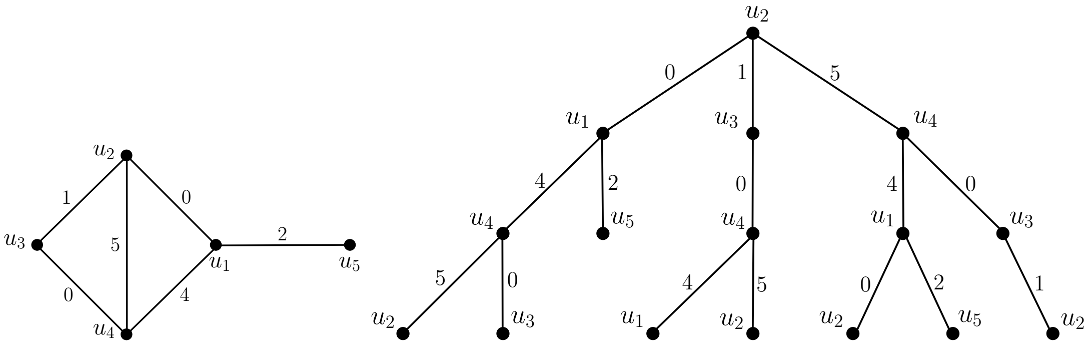

Figure 1: On the left an example edge-weighted graph and on the right the associated level-2 computation tree $T^2(u_2)$ rooted at u_2 with the node labels next to the nodes.

The definition of the computation tree is such that each non-leaf node $[x, u]$ in the computation tree has neighbors with the same labels as the neighbors of u in G. Also, the messages that the CT-root of a level-k computation tree with label u receives after k iterations of the BP algorithm on the computation tree are exactly the same as the messages that u receives after k iterations of the BP algorithm on G. The behavior of the BP algorithm on trees is well understood, in contrast to the behavior of the BP algorithm on graphs with cycles. Therefore, computation trees form a useful tool for analysis of the BP algorithm on graphs with cycles.

On a computation tree $T = (V_T, E_T)$ we can naturally define a probability distribution P_T using the node labels and the functions ϕ_u and ϕ_{uv} as defined for G (see Equation (1)):

$$P_T(\mathbf{x}) = \frac{1}{Z} \prod_{[y,u]\in V_T} \phi_u(x_y) \prod_{([y,u],[z,v])\in E_T} \phi_{uv}(x_y, x_z), \quad \mathbf{x} \in \mathcal{X}_T. \tag{2}$$

In the above, analogously to Equation (1), we have $V_T = \{1_T, 2_T, \ldots, n_T\}$, we associate a random variable X_y with each $[y, u] \in V_T$ which takes values in $\mathcal{X}_y = \mathcal{X}_u$, and we define $\mathcal{X}_T = \mathcal{X}_{1_T} \times \mathcal{X}_{2_T} \times \ldots \times \mathcal{X}_{n_T}$.

If BP converges, then the MAP assignment (given by the MAP estimate of P_T) of all nodes in the computation tree that are sufficiently far away from the leafs of the tree is according to the assignment that the BP algorithm converged to. This follows, for instance, from the periodic assignment lemma by Weiss [15]. Nodes that are close to the leafs do not necessarily take the assignment that BP converged to. (In the above we mean by 'leafs' only those leafs of the computation tree that are in the lowest level of the computation tree, not the nodes in the higher levels of the computation tree that are leafs only because the nodes that they are labeled with have degree 1 in the original graph G. For example, the u_5-labeled node at distance 2 from the CT-root in the computation tree in Figure 1 is not considered a leaf, while the u_3-labeled node at distance 3 from the CT-root is considered a leaf.)

Theorem 2.1 (Weiss [15]). *Assume that the BP algorithm converges after K iterations. Each node $[x, v]$ in the computation tree $T^k(u)$ $(k \geq K)$ that is at distance at most $k - K$ from the CT-root of $T^k(u)$ has MAP assignment equal to the assignment that v converged to.*

2.2 Belief Propagation for the Maximum-Weight Independent Set Problem

Sanghavi et al. [12] developed a variant of the BP algorithm for the MWIS problem, which we denote by BP-MWIS. For a graph $G = (V, E)$ they associate with each node $u \in V$ a random variable X_u which takes values from the set $\{0, 1\}$. A value of '0' for X_u can be interpreted as u not being part of independent set S, while a value of '1' can be interpreted as u being part of S. They define $\phi_u(x_u) = e^{w(u)x_u}$, $\phi_{uv}(x_u, x_v) = 0$ if $x_u + x_v > 1$, and $\phi_{uv}(x_u, x_v) = 1$ otherwise. Let the probability distribution P_{IS} be given by

$$P_{\mathrm{IS}}(\mathbf{x}) = \frac{1}{Z} \prod_{u \in V} \phi_u(x_u) \prod_{(u,v) \in E} \phi_{uv}(x_u, x_v), \qquad \mathbf{x} \in \{0, 1\}^{|V|}.$$

For distribution P_{IS}, only $\mathbf{x}$ corresponding to independent sets of G have positive probability. Since the MAP estimate of P_{IS} corresponds to the MWIS of G, BP can be used as a heuristic for computing the MWIS of G. BP-MWIS is the BP algorithm by Sanghavi et al. for the graphical model given by graph G and probability distribution P_{IS}. In each iteration of BP-MWIS each node sends two messages $m_{u \to v}(0)$ and $m_{u \to v}(1)$ to each of the nodes v in its neighborhood $N(u)$. Since the exact structure of the messages does not play a role in our analysis, we will not further specify them and refer to the original paper. At the end of each iteration each node estimates whether it should be in the MWIS. We denote the estimate of node u in iteration k by $x_u^k \in \{0, 1, ?\}$. An estimate of '0' can be interpreted as u believing that it should not be in the MWIS, an estimate of '1' can be interpreted as u believing that it should be in the MWIS, and an estimate of '?' as u considering it equally likely that it is part of the MWIS or not. Sanghavi et al. showed that if BP-MWIS converges (that is, there exists a number of iterations K such that the estimate x_u^k is equal to x_u^K for all $u \in V$ and $k \geq K$), the estimates correspond to the optimum of LP-MWIS (see Section 1.1). That is, if for all $u \in V$ we set $x_u = 0$ if $x_u^K =$ '0', $x_u = 1$ if $x_u^K =$ '1', and $x_u = 1/2$ if $x_u^K =$ '?', the vector x is an optimum of LP-MWIS.

