An empirical analysis of loopy belief propagation in three topologies: grids, small-world networks and random graphs

R. Santana, A. Mendiburu and J. A. Lozano
Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country
Paseo Manuel de Lardizábal, 20018 San Sebastian - Donostia, Spain.
rsantana,alexander.mendiburu,ja.lozano@ehu.es

Abstract

Recently, much research has been devoted to the study of loopy belief propagation algorithm. However, little attention has been paid to the change of its behavior in relation with the problem graph topology. In this paper we empirically study the behavior of loopy belief propagation on different network topologies which include grids, small-world networks and random graphs. In our experiments, several descriptors of the algorithm are collected in order to analyze its behavior. We show that the performance of the algorithm is highly sensitive to changes in the topologies. Furthermore, evidence is given showing that the addition of shortcuts to grids can determine important changes in the dynamics of the algorithm.

1 Introduction

Loopy belief propagation (LBP) (Pearl, 1988) is a very efficient message-passing algorithm that has been applied to a variety of inference and optimization problems. One of the factors that influences the accuracy and efficiency of LBP and other message-passing algorithms is the underlying graphical structure (or topology) of the graphical model where the inference algorithm is applied. Although it is known that the existence of cycles in the graph has an impact on the behavior of LBP, little attention has been given to the study of the relationship between other characteristics of the graph topology and the LBP behavior. Moreover, few papers consider possible ways of using the graph topology to adapt the LBP implementation.

A paper that can be considered an exception to this trend is (Ohkubo et al., 2005). It describes a modification of LBP that takes into account the information about the topological heterogeneity of the complex networks where it is applied. It turns out that by modifying the asynchronous message-passing schedule according to the degree of the network vertices it is possible to increase the efficiency of LBP. The use of topological information in the scheduling of the messages could also be seen as another way of conceiving *informed scheduling schemes* of which residual belief propagation (Elidan et al., 2006) is perhaps the best known example.

Natural candidates for analyzing the influence of the graph topology on LBP are complex networks, whose topological characteristics are midway between those of regular lattice or grids and random graphs. The recent surge on the study of complex networks (Watts and Strogatz, 1998) is mainly due to their suitability as a framework for the study of complex systems (Dorogovtsev et al., 2007).

One example of complex networks are small-world networks which simultaneously hold some particular characteristics of regular lattices, such as the existence of local clustering between neighboring vertices, with other attributes characteristic of random networks, such as the short average distance between pairs of vertices. This combination of attributes makes them more appropriate than grids and random graphs to represent interaction networks of real phenomena such as neuronal, social and genetic networks (Watts and Strogatz, 1998), electronic circuits (Ferrer i Cancho et al., 2001), etc.

In this paper, we analyze two kinds of problems related with the use of LBP on networks of different topology:

1. We consider the change in the dynamics of LBP when there is a variation in the graphical structure. In particular, we investigate whether there are significant differences in the behavior of LBP for problems defined in grids, small-world and random graphs.

2. We also investigate whether it is possible to influence the performance and behavior of LBP by modifying the graphical structure without changing the function values.

The paper is organized as follows: In the following section the main concepts related to the class of complex networks and factor graphs are introduced. Section 3 briefly reviews the LBP algorithm and in Section 4 the main characteristics of our implementation, FlexLBP, are explained. In Section 5, the use of LBP across the different classes of chosen topologies is analyzed. Section 6 presents the experiments and analyzes their results. The paper ends in Section 7 where the conclusions and topics for future work are presented.

2 Small-world networks and factor graphs

2.1 Small-world networks

Let $G = (V, E)$ be an undirected graph, where $V = \{v_1, \ldots, v_n\}$ is the set of nodes and $E = \{e_1, \ldots, e_m\}$ is the set of edges between the nodes. We use parameter ϵ_{ij} to represent the existence of an edge between vertices v_i and v_j. $\epsilon_{ij} = 1$ if there exists such an edge, $\epsilon_{ij} = 0$ otherwise. Two nodes connected by an edge are called adjacent and we denote k_i to the degree of a given vertex v_i, which is the number of edges connecting v_i with other nodes. The shortest path length between vertices v_i and v_j is denoted d_{ij}. We assume the graph G is connected and, therefore, d_{ij} is finite and positive $\forall i, j$.

