9

On the Power of Belief Propagation: A Constraint Propagation Perspective

R. DECHTER, B. BIDYUK, R. MATEESCU AND E. ROLLON

1 Introduction

In his seminal paper, Pearl [1986] introduced the notion of Bayesian networks and the first processing algorithm, *Belief Propagation (BP)*, that computes posterior marginals, called beliefs, for each variable when the network is singly connected. The paper provided the foundation for the whole area of Bayesian networks. It was the first in a series of influential papers by Pearl, his students and his collaborators that culminated a few years later in his book on probabilistic reasoning [Pearl 1988]. In his early paper Pearl showed that for singly connected networks (e.g., polytrees) the distributed message-passing algorithm converges to the correct marginals in a number of iterations equal to the diameter of the network. In his book Pearl goes further to suggest the use of BP for loopy networks as an approximation algorithm (see page 195 and exercise 4.7 in [Pearl 1988]). During the decade that followed researchers focused on extending BP to general loopy networks using two principles. The first is tree-clustering, namely, the transformation of a general network into a tree of large-domain variables called clusters on which BP can be applied. This led to the join-tree or junction-tree clustering and to the bucket-elimination schemes [Pearl 1988; Dechter 2003] whose time and space complexity is exponential in the tree-width of the network. The second principle is that of cutset-conditioning that decomposes the original network into a collection of independent singly-connected networks all of which must be processed by BP. The cutset-conditioning approach is time exponential in the network's loop-cutset size and require linear space [Pearl 1988; Dechter 2003].

The idea of applying belief propagation directly to multiply connected networks caught up only a decade after the book was published, when it was observed by researchers in coding theory that high performing probabilistic decoding algorithms such as turbo codes and low density parity-check codes, which significantly outperformed the best decoders at the time, are equivalent to an iterative application of Pearl's belief propagation algorithm [McEliece, MacKay, and Cheng 1998]. This success intrigued researchers and started massive explorations of the potential of these local computation algorithms for general applications. There is now a significant body of research seeking the understanding and improvement of the inference power of iterative belief propagation (IBP).

The early work on IBP showed its convergence for a single loop, provided empirical evidence of its successes and failures on various classes of networks [Rish, Kask, and Dechter 1998; Murphy, Weiss, and Jordan 2000] and explored the relationship between energy min-

imization and belief-propagation shedding light on convergence and stable points [Yedidia, Freeman, and Weiss 2000]. Current state of the art in convergence analysis are the works by [Ihler, Fisher, and Willsky 2005; Mooij and Kappen 2007] that characterize convergence in networks having no determinism. The work by [Roosta, Wainwright, and Sastry 2008] also includes an analysis of the possible effects of strong evidence on convergence which can act to suppress the effects of cycles. As far as accuracy, the work of [Ihler 2007] considers how weak potentials can make the graph sufficiently tree-like to provide error bounds, a work which is extended and improved in [Mooij and Kappen 2009]. For additional information see [Koller 2010].

While a significant progress has been made in understanding the relationship between belief propagation and energy minimization, and while many extensions and variations were proposed, some with remarkable performance (e.g., survey propagation for solving satisfiability for random SAT problems), the following questions remain even now:

- Why does belief propagation work so well on coding networks?
- Can we characterize additional classes of problems for which IBP is effective?
- Can we assess the quality of the algorithm's performance once and if it converges.

In this paper we try to shed light on the power (and limits) of belief propagation algorithms and on the above questions by explicating its relationship with constraint propagation algorithms such as arc-consistency. Our results are relevant primarily to networks that have determinism and extreme probabilities. Specifically, we show that: (1) Belief propagation converges for zero beliefs; (2) All IBP-inferred zero beliefs are correct; (3) IBP's power to infer zero beliefs is as weak and as strong as that of arc-consistency; (4) Evidence and inferred singleton beliefs act like cutsets during IBP's performance. From points (2) and (4) it follows that if the inferred evidence breaks all the cycles, then IBP converges to the exact beliefs for all variables.

Subsequently, we investigate empirically the behavior of IBP for inferred near-zero beliefs. Specifically, we explore the hypothesis that: (5) If IBP infers that the belief of a variable is close to zero then this inference is relatively accurate. We will see that while our empirical results support the hypothesis on benchmarks having no determinism, the results are quite mixed for networks with determinism.

Finally, (6) We investigate if variables that have extreme probabilities in all its domain values (i.e., extreme support) also nearly cut off information flow. If that hypothesis is true, whenever the set of variables with extreme support constitute a loop-cutset, IBP is likely to converge and, if the inferred beliefs for those variables are sound, it will converge to accurate beliefs throughout the network.

On coding networks that posses significant determinism, we do see this desired behavior. So, we could view this hypothesis as the first to provide a plausible explanation to the success of belief propagation on coding networks. In coding networks the channel noise is modeled through a normal distribution centered at the transmitted character and controlled by a small standard deviation. The problem is modeled as a layered belief network whose

sink nodes are all evidence that transmit extreme support to their parents, which constitute all the rest of the variables. The remaining dependencies are functional and arc-consistency on this type of networks is strong and often complete. Alas, as we show, on some other deterministic networks IBP's performance inferring near zero values is utterly inaccurate, and therefore the strength of this explanation is questionable.

The paper is based for the most part on [Dechter and Mateescu 2003] and also on [Bidyuk and Dechter 2001]. The empirical portion of the paper includes significant new analysis of recent empirical evaluations carried on in UAI 2006 and UAI 2008[1].

2 Arc-consistency

DEFINITION 1 (constraint network). A constraint network is a triple $\mathcal{C} = \langle X, D, C \rangle$, where $X = \{X_1, ..., X_n\}$ is a set of variables associated with a set of discrete-valued domains $D = \{D_1, ..., D_n\}$ and a set of constraints $C = \{C_1, ..., C_r\}$. Each constraint C_i is a pair $\langle S_i, R_i \rangle$ where R_i is a relation $R_i \subseteq D_{S_i}$ defined on a subset of variables $S_i \subseteq X$ and D_{S_i} is the Cartesian product of the domains of variables S_i. The relation R_i denotes all tuples of D_{S_i} allowed by the constraint. The projection operator π creates a new relation, $\pi_{S_j}(R_i) = \{x \mid x \in D_{S_j}$ and $\exists y, y \in D_{S_i \setminus S_j}$ and $x \cup y \in R_i\}$, where $S_j \subseteq S_i$. Constraints can be combined with the join operator $\bowtie$, resulting in a new relation, $R_i \bowtie R_j = \{x \mid x \in D_{S_i \cup S_j}$ and $\pi_{S_i}(x) \in R_i$ and $\pi_{S_j}(x) \in R_j\}$.

DEFINITION 2 (constraint satisfaction problem). The constraint satisfaction problem (CSP) defined over a constraint network $\mathcal{C} = \langle X, D, C \rangle$, is the task of finding a solution, that is, an assignment of values to all the variables $x = (x_1, ..., x_n), x_i \in D_i$, such that $\forall C_i \in C, \pi_{S_i}(x) \in R_i$. The set of all solutions of the constraint network $\mathcal{C}$ is $sol(\mathcal{C}) = \bowtie R_i$.

2.1 Describing Arc-Consistency Algorithms

Arc-consistency algorithms belong to the well-known class of constraint propagation algorithms [Mackworth 1977; Dechter 2003]. All constraint propagation algorithms are polynomial time algorithms that are at the center of constraint processing techniques.

DEFINITION 3 (arc-consistency). [Mackworth 1977] Given a binary constraint network $\mathcal{C} = \langle X, D, C \rangle$, $\mathcal{C}$ is arc-consistent iff for every binary constraint $R_i \in C$ s.t. $S_i = \{X_j, X_k\}$, every value $x_j \in D_j$ has a value $x_k \in D_k$ s.t. $(x_j, x_k) \in R_i$.

When a binary constraint network is not arc-consistent, arc-consistency algorithms remove values from the domains of the variables until an arc-consistent network is generated. A variety of such algorithms were developed over the past three decades [Dechter 2003]. We will consider here a simple and not the most efficient version, which we call *relational distributed arc-consistency* algorithm. Rather than defining it on binary constraint networks we will define it directly over the dual graph, extending the arc-consistency condition to non-binary networks.

