2007 Conference on Information Sciences and Systems, The Johns Hopkins University, March 14--16, 2007

Consensus Estimation via Belief Propagation

Huaiyu Dai and Yanbing Zhang
Department of Electrical and Computer Engineering
NC State University
Raleigh, NC 27695
E-mail: Huaiyu_Dai@ncsu.edu, yzhang@ncsu.edu

Abstract –**In this paper, a new problem, consensus estimation, is formulated, whose setting is complementary to the well-known CEO problem. In particular, a set of nodes are employed to sense and estimate a common source, and the purpose is to reach the best possible estimate for all nodes, through local processing and information exchange over the network. The belief propagation algorithm is adopted to provide a common information processing and dissemination framework for such a purpose. The discussion is also extended to the application of estimating a Markov random field.**

I. INTRODUCTION

Recent advances in information technology are leading to a paradigm shift towards ad hoc networking, distributed processing and pervasive computing and communications. Examples include mobile ad hoc networks, wireless mesh networks, and sensor networks for various military, commercial, environmental and emergency applications.

The well-known CEO problem [1][2][3] in a decentralized communication setting is formulated as follows. A team of agents are deployed to observe a common source and report to the CEO (chief executive officer or central estimation officer) *independently* (i.e., they are not allowed to convene) with a sum rate constraint. The CEO reconstructs the source based on these reports, and the goal is to achieve the best distortion-rate tradeoff.

In this paper we consider a new problem, consensus estimation, in a complementary setting. The motivation is that, in many interesting scenarios, a central point is not available due to lack of infrastructure. In principle, such infrastructure could be set up, but the associated cost and maintenance would be costly or even infeasible. Furthermore, a network with such a single point may be vulnerable to environmental dynamism or malicious attack. Therefore, it is preferable to have a distributed network where individual nodes of limited capability can self-organize to form a cooperative powerful system in a scalable and energy-efficient way. Instead of communicating to a central point, nodes talk to their neighbors to share information, and hopefully due to dense deployment, the whole network is connected so eventually each node can collect enough information to make a good estimate of the underlying source. A side benefit is that, since every node has the result, there is an added layer of robustness and flexibility to outside query.

The remainder of this paper is organized as follows: Section II presents the system model and problem statement. The belief propagation (BP) algorithm is introduced in Section III, and applied to the consensus estimation problem. The discussion of Section III is concretized with Gaussian distribution in Section IV, and extended to the field estimation scenario. Finally Section V concludes the paper.

II. SYSTEM MODEL AND PROBLEM STATEMENT

For simplicity, consider a temporally memory-less source X with distribution function $p(x)$. Each of the N nodes of a distributed system observed independently corrupted versions of X, $\{Y_i\}_{i=1}^N$, with conditional probability $f_i(y_i \mid x)$. The goal of the consensus estimation problem is to reach a common estimate of X at each node, through local processing and information exchange. In certain scenarios, such a goal can be achieved via distributed averaging or gossiping algorithms [4][5]. Here we would like to take a more general Bayesian approach. A somewhat stronger condition is imposed that, after information processing and dissemination, each node can obtain the posterior distribution

$$p\left(x \middle| \{y_i\}_{i=1}^N\right) \propto p(x) \prod_{i=1}^N f_i(y_i \mid x) . \tag{1}$$

Then various estimates such as those corresponding to MAP or MMSE criteria can be easily obtained. We will discuss a common information processing and dissemination framework for this purpose in Section III.

Analysis with respect to the tradeoff between distortion and communication constraints can be pursued. Nonetheless, such a study is potentially much more involved due to the following reasons. First, the time it takes to reach consensus (the convergence time of the algorithm) is certainly non-trivial. We may typically need to stop at some point and there is an additional distortion due to this approximation. Second, in many scenarios of interest, each node is hard energy-limited, so it might make more sense to directly consider distortion-energy tradeoff. Compared with the CEO problem, consensus estimation avoids long-haul transmission to the central point, but requires more iterations for consensus. Finally this problem involves the interaction among source, channel and networking coding. We leave this study to our future study.

This work was supported in part by the National Science Foundation under Grant CCF-0515164.

III. BP ALGORITHM AND ITS APPLICATION TO CONSENSUS ESTIMATION

In this section, we first introduce a powerful probabilistic framework, belief propagation, for collaborative information processing. Then we apply it to our consensus estimation problem and discuss some convergence properties.

