# Exact loopy belief propagation on Euler graphs

**Chen-Hsiang Yeang**[1]
[1]Institute of Statistical Science, Academia Sinica, Taipei, Taiwan

**Abstract**— *Belief propagation is widely used in inference of graphical models. It yields exact solutions when the underlying graph is singly connected. When the graph contains loops, double-counting of evidence may degrade the accuracy of converged beliefs or even prohibit the convergence of messages.*

*In this paper, we propose an exact belief propagation algorithm on Euler graphs. An Euler graph contains paths (Euler paths) which traverse the entire graph by visiting each edge exactly once. By exploiting this property, messages are propagated along an Euler path without double counting local evidence. Unlike standard belief propagation, it postpones marginalization over a variable until all edges incident to the variable have been traversed along the Euler path.*

*The primary bottleneck of the algorithm is the marginalization over multiple variables which may have exponential time complexity. We propose two conditions when it is tractable. If $O(\log n)$ nodes have loops and each node has loop size $O(c)$, then the marginalization of almost every variable requires polynomial time. Furthermore, if all nodes have loop size no greater than $O(\log n)$, then only $O(\log n)$ variables require marginalization over an exponential number of configurations. Empirical results demonstrate the Euler belief propagation can efficiently evaluate the exact marginal probabilities when the graph has a clustered structure.*

**Keywords:** belief propagation, Euler graph

## 1. Introduction

Belief propagation (BP) algorithms [1], [2] are widely used in evaluating marginal probabilities and finding the maximum a posteriori configurations of graphical models such as Bayesian networks, Markov random fields, or factor graphs. A graphical model can be represented as a graph where terms in the joint distribution correspond to nodes or edges. Message functions are defined on graph edges. At each iteration, each message is updated according to the terms of the corresponding edge and nodes and its neighboring messages. Message updates continue until all messages converge to a fixed point. The belief function of a variable is the product of messages incident to the variable. Because the algorithms are efficient in many problems and easy to implement, they have become one of the major tools in machine learning and been applied in many different problems. Examples include probabilistic inference of Bayesian networks [3], decoding complex error-correcting codes [4], solving Boolean satisfiability problems [5], computer vision [6], statistical physics [2], and bioinformatics [7].

Belief propagation converges to the exact marginal or max-marginal probabilities when the underlying graph is a tree or a forest. When the graph contains loops, messages may circulate around loops and local evidence is counted multiple times [1]. Thus the resulting beliefs may be no longer accurate, or may not even converge to a fixed point.

Empirical studies indicate that belief propagation achieves great performance in some practical problems of complex loopy graphs [8]. Nevertheless, finding better approximations for loopy belief propagation and demarcating the scenarios when accuracy is achieved are still of theoretical and practical interests. Many previous works toward these two directions have been pursued. For instance, Weiss cast belief propagation on a loopy graph as belief propagation on an unwrapped tree converted from the original graph, and concluded that BP on graphs with a single loop is exact [9]. He further showed that BP on Gaussian graphical models is exact regardless of the graph structure [10]. Yedidia et al. cast model inference as the problem of minimizing the free energy in a statistical physical system, and showed that BP minimizes an approximated bethe free energy [2]. They consequently generalized BP to optimize a better approximation – Kikuchi free energy – to the free energy. Wainwright et al. decomposed the joint distribution on an arbitrary graph as a mixture of trees, and devised the tree reweighting algorithm which accurately identifies the MAP configurations [11].

In this paper, we propose a belief propagation algorithm on a special type of graphs – Euler graphs. An Euler graph contains paths which traverse the entire graph by visiting each edge exactly once. By exploiting this property, we propagate messages along an Euler path without double counting local evidence. Different from the standard BP, marginalization over a variable is postponed until all edges incident to this variable have been traversed. This algorithm gives exact marginal probabilities.

The major bottleneck of this algorithm is the marginalization over multiple variables. If the Euler path contains loops, then the messages along the loops have multiple variables. Direct marginalization over multiple variables has exponential time complexity in terms of variable size. Therefore, it is necessary to control the number and size of loops in order to make the algorithm tractable. We specify two conditions when the Euler BP is tractable: when the number of the loops is $O(\log n)$ and the size of loops is $O(c)$, and when the size of loops is $O(\log n)$.

