Convergence and Correctness of Belief Propagation for Weighted Min-Max Flow *

Guowei Dai[1], Longkun Guo[2], Gregory Gutin[3], Xiaoyan Zhang[1], Zanbo Zhang[4]

[1] Mathematical Science & Institute of Mathematics, Nanjing Normal University, Nanjing, China.
[2] Computer Science, Qilu University of Technology, Jinan, China.
[3] Department of Computer Science, Royal Holloway, University of London, Egham, UK.
[4] Statistics & Mathematics, and Institute of Artificial Intelligence & Deep Learning, Guangdong University of Finance & Economics, Guangzhou, China.

Abstract. In this paper, we investigate the performance of message-passing algorithms for the weighted min-max flow ($\mathcal{WMMF}$) problem which was introduced by Ichirnori et al. (1980). $\mathcal{WMMF}$ was well studied in combinational optimization, as it provides important applications in time transportation problem and the storage management problem. We develop a message-passing algorithm called min-max belief propagation (BP) for determining the optimal solution of $\mathcal{WMMF}$. As the main result of this paper, we prove that for a digraph of size n, BP converges to the optimal solution within $O(n^3)$ time after $O(n)$ iterations if the optimal solution of the underlying min-max flow problem instance is unique. To the best of our knowledge, the fastest polynomial time algorithm for $\mathcal{WMMF}$ runs in essentially $O(n^6)$ time among the known algorithms, where n represents the number of vertices. On the other hand, it is one of a very few instances where BP are proved correct with fully-polynomial running time.

Key words. Belief Propagation, Min-Max BP algorithm, message-passing algorithm, min-max flow.

1. Introduction

As an algorithmic framework, message passing is extremely powerful and has been widely used on various graphical models (GMs). Belief propagation (BP), proposed by Pearl in 1988 [17], is a message-passing heuristic algorithm and has wide applications in the context of variety of disciplines including satisfiability in discrete optimization [1,16], error correcting code in information theory [15,18], and data clustering in machine learning [9]. The great popularity of BP can be attributed to two main reasons. Firstly, it is easy to implement due to its simple and message-passing nature. Secondly, it performs well in many practical applications. The wide scope of application, simplicity, and experimental success of BP has gained a lot of attention recently [16,18,23].

BP is known to converge to the correct solutions on GMs with no cycles [17]. When the underlying graph is a tree, the BP algorithm essentially performs the recursion of

* Financially supported by National Natural Science Foundation of China grants No.11871280, 11971349, 61772005, U1811461 and Natural Science Foundation of Guangdong Province grants No.2020B1515310009.
Correspondence to: Xiaoyan Zhang (email: zhangxiaoyan@njnu.edu.cn)

dynamic programming (DP) leading to a correct solution. Specifically, BP provides a natural parallel iterative version of the DP in which message passing occurs between the variable nodes along edges of the graphical model. Surprisingly, even for GMs with cycles, the BP heuristic performs well in many case, some of which are with rigorous analysis of optimality and convergence [4, 7, 11, 19, 20], while the correctness and convergence properties of BP for general combinatorial optimization problems are still open.

As a major breakthrough, Bayati et al. [4] and Cheng et al. [6] were the first to simplify the BP algorithm independently to obtain essentially the same algorithms for the maximum weight matching (MWM) in a bipartite graph. They established the correctness and convergence of BP algorithm for MWM in pseudo-polynomial time. Sanghavi et al. [19] as well as Bayati et al. [2] generalized the result to the minimum cost b-matching problem on arbitrary graphs and established that BP algorithm converges to the optimal solution, in pseudo-polynomial time, as long as the corresponding linear programming relaxation has no fractional solutions. Furthermore, MWM can be viewed as a special case of the minimum cost flow (MCF) problem. Recently, Gamarnik et al. [11] studied the performance of BP algorithm for finding the optimal solution of MCF and proved that BP algorithm converges to the optimal solution in the pseudo-polynomial time, provided that the optimal solution is unique. Brunsch et al. [5] studied BP algorithm in the framework of smoothed analysis and proved that with high probability the number of iterations needed to compute maximum-weight matchings and min-cost flows is bounded by a polynomial if the weights or costs of the edges are randomly perturbed.