A. BP Algorithm

Belief propagation (also known as sum-product algorithm) refers to a general class of message-passing algorithms originally intended to solve the NP-hard probabilistic inference problems by exploiting "partial independence" existing among random variables [6]. It became known to researchers in the communications and signal processing society after the discovery and rediscovery of capacity-achieving turbo codes and LDPC codes, whose success has not been well understood yet. It is exciting to find that decoding of these powerful codes naturally results when applying BP on graphs with cycles. One of the most celebrated algorithms in digital communications, the Forward-Backward algorithm, is also a direct application of BP to the hidden Markov model or trellis diagram [7][8].

We briefly introduce the basic BP algorithm here to facilitate the following discussion. Given a joint distribution function $P(\mathbf{X})$ [1] of a random vector $\mathbf{X}=(X_1,\cdots,X_N)^T$, the NP-hard inference tasks like marginalization or posterior calculation can be made much easier when $P(\mathbf{X})$ assumes a product form of local functions; the implied conditional independence can be conveniently captured by graphical models including Markov random fields (MRF), Bayesian networks (BN), and factor graphs (FG) [9][10]. The intuition is that each node (variable) only needs to interact with its neighbors (dependent variables) whose number could be substantially smaller (e.g., $O(\log N)$), indicating potentials of exponential speedup on inference tasks. BP algorithms are efficient techniques for such a purpose, assuming different forms on various graphical models. For ease of exposition, we will focus on MRF with only pairwise interactions, since BN, FG, and MRF with higher-order cliques (i.e., fully-connected subgraph) can always be converted to an equivalent pairwise MRF. BP algorithms on MRF also turn out a bit simpler; while messages assume two forms in BN due to directedness of edges (reflecting parent-child relationship), and in FG due to two types of nodes (function and variable), they take one uniform form in MRF. MRF also exhibits modeling convenience in wireless networks mostly consisted of homogeneous nodes.

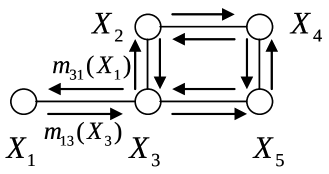

Figure 1 Pairwise Markov Random Field

A pairwise MRF, as shown in Figure 1, is an undirected graph (V,E) with maximum cliques of size 2, where each node $i\in V$ is associated with a random variable (or a more general random vector). The Hammersley-Clifford theorem [6] dictates that, if a joint distribution can be represented by a pairwise MRF, it should admit the following form (and vice versa)

$$p(\mathbf{X})=\frac{1}{Z}\prod_{(i,j)\in E}\psi_{ij}(X_i,X_j)\prod_{i\in V}\phi_i(X_i)\,, \qquad (2)$$

for a set of single-node functions $\{\phi_i(X_i)\}$ (called local functions, defined for each $i\in V$), and a set of pairwise functions $\{\psi_{ij}(X_i,X_j)\}$ (called compatibility functions, defined for each $(i,j)\in E$), and a normalization factor Z (called partition functions in physics).

The essence of the BP algorithm is the message-passing rule and belief-updating rule. The message from node i to j at the nth iteration is a function of X_j, defined as

$$m_{ij}^n(X_j)=\sum\nolimits_{X_i}\psi_{ij}(X_i,X_j)\phi_i(X_i)\prod\nolimits_{k\in N(i)\backslash\{j\}}m_{ki}^{n-1}(X_i)\,, \qquad (3)$$

where $N(i)$ is the set of neighbors of node i. The sum in (3) is replaced with the integral when continuous random variables are considered. This message is often normalized for numerical stabilization though not necessarily. Roughly speaking, it represents the current *belief* (approximated posterior probability distribution) that node i has about X_j, given its own observations and received messages from other parts of the graph in the last round. The *belief* node i has about its own variable is updated as (with normalization factor α)

$$b_i^n(X_i)=\alpha\phi_i(X_i)\prod\nolimits_{k\in N(i)}m_{ki}^n(X_i)\,. \qquad (4)$$

Usually the messages are initialized with unbiased (constant) ones to trigger the iteration. If the computing graph is a tree, it is known that the BP algorithm is guaranteed to converge to the true marginals, i.e., $b_i^n(X_i)\to P(X_i)=\sum\nolimits_{\mathbf{X}\backslash X_i}P(\mathbf{X})$. BP can be naturally applied on graphs with cycles as well. In this scenario, the iteration is typically stopped when improvement on beliefs is marginal, or sufficiently many numbers of iteration have passed. However, little is known about the convergence and correctness of BP on loopy graphs, though its effectiveness has been verified through experiments in various areas.

