Smoothed Analysis of Belief Propagation for Minimum-Cost Flow and Matching*

Tobias Brunsch[1], Kamiel Cornelissen[2], Bodo Manthey[2], Heiko Röglin[1]

[1]University of Bonn, Department of Computer Science
brunsch@cs.uni-bonn.de, heiko@roeglin.org
[2]University of Twente, Department of Applied Mathematics
b.manthey@utwente.nl, k.cornelissen@utwente.nl

Belief propagation (BP) is a message-passing heuristic for statistical inference in graphical models such as Bayesian networks and Markov random fields. BP is used to compute marginal distributions or maximum likelihood assignments and has applications in many areas, including machine learning, image processing, and computer vision. However, the theoretical understanding of the performance of BP remains limited.

Recently, BP has been applied to combinatorial optimization problems. It has been proved that BP can be used to compute maximum-weight matchings and minimum-cost flows for instances with a unique optimum. The number of iterations needed for this is pseudo-polynomial and hence BP is not efficient in general.

We study BP in the framework of smoothed analysis and prove that with high probability the number of iterations needed to compute maximum-weight matchings and minimum-cost flows is bounded by a polynomial if the weights/costs of the edges are randomly perturbed. To prove our upper bounds, we use an isolation lemma by Beier and Vöcking (SIAM Journal on Computing, 2006) for the matching problem and we generalize an isolation lemma by Gamarnik, Shah, and Wei (Operations Research, 2012) for the min-cost flow problem. We also prove lower tail bounds for the number of iterations that BP needs to converge that almost match our upper bounds.

1 Belief Propagation

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 [9]. Typical graphical models to which BP is applied are Bayesian networks and Markov random fields. There are two variants of BP. The sum-product variant is used to compute marginal probabilities. The max-product variant (or the functionally equivalent min-sum variant) is used to compute maximum *a posteriori* (MAP) probability estimates.

*This research was supported by ERC Starting Grant 306465 (BeyondWorstCase) and NWO grant 613.001.023 (Smoothed Analysis of Belief Propagation).

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. [18]. 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 [16, 17].

If the graphical model is tree-structured, BP computes exact marginals or 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 [2, 4, 11, 12], the minimum spanning tree (MST) problem [3], the minimum-cost flow (MCF) problem [7], the maximum-weight independent set problem [13], and the 3-coloring problem [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.

Bayati et al. [4] have shown that max-product BP correctly computes the MWM in bipartite graphs in pseudo-polynomial time if the MWM is unique. For MST, it is known that BP converges to the correct optimal solution, if it converges at all (which is not guaranteed) [3]. Gamarnik et al. [7] have shown that the max-product BP algorithm computes the MCF in pseudo-polynomial time if the MCF is unique.

1.1 Belief Propagation for Matching and Flow Problems

In this section, we discuss the previous results about the BP algorithm for computing maximum-weight matchings and minimum-cost flows in more detail. Bayati et al. [4] have shown that the max-product BP algorithm correctly computes the maximum-weight matching in bipartite graphs if the MWM is unique. Convergence of BP takes pseudo-polynomial time and depends linearly on both the weight of the heaviest edge and $1/\delta$, where δ is the difference in weight between the best and second-best matching. In Section 2.3, we describe the BP algorithm for MWM in detail.

Belief propagation has also been applied to finding maximum-weight perfect matchings in arbitrary graphs and to finding maximum-weight perfect b-matchings [2, 12], where a perfect b-matching is a set of edges such that every vertex i is incident to exactly b_i edges in the set. For arbitrary graphs, the BP algorithm for MWM does not necessarily converge [12]. However, Bayati et al. [2] and Sanghavi et al. [12] have shown that the BP algorithm converges to the optimal matching if the relaxation of the corresponding linear program has an optimal solution that is unique and integral. The number of iterations needed until convergence depends again linearly on the reciprocal of the parameter δ. Bayati et al. [2] have also shown that the same result holds for the problem of finding maximum-weight b-matchings that do not need to be perfect.

Gamarnik et al. [7] have shown that BP can be used to find a minimum-cost flow, provided that the instance has a unique optimal solution. The number of iterations until convergence is pseudo-polynomial and depends again linearly on the reciprocal of the difference in cost between the best and second-best integer flow. In addition, they have proved a discrete isolation lemma [7, Theorem 8.1] that shows that the edge costs can be slightly randomly perturbed to ensure that, with a probability of at least $1/2$, the perturbed MCF instance has a unique optimal solution. Using this result, they constructed an FPRAS for MCF using BP.

1.2 Smoothed Analysis

Smoothed analysis was introduced by Spielman and Teng [14] in order to explain the performance of the simplex method for linear programming. It is a hybrid of worst-case and average-case analysis and an alternative to both. An adversary specifies an instance, and this instance is then slightly randomly perturbed. The perturbation can, for instance, model noise from measurement. For many algorithms worst-case instances are contrived and are unlikely to occur in practice. Also, adding a little noise to a worst-case instance often dramatically improves the performance of an algorithm on this instance. A problem with average-case analysis is that it is sometimes not clear what probability distributions for the input instances should be used in order to obtain typical instances. Good performance bounds of an algorithm in the smoothed setting usually indicate good performance on instances encountered in practice since instances from practice are usually subject to a certain amount of noise because of, for example, measurement errors. For these reasons, smoothed analysis has been applied in a variety of contexts since its invention in 2001. We refer to two recent surveys [8,15] for a broader picture.

We apply smoothed analysis to belief propagation for min-cost flow and maximum-weight matching. The graph and (for MCF) the capacities of the edges and the budgets of the nodes are adversarial. The costs or weights of the edges are random according to the one-step model introduced by Beier and Vöcking [5]. The one-step model generalizes the model for smoothed analysis originally introduced by Spielman and Teng [14] by allowing the adversary to not just choose the mean of each input parameter, but even its distribution. We consider the general probabilistic model described below.

- The adversary specifies the graph $G = (V, E)$ and, in the case of min-cost flow, the integer capacities of the edges and the integer budgets (both are not required to be polynomially bounded). Additionally the adversary specifies a probability density function $f_e : [0, 1] \to [0, \phi]$ for every edge e.

- The costs (for min-cost flow) or weights (for matching) of the edges are then drawn independently according to their respective density functions.

The parameter ϕ controls the adversary's power: If $\phi = 1$, then we have the average case. The larger ϕ, the more powerful the adversary. Note that one option for the adversary is to specify a density function f_e such that it takes value ϕ on some interval $[t_e, t_e + \frac{1}{\phi}] \subseteq [0, 1]$ and takes value 0 outside this interval. This choice ensures that the weight of edge e takes its value in an interval of width $1/\phi$ and most resembles worst-case analysis. The role of ϕ is the same as the role of $1/\sigma$ in the classical model of smoothed analysis, where instances are perturbed by independent Gaussian noise with standard deviation σ. In that model the maximum density ϕ is proportional to $1/\sigma$.

1.3 Our Results

We prove upper and lower tail bounds for the number of iterations that BP needs to solve maximum-weight matching problems and min-cost flow problems. Our upper bounds match our lower bounds up to a small polynomial factor. While previous bounds on the worst-case running time for BP for MWM and MCF are pseudopolynomial [2,4,7,12], we show that in the framework of smoothed analysis with high probability BP only requires a polynomial number of iterations to converge to the correct solution. This suggests that for instances encountered in practice, BP is unlikely to require a superpolynomial number of iterations. In the following, n and m are the number of nodes and edges of the input graph, respectively. In summary, we prove the following results:

- For the min-cost flow problem, the probability that BP needs more than t iterations is upper bounded by $O(n^2 m\phi/t)$ (Sections 3.2 and 4.2). There are smoothed instances for which the probability that BP needs more than t iterations is lower bounded by $\Omega(n\phi/t)$ (Section 5.3).
- For the maximum-weight matching problem, the probability that BP needs more than t iterations is upper bounded by $O(nm\phi/t)$ (Sections 3.1 and 4.1). There are smoothed instances for which the probability that BP needs more than t iterations is lower bounded by $\Omega(n\phi/t)$ (Section 5.3).

The upper bound for matching problems holds for the variants of BP for the maximum-weight matching problem in bipartite graphs [4] as well as for the maximum-weight (perfect) b-matching problem in general graphs [2, 12]. For the latter it is required that the polytope corresponding to the relaxation of the matching LP is integral.