The rest of the paper is organized as follows. Section 2 describes the motivation and procedures of the Euler belief propagation algorithm. Section 3 proves the exactness of the algorithm. In Section 4, we discuss the bottleneck of the algorithm and give two propositions to the conditions when the Euler BP is tractable. In Section 5, we then compare the performance of the standard BP and the Euler BP on simulated data. Finally, in section 6 we summarize the contribution of our work and discuss extensions of the current algorithm.

## 2. Euler belief propagation algorithm

We first define a specific class of graphical models and message update rules, and use them throughout the paper. Denote $G = (V, E)$ an undirected graph. We define a Markov random field $M_G$ over a collection of binary random variables $X_V$, where each variable corresponds to a node in $V$. The joint probability mass function of $M_G$ can be expressed as the product of pairwise *potential* terms associated with edges in $G$:

$$P(X_V|M_G) \propto \prod_{(x_i,x_j)\in E} \phi(x_i, x_j). \quad (1)$$

This definition deviates from a typical Markov random field, which also contains potential terms associated with nodes in $G$. This difference is unimportant for our purpose since we can absorb a singlet potential term into a pairwise potential term and express the typical Markov random field as equation 1.

In this paper we consider the marginal probabilities of single variables:

$$P(x_i|M_G) = \sum_{X_V \setminus x_i} P(X_V|M_G). \quad (2)$$

$P(x_i|M_G)$ can be exactly calculated by the standard belief propagation algorithm if $G$ is singly connected (trees or forests). Define $2|E|$ messages on the graph, where each message $m_{x_i\to x_j}(x_j)$ is associated with an edge $(x_i, x_j) \in E$. Initially set each message $m^0_{x_i\to x_j}(x_j) = 1$. At each iteration, update messages according to the following rules:

$$m^{t+1}_{x_i\to x_j}(x_j) = \sum_{x_i} \phi(x_i, x_j) \prod_{x_k\in N(x_i)\setminus x_j} m^t_{x_k\to x_i}(x_i), \quad (3)$$

where $N(x_i)$ denotes neighbors of $x_i$. Message updates continue until all messages converge. The belief function of variable $x_j$ is the product of messages incident to $x_j$:

$$b(x_j) \propto \prod_{x_i\in N(x_j)} m_{x_i\to x_j}(x_j). \quad (4)$$

When $G$ contains loops, the standard BP may not converge to the exact marginal probabilities. Inaccuracy arises from two related properties of the standard BP. First, since all messages are simultaneously updated, local evidence of potential terms may appear multiple times in messages. Second, because each message is a function of a single variable (the destination of the message), each message update marginalizes over one variable (the source of the message). When messages are propagated around a loop, some variables can be locally marginalized multiple times. Both problems are illustrated in Figure 1. The message from $x_5$ to $x_1$ can be expressed as

$$\begin{aligned} &m^t_{x_5\to x_1}(x_1) = \textstyle\sum_{x_5} \phi(x_1, x_5) m^{t-1}_{x_4\to x_5}(x_5) m^{t-1}_{x_3\to x_5}(x_5) \\ &= \textstyle\sum_{x_5} \phi(x_1, x_5) \sum_{x_4} \phi(x_4, x_5) \sum_{x_3} \phi(x_3, x_5) \cdot \\ &\textstyle\sum_{x_2} \phi(x_2, x_3) \sum_{x_1} \phi(x_1, x_2) m^{t-4}_{x_5\to x_1}(x_1). \end{aligned} \quad (5)$$

If message updates continue, each potential term along the loop will keep on multiplying to the messages. Moreover, even if we stop message update at $t = 4$ to let each $\phi$ appears in $m^t_{x_5\to x_1}(x_1)$ exactly once, it is still not the marginal probability $P(x_1)$, since marginalization over $x_1$ appears in equation 5.

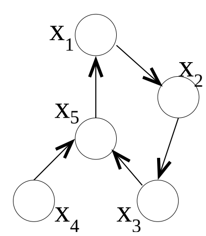


Fig. 1: Problem of message circulation

One way to make sure each potential term appears exactly once in messages is to update messages sequentially along a path. This is by no means a new concept. Forward-backward and Viterbi algorithms on hidden Markov models [13] sequentially update probability functions along the Markov chain, and they can be viewed as special cases of BP. However, to make sequential updates on a general graph we have to find a path that traverses each edge exactly once.

Finding a path which traverses each edge of the graph exactly once is the famous Euler path problem. It was first studied by Euler in 1736, and is considered as one of the first problems in graph theory. Euler showed the following conditions in which a graph has an edge-traversing path.

**Theorem** (Euler 1736)

An undirected graph $G$ has an Euler path iff it has either no nodes of odd degrees or only two nodes of odd degrees.