B. Application to Consensus Estimation

The BP algorithm was originally proposed to solve the NP-hard inference problems: we are given a joint distribution, and certain graphical models (such as MRF, BN, and FG) are constructed and BP algorithms are employed for efficient computation. For example, if a joint distribution of the form (2) is given, a corresponding pairwise MRF can be set up and iterations (3) and (4) are invoked for efficient calculations of marginals or posteriors. A *reverse* thinking is required when it is applied to wireless networks, intended to serve as a general framework for collaborative information processing and dissemination. We propose a simple and effective approach for

---

[1] In this paper we directly use random quantities as arguments of functions to denote a generic case.

this novel application of great potential as follows. The real communication graph is treated as a Markov random field: each (active) node is taken as a vertex and there is an edge between two nodes when there is a feasible communication link between them. The key step lies in associating some "virtual" state variable(s) to each node, and building some statistical models indicating relationship among them, based on application characteristics and communication models; for a pairwise MRF setting this is equivalent to defining compatibility functions $\{\psi_{ij}\}$ for each edge and (possibly) partial local functions $\{\phi_i\}$ for each node. Local functions could also include prior knowledge of the state variables and their likelihood functions given node observations. Then, assuming a joint distribution of state variables in the form (2), the BP algorithm described in above can be readily applied.

In the consensus estimation problem, the state variable associated with each node is the common source X. Since this variable is the same for all nodes, we have the following instantiation of the BP algorithm (c.f. (2)).

$$\phi_i(X) = f_i(y_i \mid X)\sqrt[N]{p(X)}, \quad \psi_{ij}(X_i, X_j) = 1(X_i = X_j). \tag{5}$$

In other words, we impose joint distribution of the form $P(X_v = x_v, v \in V) = 1\{x_1 = x_2 = \cdots = x_N = X\}p(X)\prod_{i=1}^{N} f_i(y_i \mid X)$ with $1(\cdot)$ denoting the indicator function.

The message-passing rule (3) is thus concretized as

$$\log m_{ij}^{n}(X_j) = \log \phi_i(X_j) + \sum\nolimits_{k \in N(i)\setminus\{j\}} \log m_{ki}^{n-1}(X_j) \tag{6}$$

which reveals a simple linear relationship (without convolution) for messages between successive rounds due to the special form of compatibility functions.

Upon collecting messages (in log domain) corresponding to each source value from all edges[2] into a column vector $\mathbf{z}_X^n$ of size $2|E| \times 1$, and similarly defining a vector $\mathbf{u}_X$ for the first term in the right-hand side (RHS) of (6), we obtain

$$\mathbf{z}_X^n = \mathbf{u}_X + \mathbf{A}\mathbf{z}_X^{n-1}, \tag{7}$$

where the square matrix $\mathbf{A}$ [3] captures the characteristics of the graph as represented by the second term in the RHS of (6). Viewing this recursion as a mapping $f(\mathbf{x}) = \mathbf{u}_X + \mathbf{A}\mathbf{x}$, with $f'(\mathbf{x}) = \mathbf{A}$ and the contraction mapping principle, it can be shown that if the spectral radius of $\mathbf{A}$, $\rho(\mathbf{A}) < 1$, $\mathbf{z}_X^n \to \mathbf{z}_X^\infty = (I - \mathbf{A})^{-1}\mathbf{u}_X$ for any initial messages. Also, upon convergence the final belief at node v is given by (c.f. (4))

$$b_v(X) = \alpha\phi_v(X)\exp\left(\mathbf{1}_v^T \mathbf{z}_X^\infty\right) = \alpha\phi_v(X)\exp\left(\mathbf{1}_v^T \sum\nolimits_{k=0}^{\infty} \mathbf{A}^k \mathbf{u}_X\right) \tag{8}$$

where $\mathbf{1}_v$ denotes a vector of the same dimension as $\mathbf{z}_X^\infty$, with ones at positions corresponding to the incoming edges of node v and zeroes otherwise. Note that $\mathbf{1}_v^T \mathbf{A}^k \mathbf{u}_X = \sum\nolimits_{v' \in N^{(k+1)}(v)} w_{v'} \log \phi_{v'}(X)$ admits a simple interpretation: *it effectively collects local information from nodes $v' \in N^{(k+1)}(v)$ that are distance-$k+1$ away, weighted by the number of paths $w_{v'}$ between them.*

Remark 1: If $w_{v'} = 1$ and the communication graph is connected, our goal is achieved, i.e., $b_v(X) \propto \coprod_{v' \in V} \phi_{v'}(X) \propto p\left(x \middle| \{y_i\}_{i=1}^{N}\right)$ (c.f. (1)). This is obviously true when the graph is a tree; in this scenario it is easy to verify that the corresponding matrix $\mathbf{A}$ is nilpotent so $\rho(\mathbf{A}) = 0$.