Although BP can converge to the optimum of some combinatorial optimization problems in finite iterations, the running time of them are actually pseudo-polynomial even if the problem itself has other fully polynomial time algorithms, like the MWM and M-CF mentioned above. Gamarnik et al. [11] also presented a simple modification of BP to obtain a fully polynomial time randomized approximation scheme for MCF. However, as they said themselves in [11], the 'near optimal' solution is 'rather fuzzy'. In order to identify the class of optimization problems solvable in fully polynomial time using the BP algorithm, we study the weighted min-max flow ($\mathcal{WMMF}$) problem and develop a min-max BP algorithm for determining the optimal solution of $\mathcal{WMMF}$. As a variant of the maximum flow problems, $\mathcal{WMMF}$ was introduced by Ichirnori et al. [12] and was well studied in combinational optimization [8,10,13], as it provides important applications in time transportation problem [3] and the storage management problem [21].

In this paper, we will investigate the convergence and correctness of the min-max BP algorithm for finding the optimal solution of $\mathcal{WMMF}$ on arbitrary digraphs. As the main result, we establish that our algorithm converges to the optimal solution of $\mathcal{WMMF}$ after at most $n/2$ iterations where n represents the number of vertices, provided that the optimal solution is unique. From the description of min-max BP algorithm, it may seem that each of the messages can be computed in $O(n^2)$ time. Then due to the distributed nature of BP algorithm, the computational cost of the algorithm is $O(n^3)$ in $O(n)$ iterations. As a result, the min-max BP algorithm we developed is a fully polynomial time algorithm. On the one hand, our algorithm is one of a very few instances where BP are proved correct with fully-polynomial running time. On the other hand, to the best of our knowledge, the fastest polynomial time algorithm for $\mathcal{WMMF}$ runs in essentially $O(n^6)$ time [14] among the known algorithms, where n represents the number of vertices. According to our theoretical analysis, it may explain why BP can perform well for most of combinational optimization problems and run fast in practice.

The rest of the paper is organized as follows. In Section 2, we introduce the the weighted min-max flow problem ($\mathcal{WMMF}$). In Section 3, we describe the min-max BP algorithm for $\mathcal{WMMF}$, and state our main result. Proofs of correctness and convergence for our algorithm are given in Section 4. Finally, Section 5 presents the conclusions and directions for future research.

2. Definitions and Problem Statement

Given a weighted digraph (or network) $G = (V, E)$ where V, E denote the set of vertices and arcs, respectively with $|V| = n$, $|E| = m$. To each arcs $e \in E$, assign a nonnegative weight w_e and a positive capacity c_e. For a given source vertex $s \in V$ and a sink $t \in V$, the value of the maximum flow from source s to sink t is denoted by f^*. For any vertex $i \in V$, let E_i^- and E_i^+ be the set of arcs incident inside and out of i, respectively. We assume for simplicity that source s has no arcs incident into s and that sink t has no arcs incident out of t. The weighted min-max flow ($\mathcal{WMMF}$) problem aims to minimize the maximum value of arc-flow (multiplied by arc-weight) among all flows of maximum flow values. Then, the $\mathcal{WMMF}$ on G can be formulated as the following linear program:

$$\min \quad \max_{e \in E} w_e x_e \tag{I}$$

$$\text{s.t.} \quad \sum_{e \in E_i^+} x_e - \sum_{e \in E_i^-} x_e = f_i = \begin{cases} f^*, & i = s \\ -f^*, & i = t \\ 0, & i \in V \setminus \{s, t\} \end{cases} \tag{1}$$

$$0 \le x_e \le c_e, \ \forall \ e \in E \tag{2}$$

where the variables x_e represent flow value assigned to each arc $e \in E$. The constraints (1) state that the difference of out-flow and in-flow equals the value of function f at each node $i \in V$, and the constraints (2) state that flow value on each arc $e \in E$ is at most its capacity c_e.

Next, we will show that (I) can be translated into a factorized optimization problem. Let $E_i = E_i^- \cup E_i^+$ and $x_{E_i} = \{x_e : e \in E_i\}$ for each $i \in V$. Define factor and variable functions ϕ, ψ for each $e \in E, i \in V$, respectively as follows: $\phi_e(x_e) = w_e x_e$ and

$$\psi_i(x_{E_i}) = \begin{cases} 0 & \text{if } \sum_{e \in E_i^+} x_e - \sum_{e \in E_i^-} x_e = f_i, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, we can formulate $\mathcal{WMMF}$ as an unconstrained optimization problem as follows.

$$min \max_{e \in E, i \in V} \left\{ \phi_e(x_e), \psi_i(x_{E_i}) \right\}$$
$$s.t. \ 0 \le x_e \le c_e, \ \forall e \in E.$$

Note that to enable the instance of network flow to be feasible, we assume w.l.o.g. that $|E_i| \ge 2$ for each $i \in V$.

3. BP Algorithm for $\mathcal{WMMF}$

In this section, we will define the message functions, derive the updating rules, develop an algorithm for $\mathcal{WMMF}$, and state our result.

