Belief Propagation for Minimum Convex-Cost Integral Network Flow

Jiaming Xu, Zheng Lu
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
Email: jxu@mail.utexas.edu, zhenglu1986@gmail.com

I. INTRODUCTION

Belief propagation (BP) is an iterative, message-passing algorithm for finding the mode of a discrete probability distribution specified by a graphical model. Recently, belief propagation (BP) is widely used to solve discrete optimization problem by reformulating the optimization problem as a mode-finding problem. In [1], BP is proposed to find maximum weight matching (MWM) in bipartite graph. It is shown that BP converges to the correct solution as long as the optimal MWM is unique. This work is generalized in [2] to consider MWM in general graph. It is shown that the performance of BP is exactly characterized by the natural linear programming (LP) relaxation of the problem. In particular, they established that BP converges to the correct solution if and only if the LP relaxation has no fractional optima. However, only one-side relation between BP and LP relaxation is found in the context of Maximum weight independent set [3], which says that if BP converges then the fixed point is the correct solution and LP relaxation is tight. More recently, BP algorithm is formulated to solve continuous optimization problem such as capacitated minimum-cost network Flow (MCF) problem [4]. Unlike the most previous work, the messages of BP in this problem are shown to be convex piecewise-linear functions and it is proved that BP converges to the optimal solution provided that the optimal solution is unique.

Inspired by the success of the BP in solving the MCF problem problem, in this paper we will study the minimum convex-cost integral network flow problem. Unlike the MCF problem studied previously in [4] where the flows may have real values, we only allow integral flows here.
Our motivation for doing so is two-fold: firstly, integral network flow is sufficient to model the practical situation such as routing in network. Secondly, for integral network flow, the messages in BP algorithm are vectors of integers and thus easy to implement in practical.

**Contributions:** Our main result in this paper is the convergence and correctness of BP algorithm for minimum convex-cost integral network flow problem provided that it has unique optimum. This result is surprising in the sense that previously BP algorithm is believed to converge only in graphs with no or small number of cycles; while our result suggests that BP algorithm might be used to solve various optimization problems in graph as well. We also consider the case of continuous relaxation and show that it always has integral optimal solution. Thus we establish the relations between our results and previous results in [4].

II. **Model and BP Algorithm**

In this section, we formally describe the convex-cost network flow problem. Consider a directed graph \( G = (V, E) \), where \( V \) and \( E \) denote the vertex set and edge set respectively with \( |V| = n \) and \( |E| = m \). For any vertex \( v \in V \), let \( E_v \) denote the set of edges incident to \( v \), and for any edge \( e \in E \), let \( \Delta(v, e) = 1 \) if \( e \) is an out-edge of \( v \) and \( \Delta(v, e) = -1 \) if \( e \) is an in-edge of \( v \). The convex-cost network flow problem on \( G \) is to route the network flow at the minimum cost and it can be formulated as an optimization problem:

\[
\begin{align*}
\min \quad & \sum_{e \in E} c_e(x_e) \\
\text{s. t.} \quad & \sum_{e \in E_v} \Delta(v, e)x_e = d_v, \forall v \in V \quad \text{(flow conservation)} \\
& x_e \in \{0, 1, \ldots, u_e\} \quad \text{(capacity)},
\end{align*}
\]

where \( x_e \) is the integral flow on edge \( e \), \( c_e(x_e) \) is an arbitrary convex cost function, \( d_v \) is the integral exogenous flow to vertex \( v \), and \( u_e \) is the integral capacity of edge \( e \). Note that in this paper, the flow variables \( x_e \) are assumed to be integral. The reason is that it is convenient to implement the BP algorithm and it is also in accordance to the practice.

In order to apply BP algorithm, we now formulate the convex-cost network flow problem as a MAP estimation problem by constructing a suitable probability distribution. Associate a discrete random variable \( x_e \in \{0, 1, \ldots, u_e\} \) with each edge \( e \), and consider the following probability
distribution,

\[ p(x) = \frac{1}{Z} \prod_{v} \psi_v(x_{E_v}) \prod_{e} \exp(-c_e(x_e)), \]  

where \( \psi_v(x_{E_v}) \) is given by

\[ \psi_v(x_{E_v}) = \begin{cases} 
1 & \text{if } \sum_{e \in E_v} \Delta(v, e)x_e = d_v \\
0 & \text{else} 
\end{cases}. \]  

It is easy to see that if flow \( x \) satisfies the flow conservation constraints, \( p(x) \propto \exp(-\sum_e c_e(x_e)) \).