A graph is called an Euler graph if it contains an Euler path. An Euler path in an Euler graph can be constructed in linear time in terms of the graph node size [12].

For an Euler graph, the double-counting problem can be resolved by updating messages sequentially along an Euler path. At each iteration, only one message is updated, and the update is according to the potential term of the current edge and the message at the preceeding edge along the path, instead of all messages incident to the source node.

The double-marginalization problem can be resolved by postponing the marginalization over a variable until all edges incident to the variable have been traversed along the Euler path. A simple example is illustrated in the top graph of Figure 2. Suppose message updates follow the order $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_1$. When updating $m_{x_2 \rightarrow x_3}$, we can marginalize over $x_2$ since all potential terms containing $x_2$ appear in $m_{x_2 \rightarrow x_3}$. Hence $x_2$ will not be double-marginalized at subsequent steps. In contrast, $x_1$ is not marginalized out in updating $m_{x_1 \rightarrow x_2}$, since the term $\phi(x_3, x_1)$ has not yet appeared. Thus the message update of $m_{x_2 \rightarrow x_3}$ is

$$m_{x_2 \rightarrow x_3}(x_1, x_3) = \sum_{x_2} \phi(x_2, x_3) m_{x_1 \rightarrow x_2}(x_1, x_2). \quad (6)$$

Notice a message can be a function of multiple variables. The arguments of a message include the destination variable and all variables which remain unmarginalized up to the current update.

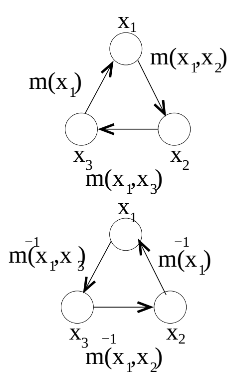


Fig. 2: An example of the Euler BP

Finally, a message incident to a variable contains the potential terms up to the current edge along the Euler path. However, the accurate belief function of a variable should contain all potential terms. To include the potential terms of the remaining edges, we have to run the message updates along the inverse direction of the original path. Two messages are complementary if they together cover every edge in the graph. The belief function of a variable is the product of two complementary messages incident to the variable, marginalized over all except the target variable. For example, in Figure 2 the complementary messages of $x_3$ are $m_{x_2 \rightarrow x_3}(x_1, x_3)$ and $m^{-1}_{x_1 \rightarrow x_3}(x_1, x_3)$, where

$$m^{-1}_{x_1 \rightarrow x_3}(x_1, x_3) = \phi(x_1, x_3). \quad (7)$$

Combining these procedures, we can now describe the Euler BP algorithm.

**Inputs**: An Euler graph $G = (V, E)$, probability mass function $P(X) = \frac{1}{Z} \prod_{e \in E} \phi_e(x_{e_1}, x_{e_2})$.

**Outputs**: Marginal probabilities $P(x_i)$ for each $x_i \in X_V$.

**Procedures**:

1) Identify an Euler path $\pi$ of $G$. Denote $\pi(1), \cdots, \pi(|E|)$ as the edge sequence along $\pi$.
2) Initialize message $m_0(X) = 1$.
   For $i = 1$ to $|E|$, update messages as follows.
$$m_{\pi(i)}(X) = \begin{cases} \sum_{x_{\pi(i)_1}} m_{\pi(i-1)}(X) \phi_{\pi(i)}(x_{\pi(i)_1}, x_{\pi(i)_2}) \\ \text{if } x_{\pi(i)_1} \text{ is traversed by } \pi \text{ at step } i \text{ ,} \\ m_{\pi(i-1)}(X) \phi_{\pi(i)}(x_{\pi(i)_1}, x_{\pi(i)_2}) \\ \text{otherwise.} \end{cases} \quad (8)$$
3) Denote $\pi^{-1}$ as the inverse path of $\pi$, $\pi^{-1}(i) = \pi(|E| - i + 1)$. Initialize inverse message $m_0^{-1}(X) = 1$.
   For $i = 1$ to $|E|$, update the inverse messages as follows.
$$m^{-1}_{\pi^{-1}(i)}(X) = \begin{cases} \sum_{x_{\pi^{-1}(i)_1}} m^{-1}_{\pi^{-1}(i-1)}(X) \cdot \\ \phi_{\pi^{-1}(i)}(x_{\pi^{-1}(i)_1}, x_{\pi^{-1}(i)_2}) \\ \text{if } x_{\pi^{-1}(i)_1} \text{ is traversed by } \pi^{-1} \text{ at step } i, \\ m_{\pi^{-1}(i-1)}(X) \phi_{\pi^{-1}(i)}(x_{\pi^{-1}(i)_1}, x_{\pi^{-1}(i)_2}) \\ \text{otherwise.} \end{cases} \quad (9)$$
4) For each variable $x_i$, identify complementary message pairs $m_a(X_r)$ and $m^{-1}_{|E|-a}(X_r)$ incident to $x_i$. If $a = 0$ or $a = |E|$, then $m^{-1}_{|E|-a}(X_r) = 1$ or $m_a(X_r) = 1$ respectively. If there are multiple pairs of complementary messages, find the one with the smallest $|X_r|$.
   The belief function of $x_i$ is
$$b(x_i) = \sum_{X_r \backslash x_i} m_a(X_r) m^{-1}_{|E|-a}(X_r). \quad (10)$$
5) Output $P(X) = \frac{1}{Z} b(X)$.