Let $|\Gamma_i|$ be the number of connections between the nearest neighbors of a node $v_i \in V$ and $C_i = \frac{|\Gamma_i|}{k_i(k_i-1)}$, the *clustering coefficient* C is calculated as $C = \frac{1}{n}\sum_i C_i$. The *path length* is calculated as $L = \frac{1}{n(n-1)}\sum_{i \neq j} d_{ij}$.

Small-world networks are characterized by a high clustering coefficient and a small path length. They can be generated by randomly replacing a fraction p of the links of a d-dimensional lattice with new random links. Since the new links decrease the shortest path length between the connected nodes, they are usually called *shortcuts*. The two limiting values of $p = 0$ and $p = 1$ respectively correspond to a regular lattice and a random graph. p is commonly called the *rewiring probability*.

Different patterns can be identified in the connectivity of small-world networks resulting in a classification (Amaral et al., 2000) of these networks in scale-free networks, broad-scale networks and single-scale networks. For more details on small-world and other complex networks, (Amaral et al., 2000; Barthélémy and Amaral, 1999; Dorogovtsev et al., 2007) can be consulted.

2.2 Factor graphs

Factor graphs (Kschischang et al., 2001) are bipartite graphs with two different types of nodes: variable nodes and factor nodes. Each variable node identifies a single variable X_i that can take values from a (usually discrete) domain, while factor nodes f_j represent different functions whose arguments are subsets of variables. This is graphically represented by edges that connect a particular function node with its variable nodes (arguments).

Factor graphs are appropriate to represent those cases in which the joint probability distribution can be expressed as a factorization of several local functions:

$$p(x_1, \ldots, x_n) = \frac{1}{Z}\prod_{j \epsilon J} f_j(\boldsymbol{x_j}) \qquad (1)$$

where $Z = \sum_{\boldsymbol{x}} \prod_{j \epsilon J} f_j(\boldsymbol{x_j})$ is a normalization constant, n is the number of variable nodes, J is a discrete index set, $\boldsymbol{X_j}$ is a subset of $\{X_1, \ldots, X_n\}$, and $f_j(\boldsymbol{x_j})$ is a function containing the variables of $\boldsymbol{X_j}$ as arguments.

The structure of a factor graph can be determined from a given undirected network by associating a factor node to each edge in the network. We use factor graphs as the base graphical for the application of LBP.

3 Loopy belief propagation

LBP works by exchanging messages between nodes. Each node sends and receives messages until a stable situation is reached. Messages, locally calculated by each node, comprise statistical information concerning neighbor nodes.

When applied on factor graphs, two kinds of messages are identified (Yedidia et al., 2005): messages $n_{i\to a}(x_i)$ sent from a variable node i to a factor node a, and messages $m_{a\to i}(x_i)$ sent from a factor node a to a variable node i.

Messages are updated according to the following rules:

$$n_{i\to a}(x_i) := \prod_{c\epsilon N(i)\backslash a} m_{c\to i}(x_i) \quad (2)$$

$$m_{a\to i}(x_i) := \sum_{\boldsymbol{x_a}\backslash x_i} f_a(\boldsymbol{x_a}) \prod_{j\epsilon N(a)\backslash i} n_{j\to a}(x_j) \quad (3)$$

$$m_{a\to i}(x_i) := \arg\max_{\boldsymbol{x_a}\backslash x_i} \{f_a(\boldsymbol{x_a}) \prod_{j\epsilon N(a)\backslash i} n_{j\to a}(x_j)\} \quad (4)$$

where $N(i)\backslash a$ represents all the neighboring factor nodes of node i excluding node a, and $\sum_{\boldsymbol{x_a}\backslash x_i}$ expresses that the sum is completed taking into account all the possible values that all variables except X_i in $\boldsymbol{X}_a$ can take –while variable X_i takes its x_i value.

Equations 2 and 3 are used when marginal probabilities are looked for (sum-product). By contrast, in order to obtain the most probable configurations (max-product), Equations 2 and 4 should be applied.

When the algorithm converges (i.e. messages do not change), marginal functions (sum-product) or max-marginals (max-product), $g_i(x_i)$, are obtained as the normalized product of all messages received by X_i:

$$g_i(x_i) \propto \prod_{a\epsilon N(i)} m_{a\to i}(x_i) \quad (5)$$

Regarding the max-product approach, when the algorithm converges to the most probable value, each variable in the optimal solution is assigned the value given by the configuration with the highest probability at each max-marginal.