In our analysis we use several results by Sanghavi et al. [12] which we list below.

Theorem 2.2 (Sanghavi et al. [12]). *If LP-MWIS has a non-integral optimal solution, then BP-MWIS does not converge.*

Just like for the original graph G, we can consider maximum-weight independent sets of a computation tree $T^k(u)$. The estimates x_u^k of BP-MWIS can be directly related to whether or not the CT-root of the computation tree $T^k(u)$ is part of a MWIS of $T^k(u)$.

Theorem 2.3 (Sanghavi et al. [12]). *For any node $u \in V$ and any number of iterations k we have:*

- $x_u^k =$ '1' *if and only if the CT-root of $T^k(u)$ is a member of every MWIS of $T^k(u)$;*
- $x_u^k =$ '0' *if and only if the CT-root of $T^k(u)$ is not a member of any MWIS of $T^k(u)$;*
- $x_u^k =$ '?' *otherwise.*

2.3 Belief Propagation for the Minimum Spanning Tree Problem

Bayati et al. [2] developed a variant of the BP algorithm for the MST problem, which we denote by BP-MST. For convenience, we give a short description of their algorithm below. Also, we state their results that we use in Section 4 to show that BP-MST does not converge for all instances of the MST problem. For a more elaborate description of the algorithm we refer to the original paper.

A spanning tree of an undirected graph $G = (V, E)$ is modeled as a rooted directed tree. One of the nodes in V is designated as the root of the tree. To distinguish the root of a rooted spanning tree from the root of a computation tree, we call the former the MST-root. Each node $u \in V$ has an associated parent node $p_u \in N(u)$ and an associated depth $d_u \in \{0, 1, \ldots, n-1\}$. (Though Bayati et al. [2] did not specify the maximum value $d_{\max}$ for the depth of a node, we make the natural choice of $d_{\max} = n - 1$. Using smaller values of $d_{\max}$, we can model the NP-hard problem of finding minimum spanning trees of bounded depth.) The MST-root has (by definition) itself as its parent and depth 0. For each other node u it has to hold that $(u, p_u) \in E$ and that $d_{p_u} = d_u - 1$. Note that every spanning tree of G can be modeled in this way and that each set $\{(p_u, d_u)_{u \in V}\}$ that satisfies the above conditions provides a spanning tree of G. For an example of an undirected spanning tree modeled as a directed spanning tree, we refer to the right image of Figure 2.

In each iteration of BP-MST each node u sends a message $m_{u \to v}(p_v, d_v)$ to each of the nodes v in its neighborhood $N(u)$ for all the possible combinations of values for p_v and d_v. Such a message $m_{u \to v}(p_v, d_v)$ can be interpreted as the likelihood according to the sending node u that the receiving node v should have parent p_v and depth d_v in the MST of G. Since the exact structure of the messages does not play a role in our analysis, we will not further specify them and refer to the original paper. At the end of each iteration, each node u uses the incoming messages to estimate its parent p_u and depth d_u in the MST. Bayati et al. showed that if BP-MST converges, it finds the MST.

Theorem 2.4 (Bayati et al. [2]). *If BP-MST converges to $(p_u, d_u)_{u \in V}$, then the set of edges $\{(u, p_u)_{u \in V \setminus \{\text{MST-root}\}}\}$ is the minimum spanning tree of G.*

For another result by Bayati et al. that we use in our analysis, we need the notion of an *Oriented Spanning Tree* (OST) on the computation tree $T^k(u)$ (see Section 2.1) for BP-MST. We assign to each node $[x, v]$ in $T^k(u)$ a depth $d_x \in \{0, 1, \ldots, n-1\}$. To each non-leaf node $[y, v]$ in $T^k(u)$ we assign a parent p_y in its neighborhood $N([y, v])$ (or $[y, v]$ itself in case v is the MST-root of G). Here 'leafs' are again only those leafs in the lowest level of the computation tree, see also Section 2.1. We call such an assignment valid if it satisfies two properties:

- Every non-leaf node $[y, v]$ of $T^k(u)$ for which v is the MST-root of G has itself as its parent and depth $d_y = 0$.
- For every non-leaf node $[y, v]$ of $T^k(u)$ for which v is not the MST-root of G, it has to hold that $d_{p_y} = d_y - 1$.

Every such valid assignment gives an OST

$$\big\{([y, v], p_y) \mid [y, v] \text{ is not a leaf in the lowest level of } T^k(u) \text{ and } v \text{ is not the MST-root of } G\big\}.$$

Among all OSTs on the computation tree, we call the one of minimum weight the *Minimum-Weight Oriented Spanning Tree* (MWOST). Bayati et al. showed that BP-MST solves the MWOST problem on the computation tree.

Theorem 2.5 (Bayati et al. [2]). *BP-MST solves the MWOST problem on the computation tree. That is, the MAP assignment of all nodes in the computation tree is such that it corresponds to the MWOST on the computation tree. In particular, for all $u \in V$, the estimates p_u^k and d_u^k at the end of iteration k are equal to the values of $p_{\text{CT-root}}$ and $d_{\text{CT-root}}$ in the MWOST of $T^k(u)$.*

Theorem 2.4 and Theorem 2.5 show that, though BP-MST actually computes the MWOST of the computation tree, if it converges, it finds the MST of G. However, convergence of BP-MST is not guaranteed. In Section 4 we show a small example graph G for which BP-MST does not converge and explain why we believe that BP-MST does not converge for most graphs in practice.

3 Convergence of BP-MWIS

In this section we characterize the graphs G for which BP-MWIS converges for all possible node weights (assuming that the MWIS is unique). We show that BP-MWIS is only guaranteed to work well for bipartite graphs that are trees plus at most one additional edge. In Section 3.1 we show that BP-MWIS converges for all possible node weights for all G that contain no odd length cycles and at most one even length cycle. In Section 3.2 we show that if G contains an odd length cycle or at least two even length cycles, there exist node weights for which BP-MWIS does not converge.