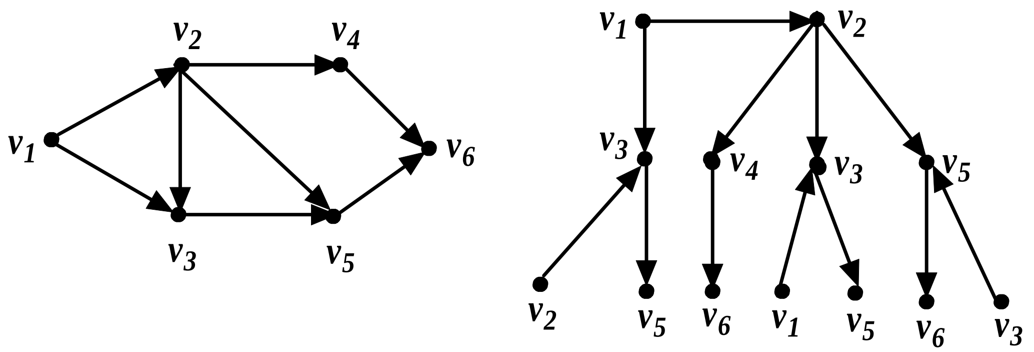

Fig. 1: An example of a 2-level computation tree $T^2_{v_1v_2}$.

3.1. Computation tree

Before define the message functions, we need to give the definition of computation tree first, which has been used in most of the previous analysis of BP algorithms [4,11,19,22]. Let T_a^N denote the N-level computation tree associated with root $a \in E$. In fact, T_a^0 is the tree consists only of arc a. Let $N(u) = \{v : uv \in E \text{ } or \text{ } vu \in E\}$. Due to the local connectivity of the graph, T_a^N can be defined by the following recursive rules:
(a) For any $a = ij \in E$, the arc between vertices labeled i and j in T_a^N is also denoted as a for simplicity and is assigned the same weight w_{ij} as that in G;
(b) T_a^N has a root $a = ij$;
(c) All labels in the set $N(i) \setminus \{j\}$ and $N(j) \setminus \{i\}$ are the children of vertices i and j, respectively;
(d) All labels in the set $N(i') \setminus \{j'\}$ are the children of each non-leaf vertex i' whose parent is j'.
T_a^N is often called the unwrapped tree rooted at a. One can view a run of BP as sending the messages in the computation tree from the leaves to the root, although the messages are sent in the original graph. An example of a computation tree is given in Figure 1.

Given a root a, let $V^o(T_a^N) \subset V(T_a^N)$ denote the set of all the vertices which are not on the N-th level of T_a^N. Then the problem of $\mathcal{WMMF}$ on T_a^N can be formulated as $\mathcal{WMMF}_a^N$:

$$\min \quad \max_{e \in E(T_a^N)} w_e y_e \qquad (II)$$

$$\text{s.t.} \quad \sum_{e \in E_k^+(T_a^N)} y_e - \sum_{e \in E_k^-(T_a^N)} y_e = f_k, \quad \forall\ k \in V^o(T_a^N) \qquad (3)$$

$$0 \le y_e \le c_e, \qquad \forall\ e \in E(T_a^N). \qquad (4)$$

3.2. Algorithm and result

For each arc $e = ij$ on the computation tree, define a message function $m_{e\to j}(x_e)$ on the subtree below e with e included. Let the function $m_{e\to j}(x_e)$ return the maximum value of arc-flow (multiplied by arc-weight) of the $\mathcal{WMMF}$ on that subtree. Similarly, define the message function $m_{i\to e}(x_e)$ which returns the maximum value of arc-flow (multiplied by arc-weight) of the $\mathcal{WMMF}$ on the subtree below i including i but not e. Due to the

Algorithm 1 Min-Max BP for $\mathcal{WMMF}$

1: Initialize $t = 0$, message $m^0_{i\to e}(x_e) = w_e x_e$ for each $e = ij \in E$.
2: **for** $t = 1, 2, ..., N$ **do**
3: For each $e = ij \in E$ update messages as follows:

$$m^t_{e\to j}(x_e) = \max\left\{\phi_e(x_e), m^{t-1}_{i\to e}(x_e)\right\},$$

$$m^t_{i\to e}(x_e) = \min_{x_{E_i \setminus e}} \left\{ \max\{\psi_i(x_{E_i}), \max_{e'\in E_i\setminus e} m^t_{e'\to i}(x_{e'})\}\right\}.$$