A toy example of the Euler BP is shown in Figure 2. The first message $m_{1 \rightarrow 2}$ qeuals to the edge potential $\phi(x_1, x_2)$. The next message $m_{2 \rightarrow 3}$ is a function of $x_1, x_3$, since $x_2$ is marginalized out but $x_1$ is not yet completely traversed.

$$m_{2 \rightarrow 3}(x_1, x_3) = \sum_{x_2} \phi(x_2, x_3) m_{1 \rightarrow 2}(x_1, x_2). \quad (11)$$

Similarly, $m_{3 \rightarrow 1}$ is a function of $x_1$. The inverse messages are updated along the opposite direction of the Euler path. The belief function of a variable is the product of two complementary messages incident to this variable, marginalized over all except the target variable. In this example,

$$b(x_2) = \sum_{x_1} m_{1 \rightarrow 2}(x_1, x_2) \cdot m^{-1}_{3 \rightarrow 2}(x_1, x_2) \quad (12)$$

## 3. Exactness of the algorithm

We have explained in Section 2 the intuition why the Euler BP outputs the exact marginal probabilities. In this section, we give a formal proof about the exactness of the algorithm.

**Theorem**

$b(x_i) \propto \sum_{X \backslash x_i} P(X)$ for each $x_i$.

**Proof**

We have to show that (1) each $\phi_e(x_{e_1}, x_{e_2})$ appears in $b(x_i)$ exactly once, (2) each variable except $x_i$ is marginalized exactly once in $b(x_i)$.

According to equations 8 and 9, $m_a(X_r)$ contains potential terms $\phi_{\pi(1)}, \cdots, \phi_{\pi(a)}$ exactly once, and $m^{-1}_{|E|-a}(X_r)$ contains potential terms $\phi^{-1}_{\pi^{-1}(1)}, \cdots, \phi^{-1}_{\pi^{-1}(|E|-a)}$ exactly once. Because $\pi^{-1}(1) = \pi(|E|), \cdots, \pi^{-1}(|E|-a) = \pi(a+1)$, $m_a(X_r)$ and $m^{-1}_{|E|-a}(X_r)$ cover each potential term exactly once. Thus (1) is true.

To show (2), we first show that a variable is not double-marginalized in $m_a(X_r)$ or in $m^{-1}_{|E|-a}(X_r)$. This is clear from message update equations 8 and 9, since we marginalize each variable at most once along $\pi$ or $\pi^{-1}$. We also have to show a variable is not marginalized in both $m_a(X_r)$ and $m^{-1}_{|E|-a}(X_r)$. If a variable $x_j$ is marginalized in $m_a(X_r)$, then according to message update rules no edges in $\pi(a+1) = \pi^{-1}(|E|-a), \cdots, \pi(|E|) = \pi^{-1}(1)$ are incident to $x_j$. Hence $x_j$ does not appear in potential terms of $m^{-1}_{|E|-a}(X_r)$ and is not marginalized in $m^{-1}_{|E|-a}(X_r)$. The same arguments hold for variables marginalized in $m^{-1}_{|E|-a}(X_r)$. (2) also guarantees $m_a$ and $m^{-1}_{|E|-a}$ have the same set of unmarginalized variables $X_r$. Each variable in $X_r$ appears in the potential terms in $m_a$ and $m^{-1}_{|E|-a}$.

## 4. Analysis of the algorithm

An exact algorithm is useful only if it is efficient. In this section, we compare the time complexities of the standard BP and the Euler BP, and give two propositions about the conditions when the Euler BP is efficient.