Therefore, the min-cost solution of the convex-cost network flow problem corresponds to the MAP estimate of \( p(x) \). We can then apply BP algorithm to solve the convex-cost network flow problem by the following steps,

1. Initialize \( t = 0 \). For each \( e \in E \), suppose \( e = (v, w) \). Initialize messages \( m_{e \rightarrow v}^0(x_e) = 0, \forall x_e \in \{0, 1, \ldots, u_e\} \) and \( m_{e \rightarrow w}^0(x_e) = 0, \forall x_e \in \{0, 1, \ldots, u_e\} \).

2. for \( t = 1, 2, \ldots, N \) do

   3. for \( \forall e \in E \), update the messages as:

   \[ m_{e \rightarrow v}^t(x_e) = \exp(-c_e(x_e)) \times \max_{z \in \mathbb{R}^{|E|}, z_e = x_e} \left\{ \psi_v(z) \prod_{\tilde{e} \in E_{w} \setminus e} m_{\tilde{e} \rightarrow w}^{t-1}(z_{\tilde{e}}) \right\}, \]  

      \( \forall x_e \in \{0, 1, \ldots, u_e\} \)  

   \[ m_{e \rightarrow w}^t(x_e) = \exp(-c_e(x_e)) \times \max_{z \in \mathbb{R}^{|E|}, z_e = x_e} \left\{ \psi_{w}(z) \prod_{\tilde{e} \in E_{v} \setminus e} m_{\tilde{e} \rightarrow v}^{t-1}(z_{\tilde{e}}) \right\}, \]  

      \( \forall x_e \in \{0, 1, \ldots, u_e\} \).

4. \( t\leftarrow t+1 \)

5. end for

6. for \( \forall e \in E \), the belief at edge \( e \) is given by

\[ b_e^N(x_e) = m_{e \rightarrow v}^N \times m_{e \rightarrow w}^N \times \exp(c_e(x_e)), \]  

      \( \forall x_e \in \{0, 1, \ldots, u_e\} \).
7. Find \( \hat{x}^N_e = \arg \max b^N_e(x_e) \) for each \( e \in E \) and return \( \hat{x}^N_e \) as the guess of the optimal solution of the convex-cost network flow problem.

The BP algorithm above is derived from the BP on factor graph, in which there are two kinds of messages, the ones passed from variable nodes to factor nodes \( (m_{e\to v}) \) and the ones passed from factor nodes to variable nodes \( (m_{v\to e}) \). In our problem, since each variable (edge) only connects with two factors (nodes), so we can simply incorporate the message \( m_{v\to e} \) into the message \( m_{e\to v} \) and obtain the above BP algorithm.

III. CONVERGENCE AND CORRECTNESS

A. Residual Network

Before formally stating the theorem of convergence and correctness, let us first give the definition of the residual network, which is useful in the proof of convergence. Define \( G(x) \) to be the residual network of \( G \) and flow \( x \) as follows:

(1) \( G(x) \) has the same vertex set as \( G \).

(2) For \( \forall e = (v, w) \in E \), if \( x_e < u_e \), then \( e \) is an arc in \( G(x) \) with cost \( c_e^x = c_e(x_e + 1) - c_e(x_e) \), and if \( x_e > 0 \), then there is an arc \( e' = (w, v) \) in \( G(x) \) with cost \( c_e^{x'} = c_e(x_e - 1) - c_e(x_e) \).

Now, let \( \delta(x) := \min_{C \in C} \{ c^x(C) = \sum_{e \in C} c^x_e \} \), where \( C \) is the set of directed cycles in \( G(x) \). We can argue that \( \delta(x^*) \) is always positive by the following lemma.

Lemma 1: If \( x^* \) is the unique optimal solution of the convex-cost network flow problem, then \( \delta(x^*) > 0 \).