Remark 2: For general graphs, some local information may be over counted (i.e., $w_{v'} > 1$) so the final beliefs may not be correct. Nonetheless, we don't need correct beliefs to make correct MAP estimates. Intuitively, we can still be on the correct side (though may be over confident) as long as all evidence are equally over counted, which dictates a certain symmetry on the communication graph.

Remark 3: If the source and observation assume Gaussian distribution[4], when the algorithm converges, each node will obtain correct posterior means and modes [12].

Remark 4: Though message synchronization is assumed for simplicity, the algorithm is guaranteed to converge when $\rho(\mathbf{A}) < 1$ even with total asynchronism (i.e., arbitrary delays in message arrival) [11].

Remark 5: For the special case of consensus detection (i.e., X is discrete and finite) [13], absolute convergence is not necessary either (i.e., $\rho(\mathbf{A}) \geq 1$ is allowed), as long as in the final obtained normalized beliefs, the one corresponding to the true MAP estimate dominates.

IV. GUASSIAN CASE

Gaussian distribution is a widely adopted assumption in theoretical studies. It is a good approximation of practical situations in many scenarios of interest, while amenable to analysis and often can provide useful insights.

The following result is useful for message passing with Gaussian distribution. Let $X \sim \mathcal{N}(\mu, \Sigma)$ be a Gaussian random vector with mean μ and positive definite covariance Σ. One can define a new set of parameters (ϑ, Λ) by $\vartheta = \Sigma^{-1}\mu$, $\Lambda = \Sigma^{-1}$, and alternatively denote $X \sim \mathcal{N}^{-1}(\vartheta, \Lambda)$. Let $p_1(x) = \mathcal{N}^{-1}(\vartheta_1, \Lambda_1)$ and $p_2(x) = \mathcal{N}^{-1}(\vartheta_2, \Lambda_2)$ be two different distributions on the same random Gaussian random vector *x*, and consider the product density $p_{12}(x) = \alpha p_1(x) p_2(x)$. Then

---

[2] Note that each edge has two messages of opposite directions.

[3] Entry $a_{ee'}^{(k)}$ of the *k*th power of matrix **A,** $\mathbf{A}^k = \left(a_{ee'}^{(k)}\right)_{2|E|\times 2|E}$ is the number of directed paths of length $k+1$ which start from the source node of link e' and end at the destination node of e, and do not contain consecutive sections of the same arc with opposite directions.

[4] This is a common assumption and will be further discussed in Section IV.

$p_{12}(x) = \mathcal{N}^{-1}(\vartheta_{12}, \Lambda_{12})$ with $\vartheta_{12} = \vartheta_1 + \vartheta_2$ and $\Lambda_{12} = \Lambda_1 + \Lambda_2$. Similarly, the quotient $p_1(x)/p_2(x)$ produces an exponential quadratic form with parameters $(\vartheta_1 - \vartheta_2, \Lambda_1 - \Lambda_2)$. However, this quotient will define a valid probability density only if $\Lambda_1 - \Lambda_2$ is positive definite.

A. Single Source

Assume the source $X \sim \mathcal{N}^{-1}(\mu_S/\sigma_S^2, 1/\sigma_S^2)$. Each node makes a noisy linear observation

$$y_i = H_i x + n_i,\ i = 1, \cdots, N, \tag{9}$$

where for generality we consider a vector observation of $y_i \in \mathbb{R}^{d_i}$ for each sensor[5], channel gain matrix H_i is assumed known, and noise n_i is Gaussian with zero mean and variance Σ_i. It is easy to derive that, the conditional probability $f_i(y_i \mid x)$, viewed as a function of x, assumes the form of

$$\mathcal{N}^{-1}(H_i^T \Sigma_i^{-1} y_i, H_i^T \Sigma_i^{-1} H_i) \tag{10}$$

up to some scaling constant.