The time complexity of the standard BP depends on the number of iterations it takes to converge. Consider a graph of $n$ edges. Each iteration has $2n$ message updates. For a tree, it takes $O(n)$ iterations to converge. Hence the time complexity is $O(n^2)$. The time complexity on loopy graphs can be much higher. There are $2n$ message updates in the Euler BP. However, the marginalization in belief function calculation (equation 10) is the bottleneck of the algorithm. For an arbitrary function of $k$ binary variables, the time complexity of marginalizing it over $k-1$ variables is $O(2^{k-1})$. Therefore, the time complexity of the Euler BP is $O(2^{m-1}n)$, where $m$ is the maximum number of unmarginalized variables in each message and inverse message.

Under what condition is the Euler BP tractable? If the maximum number of unmarginalized variables is $O(\log n)$, then the time complexity of the algorithm is polynomial. This criterion is often too stringent. Thus we relax it by allowing a small fraction of messages with more than $O(\log n)$ unmarginalized variables. We can choose not to marginalize these messages at the last step of the algorithm (equation 10). The price to pay is not being able to calculate the marginal probabilities of all variables. However, only a negligible fraction of variables are skipped if they scale as $O(c)$ or $O(\log n)$ and $n$ is large.

The number of unmarginalized variables of each message depends on the graph topology and the Euler path. A variable is unmarginalized in a message if along the Euler path the first edge containing this variable appears before the current message, and the last edge containing this variable does not yet appear. Intuitively, a highly connected graph tends to have many unmarginalized variables. For instance, along the Euler path of a clique $K_n$, each message has $n$ unmarginalized variables. Thus the time complexity of the Euler BP on $K_n$ is identical to the brute force calculation of marginal probabilities. We quantize this intuition with the size of loops along an Euler path.

We define a loop $L_i$ of node $x_i$ along an Euler path $\pi$ as the segment of $\pi$ between the first edge emanating from $x_i$ and the last edge incident to $x_i$. $L_i$ can visit $x_i$ multiple times if $x_i$ has more than 2 edges. If $x_i$ has 2 edges, then $\pi$ will never visit back $x_i$ after it leaves $x_i$. In this case $x_i$ does not have a loop. If $x_i$ has an odd number of edges, then $x_i$ is an end point of $\pi$. We specify the loop of the starting node as the entire path and the end node does not have a loop. Furthermore, we define the loop size $|L_i|$ as the number of distinct nodes along $L_i$. We also denote $x_i$ influences another node $x_j$ if $L_i$ contains $x_j$. Figure 3 illustrates several examples of loops along an Euler path. A loop is marked by thick lines in a graph.

The number and size of loops along an Euler path determines the number of unmarginalized variables. A variable with a long loop appears in many messages, and more variables are accumulated as unmarginalized as there are more loops. We give two propositions on the number and size of loops when the Euler BP is tractable.

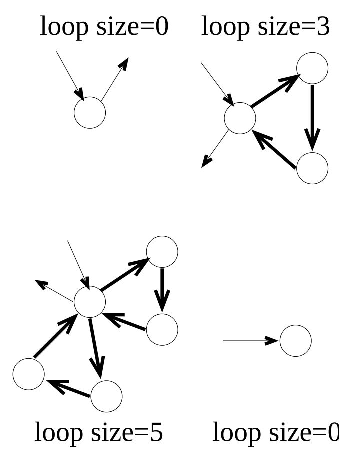


Fig. 3: Loops along Euler paths

**Proposition 1**

Given an Euler graph $G$ with $n$ nodes and an Euler path $\pi$. If the number of nodes which have loops is $O(\log n)$, and the size of each loop is $O(c)$, then there are $O(c)$ nodes which are influenced by more than $O(\log n)$ nodes. Therefore, all but a constant number of variables can be marginalized using the Euler BP in polynomial time.

**Proposition 2**

Given an Euler graph $G$ with $n$ nodes and an Euler path $\pi$. If the size of each loop is $O(\log n)$, then there are $O(\log n)$ nodes which are influenced by $O(n)$ nodes. Therefore, $O(\frac{\log n}{n})$ fraction of variables cannot be marginalized using the Euler BP in polynomial time.