To prove the upper tail bound for BP for MCF, we use a continuous isolation lemma that is similar to the discrete isolation lemma by Gamarnik et al. [7, Theorem 8.1]. We need the continuous version since we do not only want to have a unique optimal solution, but we also need to quantify the gap between the best and the second-best solutions.

Though our upper tail bounds show that with high probability BP needs only a polynomial number of iterations, our bounds are not strong enough to yield any bound on the expected number of iterations. Indeed, using the lower bound of $\Omega(n\phi/t)$ for the probability that t iterations are insufficient to find a maximum-weight matching in bipartite graphs, we show that this expectation is not finite. The lower bound even holds in the average case, i.e., if $\phi = 1$ (Section 5.2), and it carries over to the variants of BP for the min-cost flow problem and the minimum/maximum-weight (perfect) b-matching problem in general graphs mentioned above [2, 4, 7, 12]. The lower bound matches the upper bound up to a factor of $O(m)$ for matching and up to a factor of $O(nm)$ for min-cost flow (Section 5.3). The smoothed lower bound even holds for complete (i.e., non-adversarial) bipartite graphs.

Finally, let us remark that, for the min-cost flow problem, we bound only the number of iterations that BP needs until convergence. The messages might be super-polynomially long. However, for all matching problems, the length of the messages is always bounded by a small polynomial.

2 Definitions and Problem Statement

In this section, we define the maximum-weight matching problems that we consider and the min-cost flow problem. We also describe the BP algorithms that we apply.

2.1 Maximum-Weight Matching and Minimum-Cost Flow

First, we define the maximum-weight matching problem. For this, consider an undirected weighted graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, and $E \subseteq \big\{\{v_i, v_j\} = e_{\{i,j\}}, 1 \leq i < j \leq n\big\}$. Each edge $e_{\{i,j\}}$ has weight $w_{\{i,j\}} \in \mathbb{R}^+$. A collection of edges $M \subseteq E$ is called a matching if each node of G is incident to at most one edge in M. We define the weight of a matching M by

$$w(M) = \sum_{e_{\{i,j\}} \in M} w_{\{i,j\}}.$$

The maximum-weight matching $M^\star$ of G is defined as

$$M^\star = \text{argmax}\{w(M) \mid M \text{ is a matching of } G\}.$$

The bipartite maximum-weight matching problem is defined analogously. The only difference is that for this problem it is required that the graph G is bipartite, i.e., its vertex set V can be partitioned in two sets V_1 and V_2 such that all edges in its edge set E have exactly one endpoint in V_1 and exactly one endpoint in V_2.

A b-matching $M \subseteq E$ in an arbitrary graph $G = (V, E)$ is a set of edges such that every node $i \in V$ is incident to at most b_i edges from M, where $b_i \geq 0$. A b-matching is called perfect if every node $i \in V$ is incident to exactly b_i edges from M. Also for these problems we assume that each edge $e \in E$ has a certain weight w_e and we define the weight of a b-matching M accordingly.

2.2 Min-Cost Flow Problem

In the min-cost flow problem (MCF), the goal is to find a cheapest flow that satisfies all capacity and budget constraints. We are given a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$. In principle we allow multiple edges between a pair of nodes, but for ease of notation we consider simple directed graphs. Each node v has a budget $b_v \in \mathbb{Z}$. Each directed edge $e = e_{ij}$ from v_i to v_j has capacity $u_e \in \mathbb{N}_0$ and cost $c_e \in \mathbb{R}^+$. For each node $v \in V$, we define E_v as the set of edges incident to v. For each edge $e \in E_v$ we define $\Delta(v, e) = 1$ if e is an out-going edge of v and $\Delta(v, e) = -1$ if e is an in-going edge of v. In the MCF, one needs to assign a flow f_e to each edge e such that the total cost $\sum_{e \in E} c_e f_e$ is minimized and the flow constraints

$$0 \leq f_e \leq u_e \quad \text{for all } e \in E,$$

and budget constraints

$$\sum_{e \in E_v} \Delta(v, e) f_e = b_v \quad \text{for all } v \in V$$

are satisfied. We refer to Ahuja et al. [1] for more details about MCF.

Note that we could also have allowed rational values for the budgets and capacities. Since our results do not depend on the values of the budgets and capacities, our results are not affected by scaling all capacities and budgets by the smallest common denominator.

Also note that finding a perfect minimum-weight matching in a bipartite graph $G = (U \cup V, E)$ is a special case of the min-cost flow problem; see Ahuja et al. [1] for details.

2.3 Belief Propagation

For the sake of completeness, we describe the BP algorithm for bipartite maximum-weight matching used by Bayati et al. [4]. We also provide a short description of the BP algorithm for min-cost flow by Gamarnik et al. [7]. For the details of this version of BP and for the versions of BP for the (perfect) maximum-weight b-matching problem, we refer to the original papers [2,7,12]. Note that in order to understand the results and proofs that follow in the rest of the paper, it is not necessary to know the details of the BP algorithm. When necessary, we discuss the differences between the various versions of BP in Sections 4 and 5.

The BP algorithm used by Bayati et al. [4] is an iterative message-passing algorithm for computing maximum-weight matchings (MWM). Bayati et al. define their algorithm for complete bipartite graphs $G = (U \cup V, E)$ with $|U| = |V| = n$. In each iteration t, each node u_i sends a message vector

$$\vec{M}^t_{ij} = [\vec{m}^t_{ij}(1), \vec{m}^t_{ij}(2), \ldots, \vec{m}^t_{ij}(n)]$$

to each of its neighbors v_j. The messages can be interpreted as how "likely" the sending node thinks it is that the receiving node should be matched to a particular node in the MWM. The greater the value of the message $\vec{m}^t_{ij}(r)$, the more likely it is according to node u_i in iteration t that node v_j should be matched to node u_r. Similarly, each node v_j sends a message vector $\overleftarrow{M}^t_{ji}$ to each of its neighbors u_i. The messages are initialized as

$$\vec{m}^0_{ij}(r) = \begin{cases} w_{ij} & \text{if } r = i, \\ 0 & \text{otherwise and} \end{cases}$$

$$\overleftarrow{m}^0_{ji}(r) = \begin{cases} w_{ij} & \text{if } r = j, \\ 0 & \text{otherwise.} \end{cases}$$

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

$$\vec{m}^t_{ij}(r) = \begin{cases} w_{ij} + \sum_{k \neq j} \overleftarrow{m}^{t-1}_{ki}(j) & \text{if } r = i, \\ \max_{q \neq j} \left[w_{iq} + \sum_{k \neq j} \overleftarrow{m}^{t-1}_{ki}(q) \right] & \text{otherwise and} \end{cases}$$

$$\overleftarrow{m}^t_{ji}(r) = \begin{cases} w_{ij} + \sum_{k \neq i} \vec{m}^{t-1}_{kj}(i) & \text{if } r = j, \\ \max_{q \neq i} \left[w_{qj} + \sum_{k \neq i} \vec{m}^{t-1}_{kj}(q) \right] & \text{otherwise.} \end{cases}$$

The beliefs of nodes u_i and v_j in iteration t are defined as

$$b^t_{u_i}(r) = w_{ir} + \sum_k \overleftarrow{m}^t_{ki}(r),$$

$$b^t_{v_j}(r) = w_{rj} + \sum_k \vec{m}^t_{kj}(r).$$

These beliefs can be interpreted as the "likelihood" that a node should be matched to a particular neighbor. The greater the value of $b^t_{u_i}(j)$, the more likely it is that node u_i should be matched to node v_j. We denote the estimated MWM in iteration t by $\tilde{M}^t$. The estimated matching $\tilde{M}^t$ matches each node u_i to node v_j, where $j = \operatorname{argmax}_{1 \le r \le n}\{b_{u_i}^t(r)\}$. Note that $\tilde{M}^t$ does not always define a matching, since multiple nodes may be matched to the same node. However, Bayati et al. [4] have shown that if the MWM is unique, then if t is sufficiently large, $\tilde{M}^t$ is a matching and equal to the MWM.

The BP algorithm for min-cost flow uses the same idea of iterative message-passing as the BP algorithm for bipartite matching. However, the messages sent between edges and their endpoints in the min-cost flow version are functions instead of vectors. These functions represent estimates of the cost of sending a certain amount of flow along the edge, taking into account the cost of the edge, the capacity of the edge, the fact that flow has to be conserved at the endpoints of the edge, and the messages received from neighboring edges in the previous iteration. For a detailed description of the BP algorithm for MCF we refer to the original work [7].