3.1 Graphs for Which BP-MWIS Converges

In this section we show that BP-MWIS converges to the correct solution for all possible node weights for graphs that contain no odd cycles and at most one even cycle. The idea is that computing an MWIS in such graphs boils down to computing an MWIS in a cycle of even length, where the weight of a node v is the weight of an MWIS of the tree, whose root v is, including v minus the weight of an MWIS of the tree excluding v.

Theorem 3.1. *Let $G = (V, E)$ be a graph that contains no odd cycle and at most one even cycle. Then BP-MWIS converges to the correct solution for all possible node weights w for which the MWIS of G is unique.*

Proof. If G is a tree, then after at most n iterations, the computation tree T is equal to G. Since the MWIS of G is unique, the MWIS of T is unique as well and according to Theorem 2.3 BP-MWIS converges to the correct solution.

Next we consider the case that G contains exactly one even cycle $C = (W, F)$ and no odd cycles. Let $q = |W|$. We denote the nodes in C by $v_0, v_1, v_2, \ldots, v_q = v_0$ such that $(v_i, v_{i+1}) \in F$. Furthermore, we define sets $V_1, V_2, \ldots, V_q$ where V_i consists of node v_i plus all nodes u that are not on the cycle C and for which the shortest path from u to one of the cycle nodes ends in v_i. We also define weights

$$\begin{aligned} w_i^+ &= \max\{w(B) \mid v_i \in B, B \subset V_i, B \text{ is an independent set on } G\} \quad \text{and} \\ w_i^- &= \max\{w(B) \mid v_i \notin B, B \subset V_i, B \text{ is an independent set on } G\}. \end{aligned}$$

We denote by $V_i^+ \subseteq V_i$ and by $V_i^- \subseteq V_i$ the subsets for which the weights w_i^+ and w_i^- are obtained, respectively, breaking ties arbitrarily. Using the above definitions, the problem of finding the MWIS of G can be reduced to finding the independent set $D \subset W$ for which

$$\sum_{i:v_i \in D} w_i^+ + \sum_{i:v_i \notin D} w_i^- \quad \text{is maximized.}$$

We denote the MWIS of G by I. Also, we denote by I' an arbitrary second-heaviest independent set, that is

$$I' \in \text{argmax}\{w(S) \mid S \subset V, S \neq I, S \text{ is an independent set of } G\}.$$

Since the MWIS of G is unique, there is a strictly positive difference between the weight of I and the weight of I'. We define $\delta = w(I) - w(I') > 0$. We denote the weight of the heaviest node of G by w^*.

Let $T = (V_T, E_T)$ be a computation tree for G and let $R \subseteq V_T$ be a subset of W-labeled nodes of the computation tree. In the following we denote by $M[R]$ the subgraph of T that is induced by R plus all nodes u in T that are not W-labeled and for which a path from u to some $v \in R$ exists for which all nodes except for v are not W-labeled.

Note that from the above definitions we immediately obtain

$$w_i^+ \leq w_i^- + w^*, \tag{3}$$

since $B = V_i^+ \setminus \{v_i\}$ is an independent set on G, $B \subset V_i$, and $v_i \notin B$. Also, we have

$$w_i^+ \geq w_i^- \text{ if } v_i \in I, \tag{4}$$

because otherwise, we can improve I by removing the nodes in V_i^+ and then adding the nodes in V_i^-.

We first show that BP-MWIS converges to the correct solution for nodes $v \in W, v \in I$. Assume to the contrary that BP-MWIS does not converge to the correct solution for v. We define $K^* = \frac{n^2 w^*}{\delta}$. Then, according to Theorem 2.3 there exists a $k > K^* + 3n$ such that the CT-root of the computation tree $T = T^k(v)$ is not a part of every MWIS of T. Let J be an MWIS of T that does not include the CT-root. We now define sets S^+ and S^- on T recursively. We start by adding the CT-root to S^+. Each time we add a node to S^+, we add to S^- each of its neighbors in the computation tree that is W-labeled, is in J, and is at distance at most $K^* + 2n + 1$ from the CT-root. Each time we add a node to S^-, we add to S^+ each of its neighbors in the computation tree that is W-labeled, is I-labeled, and is at distance at most $K^* + 2n$ from the CT-root.

Note that the nodes in $S^+ \cup S^-$ induce a path P that starts at a v_i-labeled node, continues to a v_{i+1}-labeled node, etc., and ends in a v_j-labeled node. We can partition this path in shorter paths, such that p parts $P_1, \ldots, P_p$ are equal to $(v_i, v_{i+1}, \ldots, v_{i-1})$, that is, every P_ℓ is equal to cycle C with edge (v_{i-1}, v_i) removed. In addition, the partition consists at most one part P^* of length less than $|W|$ which is equal to $(v_i, v_{i+1}, \ldots, v_j)$.

Next we show that we can construct an independent set $\tilde{J}$ on T of weight greater than $w(J)$ as follows. We set $\tilde{J} = J$. For each node $[u, v_i]$ in S^+ we first remove from $\tilde{J}$ all nodes in $M[\{u\}]$, then add to $\tilde{J}$ all V_i^+-labeled nodes in $M[\{u\}]$. In addition, for each node $[u, v_i]$ in S^- we first remove from $\tilde{J}$ all nodes in $M[\{u\}]$, then add to $\tilde{J}$ all V_i^--labeled nodes in $M[\{u\}]$. Note that $\tilde{J}$ is again an independent set on T, since the W-labeled neighbors of each node $[u, v_i] \in S^+$ are either in S^- and therefore not in $\tilde{J}$, or they were not in J (otherwise they would have been added to S^-) and therefore not in $\tilde{J}$ either.