4 FlexLBP program

As described in the previous sections, LBP is a widely studied and used algorithm and has been rediscovered and adapted repeatedly to particular problems. Thus, different implementations have been developed since the algorithm was first proposed, although most of them focus on a particular scheduling policy, stopping criterion, etc.

In our LBP implementation (FlexLBP), we have designed a flexible tool, so that researchers can tune different parameters according to the characteristics of the problem they are facing. The FlexLBP implementation has been done following a distributed scheme. That is, each node runs independently, triggered by the message(s) it receives. Additionally, different scheduling policies (asynchronous, synchronous, and even particular rules for individual nodes) can be selected. In addition, the set of nodes that start sending messages can be customized, the order in which messages are processed can be changed, different values can be set for the initial messages, and max-product or sum-product algorithms can be selected. Finally, and due to the distributed scheme, stopping conditions are checked locally (in each node). We have established four different stopping situations: (1) a given maximum number of iterations is reached (that is, calculated messages are different), (2) a node stops because all its neighbors have stopped, (3) message values calculated by a node have not changed in the last

i iterations, and (4) message values calculated by a node follow a periodic (cyclic) sequence.

To investigate the behavior of LBP, we collect a number of statistics that will be used in the analysis of the algorithm behavior. These statistics include information about the four different stopping criteria previously explained.

5 LBP on networks of different topologies

Key to the evaluation of LBP is the determination of the network topologies from which the factor graphs are constructed by associating a factor node to each edge in the graph. We consider two different scenarios to investigate the effects of the network topology. These scenarios are defined by the way the networks are generated.

5.1 From grids to random graphs

We start from a factor graph G^i constructed from a 2-dimensional grid with periodic boundary conditions. In G^i, there is a factor between any pair of nodes that are neighbors in the grid. The number of nodes is $n = \times m$ where m is the dimension of the grid. The number of factors is $2n$. The function values for each of the factors are independently generated.

¿From graph G^i, a collection of factor graphs is generated by rewiring the original edges in the grid of G^i with probability p. To generate a rewired graph from G^i, each edge is visited and a decision about rewiring is made with probability p. If the edge if rewired, the variable nodes in the corresponding factor node are modified but the factor node's function values are kept intact.

5.2 Adding shortcuts

Similar to the previous section, we start from a factor graph G^i constructed from a 2-dimensional grid with periodic boundary conditions. In this case, a collection of factor graphs is generated from graph G^i by adding e edges to the grid of G^i and associating a factor to each edge. Three classes of graphs are created according to the way shortcuts are added:

1. Randomly: Edges are added between two randomly selected nodes.

2. Max. distance: At each step, an edge is added between a pair of nodes at maximum distance in the current network. Every time an edge is added, distances are recalculated.

3. Min. distance: At each step, an edge is added between a pair of nodes at distance 2 in the current network. Every time an edge is added, distances are recalculated.

The different procedures used to add the edges determine differences between the path lengths of graphs belonging to the different groups. Regarding the factor nodes, no matter which method was used for adding the shortcuts, the function values for all the configurations of each of the added factors are set to 1. This means that the contribution of all possible configurations of these factors to the global function will be the same and therefore the maximum configuration of the original graph (respectively the marginals) will not be modified by the addition of the factors. However, the introduction of the new factors (or shortcuts) may have an effect on the LBP dynamics and this is precisely what we would like to identify.

6 Experiments

6.1 Design of the experiments

The starting graph structures used in our experiments are 2-dimensional grids ($m = 7$) with periodic conditions. We use binary variables ($n = 49$) and the maximum size of the factor nodes is 2. Random functions are used. The values corresponding to each factor node entry are generated as $J_{ij} = e^{\beta}$, where β is a value uniformly chosen from $(0, 1)$. To determine whether differences between the behavior of LBP on the different classes of networks are statistically significant the Kruskal-Wallis test (Hsu, 1996) has been employed.

In all the following experiments, the maximum number of messages calculated by each node was set to 2500 and a node is said to converge when the same message value is repeated 500 times.

6.2 Investigating LBP when the rewiring probability is increased

In these experiments we investigate how LBP is affected by the changes in the rewiring probability p. Since the transition to the small-network topology is known to occur for small p values, we generate networks for values of $p \in \{0.01, 0.02, \ldots, 0.1\}$. In addition, and in order to observe the behavior of LBP when p is further increased, we also generate networks for $p \in \{0.2, 0.3, \ldots, 1.0\}$.