Proof: Suppose not, i.e., \( \delta(x^*) \leq 0 \). By definition, there exists a directed cycle \( C \) with \( c^{x^*}(C) \leq 0 \). Let \( C_F := \{ e \in C \mid c^x_e = c_e(x_e + 1) - c_e(x_e) \} \) and \( C_B := \{ e \mid e \in C, c^x_e = c_e(x_e - 1) - c_e(x_e) \} \), we have \( c^{x^*}(C_F) + c^{x^*}(C_B) \leq 0 \). Let us define \( \tilde{x} \) as

\[
\tilde{x}_e = \begin{cases} 
  x^*_e + 1 & \text{if } e \in C_F \\
  x^*_e - 1 & \text{if } e \in C_B \\
  x^*_e & \text{otherwise}
\end{cases}
\]

(8)

Since for \( \forall e \in C_F, x^*_e < u_e \), we have \( \tilde{x}_e \leq u_e \), and for \( \forall e \in C_B, x^*_e > 0 \), we have \( \tilde{x}_e \geq 0 \). Therefore, the flow \( \tilde{x} \) satisfy the capacity constraints. Also, it can be seen that \( \tilde{x} \) satisfies the
flow conservation constraints. Thus $\bar{x}$ is a feasible solution, but

$$\sum_e c_e(x'_e) - \sum_e c_e(\bar{x}_e) = \sum_{e \in C_B} (-c'_e) - \sum_{e \in C_F} c'_e$$

$$= -c'(C_B) - c'(C_F) \geq 0,$$

which contradicts the uniqueness of the optimal solution.  

\[ \tag{9} \]

\[ \]

B. Main Theorem and Computation Tree

The convergence and correctness of BP is established by the following theorem.

\textbf{Theorem 1:} Suppose the convex-cost network flow problem has a unique optimal solution $x^*$. Define $L$ to be the maximum cost of a simple directed path in $G(x^*)$. Then for any $N \geq (\lfloor \frac{L}{2\delta(x^*)} \rfloor + 1)n$, we have $\hat{x}^N = x^*$.

Before going to the proof, let us first introduce the concept of the computation tree and some notations briefly. Let $T^N_e$ be the $N + 1$ level computation tree corresponding to variable $x_e$ as follows:

1. Let $V(T^N_e)$ and $E(T^N_e)$ be the vertex set and edge set of $T^N_e$ respectively.
2. Divide $V(T^N_e)$ into $N + 1$ levels and on the 0-th level, we have a ‘root’ edge $r$.
3. For any $v \in V(T^N_e)$, let $E_v$ be the set of all edges incident to $v$, and for the leaf nodes in $T^N_e$, $v$ is incident to exactly one edge.
4. Let $\Gamma$ be the map which maps each edge $e' \in E(T^N_e)$ to its corresponding edge $e \in E$ in the original graph $G$.

C. Proof of the Main Theorem

\textbf{Proof:} suppose the statement does not hold, i.e., there exists $e_0 \in E$ and $N \geq (\lfloor \frac{L}{2\delta(x^*)} \rfloor + 1)n$ such that $\hat{x}^N_{e_0} \neq x^*_{e_0}$. Since BP algorithm is accurate on the tree structure, there must exists an optimal solution $y^*$ for the computation tree $T^N_{e_0}$ with $y^*_{e_0} = \hat{x}^N_{e_0}$, where $\Gamma(e'_0) = e_0$.

Now, without loss of generality, suppose $y^*_{e'_0} > x^*_{e_0}$. Let $e'_0 = (v_\alpha, v_\beta)$. Because both $y^*$ and $x^*$ are feasible solutions, they satisfy the flow conservation constraints. Therefore, we can find edge $e'_1 \neq e'_0$ incident to $v_\alpha$ such that $y^*_{e'_1} > x^*_{e_1}$ if $e'_1$ has the same orientation as $e'_0$ and $y^*_{e'_1} < x^*_{e_1}$ if $e'_1$ has the opposite orientation as $e'_0$. Similarly, we can find edge $e'_{-1}$ incident to $v_\beta$ satisfying the same constraint.
If we continue this all the way down to the leaf nodes of $T_{e_0}^N$, we can find the set of edges 
\{e_N', e_{N-1}', \ldots, e_1', e_0', e_{-1}', \ldots, e_{-N}'\} such that
\[ y_{e_i}' > x_{e_i}' \iff e_i' \text{ has the same orientation as } e_0' \] (10)
\[ y_{e_i}' < x_{e_i}' \iff e_i' \text{ has the opposite orientation as } e_0' \] (11)

Let $X = \{e_N', e_{N-1}', \ldots, e_1', e_0', e_{-1}', \ldots, e_{-N}'\}$. For any $e' = (v_p, v_q) \in X$, let $\text{Aug}(e') = (v_p, v_q)$ if $y_{e_i}' > x_{e_i}'$ and $\text{Aug}(e') = (v_q, v_p)$ if $y_{e_i}' < x_{e_i}'$. Then, each $\Gamma(\text{Aug}(e'))$ is an edge on the residual graph $G(x^*)$. Note that $W = \{\text{Aug}(e_N'), \ldots, \text{Aug}(e_0'), \ldots, \text{Aug}(e_{-N}')\}$ is a directed path on $T_{e_0}^N$, and $\Gamma(W)$ is a directed walk on $G(x^*)$. We can argue that $c^x^*(\Gamma(W)) > 0$, which is shown by Lemma 2 in the sequel.