Now we consider one path P_ℓ and the graph $M_\ell = (V_{M_\ell}, E_{M_\ell}) = M[P_\ell]$. Note that M_ℓ is a copy of G, except for the missing edge (v_{i-1}, v_i). The set of labels of the nodes in $V_{M_\ell} \cap \tilde{J}$ is exactly equal to I. Also, the set of labels of the nodes in $V_{M_\ell} \cap J$ is equal to some other independent set $\tilde{I}$ of G. Since I is at least δ heavier then any other independent set of G, we have that

$$w(V_{M_\ell} \cap \tilde{J}) \geq w(V_{M_\ell} \cap J) + \delta. \qquad (5)$$

In the following we denote by $M^* = (V_{M^*}, E_{M^*}) = M[P^*]$. We now distinguish two cases.

Case 1: $|P| > K^* + n$. Since $|P| > K^* + n$, we have $p \geq K^*/n + 1$. By Equation (5), we have $w(V_{M_\ell} \cap \tilde{J}) \geq w(V_{M_\ell} \cap J) + \delta$ for all ℓ. By Equations (3) and (4) we have $w(V_{M^*} \cap J) \leq w(V_{M^*} \cap \tilde{J}) + (n-1)w^*$. Combining these two inequalities yields

$$w(\tilde{J}) - w(J) \geq p\delta - (n-1)w^* > 0.$$

Since $\tilde{J}$ is heavier than J, our assumption that BP-MWIS does not converge to the correct solution for node v, graph G, and weights w was incorrect.

Case 2: $|P| \leq K^* + n$. Let $[x, v_i]$ and $[y, v_j]$ be the endpoints of P. Suppose $x \in S^+$. Since $x \in S^+$, we have $v_i \in I$ by definition and therefore $v_{i-1} \notin I$. Suppose now $x \in S^-$. Since the v_{i-1}-labeled neighbor u of x was not added to S^+, u cannot be I-labeled by definition, so $v_{i-1} \notin I$. Similarly, the v_{j+1}-labeled neighbor of y that is not in P cannot be I-labeled, so $v_{j+1} \notin I$. Consider now P^*. Suppose that $J \cap V_{M^*}$ is at least as heavy as $\tilde{J} \cap V_{M^*}$. Since neither v_{i-1} nor v_{j+1} is in I, we can define a new independent set $\hat{I}$ of G of weight at least $w(I)$. We set $\hat{I} = I$. Next, we remove from $\hat{I}$ all nodes in the sets V_i for which v_i is used to label one of the nodes in P^*. By doing so, $w(\hat{I})$ decreases by $w(\tilde{J} \cap V_{M^*})$. Then we add to $\hat{I}$ all nodes that are used to label one of the nodes in $J \cap V_{M^*}$. By doing so, $w(\hat{I})$ increases by $w(J \cap V_{M^*})$. Since the nodes in $J \cap V_{M^*}$ are at least as heavy as the nodes in $\tilde{J} \cap V_{M^*}$, we have that $\hat{I}$ is at least as heavy as I. This contradicts the fact that I is the unique MWIS of G. Therefore, our assumption that $J \cap V_{M^*}$ is at least as heavy as $\tilde{J} \cap V_{M^*}$ was wrong. Since also $\tilde{J} \cup V_{M_\ell}$ is heavier than $J \cup V_{M_\ell}$ for all ℓ, we have that $\tilde{J}$ is heavier than J.

We set $\hat{I} = I$. Next, we remove from $\hat{I}$ all nodes in the sets V_i for which v_i is used to label one of the nodes in P^*. By doing so, $w(\hat{I})$ decreases by $w(\tilde{J} \cap V_{M^*})$. Then we add to $\hat{I}$ all nodes that are used to label one of the nodes in $J \cap V_{M^*}$. By doing so, $w(\hat{I})$ increases by $w(J \cap V_{M^*})$. Since the nodes in $J \cap V_{M^*}$ are at least as heavy as the nodes in $\tilde{J} \cap V_{M^*}$, we have that $\hat{I}$ is at least as heavy as I. This contradicts the fact that I is the unique MWIS of G. Therefore, our assumption that $J \cap V_{M^*}$ is at least as heavy as $\tilde{J} \cap V_{M^*}$ was wrong. Since also $\tilde{J} \cup V_{M_\ell}$ is heavier than $J \cup V_{M_\ell}$ for all ℓ, we have that $\tilde{J}$ is heavier than J.

Since $\tilde{J}$ is heavier than J, our assumption that BP-MWIS does not converge to the correct solution for node v, graph G, and weights w was incorrect.

We showed convergence of BP-MWIS for nodes $v \in W, v \in I$. Next we consider nodes $v \in W, v \notin I$. The proof that BP-MWIS converges to the correct solution for these nodes is very similar to the proof for nodes that are in I. Assume that BP-MWIS does not converge to the correct solution for v. Then, according to Theorem 2.3, there exists a $k > K^* + 3n$ such that the CT-root of the computation tree $T = T^k(v)$ is part of some MWIS J on T. We now define sets S^+ and S^- analogously to the proof for $v \in I$ and start the recursive definition of these sets by including the CT-root in S^-. We can then show that J is not an MWIS of T, so the assumption that BP-MWIS does not converge for $v \in W, v \notin I$ was wrong. We omit the rest of the proof, since it is very similar to the proof for $v \in I$.

Finally, we show that BP-MWIS converges to the correct solution for nodes $v \notin W$. Assume w.l.o.g. that $v \in V_1$ and let d be the length of the shortest path from v to v_1 in G. Note that $T = T^{k+d}(v)$ is exactly the same as $\hat{T} = T^k(v_1)$ for $k \geq n$ (except that the CT-root is different). Since these two computation trees are the same, also the MWISs on these trees are the same. We denote the v_1-labeled node that is closest to the CT-root in T by u. Since u corresponds to the CT-root of $\hat{T}$, $v_1 \in W$, and BP-MWIS converges to the correct solution for nodes in W, u is in every MWIS of T if $v_1 \in I$ and it is in no MWIS of T if $v_1 \notin I$. Let $M = M[\{u\}]$. We now consider the case where u is in every MWIS of T. In computation tree T, all nodes in $M \setminus \{u\}$ are only connected to other nodes in M. Therefore, every MWIS J on T with $u \in J$ includes each $[x, y] \in M$ if and only if $y \in V_1^+$. If $y \in V_1^+$ and $v_1 \in I$, then also $y \in I$. On the other hand, if $y \notin V_1^+$ and $v_1 \in I$, then also $y \notin I$. This holds in particular for the CT-root. It will be in every MWIS of T if $v \in I$ and in no MWIS of T if $v \notin I$. The case that u is in no MWIS of T is similar and we therefore omit the proof. □