We identify each member of the graph collection generated from G^i with a unique assignment to parameters $p \in \{0.01, 0.02, \ldots, 0.1, 0.2, \ldots, 1.0\}$, $inst \in \{1, \ldots, 100\}$. Thus, for each value of p, 100 different graphs are generated by rewiring the edges from G^i with probability p. The total number of factor graphs generated starting from G^i is 1900. Since in our experiments we conducted experiments with 10 initial graphs, i.e. $i \in \{1, \ldots, 10\}$, and the max and sum versions of LBP were used, the total number of LBP runs was $38,000$.

Figures 1, 2 and 3 show the results of max-LBP for different values of the rewiring probability. For each of the descriptors employed, the results were computed as the average among all the runs for the 10 different instances. The used descriptors were: the function value of the best solution found by LPB, the number of nodes that converged and the number of iterations for nodes that converged.

An analysis of Figure 1 reveals the decrease in the average value of the best solution found by max-LBP when the rewiring probability is increased. This may indicate that the type of constraints introduced by higher values of p determine smaller values of the optimum and/or that it is more difficult for max-LBP to find the actual optimum of the functions. Evidence of the difficulties of max-LBP for converging when p is increased emerges from the analysis of Figures 2 and 3. It can be seen that the number of

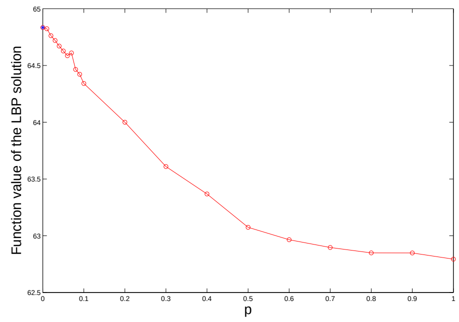


Figure 1: Value of the best solution found by max-LBP for different values of the rewiring probabilities.

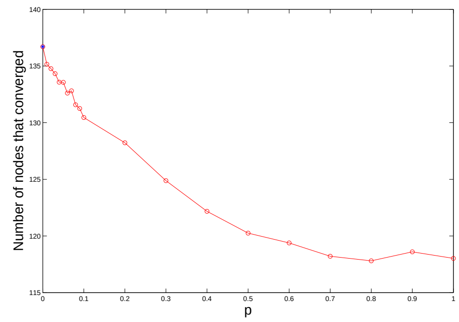


Figure 2: Number of nodes that converged when using max-LBP with different values of the rewiring probabilities.

nodes that converged decreases with p. On the other hand, the number of iterations needed by the nodes that converged is higher. Although the curve describing the number of nodes that converged is clearly monotonically decreasing, it seems to be more pronounced as p goes from 0.1 to 0.5 than for $p > 0.5$. This fact indicates that the behavior of max-LBP is more sensitive to changes around certain values or p.

We conducted similar experiments using sum-LBP for the same set of graphs. As it has been previously reported (Mooij et al., 2007), LBP convergence is easier to achieve for the sum case than for the max case. Figures 4 and 5 respectively show the number of nodes that converge and the average number of iterations for different values of the rewiring probability. Al-

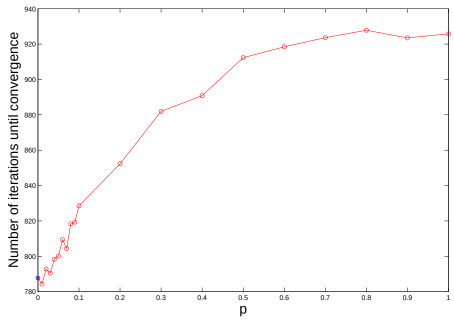


Figure 3: Number of iterations for nodes that converged when using max-LBP with different values of the rewiring probabilities.

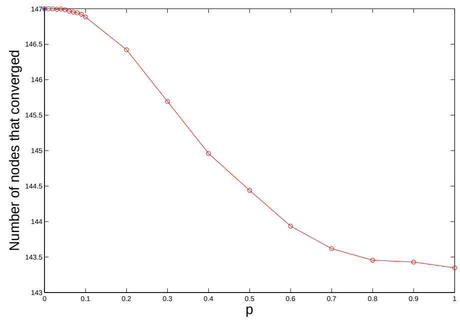


Figure 4: Number of nodes that converged when using sum-LBP with different values of the rewiring probabilities.

though the number of nodes that converge is higher, the curve is also monotonically decreasing. Conversely, the number of iterations until convergence is increased. Therefore, the influence of the rewiring probability seems to be the same for max-LBP and sum-LBP.