3 Isolation Lemma for Maximum-Weight Matchings and Min-Cost Flows

Before we turn to proving the upper tail bounds for the number of iterations of the BP algorithm in Section 4, we take a closer look at the quantity δ, which we defined above as the difference in weight or cost between the best and second-best matching or integer flow, respectively. The previous results discussed in Section 1.1 indicate that in order for the BP algorithm to be efficient δ must not be too small. While δ can be arbitrarily small for weights or costs that are chosen by an adversary, it is a well-known phenomenon that δ is with high probability not too small when the weights or costs are drawn randomly.

3.1 Maximum-Weight Matchings

Beier and Vöcking [5, Section 2.1] have considered a general scenario in which an arbitrary set $S \subseteq \{0,1\}^m$ of feasible solutions is given and to every $x = (x_1, \ldots, x_m) \in S$ a weight $w \cdot x = w_1x_1 + \ldots + w_mx_m$ is assigned by a linear objective function. As in our model they assume that every coefficient w_i is drawn independently according to an adversarial density function $f_i : [0,1] \to [0,\phi]$ and they define δ as the difference in weight between the best and the second-best feasible solution from S, i.e., $\delta = w \cdot x^\star - w \cdot \hat{x}$ where $x^\star = \operatorname{argmax}_{x \in S} w \cdot x$ and $\hat{x} = \operatorname{argmax}_{x \in S \setminus \{x^\star\}} w \cdot x$. They prove a strong isolation lemma that, regardless of the adversarial choices of S and the density functions f_i, the probability of the event $\delta \le \varepsilon$ is bounded from above by $2\varepsilon\phi m$ for any $\varepsilon \ge 0$.

If we choose S as the set of incidence vectors of all matchings or (perfect) b-matchings in a given graph, Beier and Vöcking's results yield for every $\varepsilon \ge 0$ an upper bound on the probability that the difference in weight δ between the best and second-best matching or the best and second-best (perfect) b-matching is at most ε. Combined with the results in Section 1.1, this can immediately be used to obtain an upper tail bound on the number of iterations of the BP algorithm for these problems. We prove this upper tail bound in Section 4.1.

3.2 Min-Cost Flows

The situation for the min-cost flow problem is significantly more difficult because the set S of feasible integer flows cannot naturally be expressed with binary variables. If one introduce a variable for each edge corresponding to the flow on that edge, then $S \subseteq \{0, 1, 2, \dots, u_{\max}\}^m$ where $u_{\max} = \max_{e \in E} u_e$. Röglin and Vöcking [10] have extended the isolation lemma to the setting of integer, instead of binary, vectors. However, their result is not strong enough for our purposes as it bounds the probability of the event $\delta \le \varepsilon$ by $\varepsilon \phi m (u_{\max} + 1)^2$ from above for any $\varepsilon \ge 0$. As this bound depends on $u_{\max}$ it would only lead to a pseudo-polynomial upper tail bound on the number of iterations of the BP algorithm when combined with the results of [7]. Our goal is, however, to obtain a polynomial tail bound that does not depend on the capacities. In the remainder of this section, we prove that the isolation lemma from [10] can be significantly strengthened when structural properties of the min-cost flow problem are exploited.

In the following we consider the residual network for a flow f. For each edge e_{ij} in the original network that has less flow than its capacity u_{ij}, we include an edge e_{ij} with capacity $u_{ij} - f_{ij}$ in the residual network. Similarly, for each edge e_{ij} that has flow greater than zero, we include the backwards edge e_{ji} with capacity f_{ij} in the residual network. We refer to Ahuja et al. [1] for more details about residual networks.

As all capacities and budgets are integers, there is always a min-cost flow that is integral. An additional property of our probabilistic model is that with probability one there do not exist two different integer flows with exactly the same cost. This follows directly from the fact that all costs are continuous random variables. Hence, without loss of generality we restrict our presentation in the following to the situation that the min-cost flow is unique.

In fact, Gamarnik et al. [7] have not used δ, the difference in cost between the best and second-best integer flow, to bound the number of iterations needed for BP to find the unique optimal solution of MCF, but they have used another quantity Δ. They have defined Δ as the length of the cheapest cycle in the residual network of the min-cost flow $f^\star$. Note that Δ is always non-negative. Otherwise, we could send one unit of flow along a cheapest cycle. This would result in a feasible integral flow with lower cost. With the same argument we can argue that Δ must be at least as large as δ because sending one unit of flow along a cheapest cycle results in a feasible integral flow different from $f^\star$ whose cost exceeds the cost of $f^\star$ by exactly Δ. Hence any lower bound for δ is also a lower bound for Δ and so it suffices for our purposes to bound the probability of the event $\delta \le \varepsilon$ from above.

The isolation lemma we prove is based on ideas that Gamarnik et al. [7, Theorem 8.1] have developed to prove that the optimal solution of a min-cost flow problem is unique with high probability if the costs are randomly drawn integers from a sufficiently large set. We provide a continuous counterpart of this lemma, where we bound the probability that the second-best integer flow is close in cost to the optimal integer flow.

Lemma 3.1. *The probability that the cost of the optimal and the second-best integer flow differs by at most $\varepsilon \ge 0$ is bounded from above by $2\varepsilon\phi m$.*

Proof. Consider any fixed edge $\tilde{e}$, and let the costs of all other edges be fixed by an adversary. The cost $c_{\tilde{e}}$ of $\tilde{e}$ is drawn according to its probability distribution, whose density is bounded by ϕ.

For $e \in E$, let $\underline{\mathsf{E}}^e_\varepsilon$ be the event that there exist two different integer flows $f^\star$ and $\hat{f}$ with the following properties:

(i) $f^\star$ is optimal.

(ii) $c \cdot f^\star$ and $c \cdot \hat{f}$ differ by at most ε, i.e., $c \cdot \hat{f} \le c \cdot f^\star + \varepsilon$.

(iii) $f^\star_e = 0$ and $\hat{f}_e > 0$.

Let $\overline{\mathsf{E}}^e_\varepsilon$ be analogously defined, except for Condition (iii) being replaced by $f^\star_e = u_e$ and $\hat{f}_e < u_e$.

Claim 3.2. *Let $e \in E$ be arbitrary. Assume that all costs except for c_e are fixed. Let $I \subseteq [0,1]$ be the set of real numbers such that $I = \{c_e \mid \underline{\mathsf{E}}^e_\varepsilon\}$. Then I is a subset of an interval of length at most ε.*

Proof. If $I \neq \emptyset$, let $\alpha = \min(I)$ and let $f^\star$ be an optimal integer flow for $c_e = \alpha$ with $f^\star_e = 0$. Due to the choice of α it is clear that $I \subseteq [\alpha, \infty)$ We claim that $I \subseteq [\alpha, \alpha + \varepsilon]$. If $c_e = \alpha + \eta$ for some $\eta > 0$, then $f^\star$ stays optimal, and, for any feasible integer solution f with $f_e > 0$, we have

$$\begin{aligned} c \cdot f = \sum_{\tilde{e} \neq e} c_{\tilde{e}} f_{\tilde{e}} + (\alpha + \eta) f_e &\geq \sum_{\tilde{e} \neq e} c_{\tilde{e}} f_{\tilde{e}} + \alpha f_e + \eta && \text{as } f_e \geq 1 \\ &\geq c \cdot f^\star + \eta && \text{as } f^\star_e = 0 \text{ and } f^\star \text{ is optimal.} \end{aligned}$$

Thus, for $\eta > \varepsilon$, the event $\underline{\mathsf{E}}^e_\varepsilon$ does not occur. □

The proof of the following claim is omitted as it is completely analogous to the proof of the previous claim.

Claim 3.3. *Let $e \in E$ be arbitrary. Assume that all costs except for c_e are fixed. Let $I \subseteq [0,1]$ be the set of real numbers such that $I = \{c_e \mid \overline{\mathsf{E}}^e_\varepsilon\}$. Then I is a subset of an interval of length at most ε.*

The following claim shows that, provided no event $\underline{\mathsf{E}}^e_\varepsilon$ or $\overline{\mathsf{E}}^e_\varepsilon$ occurs, the second-best integer flow is more expensive than the best integer flow by at least an amount of ε.