3.2 Graphs for Which BP-MWIS Does Not Converge

In Section 3.1 we showed that BP-MWIS converges to the correct solution for all possible node weights for graphs with at most one even cycle and no odd cycles. In this section we show that these are the only graphs for which BP-MWIS converges to the correct solution for all possible node weights. First we show that there exist node weights such that BP-MWIS does not converge for graphs that contain an odd cycle and then we show that there exist node weights such that BP-MWIS does not converge to the correct solution for graphs that contain two or more even cycles.

In our proofs we use the concept of *heavy* nodes and *light* nodes. We denote the set of heavy nodes by H and the set of light nodes by L. The heavy nodes all have weight at least 1. We do not specify the exact weights of the light nodes, but they all have weight from the open interval $]0, 1/9n^2[$ such that the weights of all subsets of L are different, that is, $w(S) = w(T) \Rightarrow S = T$ for all $S, T \subset L$. We choose the node weights like this to ensure that the MWIS is unique.

First we consider graphs with at least one odd cycle. For these graphs our result follows directly from Theorem 2.2 and the fact that for graphs with an odd cycle we can choose node weights such that LP-MWIS does not have an integral optimal solution.

Theorem 3.2. *Let $G = (V, E)$ be a graph that contains at least one odd cycle $C = (W, F)$. Then there exist weights for the nodes such that the MWIS of G is unique, but BP-MWIS does not converge.*

Proof. Let $k = |W|$. We denote the nodes in C by $v_0, v_1, v_2, \ldots, v_k = v_0$ such that $(v_i, v_{i+1}) \in F$. We choose the node weights such that the nodes in $T = \{v_1, v_3, v_5, \ldots, v_{k-4}, v_{k-2}\}$ have weight $1 + 1/(2n)$, the nodes in $W \setminus T$ have weight 1, and all nodes in $V \setminus W$ are light nodes. We show that the optimal solution of LP-MWIS is non-integral.

The MWIS of G consists of the nodes in T plus some light nodes. This is because we can include at most $(k-1)/2$ nodes from W and including a node from $W \setminus T$ instead of a node in T costs us $1/(2n)$, while we can gain at most $(n-k)(1/(9n^2)) < 1/(9n)$ by including more of the light nodes. The weight of the MWIS of G is therefore bounded by $((k-1)/2)(1 + 1/(2n)) + (n-k)(1/(9n^2)) < k/2$. By our assumption on the weights of the light nodes, the MWIS is unique.

Let x be the solution of LP-MWIS with $x_i = 0$ if $i \notin W$ and $x_i = 1/2$ if $i \in W$. The objective value for x is clearly greater than $k/2$. This shows that LP-MWIS cannot have an integral optimal solution and according to Theorem 2.2 BP-MWIS does not converge. □

Next we consider graphs with at least two even cycles.

Theorem 3.3. *Let $G = (V, E)$ be a graph that contains at least two even cycles $C_1 = (W_1, F_1)$ and $C_2 = (W_2, F_2)$. There exist node weights such that the MWIS of G is unique, but BP-MWIS does not converge to the correct solution.*

Proof. If G contains an odd cycle, then the theorem follows from Theorem 3.2. We therefore assume in the following that G is bipartite. We now define a set X of nodes and a set Y of edges as follows. If C_1 and C_2 have at least one node in common, we define $X = W_1 \cup W_2$ and $Y = F_1 \cup F_2$. If C_1 and C_2 have no nodes in common, let $P = (W_P, F_P)$ be an arbitrary path from W_1 to W_2. In this case we define $X = W_1 \cup W_2 \cup W_P$ and $Y = F_1 \cup F_2 \cup F_P$. Note that all nodes in X have degree at least 2 in the graph $M = (X, Y)$, since either they are on one of the two cycles, or they are a non-leaf node of the path P.

Since M is a connected bipartite graph, we can uniquely partition the nodes in X into two sets X_1 and X_2 such that there are no edges (x_1, x_2) in Y between a node $x_1 \in X_1$ and an node $x_2 \in X_2$. We now distinguish two cases.

Case 1: $|X_1| \neq |X_2|$. Assume w.l.o.g. that $|X_1| > |X_2|$. We define weights $\tilde{w}$ for the nodes in X as follows. Each node $x_1 \in X_1$ has weight $\tilde{w}(x_1) = 1$ and each node $x_2 \in X_2$ has weight $\tilde{w}(x_2) = 1 + 1/(2n)$. Let $S \subset X$ be an arbitrary MWIS of M according to the weights $\tilde{w}$. Since G is bipartite, S is an independent set of G as well. We now define weights w on the nodes V of G as follows. All nodes in $V \setminus X$ are light nodes. Each node $x \in S$ has weight $w(x) = \tilde{w}(x) + 1/(4n)$. Finally, each node $x \in X \setminus S$ has weight $w(x) = \tilde{w}(x)$. By choosing the weights w like this, we ensure that the MWIS J of G is unique and consists of the nodes in S plus the heaviest subset $\hat{L}$ of light nodes such that nodes in $\hat{L}$ are not incident to nodes in S or other nodes in $\hat{L}$. The MWIS is unique, since the nodes in S have total weight at least $1/(4n)$ greater than any other subset of X and the total weight of all light nodes is at most $1/(9n)$.