Results shown in the previous figures correspond to the average for 10 different network topologies.

For each descriptor we have carried out the Kruskal-Wallis statistical test to compare the LBP results for each possible pair of values of p. For some pairs of values and descriptors, the test did not find statistically significant differences.

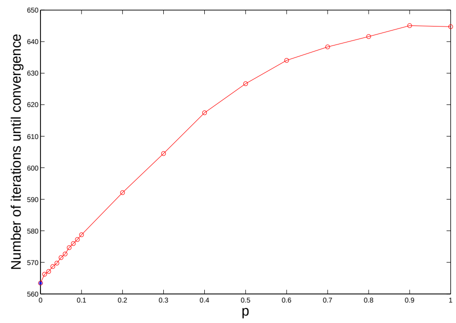


Figure 5: Number of iterations for nodes that converged when using sum-LBP for different values of the rewiring probabilities.

6.3 Investigating the influence of factor additions

The purpose of the following experiments is twofold: First, we investigate the way in which the addition of shortcuts can modify the behavior of LBP. Second, we analyze the influence that the way in which shortcuts are added has in the LBP behavior. We are particularly interested in knowing whether the addition of shortcuts leads to improvements in the results achieved by LBP, i.e. better solutions are obtained or the algorithm converges faster.

The number of added edges was fixed to $e = 10$ and the number of initial graphs to 50. An instance corresponds to a random assignment of the function values to the factor nodes defined on the 2-dimensional grid.

For each of the three methods used to add the edges described in Section 5.2, 50 different graph structures are created by considering 50 possible ways of adding 10 edges to the original grid. The total combination of the initial instances, the methods used to add the edges, and the graph structures generated for each method was $50 \times 3 \times 50 = 7500$. max-LBP and sum-LBP were run on each of these instances.

For each of the instances, we compute the average value of the best solutions found by max-LBP among the 50 graph structures for each of the three methods employed to add the shortcuts. Average values are then compared with the value found by max-LBP in the original graph (without added shortcuts). Of the 50 graphs, the Random, Max. distance and Min. distance methods respectively improve (on average) the values of the function for 9, 9, and 10 of the instances. They respectively converged to the same optimal solution for 22, 22, and 24 instances. These results show that adding shortcuts may have an effect on the quality of the solution obtained, at least in some cases.

To find statistical differences between the behavior of the three methods, the Kruskal-Wallis test was applied using the values of the best solution found. It found statistical significant differences between at least two of the three methods in 13 of the instances. The average gain in the value of the function is shown in Figure 6a. Negative values indicate that the solution found by LBP in the graphs with shortcuts was worse than in the original graph. Of the 13 instances, the method that reduces the path length the most (i.e. Min. distance) was the best in improving the value with respect to the other two methods in 9 of the instances.

On the 37 instances in which the statistical test did not find significant differences between any pair of the methods regarding the quality of the solution obtained, we conducted additional tests to identify significant differences in the number of (factor and vertices) nodes that converged and the number of iterations to convergence. In 21 of the instances, significant differences were found in the number of nodes that converged. The gain in the proportion of nodes that converged with respect to the initial graph is shown in Figure 6b. The Min. distance method achieved a higher proportion of nodes that converged in 18 of the 21 instances. Finally, regarding the number of iterations until convergence, there were significant differences in 20 of the 37 instances and, as shown in Figure 6c), in 12 of them Min. distance achieved a higher number of iterations.

In general, it was observed that max-LBP obtained better solutions and converged in a higher proportion of the nodes of the graphs generated by the Min. distance method than the other two methods. However, the average number of iterations also increased.

Concerning the behavior of sum-LBP, there were not significant differences in the number of nodes that converged. Nevertheless, a different trend to that shown for max-LBP was manifested. As observed in Figure 7, the Min. distance method needed fewer iterations than the other two for all of the instances.