Claim 3.4. *Assume that for every edge $e \in E$ neither $\underline{\mathsf{E}}^e_\varepsilon$ nor $\overline{\mathsf{E}}^e_\varepsilon$ occurs. Let $f^\star$ be a min-cost flow and let $\hat{f} \neq f^\star$ be a min-cost integer flow that differs from $f^\star$, i.e., a second-best integer flow. Then $c \cdot \hat{f} \geq c \cdot f^\star + \varepsilon$.*

Proof. We prove the claim by contradiction. Assume to the contrary that for every edge $e \in E$ neither $\underline{\mathsf{E}}^e_\varepsilon$ nor $\overline{\mathsf{E}}^e_\varepsilon$ occurs, but that $f^\star$ and $\hat{f}$ differ less than ε in cost. First, we prove that under our assumption that the min-cost flow is unique some edge e exists such that $f^\star_e \in \{0, u_e\}$ and $\hat{f}_e \neq f^\star_e$. Suppose that no such edge e exists and let $d = f^\star - \hat{f}$. Then $d_e > 0$ only if $f^\star_e < u_e$ (otherwise event $\overline{\mathsf{E}}^e_\varepsilon$ occurs) and $d_e < 0$ only if $f^\star_e > 0$ (otherwise event $\underline{\mathsf{E}}^e_\varepsilon$ occurs). From this, we can conclude that there exists a $\lambda > 0$ such that $f^\star + \lambda d$ is a feasible flow. Let $\lambda_0 = \max\{\lambda \mid f^\star + \lambda d \text{ is feasible}\}$ and $\check{f} = f^\star + \lambda_0 d$. From the assumption that the min-cost flow is unique it follows that $c \cdot d = c \cdot f^\star - c \cdot \hat{f} < 0$. Hence, $c \cdot \check{f} < c \cdot f^\star$, contradicting the choice of $f^\star$ as min-cost flow.

This argument shows that there always exists an edge e such that $f^\star_e \in \{0, u_e\}$ and $\hat{f}_e \neq f^\star_e$. As none of the events $\underline{\mathsf{E}}^e_\varepsilon$ and $\overline{\mathsf{E}}^e_\varepsilon$ occurs for this edge e, it follows that $c \cdot \hat{f} \geq c \cdot f^\star + \varepsilon$, contradicting our assumption at the start of the proof that $f^\star$ and $\hat{f}$ differ less than ε in cost. □

From Claims 3.2 and 3.3, we obtain $\mathbb{P}(\underline{\mathsf{E}}^e_\varepsilon) \leq \varepsilon\phi$ and $\mathbb{P}(\overline{\mathsf{E}}^e_\varepsilon) \leq \varepsilon\phi$: We fix all edge costs except for c_e and then $\underline{\mathsf{E}}^e_\varepsilon$ can only occur if c_e falls into an interval of length at most ε. Since the density function of c_e is bounded from above by ϕ, this happens with a probability of at most $\varepsilon\phi$. The same holds for any $\overline{\mathsf{E}}^e_\varepsilon$. Since for Claim 3.4 we need that none of the events $\underline{\mathsf{E}}^e_\varepsilon$ and $\overline{\mathsf{E}}^e_\varepsilon$ occur, the lemma follows by a union bound over all $2m$ events $\underline{\mathsf{E}}^e_\varepsilon$ and $\overline{\mathsf{E}}^e_\varepsilon$. □

The isolation lemma (Lemma 3.1) together with the discussion about the relation between δ, the difference in cost between the best and second-best integer flow, and Δ, the length of the cheapest cycle in the residual network of the min-cost flow $f^\star$, immediately imply the following upper bound for the probability that Δ is small.

Corollary 3.5. *For any $\varepsilon > 0$, we have $\mathbb{P}(\Delta \leq \varepsilon) \leq 2\varepsilon\phi m$.*

4 Upper Tail Bounds

In this section we prove upper tail bounds for the number of iterations that BP needs to compute maximum-weight matchings and min-cost flows. We show that in the smoothed analysis framework, with high probability a polynomial number of iterations is sufficient, both for MWM and MCF. This result can be interpreted as an indication that instances encountered in practice are unlikely to require a superpolynomial number of iterations.

4.1 Maximum-Weight Matching

We first consider the BP algorithm of Bayati et al. [4], which computes maximum-weight matchings in complete bipartite graphs G in $O(nw^\star/\delta)$ iterations on all instances with a unique optimum. Here $w^\star$ denotes the weight of the heaviest edge and δ denotes the difference in weight between the best and the second-best matching. Even though it is assumed that G is a complete bipartite graph, this is not strictly necessary. If a non-complete graph is given, missing edges can just be interpreted as edges of weight 0.

With Beier and Vöcking's isolation lemma (see Section 3.1), we obtain the following tail bound for the number of iterations needed until convergence when computing maximum-weight perfect matchings in bipartite graphs using BP.

Theorem 4.1. *Let τ be the number of iterations until Bayati et al.'s BP [4] for maximum-weight perfect bipartite matching converges. Then $\mathbb{P}(\tau \geq t) = O(nm\phi/t)$.*

Proof. The number of iterations until BP converges is bounded from above by $O(nw^\star/\delta)$ [4]. The weight of each edge is at most 1, so $w^\star \leq 1$. The upper bound exceeds t only if $\delta \leq O(n/t)$. By Beier and Vöcking's isolation lemma, we have $\mathbb{P}(\delta \leq O(n/t)) \leq O(nm\phi/t)$, which yields the bound claimed. $\square$

This tail bound is not strong enough to yield any bound on the expected running-time of BP for bipartite matching. But it is strong enough to show that BP has smoothed polynomial running-time with respect to the relaxed definition adapted from average-case complexity [5], where it is required that the expectation of the running-time to some power $\alpha > 0$ is at most linear. However, a bound on the expected number of iterations is impossible, and the tail bound proved above is tight up to a factor of $O(m)$ (Section 5).

As discussed in Section 1.1, BP has also been applied to finding maximum-weight (perfect) b-matchings in arbitrary graphs [2,12]. The result is basically that BP converges to the optimal matching if the optimal solution of the relaxation of the corresponding linear program is unique and integral. The number of iterations needed until convergence depends again on "how unique" the optimal solution is. For Bayati et al.'s variant [2], the number of iterations until convergence depends on $1/\delta$, where δ is again the difference in weight between the best and the second-best matching. For Sanghavi et al.'s variant [12], the number of iterations until convergence depends on $1/c$, where c is the smallest rate by which the objective value will decrease if we move away from the optimal solution.

However, the technical problem in transferring the upper bound for bipartite graphs to arbitrary graphs is that the adversary can achieve that, with high probability or even with a probability of 1 (for larger ϕ), the optimal solution of the LP relaxation is not integral. In this case, BP does not converge. Already in the average-case, i.e., for $\phi = 1$, where the adversary has no power at all, the optimal solution of the LP relaxation has some fractional variables with high probability.

Still, we can transfer the results for bipartite matching to both algorithms for arbitrary matching if we restrict the input graphs to be bipartite, since in this case the constraint matrix of the associated LP is totally unimodular.

Theorem 4.2. *Let τ be the number of iterations until Bayati et al.'s [2] or Sanghavi et al.'s [12] BP for general matching, restricted to bipartite graphs as input, converges. Then $\mathbb{P}(\tau \geq t) = O(nm\phi/t)$.*

Proof. For Bayati et al.'s BP algorithm, this follows in the same way as Theorem 4.1 from their bound on the number of iterations until convergence, which is $O(n/\delta)$ [2, Theorem 1].

Sanghavi et al. prove that their variant of BP for general graphs converges after $O(1/c)$ iterations, provided that the LP relaxation has no fractional optimal solutions. Here, c is defined as

$$c = \min_{\hat{x} \neq x^\star \text{ is a vertex of } P} \frac{w \cdot (x^\star - \hat{x})}{\|x^\star - \hat{x}\|_1},$$

where $x^\star$ is the (unique) optimal solution to the relaxation and P is the matching polytope [12, Remark 2].