Note that at least one of the nodes in X_1 is part of J, since X_1 is an independent set of M and it has total weight greater than any subset of nodes $D \subset X_2$, because of $|X_1| > |X_2|$. Let $x_1 \in X_1$ be part of J. Assume that BP-MWIS converges to the correct solution in K iterations. We consider the computation tree $T = T^k(x_1)$ for some even $k \geq K$. Since BP-MWIS converged to the correct solution by assumption and because of Theorem 2.3, the CT-root of T is a member of every MWIS of T. We now show by induction that this is not the case and that our assumption that BP-MWIS converges to the correct solution is wrong. In particular, we show that all X_1-labeled nodes are in no MWIS of T, while all X_2-labeled nodes are in every MWIS of T. Note that a node u in T that is heavier than all of its neighbors together is in every MWIS of T, since we can always improve independent sets of T that do not include u by including u and removing all neighbors of u.

As the basis step, we consider the leafs of T. Since the leafs are at an odd distance from the CT-root, they cannot be X_1-labeled. If they are X_2-labeled, they are in every MWIS of T, since they have greater weight than their parent node.

As the induction step, we consider the nodes at distance t from the CT-root. We assume that for all nodes at distance greater than t from the CT-root it holds that they are part of no MWIS of T if they are X_1-labeled and that they are part of every MWIS of T if they are X_2-labeled. For even t, nodes cannot be X_2-labeled. X_1-labeled nodes u at distance t from the CT-root have at least one X_2-labeled neighbor v which is at distance $t + 1$ from the CT-root.

Since v is part of every MWIS of T by assumption, u is part of no MWIS of T. For odd t, nodes cannot be X_1-labeled. An X_2-labeled node u at distance t from the CT-root is in every MWIS of T, since its X_1-labeled neighbors at distance $t+1$ from the CT-root are in no MWIS of T by assumption and its parent plus its light neighbors in T have total weight less than $w(u)$.

Case 2: $|X_1| = |X_2|$. The only connected graphs for which all nodes have degree at most 2 are paths and cycles. Since M is connected and is neither a path nor a cycle, it must contain at least one node with degree at least 3. Assume w.l.o.g. that node $x \in X_1$ has degree at least 3. We define weights $\tilde{w}$ for the nodes in X as follows. Node x has weight $\tilde{w}(x) = 5/3$. Each node $x_1 \in X_1 \setminus x$ has weight $\tilde{w}(x_1) = 1$ and each node $x_2 \in X_2$ has weight $\tilde{w}(x_2) = 1 + 1/(2n)$. Let $S \subset X$ be an arbitrary MWIS of M. We now define weights w on the nodes V of G as follows. All nodes in $V \setminus X$ are light nodes. Each node $x \in S$ has weight $w(x) = \tilde{w}(x) + 1/(4n)$. Finally, each node $x \in X \setminus S$ has weight $w(x) = \tilde{w}(x)$. Again, this way we ensure that the MWIS J of G is unique and consists of the nodes in S plus the heaviest subset $\hat{L}$ of light nodes such that nodes in $\hat{L}$ are not incident to nodes in S or other nodes in $\hat{L}$. The MWIS is unique, since the nodes in S have total weight at least $1/(4n)$ greater than any other independent set on M and the total weight of all light nodes is at most $1/(9n)$.

Note that at least one of the nodes in X_1 is part of J, since X_1 is an independent set on M and it has total weight greater than any subset of nodes $D \subset X_2$. Let $x_1 \in X_1$ be part of J. Assume that BP-MWIS converges to the correct solution in K iterations. We consider the computation tree $T = T^k(x_1)$ for some even $k \geq K$. Since BP-MWIS converged to the correct solution and by Theorem 2.3, the CT-root of T is a member of every MWIS of T. We now show by induction that this is not the case and that our assumption that BP-MWIS converges to the correct solution is wrong. In particular, we show that all X_1-labeled nodes are in no MWIS of T, while all X_2-labeled nodes are in every MWIS of T.

As the basis step, we consider the leafs of T. Since the leafs are at an odd distance from the CT-root, they cannot be X_1-labeled. The X_2-labeled leafs that do not have an x-labeled node as their parent are in every MWIS of T, since they have greater weight than their parent node. Now consider an X_2-labeled leaf u that has an x-labeled node v as its parent. Since x has degree at least 3 in M, v has at least two heavy leafs as its children. Therefore, v cannot be in any MWIS of T, since we can improve any independent set of T containing v by removing v and adding its children in T. Since v is the only neighbor of u and v is in no MWIS of T, node u is in every MWIS of T.

As the induction step, we consider the nodes at distance t from the CT-root. We assume that for all nodes at distance greater than t from the CT-root it holds that they are part of no MWIS of T if they are X_1-labeled and that they are part of every MWIS of T if they are X_2-labeled. For even t, nodes cannot be X_2-labeled. X_1-labeled nodes u at distance t from the CT-root have at least one X_2-labeled neighbor v which is at distance $t+1$ from the CT-root. Since v is part of every MWIS of T by assumption, u is part of no MWIS of T. For odd t, nodes cannot be X_1-labeled. For X_2-labeled nodes u at distance t from the CT-root we again distinguish two cases. If the parent of u is not x-labeled, u is in every MWIS of T, since its X_1-labeled neighbors at distance $t+1$ from the CT-root are in no MWIS of T by assumption and its parent plus its light neighbors in T have total weight less than $w(u)$. If the parent v of u is x-labeled, it has at least two heavy X_2-labeled nodes as its children. Node v cannot be in any MWIS of T, since we can improve any independent set of T containing v by

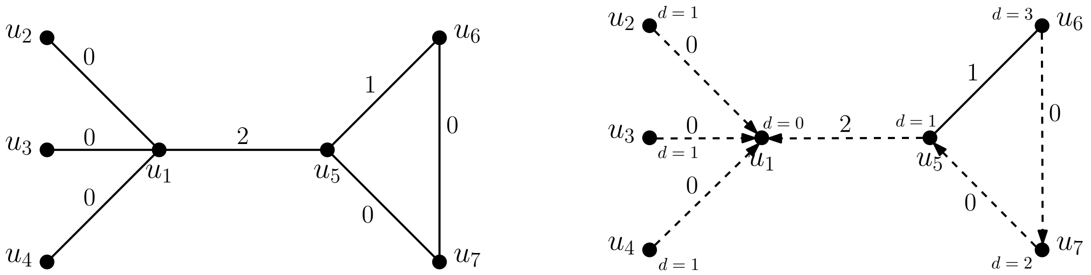

Figure 2: The left image shows the instance for which BP-MST does not converge. The right image shows the MST (dashed edges) for this instance, modeled as a directed tree rooted at u_1.