4: $t := t + 1$
5: **end for**
6: For each $e = ij \in E$, set the belief function as $b^N_e(x_e) = \max\left\{m^N_{e\to i}(x_e), m^N_{e\to j}(x_e)\right\}$.
7: Calculate the belief estimate by finding $\widehat{x}^N_e \in \arg\min_{0\le x_e\le c_e} b^N_e(x_e)$ for each $e \in E$.
8: Return $\widehat{x}^N = \{\widehat{x}^N_e \mid e \in E\}$ as an estimation of the optimal solution.

nature of tree structure, these two message functions can be recursively defined as follows: for any arc $e = ij$,

$$m_{e\to j}(x_e) = \max\left\{\phi_e(x_e), m_{i\to e}(x_e)\right\}, \tag{5}$$

$$m_{i\to e}(x_e) = \min_{x_{E_i\setminus e}} \left\{ \max\{\psi_i(x_{E_i}), \max_{e'\in E_i\setminus e} m_{e'\to i}(x_{e'})\}\right\}. \tag{6}$$

Using (5)-(6), starting from leaves, the message functions $m_{e\to j}(x_e)$ and $m_{i\to e}(x_e)$ can be computed for all $e \in E, i \in V$. Then, the update messages for each vertex and arc is as follows:

$$m^t_{e\to j}(x_e) = \max\left\{\phi_e(x_e), m^{t-1}_{i\to e}(x_e)\right\},$$

$$m^t_{i\to e}(x_e) = \min_{x_{E_i\setminus e}} \left\{ \max\{\psi_i(x_{E_i}), \max_{e'\in E_i\setminus e} m^t_{e'\to i}(x_{e'})\}\right\}.$$

Finally, combine the messages $m_{e_r\to i_r}(x_{e_r})$ and $m_{e_r\to j_r}(x_{e_r})$ at the root arc $e_r = i_r j_r$, we can derive the estimation of $belief$ at the end of iteration t on the computation tree $T^t_{e_r}$ as

$$b^t_{e_r}(x_{e_r}) = \max\left\{m^t_{e_r\to i_r}(x_{e_r}), m^t_{e_r\to j_r}(x_{e_r})\right\}.$$

The parallel algorithm called min-max BP for solving $\mathcal{WMMF}$ is described in detail as Algorithm 1.

Next, we will state our result, the proof of which is presented in Section 4.

Theorem 1 *For a digraph G of order n, if the $\mathcal{WMMF}$ on G has a unique optimal solution x^*, then Algorithm 1 converges to x^* within $\frac{n}{2}$ iterations, i.e., $\widehat{x}^N = x^*$ after $N \ge \frac{n}{2}$ iterations.*

4. Proof of Correctness and Convergence

In this section, we will establish the convergence of BP to the optimal solution of the $\mathcal{WMMF}$ under the assumption of the uniqueness of the optimal solution, namely we shall prove Theorem 1. Note that our strategy is somewhat similar to that of [11], but the technical details are quite different.

Lemma 2 *Let $\widehat{x}_a^N$ be the value of the output of the BP algorithm at the end of iteration N on arc $a \in E$. Then there exists an optimal solution y^* of $\mathcal{WMMF}_a^N$ such that $y_a^* = \widehat{x}_a^N$ where a is the root of T_a^N.*

Proof. Let $a = ij$ be the root of T_a^N. By definition, T_a^N has two components connected by the arc a. Denote the component containing i by C and $T_{a\to j}^N$ denotes C with arc a (indeed $T_{a\to j}^N$ is a tree). Let $V^0(T_{a\to j}^N)$ be the set of all the vertices which are not on the N-th level of $T_{a\to j}^N$. Define $\mathcal{WMMF}_{a\to j}^N(z)$ as follows.

$$\begin{aligned} \min \quad & \max_{e\in E(T_{a\to j}^N)} w_e y_e && (\mathcal{WMMF}_{a\to j}^N(z)) \\ \text{s.t.} \quad & \sum_{e\in E_k^+(T_{a\to j}^N)} y_e - \sum_{e\in E_k^-(T_{a\to j}^N)} y_e = f_k, \quad \forall\, k \in V^o(T_{a\to j}^N) \\ & y_a = z \\ & 0 \le y_e \le c_e, \qquad \forall\, e \in E(T_{a\to j}^N). \end{aligned}$$

Now, we show that under the BP algorithm the value of $m_{a\to j}^N(z)$ is the same as the weight of the optimal assignment for $\mathcal{WMMF}_{a\to j}^N(z)$. This can be established inductively. When $N = 1$, the statement is easy to be checked. For $N > 1$ and each $b \in E_i\backslash a$ with $b = wi$ (or iw), let $T_{b\to i}^{N-1}$ be the subtree of $T_{a\to j}^N$ that includes b and does not include i. Consider the subproblem $\mathcal{WMMF}_{b\to i}^{N-1}(z)$ as follows.