Clearly the messages and node beliefs in BP algorithms are all Gaussian distributed. Assuming that

$$m_{ij}^n(x) \sim \mathcal{N}^{-1}(\mu_{ij}^n, V_{ij}^n), \text{ and } b_i^n(x) \sim \mathcal{N}^{-1}(q_i^n, W_i^n), \tag{11}$$

we have the following message updating and belief updating rules:

$$\begin{cases} \mu_{ij}^n = \mu_S/(N\sigma_S^2) + H_i^T \Sigma_i^{-1} y_i + \sum_{k \in N(i) \backslash \{j\}} \mu_{ki}^{n-1} \\ V_{ij}^n = 1/N\sigma_S^2 + H_i^T \Sigma_i^{-1} H_i + \sum_{k \in N(i) \backslash \{j\}} V_{ki}^{n-1} \end{cases} \tag{12}$$

and

$$\begin{cases} q_i^n = \mu_S/(N\sigma_S^2) + H_i^T \Sigma_i^{-1} y_i + \sum_{k \in N(i)} \mu_{ki}^n \\ W_i^n = 1/N\sigma_S^2 + H_i^T \Sigma_i^{-1} H_i + \sum_{k \in N(i)} V_{ki}^n \end{cases} \tag{13}$$

with μ_{ij}^0 and V_{ij}^0 initialized with 0 for all i, j.

Noting the similarity of (12) and (13), the implementation of the BP algorithm in a wireless setting can exploit the *broadcast* nature of the medium. Instead of sending messages of the form (12) from each node i to its neighbors, we let node i broadcasts its belief to them with the following modified form:

$$\begin{cases} q_i^n = \mu_S/(N\sigma_S^2) + H_i^T \Sigma_i^{-1} y_i + \sum_{k \in N(i)} \mu_{ki}^{n-1} \\ W_i^n = 1/N\sigma_S^2 + H_i^T \Sigma_i^{-1} H_i + \sum_{k \in N(i)} V_{ki}^{n-1}. \end{cases} \tag{14}$$

Meanwhile, it calculates and stores its intended messages for all $j \in N(i)$ to facilitate processing in the next round (c.f. (16) below):

$$\begin{cases} \mu_{ij}^n = q_i^n - \mu_{ji}^{n-1} \\ V_{ij}^n = W_i^n - V_{ji}^{n-1}. \end{cases} \tag{15}$$

On the other hand, upon receiving q_j^n and W_j^n from some $j \in N(i)$, node i figures out the true messages from j as

$$\begin{cases} \mu_{ji}^n = q_j^n - \mu_{ij}^{n-1} \\ V_{ji}^n = W_j^n - V_{ij}^{n-1}, \end{cases} \tag{16}$$

and also store them for processing in the next round (c.f. (15)). When a node i collects all broadcast from its neighbors and figures out their intended messages, it can form its own broadcast message q_i^{n+1} and W_i^{n+1} for next iteration. Again μ_{ij}^0 and V_{ij}^0 are initialized with 0 for all i, j. In practice, node broadcasting needs to be coordinated with some MAC schemes.

B. Markov Random Field

In last subsection, we discuss the BP algorithm for consensus estimation of a single Gaussian source. This can be readily extended to multiple independent sources by treating X as a Gaussian vector.

In this subsection, we consider the application of field gathering where X is a Gaussian Markov random field and each node only observes a spatial component X_i of it. In this scenario, X_i associated with each node are not identical but nonetheless correlated through a joint distribution. Instead of achieving a common estimate at each node as previously discussed, here we intend to apply the BP algorithm to improve the estimate at each node through collecting useful information from other parts of the network.

Here we consider a good approximation for the underlying random field. Assuming that a spanning tree is formed among the distributed nodes, we only consider the pairwise interaction among $\{X_i\}$ associated with each node. In other words, we ignore the correlation among nodes that are not direct neighbors on the spanning tree. In this setting, we have [14][15]

$$p(\{x_i\}, \{y_i\}) = \frac{\prod_{(i,j) \in E} p_{ij}(x_i, x_j)}{\prod_{i \in V} p_i(x_i)^{N(i)-1}} \prod_{i \in V} f_i(y_i \mid x_i), \tag{17}$$

where

$$p_i(x_i) = \mathcal{N}^{-1}(\mu_S/\sigma_S^2, 1/\sigma_S^2), \tag{18}$$

and

$$p_{ij}(x_i, x_j) = \mathcal{N}^{-1}(C_{ij} \mu_S [1,1]^T, C_{ij}) \tag{19}$$

with

$$C_{ij} = \frac{1}{\sigma_S^2 (1 - \rho_{ij}^2)} \begin{bmatrix} 1 & -\rho_{ij} \\ -\rho_{ij} & 1 \end{bmatrix}. \tag{20}$$