adding all its heavy children C in T and removing all neighbors of nodes in C (since X_1-labeled neighbors at distance greater than t from the CT-root are in no MWIS this always leads to an improvement), leading to a contradiction. Since v is in no MWIS of T and all X_1-labeled neighbors of u at distance $t+1$ from the CT-root are in no MWIS of T by assumption, u is in every MWIS of T. □

4 Non-Convergence of BP-MST

In this section we provide an instance of the MST problem for which BP-MST does not converge to the correct solution. The instance $G = (V, E)$ is as follows (see Figure 2):

- $V = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7\}$;
- $E = \{(u_1, u_2), (u_1, u_3), (u_1, u_4), (u_1, u_5), (u_5, u_6), (u_5, u_7), (u_6, u_7)\}$;
- The weights of the edges are $w(u_1, u_5) = 2$, $w(u_5, u_6) = 1$, and the rest of the edges have weight 0.

As can easily be observed, the MST T^* of G consists of all edges except for the edge (u_5, u_6). Modeled as a directed spanning tree rooted at u_1, the set S of parents and depths corresponding to T^* is given by

$$\begin{aligned} S = \{&(p_{u_1} = u_1, d_{u_1} = 0), (p_{u_2} = u_1, d_{u_2} = 1), (p_{u_3} = u_1, d_{u_3} = 1), (p_{u_4} = u_1, d_{u_4} = 1), \\ &(p_{u_5} = u_1, d_{u_5} = 1), (p_{u_6} = u_7, d_{u_6} = 3), (p_{u_7} = u_5, d_{u_7} = 2)\}. \end{aligned} \tag{6}$$

Before we formally prove that BP-MST for G does not converge to T^*, we give an intuitive explanation of why this is the case. Note that in any spanning tree of G the expensive edge (u_1, u_5) has to be included, since this is the only edge that connects the nodes $\{u_1, u_2, u_3, u_4\}$ with the nodes $\{u_5, u_6, u_7\}$. However, copies of the edge (u_1, u_5) in the computation tree are not necessarily included in an oriented spanning tree (OST). In fact, for any u_5-labeled node in the computation tree it is cheaper to have either its u_6-labeled neighbor or its u_7-labeled neighbor as its parent than its u_1-labeled neighbor.

We show that BP-MST does not converge for G by proof by contradiction. Assume to the contrary that BP-MST for G converges. According to Theorems 2.1, 2.4, and 2.5, if we consider a sufficiently large computation tree T, the MWOST $\hat{T}$ on T should consist of copies of T^*

close to the root of T. Therefore, $\hat{T}$ contains several edges labeled (u_1, u_5). We show that we can construct an OST on T with lower costs than $\hat{T}$ by replacing an (expensive) edge labeled (u_1, u_5) by a (cheaper) edge labeled (u_5, u_6), changing the node depths where necessary. This contradicts the optimality of $\hat{T}$. We conclude that BP-MST does not converge for G.

We proceed with the formal proof.

Lemma 4.1. *If BP-MST converges for G, then it converges to the set S (see Equation* (6)*).*

Proof. Assume that BP-MST converges for G after K iterations. First we show that BP-MST converges to the correct parents p_v as given by S. For u_1 this is clear, since it is the MST-root and its parent is u_1 by definition. Now assume that for some nodes BP-MST does not converge to the correct parents. Among all these nodes, let v be one of minimum depth d_v^S as given by S and let p_v^S be the parent of v as given by S. Since S is a rooted spanning tree, p_v^S has smaller depth as given by S than v and, therefore, BP-MST converges to the correct parent for p_v^S. This means that we have neither $p_{p_v^S} = v$, nor $p_v = p_v^S$. Therefore, the edge (v, p_v^S) is not in the set $\{(v, p_v)_{v \in V \setminus \{\text{MST-root}\}}\}$, contradicting Theorem 2.4. We conclude that BP-MST converges to the correct parents for all nodes.

Finally, we show that BP-MST converges to the correct depths d_v. For u_1 this is again true by definition. Assume that for some nodes, BP-MST converges to the incorrect depths. Among all these nodes, let v be one of minimum depth d_v^S as given by S and let p_v^S be the parent of v as given by S. Consider the computation tree $T^{K+1}(v)$. According to Theorem 2.1 and the above, the neighbor $[x, p_v^S]$ of $[\text{CT-root}, v]$ has depth $d_x = d_{p_v^S}^S$. Since $p_{\text{CT-root}} = x$ and v takes the incorrect depth, we have $d_{\text{CT-root}} = d_v \neq d_v^S = d_{p_v^S}^S + 1 = d_x + 1$, contradicting Theorem 2.5. We conclude that BP-MST converges to the correct depths for all nodes. □

Theorem 4.2. *BP-MST does not converge for G.*

Proof. Assume to the contrary that BP-MST converges for G after K iterations. According to Lemma 4.1, BP-MST converges to the set S. We now consider the computation tree $T = T^{K+4}(u_5)$. According to Theorem 2.1, all nodes in T that are at distance at most 4 from the CT-root $[\text{root}, u_5]$ take MAP assignment according to S. We denote the OST that corresponds to the MAP assignment on T by T_1. We now show that we can change the parents and depths for some nodes in T such that we obtain another OST T_2 of weight less than the weight of T_1. Consider all nodes in T at distance at most 4 from the CT-root and all edges between them (see Figure 3). We make the following changes to the assignments of the nodes. We change p_{root} to $[x_2, u_6]$. We change d_{root} to 4, d_{x_3} to 5, and d_{x_8} to 6. Note that the new assignment is valid. For the nodes at distance 4 or less from the CT-root this can easily be checked and for nodes further away from the CT-root it follows since we did not change their parents and depths, and also the parents and depths of nodes at distance exactly 4 from the CT-root were not changed.