DEFINITION 4 (dual graph). Given a set of functions/constraints $F = \{f_1, ..., f_r\}$ over scopes $S_1, ..., S_r$, the dual graph of F is a graph $\mathcal{D}_F = (V, E, L)$ that associates a node

[1] http://graphmod.ics.uci.edu/uai08/Evaluation/Report

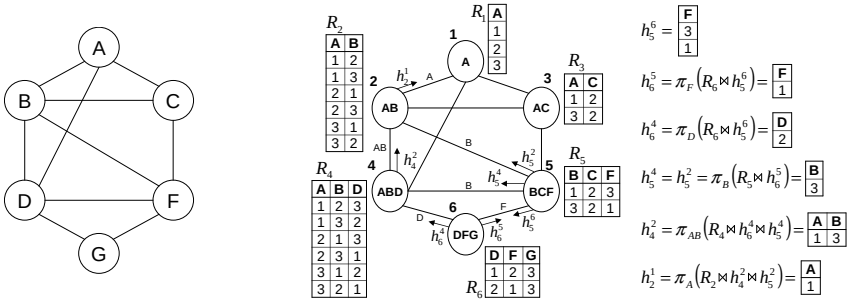


Figure 1. Part of the execution of RDAC algorithm

with each function, namely $V = F$, and an arc connects any two nodes whose scope share a variable, $E = \{(f_i, f_j)|S_i \cap S_j \neq \emptyset\}$. L is a set of labels for the arcs, where each arc is labeled by the shared variables of its nodes, $L = \{l_{ij} = S_i \cap S_j|(i,j) \in E\}$.

Algorithm *Relational distributed arc-consistency* (RDAC) is a message passing algorithm defined over the dual graph $\mathcal{D}_C$ of a constraint network $\mathcal{C} = \langle X, D, C\rangle$. It enforces what is known as *relational arc-consistency* [Dechter 2003]. Each node/constraint in $\mathcal{D}_{C_i}$, for a constraint $C_i \in \mathcal{C}$ maintains a current set of viable tuples R_i. Let $ne(i)$ be the set of neighbors of C_i in $\mathcal{D}_C$. Every node C_i sends a message to any node $C_j \in ne(i)$, which consists of the tuples over their label variables l_{ij} that are allowed by the current relation R_i. Formally, let R_i and R_j be two constraints sharing scopes, whose arc in $\mathcal{D}_C$ is labeled by l_{ij}. The message that R_i sends to R_j denoted h_i^j is defined by:

$$(1) \quad h_i^j \leftarrow \pi_{l_{ij}}(R_i \bowtie (\bowtie_{k \in ne(i)} h_k^i))$$

and each node updates its current relation according to:

$$(2) \quad R_i \leftarrow R_i \bowtie (\bowtie_{k \in ne(i)} h_k^i)$$

EXAMPLE 5. Figure 1 describes part of the execution of RDAC for a graph coloring problem, having the constraint graph shown on the left. All variables have the same domain, {1,2,3}, except for variable C whose domain is 2, and variable G whose domain is 3. The arcs correspond to *not equal* constraints, and the relations are R_A, R_{AB}, R_{AC}, R_{ABD}, R_{BCF}, R_{DFG}, where the subscript corresponds to their scopes. The dual graph of this problem is given on the right side of the figure, and each table shows the initial constraints (there are unary, binary and ternary constraints). To initialize the algorithm, the first messages sent out by each node are universal relations over the labels. For this example, RDAC actually solves the problem and finds the unique solution A=1, B=3, C=2, D=2, F=1, G=3.

Relational distributed arc-consistency algorithm converges after $O(r \cdot t)$ iterations to the largest relational arc-consistent network that is equivalent to the original network, where r is the number of constraints and t bounds the number of tuples in each constraint. Its complexity can be shown to be $O(r^2 t^2 \log t)$ [Dechter 2003].

3 Iterative Belief Propagation

DEFINITION 6 (belief network). A belief network is a quadruple $\mathcal{B} = \langle X, D, G, P \rangle$ where $X = \{X_1, \ldots, X_n\}$ is a set of random variables, $D = \{D_1, ..., D_n\}$ is the set of the corresponding domains, $G = (X, E)$ is a directed acyclic graph over X and $P = \{p_1, ..., p_n\}$ is a set of conditional probability tables (CPTs) $p_i = P(X_i|pa(X_i))$, where $pa(X_i)$ are the parents of X_i in G. The belief network represents a probability distribution over X having the product form $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_{pa(X_i)})$. An evidence set e is an instantiated subset of variables. The family of X_i, denoted by $fa(X_i)$, includes X_i and its parent variables. Namely, $fa(X_i) = \{X_i\} \cup pa(X_i)$.

DEFINITION 7 (belief updating problem). The belief updating problem defined over a belief network $\mathcal{B} = \langle X, D, G, P \rangle$ is the task of computing the posterior probability $P(Y|e)$ of *query* nodes $Y \subseteq X$ given evidence e. We will sometime denote by P_B the exact probability according the Baysian network B. When Y consists of a single variable X_i, $P_B(X_i|e)$ is also denoted as $Bel(X_i)$ and called belief, or posterior marginal, or just marginal.

3.1 Describing Iterative Belief Propagation

Iterative belief propagation (IBP) is an iterative application of Pearl's algorithm that was defined for poly-trees [Pearl 1988]. Since it is a distributed algorithm, it is well defined for any network. We will define IBP as operating over the belief network's dual join-graph.

DEFINITION 8 (dual join-graph). Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, a dual join-graph is an arc subgraph of the dual graph $\mathcal{D}_B$ whose arc labels are subsets of the labels of $\mathcal{D}_B$ satisfying the *running intersection property*, namely, that any two nodes that share a variable in the dual join-graph be connected by a path of arcs whose labels *contain* the shared variable. An *arc-minimal* dual join-graph is one for which none of the labels can be further reduced while maintaining the running intersection property.

In IBP each node in the dual join-graph sends a message over an adjacent arc whose scope is identical to its label. Pearl's original algorithm sends messages whose scopes are singleton variables only. It is easy to show that any dual graph (which itself is a dual join-graph) has an arc-minimal singleton dual join-graph which can be constructed directly by labeling the arc between the CPT of a variable and the CPT of its parent, by its parent variable. Algorithm IBP defined for any dual join-graph is given in Figure 2. One iteration of IBP is time and space linear in the size of the belief network, and when IBP is applied to the singleton labeled dual graph it coincides with Pearl's belief propagation. The inferred approximation of belief $P(X|e)$ output by IBP, will be denoted by $P_{IBP}(X|e)$.

4 Belief Propagation's Inferred Zeros

We will now make connections between distributed relational arc-consistency and iterative belief propagation. We first associate any belief network with a constraint network that captures its zero probability tuples and define algorithm IBP-RDAC, an IBP-like algorithm that achieves relational arc-consistency on the associated constraint network. Then, we show that IBP-RDAC and IBP are equivalent in terms of removing inconsistent domain

Algorithm **IBP**
Input: An arc-labeled dual join-graph $DJ = (V, E, L)$ for a belief network $\mathcal{B} = \langle X, D, G, P \rangle$. Evidence e.
Output: An augmented graph whose nodes include the original CPTs and the messages received from neighbors. Approximations of $P(X_i|e)$ and $P(fa(X_i)|e), \forall X_i \in X$.
Denote by: h_u^v the message from u to v; $ne(u)$ the neighbors of u in V; $ne_v(u) = ne(u) - \{v\}$; l_{uv} the label of $(u, v) \in E$; $elim(u, v) = fa(X_i) - fa(X_j)$, where u and v are the vertexs of family $fa(X_i)$ and $fa(X_j)$ in DJ, respectively.
• **One iteration of IBP**
For every node u in DJ in a topological order and back, do:
1. **Process observed variables**
Assign evidence variables to each p_i and remove them from the labeled arcs.
2. **Compute and send to v the function:**

$$h_u^v = \sum_{elim(u,v)} (p_u \cdot \prod_{\{h_i^u, i \in ne_v(u)\}} h_i^u)$$

EndFor
• **Compute approximations of $P(X_i|e)$ and $P(fa(X_i)|e)$:**
For every $X_i \in X$ (let u be the vertex of family $fa(X_i)$ in DJ), do:
$P(fa(X_i)|e) = \alpha(\prod_{h_i^u, u \in ne(i)} h_i^u) \cdot p_u$
$P(X_i|e) = \alpha \sum_{fa(X_i) - \{X_i\}} P(fa(X_i)|e)$
EndFor

Figure 2. Algorithm Iterative Belief Propagation

values and computing zero marginal probabilities, respectively. Since arc-consistency algorithms are well understood, this correspondence between IBP-RDAC and IBP yields the main claims and provides insight into the behavior of IBP for inferred zero beliefs. In particular, this relationship justifies the iterative application of belief propagation algorithms, while also illuminates their "distance" from being complete.