$$\begin{aligned} \min \quad & \max_{e\in E(T_{b\to i}^{N-1})} w_e y_e && (\mathcal{WMMF}_{b\to i}^{N-1}(z)) \\ \text{s.t.} \quad & \sum_{e\in E_k^+(T_{b\to i}^{N-1})} y_e - \sum_{e\in E_k^-(T_{b\to i}^{N-1})} y_e = f_k, \quad \forall\, k \in V^o(T_{b\to i}^{N-1}) \\ & y_b = z \\ & 0 \le y_e \le c_e, \qquad \forall\, e \in E(T_{b\to i}^{N-1}). \end{aligned}$$

By induction hypothesis, it is easy to see that the value of $m_{b\to i}^{N-1}(z)$ equals the weight of the solution of $\mathcal{WMMF}_{b\to i}^{N-1}(z)$. Given this hypothesis and the relation of sub-tree $T_{b\to i}^{N-1}$ for all $b \in E_i\backslash a$ with $T_{a\to j}^N$, it follows that the problem $\mathcal{WMMF}_{a\to j}^N(z)$ is equivalent to

$$\begin{aligned} \min \quad & \max\left\{w_a z, \max_{b\in E_i\backslash a} m_{b\to i}^{N-1}(y_b)\right\} \\ \text{s.t.} \quad & \sum_{e\in E_i^+(T_{a\to j}^N)} y_e - \sum_{e\in E_i^-(T_{a\to j}^N)} y_e = f_i, \\ & y_a = z \\ & 0 \le y_b \le c_b,\ \forall\, b \in E_i\backslash a. \end{aligned}$$

This is exactly the same as the relation between $m_{a\to j}^N(z)$ and message function $m_{b\to i}^{N-1}(\cdot)$ for $b \in E_i\backslash a$ as

$$m_{a\to j}^N(z) = \max\left\{w_a z, \max_{b\in E_i\backslash a} m_{g\to i}^{N-1}(y_b)\right\}.$$

That is, $m_{a\to j}^N(z)$ is exactly the same as the weight of optimal assignment of $\mathcal{WMMF}_{a\to j}^N(z)$. Using this equivalence, we will complete the proof of Lemma 2.

For given $a = ij$ with $0 \le z \le c_a$, the problem $\mathcal{WMMF}_a^N(z)$ is equivalent to

$$\begin{aligned}
\min \quad & \max\Big\{\max_{e\in E(T_{a\to i}^N)} w_e y_e, \max_{e\in E(T_{a\to j}^N)} w_e y_e\Big\} \\
\text{s.t.} \quad & \sum_{e\in E_k^+(T_a^N)} y_e - \sum_{e\in E_k^-(T_a^N)} y_e = f_k, \qquad \forall\, k \in V^o(T_a^N) \cap (V^o(T_{a\to k}^N) \cup V^o(T_{a\to j}^N)) \\
& 0 \le y_e \le c_e, \qquad \forall\, e \in E(T_{a\to i}^N) \cup E(T_{a\to j}^N).
\end{aligned}$$

That is, the maximum value of arc-flow (multiplied by arc-weight) of an optimal solution of the problem $\mathcal{WMMF}_a^N(z)$ equals $\max\Big\{m_{a\to i}^N(z), m_{a\to j}^N(z)\Big\}$, for any $0 \le z \le c_a$. Now the claim of Lemma 2 follows immediately. □

It is clear that Lemma 2 establishes the relation between BP algorithm and computation tree T_a^N. Next, we shall show the correctness and convergence of min-max BP algorithm for $\mathcal{WMMF}$ as follows.

Proof of Theorem 1: To the contrary, we suppose that there is an arc $e_0 \in E$ such that $\widehat{x}_{e_0}^N \ne x_{e_0}^*$ where $N > \frac{n}{2}$. We assume w.l.o.g. $\widehat{x}_{e_0}^N > x_{e_0}^*$. Then, by Lemma 2, there is an optimal solution y^* of $\mathcal{WMMF}_{e_0}^N$ such that $y_{e_0}^* > x_{e_0}^*$.