Proposition 1 is obviously stronger. It limits the number of loops to $O(\log n)$ and requires loop sizes do not scale with the graph size. Under this condition almost all variables can be marginalized in polynomial time. Only a constant number of variables may take exponential time. They can be either skipped or evaluated by constructing other Euler paths. Proposition 2 relaxes the stringent condition in proposition 1 by restricting loop sizes to $O(\log n)$ and removing the limit of loop numbers. Consequently, $O(\log n)$ nodes may take exponential time to marginalize. Although this is a small fraction when $n$ is large, the total number of skipped variables can be substantial.

What type of graphs satisfy conditions in propositions 1 and 2? For proposition 1, a graph is linear in most part, and it can make "detours" to a relatively small number of loops ($O(\log n)$) with constant size. For proposition 2, a graph can have a hierarchical structure. It comprises many clusters of size $O(\log n)$. Each cluster can have a highly connected topology (e.g., fully connected), but clusters are connected by simple paths.

## 5. Proof of propositions

To prove propositions 1 and 2 we first show an important lemma.

**Lemma 1** Suppose there are $N_0$ nodes with loop size 0, $N_1$ nodes with loop size 1, $\cdots$, $N_l$ nodes with loop size $l$, where $l$ is the maximum loop size. The number of nodes influenced by $\geq k$ nodes $\leq \frac{1}{k}(N_1 + 2N_2 + \cdots + lN_l)$.

**Proof** We define an $n \times n$ matrix $T$ as follows:

$$T_{ij} = \begin{cases} 1 & \text{if } x_i \text{ influences } x_j, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Row $i$ represents nodes influenced by node $i$, and column $j$ represents nodes influencing node $j$. The total number of 1s in $T$ is $\sum_{i=1}^{l} iN_i$. Denote $n_k$ the number of nodes influenced by $\geq k$ nodes, i.e., the number of columns with $\geq k$ 1's. The number of 1s in $T$ contributed by nodes influenced by $\geq k$ nodes is greater than or equal to $k \cdot n_k$. Thus

$$\sum_{i=1}^{l} iN_i \geq kn_k; n_k \leq \frac{1}{k}\sum_{i=1}^{l} iN_i. \tag{14}$$

**Proof of proposition 1**

From the conditions in proposition 1, the maximum loop size $l$ is a constant, and $\sum_{i=1}^{l} N_i \leq \#(\text{nodes with loops})$ $\leq k' \log n$, where $k'$ is another constant. Thus

$$\sum_{i=1}^{l} iN_i \leq l \cdot \sum_{i=1}^{l} N_i \leq lk' \log n. \tag{15}$$

By applying equation 15 to lemma 1,

$$\begin{aligned} &(\#\text{ nodes influenced by } \geq k \log n \text{ nodes }) \\ &\leq \frac{1}{k \log n}\sum_{i=1}^{l} iN_i \leq \frac{lk'}{k} = \text{ constant }, \end{aligned} \tag{16}$$

which proves proposition 1.

**Proof of proposition 2**

From the condition in proposition 2, $l = k' \log n$, where $k'$ is a constant. By applying lemma 1,

$$\begin{aligned} &(\#\text{ nodes influenced by } \geq kn \text{ nodes }) \leq \frac{1}{kn}\sum_{i=1}^{l} iN_i \\ &\leq \frac{1}{kn} \cdot l \cdot \sum_{i=1}^{l} N_i \leq \frac{l}{kn} \cdot n \leq \frac{k'}{k}\log n. \end{aligned} \tag{17}$$

Hence proposition 2 is proved.

## 6. Experiments

We compared the accuracy and time complexity between the standard belief propagation and the Euler belief propagation algorithms on artificial data. The data was generated by creating random Euler graphs with given numbers of nodes and edges and assigning random potential functions to each edge of the graph. For simplicity we considered all random variables as binary. The accuracy of the inference results was gauged by the $L_1$ norm of the difference between the marginal probabilities and the (normalized) belief functions. We reported both the largest $L_1$ difference among all variables and the fraction of variables whose $L_1$ difference exceeds a threshold (0.05). The time complexity was measured by the number of summations and multiplications in the marginalization steps of the algorithms.