For any $\hat{x} \neq x^\star$, we have $\|x^\star - \hat{x}\|_1 \leq n$. Furthermore, $w \cdot (x^\star - \hat{x})$ is just the difference in weights between $x^\star$ and $\hat{x}$. Since the input graph is bipartite, all vertices of P are integral. Thus, $w \cdot (x^\star - \hat{x}) \geq \delta$, where (again) δ is the difference in weight between the best and the second-best matching. Thus, $c \geq \delta/n$, which proves the theorem. $\square$

Bayati et al. [2] and Sanghavi et al. [12] have also shown how to compute b-matchings with BP. If all b_i are even, then the unique optimum to the LP relaxation is integral. Thus, we circumvent the problem that the optimal solution might be fractional. Hence, following the same reasoning as above, the probability that BP for b-matching, where all b_i are even, runs for more than t iterations until convergence is also bounded by $O(mn\phi/t)$.

Theorem 4.3. *Let τ be the number of iterations until Bayati et al.'s [2] or Sanghavi et al.'s [12] BP for (perfect) b-matching, where all b_i are even, converges. Then $\mathbb{P}(\tau \geq t) = O(nm\phi/t)$.*

Proof. The theorem follows directly from Beier and Vöcking's isolation lemma and the bounds on the number of iterations of BP by Bayati et al. [2, Theorems 2 and 3] and Sanghavi et al. [12, Theorem 3], as in the proof of Theorem 4.2. $\square$

4.2 Min-Cost Flow

The bound for the probability that Δ is small (Corollary 3.5) and the pseudo-polynomial bound by Gamarnik et al. [7] directly yield a tail bound for the number of iterations that BP for MCF needs until convergence.

Theorem 4.4. *Let τ be the number of iterations until BP for min-cost flow [7] converges. Then $\mathbb{P}(\tau \geq t) = O(n^2 m\phi/t)$.*

Proof. The number of iterations until BP for min-cost flow converges is bounded from above by cLn/Δ for some constant c, where L is the maximum cost of a simple directed path in the residual network for the optimal flow [7, Theorem 4.1]. The cost of each edge is at most 1, so $L \leq n$. The upper bound exceeds t only if $\Delta \leq cn^2/t$. By Corollary 3.5, we have $\mathbb{P}(\Delta \leq cn^2/t) \leq 2cn^2 m\phi/t$, which yields the bound claimed. □

5 Lower Tail Bounds

We show that the expected number of iterations necessary for the convergence of BP for maximum-weight matching (MWM) is unbounded. To do this, we prove a lower tail bound on the number of iterations that matches the upper tail bound as described in Section 4. For our lower bounds we use bipartite graphs since for non-bipartite graphs the convergence of BP is not guaranteed. Our lower bound holds even for a two-by-two complete bipartite graph with edge weights drawn independently and uniformly from the interval $[0, 1]$. In the following analysis, we consider the BP variant introduced by Bayati et al [4]. However, our results can be extended to other versions of BP for matching and min-cost flow [2, 7, 12] in a straightforward way. We discuss such extensions in Section 5.4.

We first discuss the average case, i.e., $\phi = 1$. We consider the average case separately since for our lower bounds in the smoothed setting we need that ϕ is sufficiently large ($\phi \geq 26$). For the average case we obtain a lower tail bound of $\Omega(n/t)$ for the probability that more than t iterations are needed for convergence (Section 5.2). For this lower bound, we use a simple adversarial graph. We leave it as an open problem whether or not the lower bound also holds for the (non-adversarial) complete bipartite graph on n vertices. After that, we consider (non-adversarial) complete bipartite graphs with smoothed weights and prove a lower bound of $\Omega(n\phi/t)$ for the probability that more than t iterations are needed for convergence (Section 5.3). We conclude this section with a discussion of how to extend our results to the other variants of BP for matching and min-cost flow (Section 5.4).

5.1 Computation Tree

For proving the lower bounds, we need the notion of a *computation tree*, which we define analogously to Bayati et al. [2].

Let $G = (V, E)$ be an arbitrary graph with $V = \{v_1, \ldots, v_n\}$. We denote the level-k *computation tree* with the root labeled $x \in V$ by $T^k(x)$. The tree $T^k(x)$ is a weighted rooted tree of height $k + 1$. The root node in $T^0(x)$ has label x, its degree is the degree of x in G, and its children are labeled with the adjacent nodes of x in G. $T^{k+1}(x)$ is obtained recursively from $T^k(x)$ by attaching children to every leaf node in $T^k(x)$. Each child of a former leaf node labeled y is assigned one vertex adjacent to y in G as a label, but the label of the former leaf node's parent is not used. (Thus, the number of children is the degree of y minus 1.) Edges between nodes with label v_i and label v_j in the computation tree have a weight of w_{ij}.

We call a collection Λ of edges in the computation tree $T^k(x)$ a *T-matching* if no two edges of Λ are adjacent in $T^k(x)$ and each non-leaf node of $T^k(x)$ is the endpoint of exactly one edge from Λ. Leaves can be the endpoint of either one or zero edges from Λ. We define $t^k(v_i; r)$ as the weight of a maximum weight T-matching in $T^k(v_i)$ that uses the edge $\{v_i, v_r\}$ at the root.

5.2 Average-Case Analysis

Consider the undirected weighted complete bipartite graph $K_{2,2} = (U \cup V, E)$, where $U = \{u_1, u_2\}$, $V = \{v_1, v_2\}$, and $\{u_i, v_j\} \in E$ for $1 \leq i, j \leq 2$. Each edge $\{u_i, v_j\} = e_{ij}$ has weight w_{ij} drawn independently and uniformly from $[0, 1]$. We define the event E_ε for $0 < \varepsilon \leq \frac{1}{8}$ as the event that $w_{11} \in \left[\frac{7}{8}, 1\right]$, $w_{12} \in \left(\frac{1}{2}, \frac{5}{8}\right]$, $w_{21} \in \left(\frac{5}{8}, \frac{3}{4}\right]$, and $w_{22} \in [w_{12} + w_{21} - w_{11} - \varepsilon, w_{12} + w_{21} - w_{11})$. Consider the two possible matchings $M_1 = \{e_{11}, e_{22}\}$ and $M_2 = \{e_{12}, e_{21}\}$. If event E_ε occurs, then the weight of M_2 is greater than the weight of M_1 and the weight difference is at most ε. In addition, w_{11} is greater than w_{12} and the weight difference is at least $1/4$. See Figure 1 for a graphical illustration of the graph $K_{2,2}$ and the event E_ε.

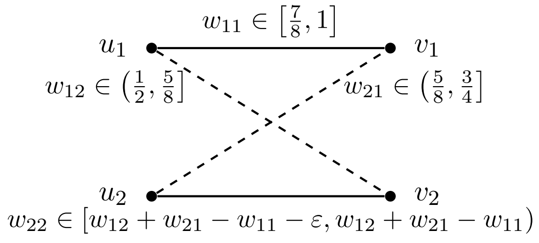
$w_{11} \in \left[\frac{7}{8}, 1\right]$
u_1
v_1
$w_{12} \in \left(\frac{1}{2}, \frac{5}{8}\right]$
$w_{21} \in \left(\frac{5}{8}, \frac{3}{4}\right]$
u_2
v_2
$w_{22} \in [w_{12} + w_{21} - w_{11} - \varepsilon, w_{12} + w_{21} - w_{11})$

Figure 1: If event E_ε occurs, then the weight of the dashed matching $M_2 = \{e_{12}, e_{21}\}$ is greater than the weight of the solid matching $M_1 = \{e_{11}, e_{22}\}$ and the weight difference is at most ε. In addition w_{11} is greater than w_{12} and the weight difference is at least $\frac{1}{4}$.