Now, let $\text{Forw} = \{e | e' \in X, y_{e_i}' > x_{e_i}'\}$ and $\text{Back} = \{e | e' \in X, y_{e_i}' < x_{e_i}'\}$. Define $\tilde{y}_e$ as
\[
\tilde{y}_e = \begin{cases} 
  y_{e_i}' - 1 & \text{if } e \in \text{Forw} \\
  y_{e_i}' + 1 & \text{if } e \in \text{Back} \\
  y_{e_i}' & \text{otherwise}
\end{cases}
\] (12)

Then, we can see that $\tilde{y}$ satisfies the capacity constraints and the flow conservation constraints for all non-leaf nodes. Therefore, $\tilde{y}$ is a feasible solution for computation tree $T_{e_0}^N$. However,
\[
\sum_{e \in E(T_{e_0}^N)} c_e(y_{e_i}') - \sum_{e \in E(T_{e_0}^N)} c_e(\tilde{y}_e) = \sum_{e \in \text{Forw}} c_{\tilde{y}_e} + \sum_{e \in \text{Back}} c_{\tilde{y}_e} 
\geq \sum_{e \in \text{Forw}} c^x_e + \sum_{e \in \text{Back}} c^x_e 
= c^x(\Gamma(W)) > 0.
\] (13)
The inequality (a) follows from the convexity of the cost function $c_e(x_e)$. This result contradicts the optimality of $y^*$, and thus completes the proof.

**Lemma 2:** If $N \geq \left(\lfloor \frac{L}{2\delta(x^*)} \rfloor + 1\right)n$, $c^x^*(\Gamma(W)) > 0$.

**Proof:** Since $\Gamma(W)$ is a directed random walk on the residual network $G(x^*)$, it can be decomposed into a set of directed cycles $Cyc(\Gamma(W))$ and one directed path $P$. Because $\Gamma(W)$ is of length $2N + 1$ and each directed cycle and directed path consists of at most $n$ edges, the

\[\text{This procedure can be proven to be successful till reaching leaf nodes.}\]
number of directed cycles \(|Cyc(\Gamma(W))| > \frac{L}{\delta(x^*)}\). Now, we have
\[
c^x(\Gamma(W)) = \sum_{C \in Cyc(\Gamma(W))} c^x(C) + c^x(P) \geq \sum_{C \in Cyc(\Gamma(W))} c^x(C) - L > \frac{L}{\delta(x^*)}\delta(x^*) - L \]
\[
= 0
\]

IV. RELATION TO CONTINUOUS NETWORK FLOW

In previous sections, we focus on the integral network flow problem. In this section, we consider the continuous relaxation where the network flow can take real values.

The continuous relaxation of the integral network flow is given by
\[
\begin{align*}
\min & \sum_e c_e(x_e) \\
\text{s.t.} & \sum_{e \in E_v} \Delta(v,e)x_e = d_v, \forall v \in V \text{ (flow conservation)} \\
& 0 \leq x_e \leq u_e \text{ (capacity)} ,
\end{align*}
\]

Remark 1: Objective function \(c_e(x_e)\) now is a convex piece-wise linear function and the corner points are at the integral value of \(x_e\). In this case, it can be shown that the continuous relaxation always has integral optimal solution. Therefore, we can use BP algorithm for the continuous relaxation problem to solve the integral case. However, it is better to consider the integral case directly since BP algorithm is easy to derive and implement in discrete case.

V. CONCLUSION

In this paper, we derive the BP algorithm for the minimum convex-cost integral network flow problem and show that it converges to the unique optimal solution. We also establish the relation between integral network flow and continuous one by considering the continuous relaxation and arguing that the continuous relaxation always has integral optimal solution. One possible future work is to consider the network flow problem with leakage or gain which is used to model
packet loss in networks or money circulation in financial market. Another possible direction is to analyze and implement BP algorithm for the particular case of network flow problem such as routing in networks.

REFERENCES