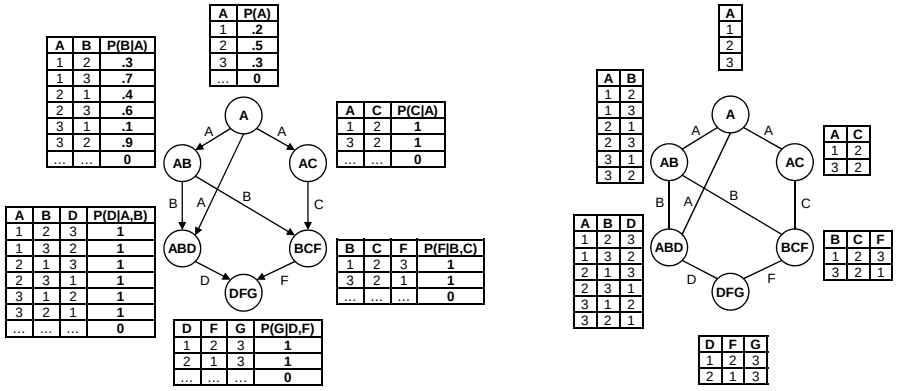
More precisely, in this section we will show that: (a) If a variable-value pair is assessed in some iteration by IBP as having a zero-belief, it remains zero in subsequent iterations; (b) Any IBP-inferred zero-belief is correct with respect to the corresponding belief network's marginal; and (c) IBP converges in finite time for all its inferred zeros.

4.1 Flattening the Belief Network

Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, we define the flattening of a belief network $\mathcal{B}$, called $flat(\mathcal{B})$, as the constraint network where all the zero entries in a probability table are removed from the corresponding relation. Formally,

DEFINITION 9 (flattening). Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, its flattening is a constraint network $flat(\mathcal{B}) = \langle X, D, flat(P) \rangle$. Each CPT $p_i \in P$ over $fa(X_i)$ is associated with a constraint $\langle S_i, R_i \rangle$ s.t. $S_i = fa(X_i)$ and $R_i = \{(x_i, x_{pa(X_i)}) \in D_{S_i} | P(x_i|x_{pa(X_i)}) > 0\}$ The set $flat(P)$ is the set of the constraints $\langle S_i, R_i \rangle, \forall p_i \in P$.

EXAMPLE 10. Figure 3 shows (a) a belief network and (b) its corresponding flattening.



(a) Bayesian network (b) Flat constraint network

Figure 3. Flattening of a Bayesian network

THEOREM 11. *Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, where $X = \{X_1, \ldots, X_n\}$, for any tuple $x = (x_1, \ldots, x_n)$: $P_{\mathcal{B}}(x) > 0 \Leftrightarrow x \in sol(flat(\mathcal{B}))$, where $sol(flat(\mathcal{B}))$ is the set of solutions of $flat(B)$.*

Proof. $P_{\mathcal{B}}(x) > 0 \Leftrightarrow \Pi_{i=1}^{n} P(x_i|x_{pa(X_i)}) > 0 \Leftrightarrow \forall i \in \{1, \ldots, n\},\ P(x_i|x_{pa(X_i)}) > 0 \Leftrightarrow \forall i \in \{1, \ldots, n\},\ (x_i, x_{pa(X_i)}) \in R_{F_i} \Leftrightarrow x \in sol(flat(\mathcal{B}))$. □

Clearly this can extend to Bayesian network with evidence:

COROLLARY 12. *Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, and evidence e $P_B(x|e) > 0 \Leftrightarrow x \in sol(flat(B) \wedge e)$.*

We next define algorithm IBP-RDAC and show that it achieves relational arc-consistency on the flat network.

DEFINITION 13 (Algorithm IBP-RDAC). Given $\mathcal{B} = \langle X, D, G, P \rangle$ and evidence e, let $\mathcal{D}_{\mathcal{B}}$ be a dual join-graph and $\mathcal{D}_{flat(\mathcal{B})}$ be a corresponding dual join-graph of the constraint network $flat(\mathcal{B})$. Algorithm IBP-RDAC applied to $\mathcal{D}_{flat(\mathcal{B})}$ is defined using IBP's specification in Figure 2 with the following modifications:

1. Pre-processing evidence: when processing evidence, we remove from each $R_i \in flat(P)$ those tuples that do not agree with the assignments in evidence e.
2. Instead of $\prod$, we use the join operator $\bowtie$.
3. Instead of $\sum$, we use the projection operator π.
4. At the termination, we update the domains of variables by:

$$D_i \leftarrow D_i \cap \pi_{X_i}((\bowtie_{v \in ne(u)} h_{(v,u)}) \bowtie R_i)$$

By construction, it should be easy to see that,

PROPOSITION 14. *Given a belief network $\mathcal{B} = \langle X, D, G, P \rangle$, algorithm IBP-RDAC is identical to algorithm $RDAC$ when applied to $\mathcal{D}_{flat(\mathcal{B})}$. Therefore, IBP-RDAC enforces relational arc-consistency over $flat(\mathcal{B})$.*

Due to the convergence of RDAC, we get that:

PROPOSITION 15. *Given a belief network $\mathcal{B}$, algorithm IBP-RDAC over $flat(\mathcal{B})$ converges in $O(n \cdot t)$ iterations, where n is the number of nodes in $\mathcal{B}$ and t is the maximum number of tuples over the labeling variables between two nodes that have positive probability.*

4.2 The Main Claim

In the following we will establish an equivalence between IBP and IBP-RDAC in terms of zero probabilities.

PROPOSITION 16. *When IBP and IBP-RDAC are applied in the same order of computation to $\mathcal{B}$ and $flat(\mathcal{B})$ respectively, the messages computed by IBP are identical to those computed by IBP-RDAC in terms of zero / non-zero probabilities. That is, for any pair of corresponding messages, $h_{(u,v)}(t) \neq 0$ in IBP iff $t \in h_{(u,v)}$ in IBP-RDAC.*

Moving from tuples to domain values, we will show that whenever IBP computes a marginal probability $P_{IBP}(x_i|e) = 0$, IBP-RDAC removes x_i from the domain of variable X_i, and vice-versa.

PROPOSITION 17. *Given a belief network $\mathcal{B}$ and evidence e, IBP applied to $\mathcal{B}$ derives $P_{IBP}(x_i|e) = 0$ iff IBP-RDAC over $flat(\mathcal{B})$ decides that $x_i \notin D_i$.*

We can now conclude that:

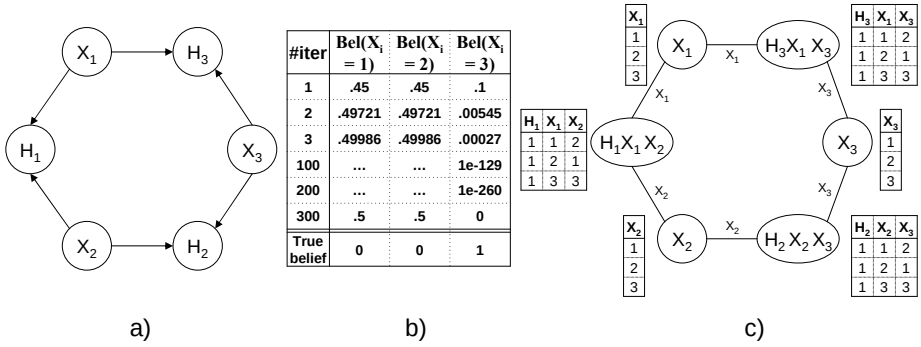
THEOREM 18. *Given evidence e, whenever IBP applied to $\mathcal{B}$ infers that $P_{IBP}(x_i|e) = 0$, the marginal $Bel(x_i) = P_{\mathcal{B}}(x_i|e) = 0$.*

Proof. By Proposition 17, if IBP over $\mathcal{B}$ computes $P_{IBP}(x_i|e) = 0$, then IBP-RDAC over $flat(\mathcal{B})$ removes the value x_i from the domain D_i. Therefore, $x_i \in D_i$ is a no-good of the constraint network $flat(\mathcal{B})$ and from Theorem 11 it follows that $Bel(x_i) = 0$. □

Next, we show that the time it takes IBP to find its inferred zeros is bounded.