$f_i(y_i \mid x_i)$, viewed as a function of x_i, assumes the form of (10). Comparing (17) with (2) reveals

$$\phi_i(X_i) = \mathcal{N}^{-1}(\mu_i, V_i) \tag{21}$$

with

---

[5] For example, each node contains heterogeneous sensors.

$$\mu_i = H_i^T \Sigma_i^{-1} y_i + (1 - N(i))\mu_S \big/ \sigma_S^2 \tag{22}$$

$$V_i = H_i^T \Sigma_i^{-1} H_i + (1 - N(i)) \big/ \sigma_S^2 \tag{23}$$

and

$$\psi_{ij}(X_i, X_j) = \mathcal{N}^{-1}(C_{ij}\mu_S [1,1]^T, C_{ij}) . \tag{24}$$

After some manipulation, we have the following message updating and belief updating rules:

$$\begin{cases} \mu_{ij}^n = \dfrac{\rho_{ij}\left(\mu_i + \sum_{k\in N(i)\backslash\{j\}} \mu_{ki}^{n-1} - \rho_{ij}\mu_S \big/ \sigma_S^2(1-\rho_{ij}^2)\right)}{\left(1+\sigma_S^2(1-\rho_{ij}^2)\right)\left(P_i + \sum_{k\in N(i)\backslash\{j\}} V_{ki}^{n-1}\right)} \\ \quad + \rho_{ij}\mu_S \big/ \sigma_S^2(1-\rho_{ij}^2) \\ V_{ij}^n = \dfrac{\left(V_i + \sum_{k\in N(i)\backslash\{j\}} V_{ki}^{n-1} + 1\big/\sigma_S^2\right)}{\left(1+\sigma_S^2(1-\rho_{ij}^2)\right)\left(P_i + \sum_{k\in N(i)\backslash\{j\}} V_{ki}^{n-1}\right)} \end{cases} \tag{25}$$

and

$$\begin{cases} q_i^n = \mu_i + \sum_{k\in N(i)} \mu_{ki}^n \\ W_i^n = V_i + \sum_{k\in N(i)} V_{ki}^n \end{cases} \tag{26}$$

with μ_{ij}^0 and V_{ij}^0 initialized with 0 for all i, j.

V. CONCLUSIONS

In this paper, a new problem, consensus estimation is proposed. The belief propagation algorithm is applied to provide a Bayesian framework facilitating further exploration. We believe that this is a potentially interesting problem that deserves further investigation.

REFERENCES

[1] T. Berger, Z. Zhang, and H. Viswanathan, “The CEO problem,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 887–902, May 1996.

[2] H. Viswanathan and T. Berger, “The quadratic Gaussian CEO problem,” *IEEE Trans. Inform. Theory*, vol. 43, pp. 1549–1559, Sept. 1997.

[3] M. Gastpar and M. Vetterli, “Source-channel communication in sensor networks”, *Proc. 2nd International Workshop on Information Processing in Sensor Networks* (IPSN’03), pp. 162-177, New York, NY, 2003.

[4] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, pp. 63-70, Los Angeles, CA, Apr. 2005.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Gossip algorithms: Design, analysis and applications,” *Proc. INFOCOM*, 2005.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[7] R. J. McEliece, D. MacKay, and J.-F. Cheng, “Turbo decoding as an instance of Pearl's ‘belief propagation’ algorithm,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, February 1998.

[8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, February 2001.

[9] B. Frey. *Graphical Models for Machine Learning and Digital Communication*, MIT Press, 1998.

[10] S. L. Lauritzen, *Graphical Models*, Oxford: Clarendon Press, 1996

[11] D. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, 1989.

[12] Y. Weiss and W. Freeman, “Correctness of belief propagation in Gaussian graphical models of arbitrary topology,” *Neural Computation*, vol. 13, no. 10, pp. 2173-2200, 2001.

[13] M. Alanyali and V. Saligrama, "In-Network decision making via local message passing," *Advances in Pervasive Computing and Networking*, B. K. Szymanski and B. Yener (eds), New York: Springer, 2005.

[14] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Generalized belief propagation", *Advances in Neural Information Processing Systems* (NIPS), vol. 13, pp. 689-695, December 2000.

[15] M. J. Wainwright, T. S. Jaakkola and A. S. Willsky, “Tree-based reparameterization framework for analysis of sum-product and related algorithms,” *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1120-1146, May 2003.