We first verified the accuracy of the Euler BP by running experiments on small graphs (between 5 and 15 nodes), and compared the performance with the standard BP. We varied both node and edge sizes. For each fixed number of nodes, edge numbers ranged from the number of nodes +1 (almost a tree) to maximum number of possible edges (a clique). Under each setting of node and edge numbers, 1000 random graphical models were created. We applied both standard and Euler BPs to each model and reported the accuracy of inference results. As expected, the belief functions of the Euler BP are identical to the marginal probabilities over all settings and all random experiments. Figures 4.1 and 4.2 demonstrates the mean of the maximum $L_1$ deviations and the fractions of inaccurate variables for the standard BP. Clearly, inaccuracy increases as the graph becomes bigger (more nodes) or more connected (more edges). Both maximum deviation and the fraction of inaccurate variables are low in each setting. For cliques of 15 nodes, the mean maximum deviation is 0.2 and the mean fraction of inaccurate variables is 0.25. The high accuracy is consistent with previous observations that belief propagation achieved good empirical performance. However, the high variance of the inaccuracy over random experiments (not shown here) suggests there exists settings of graph connections and edge potentials which make the inference results inaccurate even in small models.

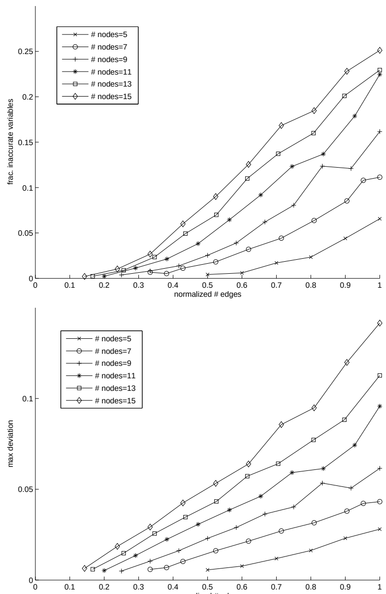


Fig. 4: Accuracy of standard BPs. Top: the fraction of incurate variables versus the number of nodes and sparsity in the graph. Bottom: the maximum $L_1$ distances between inferred and real probabilities versus the number of nodes and sparsity in the graph. Errors of Euler BPs are 0 in all experiments thus are not shown.

We then compared the time complexity of the algorithms on larger graphs. As discussed in Section 4, the Euler BP is tractable if the graph has a clustered structure. We created a large graph by first generating random clusters of the same size, then connecting these clusters in a sequence. The top diagram of Figure 5 shows the mean time complexities of standard and Euler BPs with fixed numbers of clusters. For graphs of low connectivity (the x points), complexities of both algorithms tend to scale sub-exponentially in terms of cluster size. This is reasonable since the stanard BP takes fewer iterations to converge, and the Euler BP encounters fewer loops along the Euler path. The Euler BP is more efficient than the standard BP when the cluster size is small or the graph is sparsely connected. This is sensible since the number of message updates in the Euler BP is much smaller than that in the standard BP. Thus the Euler BP is more efficient if the bottleneck of marginalization is relieved. As the connectivity increases, both algorithms tend to scale exponentially in terms of cluster size, but the Euler BP has a higher rate. As discscussed in Section 4, the Euler BP on a highly connected graph is similar to the brute force evaluation of marginal probabilities. The standard BP is much more efficient than the Euler BP on highly connected, large graphs, but its accuracy also degrades, as shown in Figure 4.

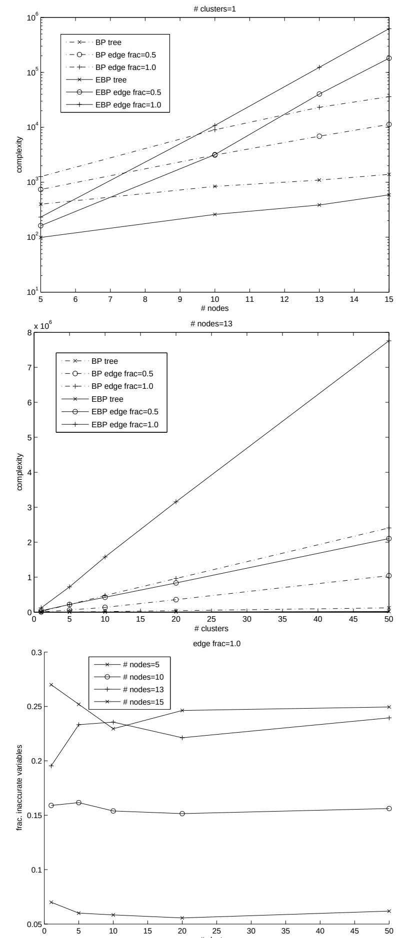


Fig. 5: Complexity and accuracy of Euler and standard BPs on large graphs. Top: time complexity versus the number of nodes and sparsity in the graph. Middle: time complexity versus the number of clusters and sparsity in the graph. Bottom: fraction of inaccurate variables versus the number of clusters.