PROPOSITION 19. *Given a belief network $\mathcal{B}$ and evidence e, IBP finds all its x_i for which $P_{IBP}(x_i|e) = 0$ in finite time, that is, there exists a number k such that no $P_{IBP}(x_i|e) = 0$ will be generated after k iterations.*

Proof. This follows from the fact that the number of iterations it takes for IBP to compute $P_{IBP}(X_i = x_i|e) = 0$ over $\mathcal{B}$ is exactly the same number of iterations IBP-RDAC needs to remove x_i from the domain D_i over $flat(\mathcal{B})$ (Propositions 16 and 17) and the fact that IBP-RDAC's number of iterations is bounded (Proposition 15). □



#iter	Bel(X_i = 1)	Bel(X_i = 2)	Bel(X_i = 3)
1	.45	.45	.1
2	.49721	.49721	.00545
3	.49986	.49986	.00027
100	...	...	1e-129
200	...	...	1e-260
300	.5	.5	0
True belief	0	0	1

Figure 4. a) A belief network; b) Example of a finite precision problem; and (c) An arc-minimal dual join-graph.

4.3 A Finite Precision Problem

Algorithms should always be implemented with care on finite precision machines. In the following example we show that IBP's messages converge in the limit (i.e. in an infinite number of iterations), but they do not stabilize in any finite number of iterations.

EXAMPLE 20. Consider the belief network in Figure 4a defined over 6 variables X_1, X_2, X_3, H_1, H_2, H_3. The domain of the X variables is $\{1, 2, 3\}$ and the domain of the H variables is $\{0, 1\}$. The priors on X variables are:

$$P(X_i) = \begin{cases} 0.45, & if \ \ X_i = 1; \\ 0.45, & if \ \ X_i = 2; \\ 0.1, & if \ \ X_i = 3; \end{cases}$$

There are three CPTs over the scopes: $\{H_1, X_1, X_2\}$, $\{H_2, X_2, X_3\}$, and $\{H_3, X_1, X_3\}$. The values of the CPTs for every triplet of variables $\{H_k, X_i, X_j\}$ are:

$$\begin{aligned} P(h_k = 1|x_i, x_j) &= \begin{cases} 1, & if \ \ (3 \neq x_i \neq x_j \neq 3); \\ 1, & if \ \ (x_i = x_j = 3); \\ 0, & otherwise \ ; \end{cases} \\ P(h_k = 0|x_i, x_j) &= 1 - P(h_k = 1|x_i, x_j). \end{aligned}$$

Consider the evidence set $e = \{H_1 = H_2 = H_3 = 1\}$. This Bayesian network expresses the probability distribution that is concentrated in a single tuple:

$$P(x_1, x_2, x_3|e) = \begin{cases} 1, & if \ \ x_1 = x_2 = x_3 = 3; \\ 0, & otherwise. \end{cases}$$

The belief for any of the X variables as a function of the number of iteration is given in Figure 4b. After about 300 iterations, the finite precision of our computer is not able to represent the value for $Bel(X_i = 3)$, and this appears to be zero, yielding the final updated belief $(.5, .5, 0)$, when in fact the true updated belief should be $(0, 0, 1)$. Notice

that $(.5, .5, 0)$ cannot be regarded as a legitimate fixed point for IBP. Namely, if we would initialize IBP with the values $(.5, .5, 0)$, then the algorithm would maintain them, appearing to have a fixed point. However, initializing IBP with zero values cannot be expected to be correct. Indeed, when we initialize with zeros we forcibly introduce determinism in the model, and IBP will always maintain it afterwards.

However, this example does not contradict our theory because, mathematically, $Bel(X_i = 3)$ never becomes a true zero, and IBP never reaches a quiescent state. The example shows however that a close to zero inferred belief by IBP can be arbitrarily inaccurate. In this case the inaccuracy seems to be due to the initial prior belief which are so different from the posterior ones.

4.4 Zeros Inferred by Generalized Belief Propagation

Belief propagation algorithms were extended yielding the class of *generalized belief propagation* (GBP) algorithms [Yedidia, Freeman, and Weiss 2000]. These algorithms fully process subparts of the networks, transforming it closer to a tree structure on which IBP can be more effective [Dechter, Mateescu, and Kask 2002; Mateescu, Kask, Gogate, and Dechter 2010]. The above results for IBP can now be extended to GBP and in particular to the variant of *iterative join-graph propagation*, IJGP [Dechter, Mateescu, and Kask 2002]. The algorithm applies message passing over a partition of the CPTs into clusters, called a join-graph, rather than over the dual graph. The set of clusters in such a partition defines a unique dual graph (i.e., each cluster is a node). This dual graph can be associated with various dual join-graphs, each defined by the labeling on the arcs between neighboring cluster nodes.

Algorithm IJGP has an accuracy parameter i, called i-bound, which restricts the maximum number of variables that can appear in a cluster and it is more accurate as i grows. The extension of all the previous observations regarding zeros to IJGP is straightforward and is summarized next, where the inferred approximation of the belief $P_{calB}(X_i|e)$ computed by IJGP is denoted by $P_{IJGP}(X_i|e)$.

THEOREM 21. *Given a belief network $\mathcal{B}$ to which IJGP is applied then:*

1. *IJGP generates all its $P_{IJGP}(x_i|e) = 0$ in finite time, that is, there exists a number k, such that no $P_{IJGP}(x_i) = 0$ will be generated after k iterations.*

2. *Whenever IJGP determines $P_{IJGP}(x_i|e) = 0$, it stays 0 during all subsequent iterations.*

3. *Whenever IJGP determines $P_{IJGP}(x_i|e) = 0$, then $Bel(x_i) = 0$.*

5 The Impact of IBP's Inferred Zeros

This section discusses the ramifications of having sound inferred zero beliefs.

5.1 The Inference Power of IBP

We now show that the inference power of IBP for zeros is sometimes very limited and other times strong, exactly wherever arc-consistency is weak or strong.

Cases of weak inference power. Consider the belief network described in Example 20. The flat constraint network of that belief network is defined over the scopes S_1=$\{H_1, X_1, X_2\}$, S_2=$\{H_2, X_2, X_3\}$, S_3=$\{H_3, X_1, X_3\}$. The constraints are defined by: $R_{S_i} = \{(1,1,2), (1,2,1), (1,3,3), (0,1,1), (0,1,3), (0,2,2), (0,2,3), (0,3,1), (0,3,2)\}$. The prior probabilities for X_i's imply unary constraints equal to the full domain $\{1,2,3\}$. An arc-minimal dual join-graph that is identical to the constraint network is given in Figure 4b. In this case, IBP-RDAC sends as messages the full domains of the variables and thus no tuple is removed from any constraint. Since IBP infers the same zeros as arc-consistency, IBP will also *not* infer any zeros. Since the true probability of most tuples is zero, we can conclude that the inference power of IBP on this example is weak or non-existent.

The weakness of arc-consistency in this example is not surprising. Arc-consistency is known to be far from complete. Since every constraint network can be expressed as a belief network (by adding a variable for each constraint as we did in the above example) and since arc-consistency can be arbitrarily weak on some constraint networks, so could be IBP.

Cases of strong inference power. The relationship between IBP and arc-consistency ensures that IBP is zero-complete, whenever arc-consistency is. In general, if for a flat constraint network of a belief network $\mathcal{B}$, arc-consistency removes all the inconsistent domain values, then IBP will also discover all the true zeros of $\mathcal{B}$. Examples of constraint networks that are complete for arc-consistency are max-closed constraints. These constraints have the property that if 2 tuples are in the relation so is their intersection. Linear constraints are often max-closed and so are Horn clauses (see [Dechter 2003]). Clearly, IBP is zero complete for acyclic networks which include binary trees, polytrees and networks whose dual graph is a hypertree [Dechter 2003]. This is not too illuminating though as we know that IBP is fully complete (not only for zeros) for such networks.

An interesting case is when the belief network has no evidence. In this case, the flat network always corresponds to the *causal constraint network* defined in [Dechter and Pearl 1991]. The inconsistent tuples or domain values are already explicitly described in each relation and no new zeros can be inferred. What is more interesting is that in the absence of evidence IBP is also complete for non-zero beliefs for many variables as we show later.