Lemma 5.1. *The probability of event E_ε is $\varepsilon/8^3$.*

Proof. The intervals in which w_{11}, w_{12}, and w_{21} have to assume values in order for event E_ε to occur all have a length of $1/8$. The interval in which w_{22} has to take a value in order for event E_ε to occur, has a length of ε and it is contained completely in the interval $\left(0, \frac{1}{2}\right]$, since

$$w_{12} + w_{21} - w_{11} - \varepsilon > \frac{1}{2} + \frac{5}{8} - 1 - \frac{1}{8} = 0$$

and

$$w_{12} + w_{21} - w_{11} \leq \frac{5}{8} + \frac{3}{4} - \frac{7}{8} = \frac{1}{2}.$$

Now the probability that w_{11}, w_{12}, w_{21}, and w_{22} all take values in the interval necessary for event E_ε to occur is $\varepsilon/8^3$. □

Lemma 5.2. *If event E_ε occurs, then the belief of node u_1 of $K_{2,2}$ at the end of the $4k$-th iteration is incorrect for all integers $k \leq \frac{1}{8\varepsilon} - 1$.*

Proof. Consider the computation tree $T^{4k}(u_1)$ (see Figure 2). According to Bayati et al. [4, Lemma 1], the belief of node u_1 of $K_{2,2}$ after $4k$ iterations is given by the two-dimensional vector $b_{u_1}^{4k} = \left[2t^{4k}(u_1; 1) \quad 2t^{4k}(u_1; 2)\right]^t$. This means that, after $4k$ iterations, the belief of node u_1 that it should be matched to v_1 is equal to twice the weight of the maximum-weight T-matching of $T^{4k}(u_1)$ that selects edge $\{u_1, v_1\}$ at the root. Analogously, after $4k$ iterations, the belief of node u_1 that it should be matched to v_2 is equal to twice the weight of the

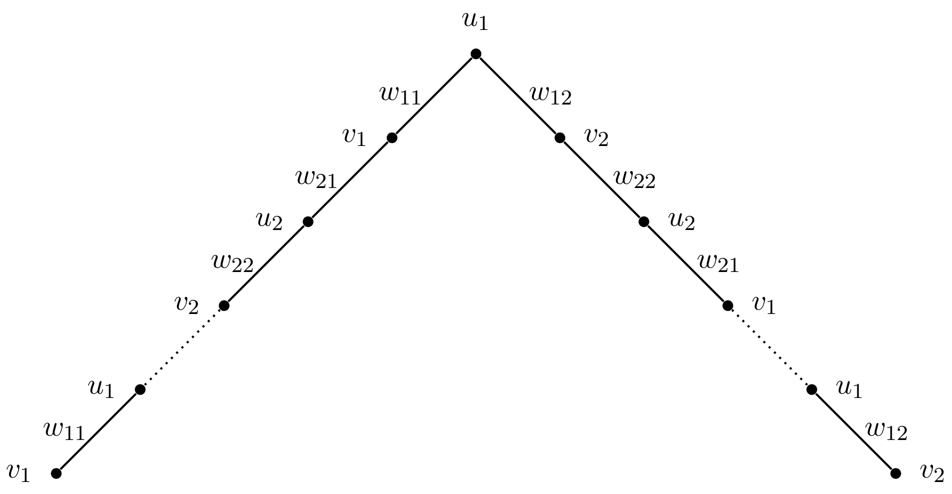

Figure 2: The computation tree $T^{4k}(u_1)$.

maximum-weight T-matching of $T^{4k}(u_1)$ that selects edge (u_1, v_2) at the root. The maximum-weight T-matching $\hat{\Lambda}$ that matches the root node to its child labeled v_2 matches each node labeled u_1 to a node labeled v_2 and each node labeled u_2 to a node labeled v_1, since this is the only possible T-matching that matches the root node to its child labeled v_2. Define $\Lambda^\star$ as the T-matching that matches each node labeled u_1 to a node labeled v_1 and each node labeled u_2 to a node labeled v_2. We show that $\Lambda^\star$ has larger weight than $\hat{\Lambda}$, which implies that the belief at node u_1 after $4k$ iterations is incorrect. We have

$$\begin{aligned} w(\Lambda^\star) - w(\hat{\Lambda}) &= (2k+1)w_{11} + 2kw_{22} - (2k+1)w_{12} - 2kw_{21} \\ &= 2k(w_{11} + w_{22} - w_{12} - w_{21}) + w_{11} - w_{12} \\ &\geq -2k\varepsilon + 1/4. \end{aligned}$$

Now $-2k\varepsilon + 1/4$ is greater than zero if $k \leq \frac{1}{8\varepsilon} - 1$. □

Theorem 5.3. *The probability that BP for MWM needs at least t iterations to converge for $K_{2,2}$ with edge weights drawn independently and uniformly from $[0,1]$ is at least $\frac{1}{ct}$ for some constant $c > 0$.*

Proof. We denote the number of iterations necessary for convergence of BP for MWM by τ. Using Lemma 5.1 and Lemma 5.2, we have

$$\begin{aligned} \mathbb{P}(\tau \geq t) \geq \mathbb{P}(\tau \geq 4\lceil t/4 \rceil) &\geq \mathbb{P}\left(E_{\frac{1}{8(\lceil t/4 \rceil + 1)}}\right) \\ &= \frac{1}{8^4(\lceil t/4 \rceil + 1)} \geq \frac{1}{ct} \end{aligned}$$

for some constant $c > 0$. □

Corollary 5.4. *There exist bipartite graphs on $n \geq 4$ nodes, where n is a multiple of 4, with edge weights drawn independently and uniformly from $[0,1]$, for which the probability that BP for MWM needs at least t iterations to converge is $\Omega\left(\frac{n}{t}\right)$ for $t \geq n/c'$ for some constant $c' > 0$.*

Proof. The bipartite graph consists of $n/4$ copies of $K_{2,2}$ and there are no edges between nodes in different copies of $K_{2,2}$. If BP does not converge in fewer than t iterations for at least one of the $n/4$ copies of $K_{2,2}$, then BP does not converge in fewer than t iterations. This holds since a run of BP on this bipartite graph corresponds to $n/4$ parallel runs of BP on the $n/4$ copies of $K_{2,2}$. Using Theorem 5.3, we have that a constant $c > 0$ exists such that

$$\begin{aligned}\mathbb{P}(\tau < t) &= \big(1 - \mathbb{P}\big(\text{BP needs at least } t \text{ iterations for a particular copy of } K_{2,2}\big)\big)^{n/4}\\ &\le \left(1 - \frac{1}{ct}\right)^{n/4} \le \exp\left(-\frac{n}{4ct}\right) \le 1 - \frac{n}{8ct},\end{aligned}$$

where the second inequality follows from $1 - x \le \exp(-x)$ and the final inequality follows from $\exp(-x) \le 1 - \frac{x}{2}$ for $x \in [0, 1]$ and from $\frac{n}{4ct} \le 1$ which holds if $t \ge \frac{n}{4c}$. □

5.3 Smoothed Analysis

In this section we consider (non-adversarial) complete bipartite graphs $K_{n,n}$ in the smoothed setting. We denote by $X \sim U[a, b]$ that random variable X is uniformly distributed on interval $[a, b]$. In the following, we assume both that $\phi \ge 26$ and $n \ge 2$ and even. Similarly to the average case (Section 5.2), we define the event E_ε^ϕ for $K_{2,2}$ and for $0 < \varepsilon \le 1/\phi$ as the event that $w_{11} \in \big[1 - \frac{1}{\phi}, 1\big]$, $w_{12} \in \big(\frac{23}{26}, \frac{23}{26} + \frac{1}{\phi}\big]$, $w_{21} \in \big(\frac{23}{26}, \frac{23}{26} + \frac{1}{\phi}\big]$, and $w_{22} \in [w_{12} + w_{21} - w_{11} - \varepsilon, w_{12} + w_{21} - w_{11})$. Consider the two possible matchings $M_1 = \{e_{11}, e_{22}\}$ and $M_2 = \{e_{12}, e_{21}\}$. If event E_ε^ϕ occurs, then the weight of M_2 is greater than the weight of M_1 and the weight difference is at most ε. In addition, w_{11} is greater than w_{12} and the weight difference is at least $\frac{3}{26} - \frac{2}{\phi}$.