The middle diagram of Figure 5 shows the mean time complexity of both algorithms in terms of cluster numbers. Clearly, it scales linearly with the cluster number. This is a very useful property for the inference on large graphs. In the experiments, we can calculate the exact marginal probabilities of graphs with 750 nodes and 5250 edges if they have a clustered structure. This would be intractable for the brute force evaluation. The mean inaccuracy in the bottom diagram of Figure 5 suggests that the fraction of inaccurate variables of the standard BP is independent of the number of clusters but only depends on cluster size and connectivity. This result may serve as the argument for using the standard BP to graphs of the clustered structure.

However, the total number of inaccurate variables increases as the graph size. This is undesirable if the goal is to minimize the number of inaccurate variables.

## 7. Conclusion and discussion

Calculating the marginal probabilities of graphical models has many important applications. The standard belief propagation algorithm can efficiently approximate the marginal probabilities, but the inference results may not be exact if the graph contains loops. In this paper, we propose a revised belief propagation algorithm, the Euler BP, which calculates the exact marginal probabilities on Euler graphs. The Euler BP avoids over-counting evidence by propagating messages along an Euler path and postponing marginalization until each evidence pertaining to a variable is traversed. We propose two conditions in which the Euler BP is tractable in large graphs. Experiments on artificial data verify the accuracy of the Euler BP, and show the algorithm is tractable when the graph comprises small clusters connected in a sequence.

The current version of the algorithm is tractable to a specific class of graphs. Several directions of improvement are important. First, we want the algorithm to apply to non Euler graphs. A trivial approach is to add "dummy edges" between nodes of odd degrees, and assign identity potential functions to these edges. The marginal probabilities of the augmented graph are identical to the original graph. Since every graph has an even number of nodes of odd degrees, we can convert an arbitrary graph into an Euler graph and maintain the marginal probabilities. However, the augmented graph may yield larger loops. Hence a criterion of adding new edges is needed. Second, when there are multiple Euler paths, one can be better than others because it gives smaller and fewer loops. It is important to develop a systematic approach to select the Euler path. Third, one way to avoid exponential explosion on large loops is to relax the condition for message marginalization. Instead of waiting until each evidence is collected, one can marginalize over a variable if the number of variables in a message exceeds a certain number. Empirical studies about this approximation in comparison with the standard BP is required. Fourth, the Euler BP can also calculate max-marginal probabilities when summation is replaced by maximization at marginalization. However, empirical evidence suggests the error in max-marginal probabilities rarely alters the optimal configurations. Hence exact evaluation may not be very useful in finding optimal configurations.

## References


[1] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*, San Mateo: CA: Morgan Kauffman, 1988.

[2] J.S. Yedidia, W.T. Freeman, Y. Weiss, "Understanding belief propagation and its generalizations", *Exploring Artificial Intelligence in the New Millennium*, Chap 8, pp. 239-256, 2001.

[3] C. Huang and D. Dawiche, " Inference in belief networks: a proecdural guide", *International Journal of Approximation Reasoning* 15:225-263, 1996.

[4] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Soft-output decoding algorithm in iterative decoding of turbo codes," JPL TDA, Tech. Rep. 42-124, 1996.

[5] M. Mezard and R. Zecchina, "Random k-satisfiability: from an analytic soultion to an efficient algorithm," *Physical Review E* 66:056126-1-056126-27, 2002.

[6] P. Felszenswalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IEEE Conference on Computer Vision and Pattern Recognition* 59(2):167-181, 2004.

[7] C.H. Yeang and T. Jaakkola, "Physical network models", *Journal of Computational Biology*, 11(2-3):243-262, 2004.

[8] B. Frey and D. MacKay, "A revolution : belief propagation in graphs with cycles", *Proceeding of the Neural Information Processing*, 1997.

[9] Y. Weiss, "Correctness of local probability propagation in graphical models with loops", *Neural Computation* 12(1): 1-41, 2000.

[10] Y. Weiss and W.T. Freeman, "Correctness of belief propagation in Gaussian graphical models", *Neural Computation* 13(10):2173-2200, 2001.

[11] M. Wainwright, T. Jaakkola and A. Willsky. "Exact map estimates by (hyper)tree agreement", *Proceeding of the Neural Information Processing*, 2002.

[12] H. Fleischner, *Eulerian Graphs and Related Topics*, Elsevier Science, London, 1990.

[13] L.R. Rabiner and B.H. Juang, "An introduction to hidden Markov models", *IEEE ASSP Magazine* 4-15, 1986.