5.2 IBP and Loop-Cutset

It is well-known that if evidence nodes form a loop-cutset, then we can transform any multiply-connected belief network into an equivalent singly-connected network which can be solved by belief propagation, leading to the loop-cutset conditioning method [Pearl 1988]. Now that we established that inferred zeros, and in particular inferred evidence (i.e., when only a single value in the domain of a variable has a non-zero probability) are sound, we show that evidence play the cutset role automatically during IBP's performance.

Indeed, we can show that during IBP's operation, an observed node X_i in a Bayesian network blocks the path between its parents and its children as defined in the d-separation criteria. All the proofs of claims appearing in Section 5.2 and Section 5.3 can be found in [Bidyuk and Dechter 2001].

PROPOSITION 22. *Let X_i be an observed node in a belief network $\mathcal{B}$. Then for any child Y_j of node X_i, the belief of Y_j computed by IBP is not dependent on the messages that X_i receives from its parents $pa(X_i)$ or the messages that node X_i receives from its other children $Y_k, k \neq j$.*

From this we can conclude that:

THEOREM 23. *If evidence nodes, original or inferred, constitute a loop-cutset, then IBP converges to the correct beliefs in linear time.*

5.3 IBP on Irrelevant Nodes

An orthogonal property is that unobserved nodes that have only unobserved descendents are irrelevant to the beliefs of the remaining nodes and therefore, processing can be restricted to the relevant subgraphs. In IBP, this property is expressed by the fact that irrelevant nodes send messages to their parents that equally support each value in the domain of a parent and thus do not affect the computation of marginal posteriors of its parents.

PROPOSITION 24. *Let X_i be an unobserved node without observed descendents in $\mathcal{B}$ and let $\mathcal{B}'$ be a subnetwork obtained by removing X_i and its descendents from $\mathcal{B}$. Then, $\forall Y \in \mathcal{B}'$ the belief of Y computed by IBP over $\mathcal{B}$ equals the belief of Y computed by IBP over $\mathcal{B}'$.*

Thus, in a loopy network without evidence, IBP always converges after 1 iteration since only propagation of top-down messages affects the computation of beliefs and those messages do not change. Also in that case, IBP converges to the correct marginals for any node X_i such that there exists only one directed path from any ancestor of X_i to X_i. This is because the relevant subnetwork that contains only the node and its ancestors is singly-connected and by Proposition 24 they are the same as the beliefs computed by applying IBP to the complete network. In summary,

THEOREM 25. *Let $\mathcal{B}'$ be a subnetwork obtained from $\mathcal{B}$ by recursively eliminating all its unobserved leaf nodes. If observed nodes constitute a loop-cutset of $\mathcal{B}'$, then IBP applied to $\mathcal{B}$ converges to the correct beliefs for all nodes in $\mathcal{B}'$.*

THEOREM 26. *If a belief network does not contain any observed nodes or only has observed root nodes, then IBP always converges.*

In summary, in Sections 5.2 and 5.3 we observed that IBP exploits the two properties of observed and unobserved nodes, *automatically*, without any outside intervention for network transformation. As a result, the correctness and convergence of IBP on a node X_i in a multiply-connected belief network will be determined by the structure restricted to X_i's relevant subgraph. If the relevant subnetwork of X_i is singly-connected relative to the evidence (observed or inferred), IBP will converge to the correct beliefs for node X_i.

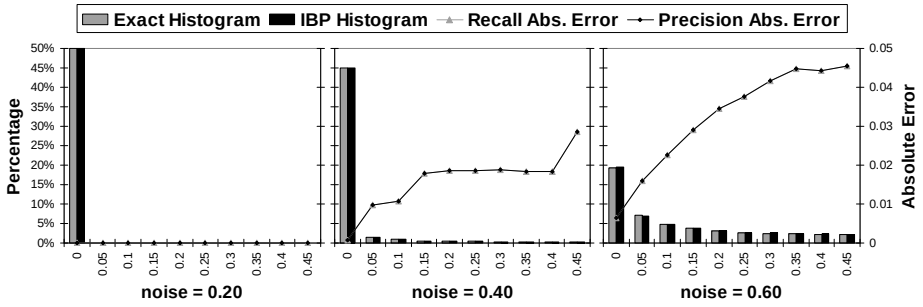


Figure 5. Coding, N=200, evidence=100, w*=15, 1000 instances.

6 Experimental Evaluation

The goal of the experiments is two-fold. First, since zero values inferred by IBP/IJGP are proved correct, we want to explore the behavior of IBP/IJGP for near zero inferred beliefs. Second, we want to explore the hypothesis that the loop-cutset impact on IBP's performance, as discussed in Section 5.2, also extends to variables with extreme support. The next two subsections are devoted to these two issues, respectively.

6.1 On the Accuracy of IBP in Near Zero Marginals

We test the performance of IBP and IJGP both on cases of strong and weak inference power. In particular, we look at networks where probabilities are extreme and investigate empirically the accuracy of IBP/IJGP across the range of belief values from 0 to 1. Since zero values inferred by IBP/IJGP are proved correct, we focus especially on the behavior of IBP/IJGP for near zero inferred beliefs.

Using names inspired by the well known measures in information retrieval, we report *Recall Absolute Error* and *Precision Absolute Error* over small intervals spanning $[0, 1]$. *Recall* is the absolute error averaged over all the exact beliefs that fall into the interval, and can therefore be viewed as capturing the level of completeness. For *precision*, the average is taken over all the belief values computed by IBP/IJGP that fall into the interval, and can be viewed as capturing soundness.

The X coordinate in Figure 5 and Figure 10 denotes the interval $[X, X + 0.05)$. For the rest of the figures, the X coordinate denotes the interval $(X - 0.05, X]$, where the 0 interval is $[0, 0]$. The left Y axis corresponds to the histograms (the bars), while the right Y axis corresponds to the absolute error (the lines). For problems with binary variables, we only show the interval [0, 0.5] because the graphs are symmetric around 0.5. The number of variables, number of evidence variables and induced width w* are reported in each graph.

Since the behavior within each benchmark is similar, we report a subset of the results (for an extended report see [Rollon and Dechter 2009].

Coding networks. Coding networks are the famous case where IBP has impressive performance. The instances are from the class of linear block codes, with 50 nodes per layer and 3 parent nodes for each variable. We experiment with instances having three differ-

ent values of channel noise: 0.2, 0.4 and 0.6. For each channel value, we generate 1000 samples.

Figure 5 shows the results. When the noise level is 0.2, all the beliefs computed by IBP are extreme. The Recall and Precision are very small, of the order of 10^{-11}. So, in this case, all the beliefs are very small (i.e., ϵ small) and IBP is able to infer them correctly, resulting in almost perfect accuracy (IBP is indeed perfect in this case for the bit error rate). As noise increases, the Recall and Precision get closer to a bell shape, indicating higher error for values close to 0.5 and smaller error for extreme values. The histograms show that fewer belief values are extreme as noise increases.

Linkage Analysis networks. Genetic linkage analysis is a statistical method for mapping genes onto a chromosome. The problem can be modeled as a belief network. We experimented with four *pedigree* instances from the UAI08 competition. The domain size ranges between 1 to 4. For these instances exact results are available. Figure 6 shows the results. We observe that the number of exact 0 beliefs is small and IJGP correctly infers all of them. The behavior of IJGP for ϵ small beliefs varies accross instances. For *pedigree1*, the Exact and IJGP histograms are about the same (for all intervals). Moreover, Recall and Precision errors are relatively small. For the rest of the instances, the accuracy of IJGP for extreme inferred marginals decreases. Notice that IJGP infers more ϵ small beliefs than the number of exact extremes in the corresponding intervals, leading to relatively high Precision error while small Recall error. The behaviour for beliefs in the 0.5 interval is reversed, leading to high Recall error while small Precision error. As expected, the accuracy of IJGP improves as the value of the control parameter i-bound increases.

Grid networks. Grid networks are characterized by two parameters (N, D), where $N \times N$ is the size of the network and D is the percentage of determinism (i.e., the percentage of values in all CPTs assigned to either 0 or 1). We experiment with *grids2* instances from the UAI08 competition. They are characterized by parameters $(\{16, \ldots, 42\}, \{50, 75, 90\})$. For each parameter configuration, there are samples of size 10 generated by randomly assigning value 1 to one leaf node.