Lemma 5.5. *There exist probability distributions on $[0, 1]$ for the weights of the edges, whose densities are bounded by $\phi \ge 26$, such that the probability of event E_ε^ϕ is at least $\varepsilon\phi/4$.*

Proof. The intervals in which w_{11}, w_{12}, and w_{21} have to assume values in order for event E_ε^ϕ to occur all have a length of $\frac{1}{\phi}$. We choose the corresponding probability distributions such that they have density ϕ on the corresponding interval and density 0 elsewhere. The interval in which w_{22} has to assume a value in order for event E_ε^ϕ to occur has a length of ε and it is contained completely in the interval $\big[\frac{20}{26} - \frac{1}{\phi}, \frac{20}{26} + \frac{3}{\phi}\big]$, since

$$w_{12} + w_{21} - w_{11} - \varepsilon > \frac{23}{26} + \frac{23}{26} - 1 - \frac{1}{\phi} = \frac{20}{26} - \frac{1}{\phi}$$

and

$$w_{12} + w_{21} - w_{11} \le \left(\frac{23}{26} + \frac{1}{\phi}\right) + \left(\frac{23}{26} + \frac{1}{\phi}\right) - \left(1 - \frac{1}{\phi}\right) = \frac{20}{26} + \frac{3}{\phi}.$$

We choose the probability distribution for w_{22} such that it has density $\frac{\phi}{4}$ on the interval $\big[\frac{20}{26} - \frac{1}{\phi}, \frac{20}{26} + \frac{3}{\phi}\big]$ and 0 elsewhere. Now the probability that w_{11}, w_{12}, and w_{21} take values in the interval necessary for event E_ε^ϕ to occur is 1. For w_{22}, this probability is $\varepsilon\phi/4$. This completes the proof of Lemma 5.5. □

Lemma 5.6. *If event E_ε^ϕ ($\phi \ge 26$) occurs, then the belief of node u_1 at the end of the $4k$-th iteration is incorrect for all integers $k \le \frac{1}{52\varepsilon} - 1$.*

Proof. As in Lemma 5.2, a maximum-weight T-matching that selects the edge labeled $\{u_1, v_1\}$ at the root has greater weight than a maximum-weight T-matching that selects the edge labeled $\{u_1, v_2\}$ at the root for these values of k. $\square$

Analogously to the proof of Theorem 5.3, Lemmas 5.5 and 5.6 above immediately yield a lower bound of $\Omega(\phi/t)$ for the probability that BP will run for at least t iterations.

Our goal in the remainder of this section is to prove an $\Omega(n\phi/t)$ lower bound for the complete bipartite graph. Thus, let us consider the complete bipartite graph $K_{n,n} = (U \cup V, E)$ with $U = \{u_p^j \mid p \in \{1,2\}, j \in \{1,\dots,n/2\}\}$ and $V = \{v_q^j \mid q \in \{1,2\}, j \in \{1,\dots,n/2\}$. Let H^j denote the subgraph induced by $\{u_1^j, u_2^j, v_1^j, v_2^j\}$ for $j \in \{1,\dots,n/2\}$. The role of the subgraphs H^j is the same as the role of the copies of $K_{2,2}$ in the proof of Corollary 5.4. Let e_{pq}^j be the edge connecting u_p^j and v_q^j ($p, q \in \{1,2\}$, $j \in \{1,\dots,n/2\}$). The weight of this edge is w_{pq}^j. We draw edge weights according to the probability distributions

$$\begin{aligned}
w_{11}^j &\sim U\left[1-\frac{1}{\phi}, 1\right], & w_{12}^j &\sim U\left(\frac{23}{26}, \frac{23}{26}+\frac{1}{\phi}\right], \\
w_{21}^j &\sim U\left(\frac{23}{26}, \frac{23}{26}+\frac{1}{\phi}\right], & w_{22}^j &\sim U\left[\frac{20}{26}-\frac{1}{\phi}, \frac{20}{26}+\frac{3}{\phi}\right], \\
w_{ab} &\sim U\left[0, \frac{1}{\phi}\right] \text{ if } u_a \in H^j \text{ and } v_b \in H^k \text{ with } j \neq k.
\end{aligned} \tag{1}$$

We call the edges between nodes in the same induced subgraph H^j *heavy edges*. Edges between nodes in different subgraphs H^j and H^k we call *light edges*. By assumption, we have $\phi \geq 26$. Thus, the weight of any light edge is at most $1/26$, while every heavy edge weighs at least $19/26$.

In contrast to the proof of Corollary 5.4, we now have to make sure that light edges are not used in any computation tree. This allows us to prove the lower bound in a similar way to that in Theorem 5.3 and Corollary 5.4.

Lemma 5.7. *Let $\Lambda^\star$ be the maximum-weight T-matching on the computation tree $T^k(u_i)$. Then $\Lambda^\star$ does not contain any light edges.*

Proof. Assume to the contrary that $\Lambda^\star$ contains a light edge $\{x, y\}$. In that case, x and y are in different subgraphs. The idea of the proof is to construct a path P from one leaf of the computation tree to another leaf that includes edge $\{x, y\}$. Path P alternately consists of edges that are in $\Lambda^\star$ and edges that are not. We show that a new T-matching of greater weight can be constructed by removing from $\Lambda^\star$ the edges in $P \cap \Lambda^\star$ and adding the edges in $P \setminus \Lambda^\star$.

We include the edge labeled $\{x, y\}$ in P and extend P on both sides. We start with node $z_0 = x$ and node $z_0 = y$, respectively, and construct the corresponding part of P as follows:

```
1. for $i = 1, 3, 5, \dots$ do
2.     if $z_{i-1}$ is a leaf node then terminate.
3.     Let $H^k$ be the subgraph that $z_{i-1}$ belongs to.
4.     Let $e_i = \{z_{i-1}, z_i\}$ be the edge incident to $z_{i-1}$ that belongs to the optimal matching
       with respect to $H^k$.
5.     Add $e_i$ to $P$.
```

```
6.    if $z_i$ is a leaf node then terminate.
7.    Let $e_{i+1} = \{z_i, z_{i+1}\}$ be the (unique) edge incident to $z_i$ that belongs to $\Lambda^\star$.
8.    Add $e_{i+1}$ to $P$.
```

It is clear that the procedure can only terminate if it finds a leaf. Moreover, the constructed sequence is alternating. Now we can show that no node will be visited twice. Otherwise, there is an index i such that $z_{i-1} = z_{i+1}$ since P is a path in a tree. However, this can not happen since the sequence is alternating. Therefore, the procedure terminates. Using the previous properties we also obtain that both paths constructed starting with $z_0 = x$ and $z_0 = y$, respectively, are disjoint since $z_1 \notin \{x, y\}$ in both cases. Consequently, we obtain one simple path P connecting two distinct leaf nodes and containing edge $\{x, y\}$.

We now show that the weight of the edges in $P \setminus \Lambda^\star$ is strictly larger than the weight of the edges in $P \cap \Lambda^\star$. For this, let P be of the form $P = (p_0, \dots, p_\ell)$, where ℓ is even and where $\{p_0, p_1\} \in \Lambda^\star$. Let $I \subseteq \{1, \dots, \ell\}$ be the set of indices i for which $\{p_{i-1}, p_i\}$ is a light edge. Clearly, $\{p_{i-1}, p_i\} \in \Lambda^\star$ for each $i \in I$ by construction (see Line 4). Since the light edge $\{x, y\}$ belongs to P we have $I \neq \emptyset$. For $i \in I$, let $P_i = (p_{i-1}, p_i, p_{i+1})$ be the subpath of P of length 2 starting at node p_{i-1}. As $\{p_i, p_{i+1}\}$ is a heavy edge, $w_{p_i p_{i+1}} - w_{p_{i-1} p_i} \geq \left(\frac{20}{26} - \frac{1}{\phi}\right) - \frac{1}{\phi} = \frac{20}{26} - \frac{2}{\phi}$. Therefore, the difference in weight between the edge of P_i that belongs to $\Lambda^\star$ and the other edge is significant.

Now remove all paths P_i from P and consider the subpaths of P (connected components) that remain. There are at most $|I| + 1$ such subpaths P'; each has even length, and they only consist of heavy edges, i.e., all their edges lie in one subgraph H^k where k depends on P'. Consider such a subpath P' and partition it into subpaths $\tilde{P}_j$ of length 4 and, if the length of P' is not a multiple of 4, into one subpath $\hat{P}$ of length 2. The $\Lambda^\star$-edges of $\tilde{P}_j$ form the non-optimal matching on H^k, whereas the other two edges form the optimal matching on H^k. Hence, the total weight of $\tilde{P}_j \cap \Lambda^\star$ is at most the total weight of $\tilde{P}_j \setminus \Lambda^\star$. Only for $\hat{P}$ might we have the case that the weight of $\hat{P} \cap \Lambda^\star$ is larger than the weight of $\hat{P} \setminus \Lambda^\star$, but since both edges are heavy, the difference is at most $1 - \left(\frac{20}{26} - \frac{1}{\phi}\right) = \frac{6}{26} + \frac{1}{\phi}$. Hence, the difference between the total weight of $P \setminus \Lambda^\star$ and the total weight of $P \cap \Lambda^\star$ is at least

$$|I| \cdot \left(\frac{20}{26} - \frac{2}{\phi}\right) - (|I| + 1) \cdot \left(\frac{6}{26} + \frac{1}{\phi}\right) = |I| \cdot \left(\frac{14}{26} - \frac{3}{\phi}\right) - \left(\frac{6}{26} + \frac{1}{\phi}\right) \geq \frac{4}{26} > 0,$$

since $|I| \geq 1$ and $\phi \geq 26$.