7 Conclusions and future work

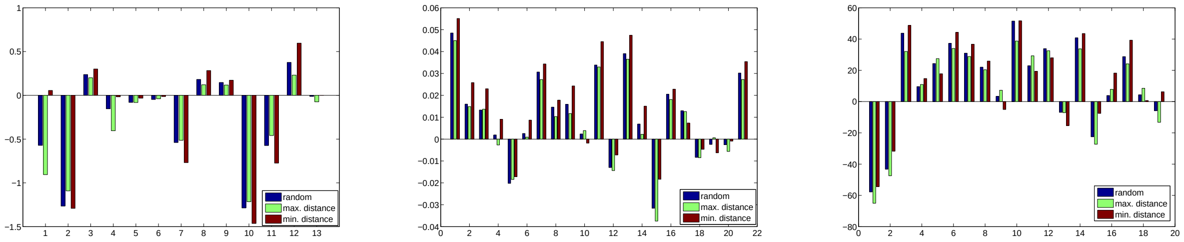
In this paper we have analyzed the effect of the network topology on the behavior of LBP. The empirical analysis of the LBP results for the different graphs considered has shown that finding optimal solutions is harder for LBP when the rewiring probability is increased.

On the other hand, we have shown that adding shortcuts to the initial lattice graphs changes the dynamics of LBP in a less clear way. These changes may determine that better solutions are achieved and/or the number of iterations to convergence can be diminished. The results obtained seem to indicate that the method which reduces the path length the most can lead to more improvements in the LBP behavior than selecting the shortcuts randomly, or choosing them in such a way that the path length is minimally reduced.

Although we have not advanced a method or heuristic for choosing the right shortcuts, i.e. those that can help LBP to escape from cycles or converge to better solutions, we speculate that an influencing factor to this respect is the reduction in the local and global distances determined by the addition of the shortcut. Furthermore, we have just investigated the addition of edges in a step previous to the algorithm start. It might be the case that the adequacy of adding a particular shortcut will change dynamically according to the current step of the LBP algorithm. An open question is then to devise ways to add shortcuts *on line* taking into account the current state of the process.

Acknowledgements

This work has been partially supported by the Etortek, Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2005-03824 and Consolider Ingenio 2010 - CSD2007-00018 projects (Span-



a) b) c)

Figure 6: Influence of the network topology in the behavior of max-LBP when shortcuts are added to an initial grid configuration. a) Improvements in the function values of the best solution found by LPB. b) Number of nodes that converged. c) Number of iterations for nodes that converged.

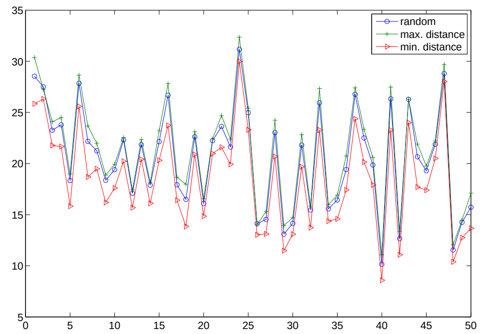


Figure 7: Influence of the network topology in the number of iterations before convergence of sum-LBP when shortcuts are added.

ish Ministry of Education and Science) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

References

L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. 2000. Classes of small-world networks. *Proceedings of the National Academy of Sciences (PNAS)*, 97(21):11149–11152.

M. Barthélémy and L. A. N. Amaral. 1999. Small-world networks: Evidence for a crossover picture. *Physical Review Letters*, 82(15):3180–3183.

S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. 2007. Critical phenomena in complex networks. *arXiv.org*, arXiv:0705.0020v6 [cond-mat.stat-mech], Nov.

G. Elidan, I. McGraw, and D. Koller. 2006. Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006)*, Boston, Massachussetts.

R. Ferrer i Cancho, C. Janssen, and R. V. Solé. 2001. Topology of technology graphs: Small world patterns in electronic circuits. *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, 64:046119.

J. C. Hsu. 1996. *Multiple Comparisons: Theory and Methods*. Chapman and Hall.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

J. M. Mooij, B. Wemmenhove, H. J. Kappen, and T. Rizzo. 2007. Loop corrected belief propagation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*.

J. Ohkubo, M. Yasuda, and K. Tanaka. 2005. Statistical-mechanical iterative algorithms on complex networks. *Physical Review E*, 72(046135):8 pages.

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

D.J. Watts and S.H. Strogatz. 1998. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. 2005. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312.