Figure 7 and Figure 8 report the results. IJGP correctly infers all 0 beliefs. However, its performance for ϵ small beliefs is quite poor. Only for networks with parameters $(16, 50)$ the Precision error is relatively small (less than 0.05). If we fix the size of the network and the i-bound, both Precision and Recall errors increase as the determinism level D increases. The histograms clearly show the gap between the number of true ϵ small beliefs and the ones inferred by IJGP. As before, the accuracy of IJGP improves as the value of the control parameter i-bound increases.

Two-layer noisy-OR networks. Variables are organized in two layers where the ones in the second layer have 10 parents. Each probability table represents a noisy OR-function. Each parent variable y_j has a value $P_j \in [0..P_{noise}]$. The CPT for each variable in the second layer is then defined as, $P(x = 0|y_1, \ldots, y_P) = \prod_{y_j=1} P_j$ and $P(x = 1|y_1, \ldots, y_P) =$

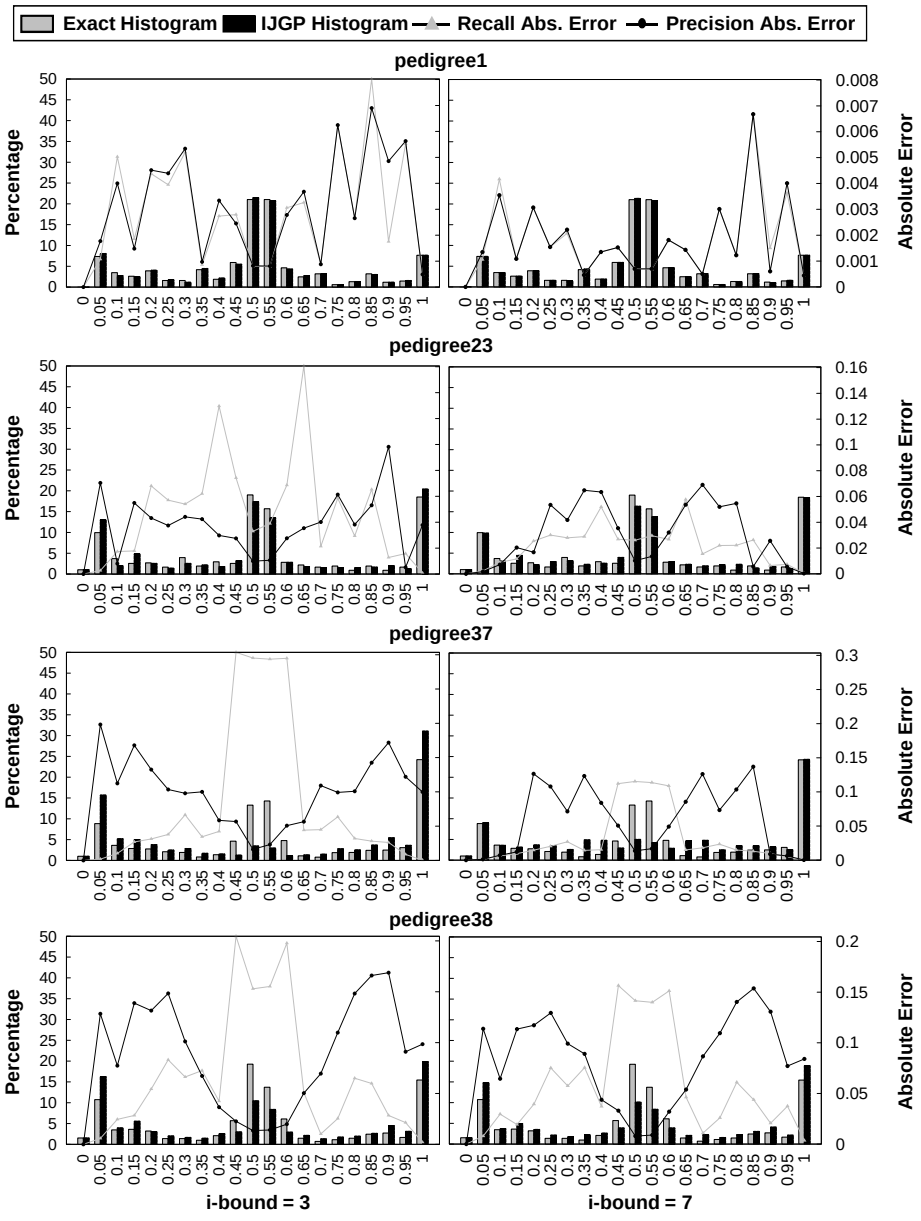


Figure 6. Results on pedigree instances. Each row is the result for one instance. Each column is the result of running IJGP with i-bound equal to 3 and 7, respectively. The number of variables N, number of evidence variables NE, and induced width w* of each instance is as follows. Pedigree1: $N = 334$, $NE = 36$ and w*=21; pedigree23: $N = 402$, $NE = 93$ and w*=30; pedigree37: $N = 1032$, $NE = 306$ and w*=30; pedigree38: $N = 724$, $NE = 143$ and w*=18.

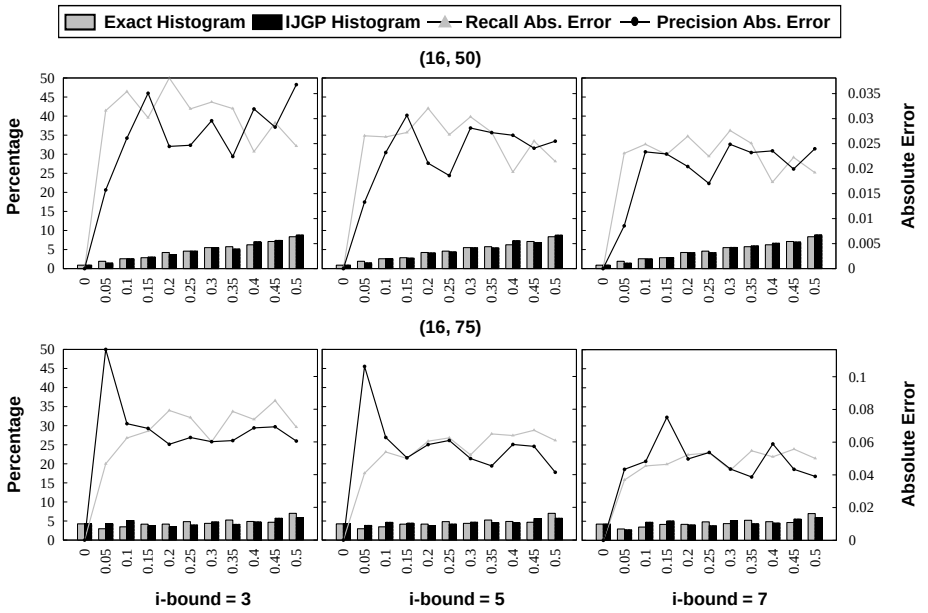


Figure 7. Results on grids2 instances. First row shows the results for parameter configuration $(16, 50)$. Second row shows the results for $(16, 75)$. Each column is the result of running IJGP with i-bound equal to 3, 5, and 7, respectively. Each plot indicates the mean value for up to 10 instances. Both parameter configurations have 256 variables, one evidence variable, and induced width w*=22.

$1 - P(x = 0|y_1, \ldots, y_P)$. We experiment on *bn2o* instances from the UAI08 competition.

Figure 9 reports the results for 3 instances. In this case, IJGP is very accurate for all instances. In particular, the accuracy in ϵ small beliefs is very high.

CPCS networks. These are medical diagnosis networks derived from the Computer-Based Patient Care Simulation system (CPCS) expert system. We tested on two networks, *cpcs54* and *cpcs360*, with 54 and 360 variables, respectively. For the first network, we generate samples of size 100 by randomly assigning 10 variables as evidence. For the second network, we also generate samples of the same size by randomly assigning 20 and 30 variables as evidence.

Figure 10 shows the results. The histograms show opposing trends in the distribution of beliefs. Although irregular, the absolute error tends to increase towards 0.5 for cpcs54. In general, the error is quite small throughout all intervals and, in particular, for inferred extreme marginals.