We can now construct a T-matching with higher weight than $\Lambda^\star$ by removing the edges in $P \cap \Lambda^\star$ from $\Lambda^\star$ and adding the edges in $P \setminus \Lambda^\star$. This contradicts the assumption that the maximum weight T-matching includes a light edge and proves the lemma. □

Theorem 5.8. *There exist probability distributions on* $[0, 1]$ *for the weights of the edges, whose densities are bounded by* $\phi \geq 26$*, such that the probability that BP for MWM needs at least t iterations to converge for* $K_{n,n}$ *is* $\Omega(n\phi/t)$ *for* $t \geq n\phi/c$ *for some constant* $c > 0$.

Proof. We choose the probability distributions for the edge weights according to (1). Let $\varepsilon = \frac{1}{52(k+1)}$ for $k = 4\lceil t/4 \rceil$ and assume that event E_ε^ϕ occurs for subgraph H^j. In this case, the weight of matching $M_2 = \{e_{12}^j, e_{21}^j\}$ is higher than the weight of matching $M_1 = \{e_{11}^j, e_{22}^j\}$, but at most by the small amount of ε. Consider the computation tree $T^{4k}(u_1^j)$. As in the proof of Lemma 5.2 we know that if the maximum weight T-matching $\Lambda^\star$ on $T^{4k}(u_1^j)$ does not include the edge labeled e_{12}^j at the root, then BP has not yet converged within the first $4k \geq t$ iterations (see Bayati et al. [4, Lemma 1]).

We show that $\Lambda^\star$ does not include edge e_{12}^j. Assume to the contrary that it does. We know from Lemma 5.7 that $\Lambda^\star$ does not contain light edges. Now we use the same procedure to create a path P from one leaf of $T^{4k}(u_1^j)$ to another leaf that contains edge e_{12}^j and alternates between edges from $\Lambda^\star$ and edges from $T^{4k}(u_1^j) \setminus \Lambda^\star$. Since $T^{4k}(u_1^j)$ has height $4k+1$ and since u_1^j is the root of $T^{4k}(u_1^j)$, path P contains exactly $8k+2$ edges, $2k+1$ of which are edges e_{12}^j, $2k+1$ of which are edges e_{11}^j, $2k$ of which are edges e_{22}^j, and $2k$ of which are edges e_{21}^j. The edges e_{12}^j and e_{21}^j are exactly the edges of $P \cap \Lambda^\star$. As in Lemma 5.2, the difference of weight between edges from $P \setminus \Lambda^\star$ and $P \cap \Lambda^\star$ is at least

$$\begin{aligned} w_{11}^j - w_{12}^j - 2k\varepsilon &\geq \left(\left(1 - \frac{1}{\phi}\right) - \left(\frac{23}{26} + \frac{1}{\phi}\right)\right) - \frac{2k}{52(k+1)} \\ &> \frac{3}{26} - \frac{2}{\phi} - \frac{1}{26} \geq 0 \end{aligned}$$

since $\phi \geq 26$. This contradicts the fact that $\Lambda^\star$ is optimal since removing from $\Lambda^\star$ the edges in $P \cap \Lambda^\star$ and adding the edges in $P \setminus \Lambda^\star$ yields a T-matching of heavier weight for $T^{4k}(u_1^j)$.

We have shown that BP does *not* converge within the first t iterations if event E_ε^ϕ occurs for some subgraph H^j. Since there are $n/2$ such subgraphs, we find that the probability that BP for MWM needs at least t iterations to converge for $K_{n,n}$ is $\Omega\left(\frac{n\phi}{t}\right)$ since

$$\begin{aligned} \mathbb{P}(\tau \leq t) &\leq \mathbb{P}\left(E_\varepsilon^\phi \text{ does not occur for any subgraph } H^j\right) \\ &\leq \left(1 - \frac{\varepsilon\phi}{4}\right)^{n/2} \leq \exp\left(-\frac{\varepsilon n \phi}{8}\right) = \exp\left(-\frac{n\phi}{8 \cdot 52 \cdot (4 \cdot \lceil t/4 \rceil + 1)}\right) \\ &\leq 1 - \frac{n\phi}{2 \cdot 8 \cdot 52 \cdot (4 \cdot \lceil t/4 \rceil + 1)} \end{aligned}$$

where the second inequality follows from Lemma 5.5. The third inequality is due to the fact that $1 - x \leq \exp(-x)$, whereas the last inequality stems from $\exp(-x) \leq 1 - \frac{x}{2}$ for $x \in [0,1]$. If $x = \frac{n\phi}{8 \cdot 52 \cdot (4 \cdot \lceil t/4 \rceil + 1)}$ is at most 1, which holds for $t \geq \frac{n\phi}{8 \cdot 52}$, then the correctness follows. $\square$

Note that the lower bound on the probability that BP for MWM converges within t iterations only differs by a factor $O(m)$ from the upper bound as proved in Section 4.1.

5.4 Concluding Remarks

The results presented in Sections 5.2 and 5.3 also hold for other versions of belief propagation for minimum/maximum-weight (perfect) b-matching and min-cost flow [2, 7, 12] applied to the matching problem on bipartite graphs. The number of iterations until convergence differs by no more than a constant factor between these versions of BP. We omit the technical details but provide some comments on how the proofs need to be adjusted.

Some of the versions of BP consider minimum-weight perfect matching [2] or min-cost flow [7] instead of maximum-weight perfect matching. For these versions, we obtain the same results if we have edge weights $\tilde{w}_e = 1 - w_e$ for all edges e.

For some versions of BP [7, 12], the root of the computation tree is an edge rather than a node. If we choose the root of this tree suitably, then we have that the difference in weight between the two matchings M_1 and M_2 of at most ε not only has to "compensate" the weight difference $\Delta w(e_1, e_2)$ between an edge e_1 in M_1 and an edge e_2 in M_2, but the entire weight w_e of an edge e in M_1 or M_2. However, the probability distributions for the edge weights described in Sections 5.2 and 5.3 are chosen such that $\Delta w(e_1, e_2)$ and w_e do not differ more than a constant factor.

Note that our lower bounds for the number of iterations until convergence of BP for MWM in the average case (see Section 5.2) do not contradict the results reported by Salez and Shah [11]. They consider complete bipartite graphs instead of the adversarial graphs that we use. Roughly speaking, Salez and Shah have proved that BP for bipartite matching requires only a constant number of iterations. However, they allow that in expectation a small constant fraction of the nodes are matched to incorrect nodes. It might even be the case that multiple nodes are matched to the same node. In our analysis, we require convergence of the BP algorithm, i.e., each node should be matched to the unique node to which it is matched in the optimal matching.

Even though the graphs we consider in Section 5.2 are different from the graphs Salez and Shah consider, and even though they consider upper bounds on the number of iterations of BP required and we consider lower bounds, some of our results are similar in flavor to their results. Theorem 5.3 only provides a constant lower bound on the number of iterations required to achieve any fixed probability of convergence of BP for a single $K_{2,2}$. By linearity of expectation, we only obtain a constant lower bound for the number of iterations required for convergence of any fixed fraction of the (independent) copies of $K_{2,2}$'s in expectation, as in the paper by Salez and Shah. On the other hand, Corollary 5.4 shows that a constant number of iterations is not sufficient to have convergence for all of the copies of $K_{2,2}$'s.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, 1993.

[2] 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.

[3] 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.

[4] 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.

[5] René Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM Journal in Computing*, 35(4):855–881, 2006.

[6] 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.

[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] Bodo Manthey and Heiko Röglin. Smoothed analysis: Analysis of algorithms beyond worst case. *it – Information Technology*, 53(6):280–286, 2011.

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

[10] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007.

[11] Justin Salez and Devavrat Shah. Optimality of belief propagation for random assignment problem. *Mathematics of Operations Research*, 34(2):468–480, 2009.

[12] 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.

[13] 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.

[14] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[15] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.

[16] 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.

[17] 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.

[18] 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.