Let $e_0 = uv$ be the root of the computation tree $T_{e_0}^N$ as above. Since x^* and y^* are the feasible solutions of $\mathcal{WMMF}$ and $\mathcal{WMMF}_{e_0}^N$, respectively,

$$f_u = x_{e_0}^* + \sum_{e\in E_i^+\setminus e_0} x_e^* - \sum_{e\in E_i^-} x_e^*, \tag{7}$$

$$f_u = y_{e_0}^* + \sum_{e\in E_i^+(T_{e_0}^N)\setminus e_0} y_e^* - \sum_{e\in E_i^-(T_{e_0}^N)} y_e^*. \tag{8}$$

Due to the inequality $y_{e_0}^* > x_{e_0}^*$, using (7)-(8), there exists an arc $e_1 \ne e_0$ incident to u such that $y_{e_1}^* > x_{e_1}^*$ if e_1 and e_0 have the same orientation at u (e_1 is ingoing from u and e_0 is outgoing from u), or $y_{e_1}^* < x_{e_1}^*$, otherwise. Similarly, we can find arc $e_{-1} \ne e_0$ incident to v such that $y_{e_{-1}}^* > x_{e_{-1}}^*$ if e_{-1} and e_0 have the same orientation at v, or $y_{e_{-1}}^* < x_{e_{-1}}^*$, otherwise. A similar argument can be applied recursively utilizing the inequalities between value of components of x^*, y^* and the equality constraint (1), (3) in linear programming (I) and (II) on each vertex, respectively. Continuing further all the way down to the leaves of $T_{e_0}^N$, we will finally obtain a path denoted by $P = \{e_{-N}, ..., e_{-1}, e_0, e_1, ..., e_N\}$ such that for $-N \le i \le N$,

$$\begin{aligned}
y_{e_i}^* > x_{e_i}^* \quad &\Leftrightarrow \quad \text{both } e_i \text{ and } e_0 \text{ have the same orientation,} \\
y_{e_i}^* < x_{e_i}^* \quad &\Leftrightarrow \quad \text{both } e_i \text{ and } e_0 \text{ have the opposite orientation.}
\end{aligned}$$

According to the definitions of x^* and y^*, such a path is guaranteed to exist. Figure 2 depicts an example of such a path given by dashed arcs.

Let $\max_E(x) = \max\{w_e x_e : e \in E\}$ where x is a feasible solution of the $\mathcal{WMMF}$. If a feasible solution y' of $\mathcal{WMMF}_{e_0}^N$ can be obtained by modifying y^* such that

$$\max_{E(T_{e_0}^N)}(y^*) > \max_{E(T_{e_0}^N)}(y'), \tag{9}$$

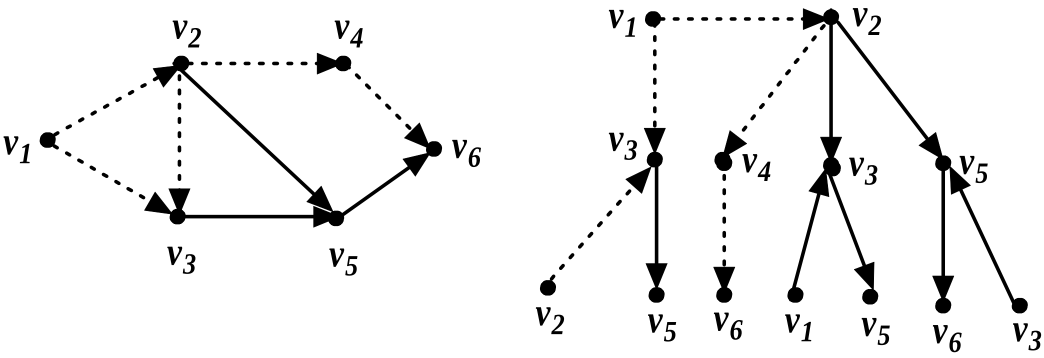

Fig. 2: An example of the path P on a computation tree $T^2_{v_1v_2}$ with dashed arcs.

then a contradiction arises to the the optimality of y^*. Define $\mathcal{A} = \{e \in P : y^*_e > x^*_e\}$ and $\mathcal{B} = \{e \in P : y^*_e < x^*_e\}$. As both $\mathcal{A}$ and $\mathcal{B}$ are finite sets, there exists $\varepsilon > 0$ such that $y^*_e - \varepsilon \geq x^*_e$ for any $e \in \mathcal{A}$ and $y^*_e + \varepsilon \leq x^*_e$ for any $e \in \mathcal{B}$. Let

$$y'_e = \begin{cases} y^*_e - \varepsilon, & e \in \mathcal{A} \\ y^*_e + \varepsilon, & e \in \mathcal{B} \\ y^*_e, & \text{otherwise.} \end{cases}$$