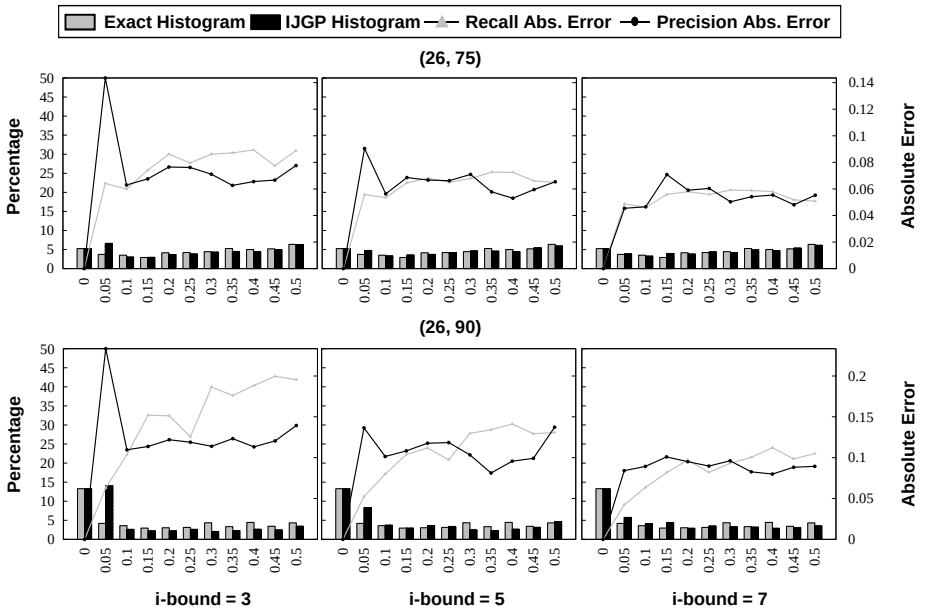


Figure 8. Results on grids2 instances. First row shows the results for parameter configuration $(26, 75)$. Second row shows the results for $(26, 90)$. Each column is the result of running IJGP with i-bound equal to 3, 5 and 7, respectively. Each plot indicates the mean value for up to 10 instances. Both parameter configurations have 676 variables, one evidence variable, and induced width w*=40.

6.2 On the Impact of Epsilon Loop-Cutset

In [Bidyuk and Dechter 2001] we explored also the hypothesis that the loop-cutset impact on IBP's performance, as discussed in Section 5.2, extends to variables with extreme support. Extreme support is expressed in the form of either extreme prior value $P(x_i) < \epsilon$ or strong correlation with an observed variable. We hypothesize that a variable X_i with extreme support nearly-cuts the information flow from its parents to its children similar to an observed variable. Subsequently, we conjecture that when a subset of variables with extreme support, called *ϵ-cutset*, form a loop-cutset of the graph, IBP converges and computes beliefs that approach exact ones.

We will briefly recap the empirical evidence supporting the hypothesis in 2-layer noisy-OR networks. The number of root nodes m and total number of nodes n was fixed in each test set (indexed $m - n$). Generating the networks, each leaf node Y_j was added to the list of children of a root node U_i with probability 0.5. All nodes were bi-valued. All leaf nodes were observed. We used average absolute error in the posterior marginals (averaged over all unobserved variables) to measure IBP's accuracy and the percent of variables for which IBP converged as a measure of convergence. In each group of experiments, the results were averaged over 100 instances.

In one set of experiments, we measured the performance of IBP while changing the

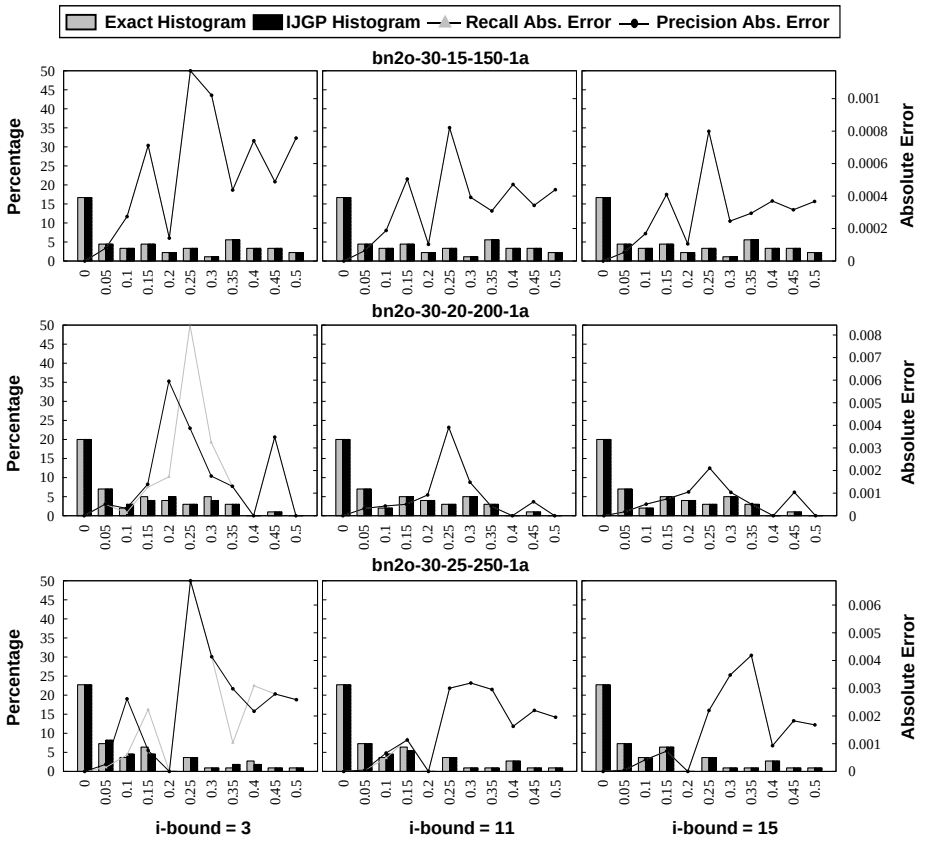


Figure 9. Results on bn2o instances. Each row is the result for one instance. Each column in each row is the result of running IJGP with i-bound equal to 3, 5 and 7, respectively. The number of variables N, number of evidence variables NE, and induced width w* of each instance is as follows. bn2o-30-15-150-1a: $N = 45$, $NE = 15$, and w*=24; bn2o-30-20-200-1a: $N = 50$, $NE = 20$, and w*=27; bn2o-30-25-250-1a: $N = 55$, $NE = 25$, and w*=26.

number of observed loop-cutset variables (we fixed all priors to $(.5, .5)$ and picked observed value for loop-cutset variables at random). The results are shown in Figure 11, top. As expected, the number of converged nodes increased and the absolute average error decreased monotonically as number of observed loop-cutset nodes increased.

Then, we repeated the experiment except now, instead of instantiating a loop-cutset variable, we set its priors to extreme (ϵ, 1-ϵ) with ϵ=$1E-10$, i.e., instead of increasing the number of observed loop-cuset variables, we increased the number of ϵ-cutset variables. If our hypothesis is correct, increasing the size of ϵ-cutset should produce an effect similar to increasing the number of observed loop-cutset variables, namely, improved convergence and better accuracy in IBP computed beliefs. The results, in Figure 11, bottom, demonstrate that initially, as the number of ϵ-cutset variables grows, the performance of IBP improves

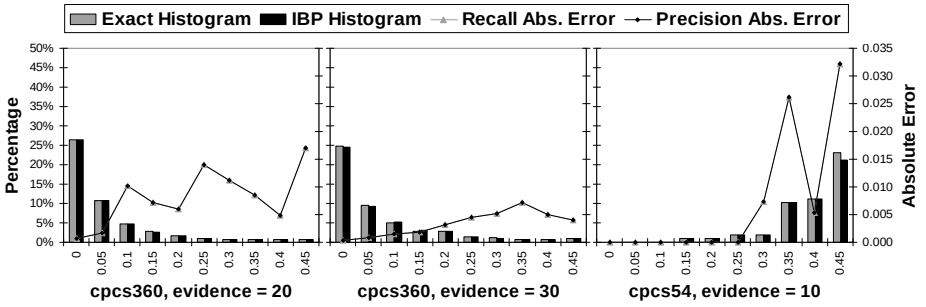


Figure 10. CPCS54, 100 instances, w*=15; CPCS360, 5 instances, w*=20

just as we conjectured. However, the percentage of nodes with converged beliefs never reaches 100% just like the average absolute error converges to some $\delta > 0$. In the case of 10-40 network, the number of converged beliefs (average absolute error) reaches maximum of $\approx 95\%$ (minimum of $\approx .001$) at 3 ϵ-cutset nodes and then drops to $\approx 80\%$ (increases to $\approx .003$) as the size of ϵ-cutset increases.