The new assignment corresponds to another OST T_2. The new tree T_2 contains exactly the same edges as T_1, except that it contains an extra edge labeled (u_5, u_6) and it does not contain one of the edges labeled (u_1, u_5) (see Figure 3). Since edge (u_5, u_6) weighs less than edge (u_1, u_5), T_2 weighs less than T_1. Therefore, BP-MST did not compute the MWOST on T, contradicting Theorem 2.5. We conclude that our initial assumption was incorrect and that BP-MST does not converge for G. □

The graph G shows that BP-MST does not converge for all graphs. Since computing the MST on a tree is trivial, G is one of the simplest non-trivial instances. Bayati et al. [2] showed

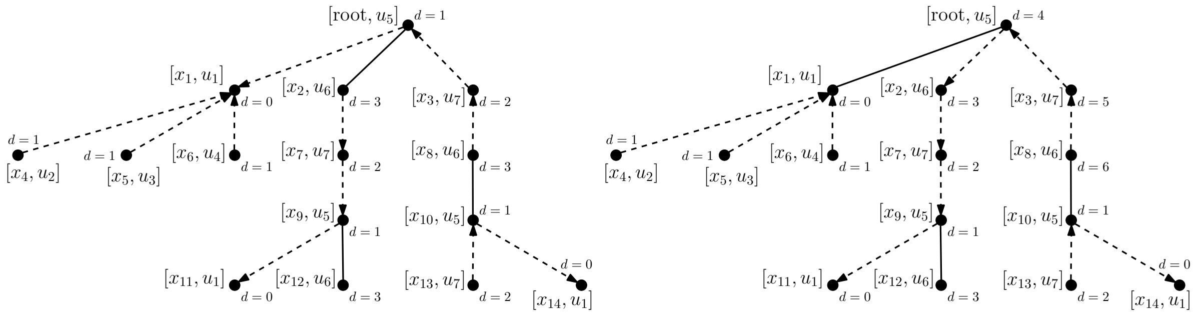

Figure 3: Both images show all nodes in the computation tree $T^{K+4}(u_5)$ that are at distance at most 4 from the CT-root $[\text{root}, u_5]$, and all edges between these nodes. The left image shows (dashed edges) the OST T_1 and the right image shows the OST T_2.

that BP-MST is correct if it converges. However, the graph G shows that there exist simple instances for which BP-MST does not converge. We believe that BP-MST does not converge for most instances encountered in practice. The reason for this is that to form the MWOST of the computation tree it is often not optimal to use copies of the MST of the input graph H. Even if the MST of H contains only one somewhat expensive edge e, an OST on the computation tree consisting of copies of the MST of H can usually be improved by leaving out a copy of edge e and adding a cheaper edge.

5 Concluding Remarks

In this paper, we have analyzed belief propagation for minimum spanning trees (BP-MST) and minimum-weight independent set (BP-MWIS).

For BP-MWIS, we completely characterized the graphs on which BP-MWIS converges for all node weights, provided that the minimum-weight independent set is unique. We remark that the node weights that we provide for showing that BP-MWIS does not converge are robust against small perturbations. This indicates that the non-convergence is not a pathological behavior, but likely to occur.

For BP-MST, we gave a small example on which BP-MST does not converge. Since this example is quite small, it is likely that such a structure occurs in many practical instances, which is an indication that BP-MST does not converge on many instances.

References

[1] Mohsen Bayati, Christian Borgs, Jennifer Chayes, and Riccardo Zecchina. Belief-propagation for weighted b-matching on arbitrary graphs and its relation to linear programs with integer solutions. *SIAM Journal on Discrete Mathematics*, 25(2):989–1011, 2011.

[2] Mohsen Bayati, Alfredo Braunstein, and Riccardo Zecchina. A rigorous analysis of the cavity equations for the minimum spanning tree. *Journal of Mathematical Physics*, 49(12):125206, 2008.

[3] Mohsen Bayati, Devavrat Shah, and Mayank Sharma. Max-product for maximum weight

matching: Convergence, correctness, and LP duality. *IEEE Transactions on Information Theory*, 54(3):1241–1251, 2008.

[4] Tobias Brunsch, Kamiel Cornelissen, Bodo Manthey, and Heiko Rglin. Smoothed analysis of belief propagation for minimum-cost flow and matching. *Journal of Graph Algorithms and Applications*, 17(6):647–670, 2013.

[5] Amin Coja-Oghlan, Elchanan Mossel, and Dan Vilenchik. A spectral approach to analysing belief propagation for 3-colouring. *Combinatorics, Probability and Computing*, 18(6):881–912, 2009.

[6] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proc. of the 47th Ann. ACM Symp. on Theory of Computing (STOC)*, pages 59–68. ACM, 2015.

[7] David Gamarnik, Devavrat Shah, and Yehua Wei. Belief propagation for min-cost network flow: Convergence and correctness. *Operations Research*, 60(2):410–428, 2012.

[8] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[9] Justin Salez and Devavrat Shah. Belief propagation: An asymptotically optimal algorithm for the random assignment problem. *Math. Oper. Res.*, 34(2):468–480, 2009.

[10] Sujay Sanghavi, Dmitry M. Malioutov, and Alan S. Willsky. Belief propagation and LP relaxation for weighted matching in general graphs. *IEEE Transactions on Information Theory*, 57(4):2203–2212, 2011.

[11] Sujay Sanghavi and Devavrat Shah. Tightness of LP via max-product belief propagation. Technical Report 0508097v2 [cs.DS], arXiv, 2008.

[12] Sujay Sanghavi, Devavrat Shah, and Alan S. Willsky. Message passing for maximum weight independent set. *IEEE Transactions on Information Theory*, 55(11):4822 –4834, 2009.

[13] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[14] Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *Proc. of the 9th IEEE International Conference on Computer Vision (ICCV 2003)*, pages 900–907. IEEE Computer Society, 2003.

[15] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41, 2000.

[16] Chen Yanover and Yair Weiss. Approximate inference and protein-folding. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems (NIPS 2002)*, pages 84–86. MIT Press, 2002.

[17] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–269. Morgan Kaufmann, 2003.