Then $y'_e = y^*_e - \varepsilon \geq x^*_e \geq 0$ for any $e \in \mathcal{A}$ and $y'_e = y^*_e + \varepsilon \leq x^*_e \leq c_e$ for any $e \in \mathcal{B}$, which satisfies all the capacity constraints of (II). Furthermore, for any vertex i on $V^o(T^N_{e_0})$, let e' and e'' be the arcs incident to i and belonging to $E(T^N_{e_0})$. Then we have that for any vertex i on $V^o(T^N_{e_0})$,

- if e' and e'' have the same orientation as e_0, then $e', e'' \in \mathcal{A}$ and

$$\sum_{e \in E^+_i(T^N_{e_0})} y'_e - \sum_{e \in E^-_i(T^N_{e_0})} y'_e = (-\varepsilon + \sum_{e \in E^+_i(T^N_{e_0})} y^*_e) - (-\varepsilon + \sum_{e \in E^-_i(T^N_{e_0})} y^*_e) = f_i.$$

- if e' and e'' have the opposite orientation as e_0, then $e', e'' \in \mathcal{B}$ and

$$\sum_{e \in E^+_i(T^N_{e_0})} y'_e - \sum_{e \in E^-_i(T^N_{e_0})} y'_e = (\varepsilon + \sum_{e \in E^+_i(T^N_{e_0})} y^*_e) - (\varepsilon + \sum_{e \in E^-_i(T^N_{e_0})} y^*_e) = f_i.$$

- if e' has the same orientation and e'' has the opposite orientation as e_0, then

$$\sum_{e \in E^+_i(T^N_{e_0})} y'_e - \sum_{e \in E^-_i(T^N_{e_0})} y'_e = \sum_{e \in E^+_i(T^N_{e_0})} y^*_e - (-\varepsilon + \varepsilon + \sum_{e \in E^-_i(T^N_{e_0})} y^*_e) = f_i.$$

  or

$$\sum_{e \in E^+_i(T^N_{e_0})} y'_e - \sum_{e \in E^-_i(T^N_{e_0})} y'_e = (-\varepsilon + \varepsilon + \sum_{e \in E^+_i(T^N_{e})} y^*_{e_0}) - \sum_{e \in E^-_i(T^N_{e_0})} y^*_e = f_i.$$

This implies y' satisfies all the other equality constraints of (II), and thus y' is a feasible solution of $\mathcal{WMMF}^N_{e_0}$.

Next, we only need to show that $\max_{E(T_{e_0}^N)}(y^*) > \max_{E(T_{e_0}^N)}(y')$. Since the only difference between y^* and y' is the arcs in P, it is sufficient to show that $\max_{E(P)}(y^*) > \max_{E(P)}(y')$. Suppose on the contrary that

$$\max_{E(P)}(y^*) \le \max_{E(P)}(y'). \tag{10}$$

Note that P can be decomposed into a simple undirected path and some simple undirected cycles in G. Let $\mathscr{C}$ denote the set of these simple undirected cycles. Since the length of the path P after N iterations is $2N+1$ and thus $\frac{2N+1}{n} \ge \frac{n+1}{n} > 1$, there must exist at least one undirected cycle $C \in \mathscr{C}$. Let

$$x'_e = \begin{cases} x^*_e + \varepsilon, & e \in \mathcal{A} \cap E(C) \\ x^*_e - \varepsilon, & e \in \mathcal{B} \cap E(C) \\ x^*_e, & \text{otherwise.} \end{cases}$$

By a similar argument as y', it is not difficult to show that x' is a feasible solution of $\mathcal{WMMF}$ on the original graph G. According to the definitions of x' and y', it follows from (10) that

$$\max_{E(P)}(x') \le \max_{E(P)}(x^*). \tag{11}$$

Due to (11), we have that

$$\max_{E}(x^*) \ge \max_{E(P)}(x^*) \ge \max_{E(P)}(x') \ge \max_{E(C)}(x').$$

Since the only difference between x^* and x' is the arcs in C, by the definitions of x', it is to see that x' is a feasible solution of $\mathcal{WMMF}$ such that $\max_E(x') \le \max_E(x^*)$. This leads to a contradiction that x^* is the unique optimal solution of $\mathcal{WMMF}$ which completes the proof. □

5. Conclusion

As a distributed, message-passing algorithm, Belief Propagation (BP) algorithm has been widely used in areas like modern statistics, coding theory, combinatorial optimization and artificial intelligence. Despite empirical successes of BP algorithm in many practical scenarios, the theoretical understanding of the performance of BP algorithm remains far from complete. In this paper, we derive a min-max BP algorithm for the weighted min-max flow ($\mathcal{WMMF}$) problem and analyze the correctness and convergence of the algorithm presented. We prove that min-max BP algorithm converges to the optimal solution with fully-polynomial running time, provided that the optimal solution is unique. Moreover, based on the research results and contributions of Gamarnik et al. [11], a simple modification of BP algorithm can be provided to obtain a fully polynomial-time randomized approximation scheme (FPRAS) without requiring the uniqueness of the optimal solution. Finally, it remains open for future research to study more general optimization problems and viability of BP algorithms for them.