To further investigate the effect of the *strength* of ϵ-support on the performance of IBP, we experimented on the same 2-layer networks varying the prior values of the loop-cutset nodes from (ϵ, 1-ϵ) to (1-ϵ, ϵ) for $\epsilon \in [1E-10, .5]$. As shown in Figure 12, initially, as ϵ decreased, the convergence and accuracy of IBP worsened. This effect was previously reported by Murphy, Weiss, and Jordan [Murphy, Weiss, and Jordan 2000]. However, as the priors of loop-cutset nodes continue to approach 0 and 1, the average error value approaches 0 and the number of converged nodes reaches 100%. Note that convergence is not symmetric with respect to ϵ. The average absolute error and percentage of converged nodes approach 0 and 1 respectively for ϵ=1-(1E-10) but not for ϵ=1E-10 (which we also observed in Figure 11, bottom).

7 Conclusion

The paper provides insight into the power of the Iterative Belief Propagation (IBP) algorithm by making its relationship with constraint propagation explicit. We show that the power of belief propagation for zero beliefs is identical to the power of arc-consistency in removing inconsistent domain values. Therefore, the strength and weakness of this scheme can be gleaned from understanding the inference power of arc-consistency. In particular we show that the inference of zero beliefs (marginals) by IBP and IJGP is always sound. These algorithms are guaranteed to converge for inferred zeros and are as efficient as the corresponding constraint propagation algorithms.

Then the paper empirically investigates whether the sound inference of zeros by IBP is extended to near zeros. We show that while the inference of near zeros is often quite accurate, it can sometimes be extremely inaccurate for networks having significant determinism. Specifically, for networks without determinism IBP's near zero inference was sound in the sense that the average absolute error was contained within the length of the 0.05 interval

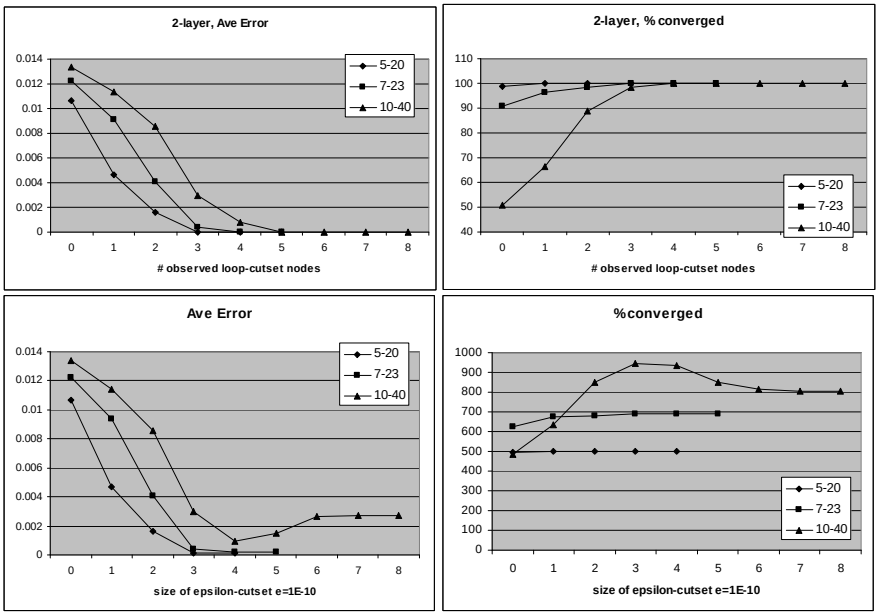


Figure 11. Results for 2-layer Noisy-OR networks. The average error and the number of converged nodes vs the number of truly observed loop-cutset nodes (top) and the size of of ϵ-cutset (bottom).

(see *two layer noisy-OR* and *CPCS* benchmarks). However, the behavior was different on benchmark networks having determinism. For example, experiments on *coding* networks show that IBP is almost perfect, while for *pedigree* and *grid* networks the results are quite inaccurate near zeros.

Finally, we show that evidence, observed or inferred, automatically acts as a cycle-cutting mechanism and improves the performance of IBP. We also provide preliminary empirical evaluation showing that the effect of loop-cutset on the accuracy of IBP extends to variables that have extreme probabilities.

References

Bidyuk, B. and R. Dechter (2001). The epsilon-cutset effect in Bayesian networks, r97, r97a in http://www.ics.uci.edu/ dechter/publications. Technical report, University of California, Irvine.

Dechter, R. (2003). *Constraint Processing*. Morgan Kaufmann Publishers.

Dechter, R. and R. Mateescu (2003). A simple insight into iterative belief propagation's success. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI'03)*, pp. 175–183.

Dechter, R., R. Mateescu, and K. Kask (2002). Iterative join-graph propagation. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*

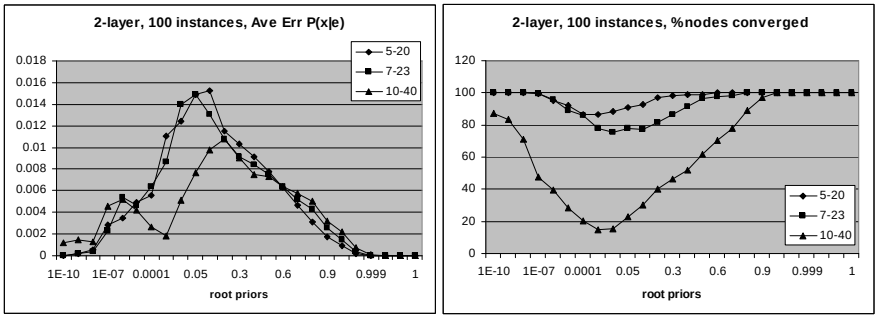


Figure 12. Results for 2-layer Noisy-OR networks. The average error and the percent of converged nodes vs ϵ-support.

(UAI'02), pp. 128–136.

Dechter, R. and J. Pearl (1991). Directed constraint networks: A relational framework for causal reasoning. In *Proceedings of the Twelfth International Joint Conferences on Artificial Intelligence (IJCAI'91)*, pp. 1164–1170.

Ihler, A. T. (2007). Accuracy bounds for belief propagation. In *Proceedings of the Twenty Third Conference on Uncertainty in Artificial Intelligence (UAI'07)*.

Ihler, A. T., J. W. Fisher, III, and A. S. Willsky (2005). Loopy belief propagation: Convergence and effects of message errors. *J. Machine Learning Research 6*, 905–936.

Koller, D. (2010). Belief propagation in loopy graphs. In *Heuristics, Probabilities and Causality: A tribute to Judea Pearl, Editors, R. Dechter, H. Gefner and J. Halpern*.

Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence 8*(1), 99–118.

Mateescu, R., K. Kask, V. Gogate, and R. Dechter (2010). Iterative join-graph propagation. *Journal of Artificial Intelligence Research (JAIR) (accepted, 2009)*.

McEliece, R. J., D. J. C. MacKay, and J. F. Cheng (1998). Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE J. Selected Areas in Communication 16*(2), 140–152.

Mooij, J. M. and H. J. Kappen (2007). Sufficient conditions for convergence of the sum-product algorithm. *IEEE Trans. Information Theory 53*(12), 4422–4437.

Mooij, J. M. and H. J. Kappen (2009). Bounds on marginal probability distributions. In *Advances in Neural Information Processing Systems 21 (NIPS'08)*, pp. 1105–1112.

Murphy, K., Y. Weiss, and M. Jordan (2000). Loopy-belief propagation for approximate inference: An empirical study. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI'00)*, pp. 467–475.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial. Intelligence 29*(3), 241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers.

Rish, I., K. Kask, and R. Dechter (1998). Empirical evaluation of approximation algorithms for probabilistic decoding. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pp. 455–463.

Rollon, E. and R. Dechter (December, 2009). Some new empirical analysis in iterative join-graph propagation, r170 in http://www.ics.uci.edu/ dechter/publications. Technical report, University of California, Irvine.

Roosta, T. G., M. J. Wainwright, and S. S. Sastry (2008). Convergence analysis of reweighted sum-product algorithms. *IEEE Trans. Signal Processing 56*(9), 4293–4305.

Yedidia, J. S., W. T. Freeman, and Y. Weiss (2000). Generalized belief propagation. In *Advances in Neural Information Processing Systems 13 (NIPS'00)*, pp. 689–695.