References

1. D. Achlioptas and F. Ricci-Tersenghi, On the solution-space geometry of random constraint satisfaction problems, in Proceedings of the thirty-eighth annual ACM symposium on theory of computing, pp. 130-139, ACM, 2006.

2. M. Bayati, C. Borgs, J. Chayes, and R. Zecchina, Belief propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions, SIAM Journal on Discrete Mathematics, vol.25, pp. 989-1011, 2011.

3. R.E. Burkard, A general hungarian method for the algebraie transpertation problem, Discrete Mathematics, vol.22, pp. 219-232, 1978.

4. M. Bayati, D. Shah, and M. Sharma, Max-product for maximum weight matching: convergence, correctness, and LP duality, IEEE Transactions on Information Theory, vol.54, pp. 1241-1251, 2008.

5. T. Brunsch, K. Cornelissen, B. Manthey, and H. Rӧglin, Smoothed Analysis of Belief Propagation for Minimum-Cost Flow and Matching, WALCOM: Algorithms and Computation, Springer Berlin Heidelberg, pp. 182-193, 2012.

6. Y. Cheng, M. Neely, and K. M. Chugg, Iterative message passing algorithm for bipartite maximum weighted matching, in Proceedings of IEEE International Symposium Information Theory, Cambridge, pp. 1934-1938, 2006.

7. G. Dai, F. Li, Y. Sun, D. Xu, and X. Zhang, Convergence and correctness of belief propagation for the Chinese postman problem, Journal of Global Optimization, vol.75, pp. 813-831, 2019.

8. H. A. Eiselt and M. Gendreau, An optimal algorithm for weighted minimax flow centers on trees, Transportation Science, vol.25, pp. 314-316, 1991.

9. B.J. Frey and D. Dueck, Clustering by passing messages between data points, Science, vol.315, pp. 972-976, 2007.

10. S. Fujishige, A. Nakayama, and W. Cui, On the equivalence of the maximum balanced flow problem and the weighted minimax flow problem, Operations Research Letters, vol.5, pp. 207-209, 1986.

11. D. Gamarnik, D. Shah, and Y. Wei, Belief propagation for min-cost network flow: convergence & correctness, Operations research, vol.60, pp. 410-428, 2012.

12. T. Ichimori, M. Murata, H. Ishii, and T. Nishida, Minimax cost flow problem, Technol. Repts. of the Osaka Univ., vol.30, pp. 39-44, 1980.

13. T. Ichirnori, H. Ishii, and T. Nishida, Finding the weighted minimax flow in a polynomial time, Journal of the Operations Research Society of Japan, vol.23, pp. 268-271, 1980.

14. T. Ichirnori, H. Ishii, and T. Nishida, Weighted minimax real-valued flows,Journal of the Operations Research Society of Japan, vol.24, pp.52-59, 1981.

15. M. Mézard, Passing messages between disciplines, Science, vol.301, pp. 1685-1686, 2003.

16. M. Mézard, G. Parisi and R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, Science, vol.297, pp. 812-815, 2002.

17. J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible reasoning, CA: Morgan Kaufmann, 1988.

18. T. Richardson and R. Urbanke, The capacity of low-density parity check codes under message-passing decoding, IEEE Transactions on Information Theory, vol.47, pp. 599-618, 2001.

19. S. Sanghavi, D. Malioutov, and A. Willsky, Belief propagation and LP relaxation for weighted matching in general graphs, IEEE Transactions on Information Theory, vol.54, pp. 2203-2212, 2011.

20. S. Sanghavi, D. Shah, A. S. Willsky, Message passing for maximum weight independent set, IEEE Transactions on Information Theory, vol.55, pp. 4822-4834, 2009.

21. D.F. Stanat and G.A. Mag*ó*, Minimizing maximum flows in linear graphs, Networks, vol.9, pp. 333-361, 1979.

22. Y. Weiss and W. Freeman, On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs, IEEE Transactions on Information Theory, vol.47, pp.736-744, 2001.

23. J. Yedidia, W. Freeman, and Y. Weiss, Constructing free-energy approximations and generalized belief propagation algorithms, IEEE Transactions on Information Theory, vol.51, pp.2282-2312, 2005.