A New Look at Survey Propagation and its Generalizations

Elitza Maneva* Elchanan Mossel† Martin J. Wainwright‡

November 1, 2005

Abstract

This paper provides a new conceptual perspective on *survey propagation*, which is an iterative algorithm recently introduced by the statistical physics community that is very effective in solving random k-SAT problems even with densities close to the satisfiability threshold. We first describe how any SAT formula can be associated with a novel family of Markov random fields (MRFs), parameterized by a real number $\rho \in [0,1]$. We then show that applying belief propagation—a well-known "message-passing" technique for estimating marginal probabilities—to this family of MRFs recovers a known family of algorithms, ranging from pure survey propagation at one extreme ($\rho = 1$) to standard belief propagation on the uniform distribution over SAT assignments at the other extreme ($\rho = 0$). Configurations in these MRFs have a natural interpretation as partial satisfiability assignments, on which a partial order can be defined. We isolate *cores* as minimal elements in this partial ordering, which are also fixed points of survey propagation and the only assignments with positive probability in the MRF for $\rho = 1$. Our experimental results for $k = 3$ suggest that solutions of random formulas typically do not possess non-trivial cores. This makes it necessary to study the structure of the space of partial assignments for $\rho < 1$ and investigate the role of assignments that are very close to being cores. To that end, we investigate the associated lattice structure, and prove a weight-preserving identity that shows how any MRF with $\rho > 0$ can be viewed as a "smoothed" version of the uniform distribution over satisfying assignments ($\rho = 0$). Finally, we isolate properties of Gibbs sampling and message-passing algorithms that are typical for an ensemble of k-SAT problems.

Keywords: Satisfiability problems; k-SAT; survey propagation; belief propagation; sum-product; message-passing; factor graph; Markov random field; Gibbs sampling.

*Department of Electrical Engineering and Computer Science, UC Berkeley, CA. Email: `elitza@eecs.berkeley.edu`. Supported by NSF grant CCR-0121555.

†Department of Statistics, UC Berkeley, CA. Email: `mossel@stat.berkeley.edu`. Supported by a Miller Fellowship in Computer Science and Statistics, NSF grant DMS-0504245 and a Sloan Fellowship in Mathematics

‡Department of Electrical Engineering and Computer Science and Department of Statistics, UC Berkeley, CA. Email: `wainwrig@eecs.berkeley.edu`. Supported by a Sloan Fellowship in Computer Science and a grant from Intel Corporation.

1 Introduction

Constraint satisfaction problems play an important role across a broad spectrum of computer science, including computational complexity theory [9], coding theory [19, 35], and artificial intelligence [34, 14]. Important but challenging problems include devising efficient algorithms for finding satisfying assignments (when the problem is indeed satisfiable), or conversely providing a certificate of unsatisfiability. One of the best-known examples of a constraint satisfaction problem is the k-SAT problem, which is a classical NP complete problem [9] for all $k \geq 3$. In trying to understand the origin of its hardness, a great deal of research has been devoted to the properties of formulas drawn from different probability distributions. One of the most natural models for random k-SAT problems is the following: for a fixed density parameter $\alpha > 0$, choose $m = \alpha n$ clauses uniformly and with replacement from the set of all k-clauses on n variables. Despite its simplicity, many essential properties of this model are yet to be understood: in particular, the hardness of deciding if a random formula is satisfiable and finding a satisfying assignment for a random formula are both major open problems [25, 42, 16].

One of the most exciting recent developments in satisfiability problems has its origins not in computer science, but rather in statistical physics. More specifically, the ground-breaking contribution of Mézard, Parisi and Zecchina [28], as described in an article published in "Science", is the development of a new algorithm for solving k-SAT problems. A particularly dramatic feature of this method, known as *survey propagation*, is that it appears to remain effective at solving very large instances of random k-SAT problems—-even with densities very close to the satisfiability threshold, a regime where other "local" algorithms (e.g., the WSAT method [37]) typically fail. Given this remarkable behavior, the survey propagation algorithm has generated a great deal of excitement and follow-up work in both the statistical physics and computer science communities [e.g., 6, 5, 7, 3, 2, 32, 33, 41]. Nonetheless, despite the considerable progress to date, the reasons underlying the remarkable performance of survey propagation are not yet fully understood.

1.1 Our contributions

This paper provides a novel conceptual perspective on survey propagation—one that not only sheds light on the reasons underlying its success, but also places it within a broader framework of related "message-passing" algorithms that are widely used in different branches of computer science. More precisely, by introducing a new family of Markov random fields (MRFs) that are associated with any k-SAT problem, we show how a range of algorithms—including survey propagation as a special case—can all be recovered as instances of the well-known belief propagation algorithm [34], as applied to suitably restricted MRFs within this family. This equivalence is important because belief propagation is a message-passing algorithm—widely used and studied in various areas, including coding theory [35, 24, 44], computer vision [17, 11] and artificial intelligence [34, 45]—-for computing approximations to marginal distributions in Markov random fields. Moreover, this equivalence motivates a deeper study of the combinatorial properties of the family of extended MRFs associated with survey propagation. Indeed, one of the main contributions of our work is to reveal the combinatorial structures underlying the survey propagation algorithm.

The configurations in our extended MRFs turn out to have a natural interpretation as particular types of *partial SAT assignments*, in which a subset of variables are assigned 0 or 1 variables in such a way that the remaining formula does not contain any empty or unit clauses. To provide some geometrical intuition for our results, it is convenient to picture these partial assignments as

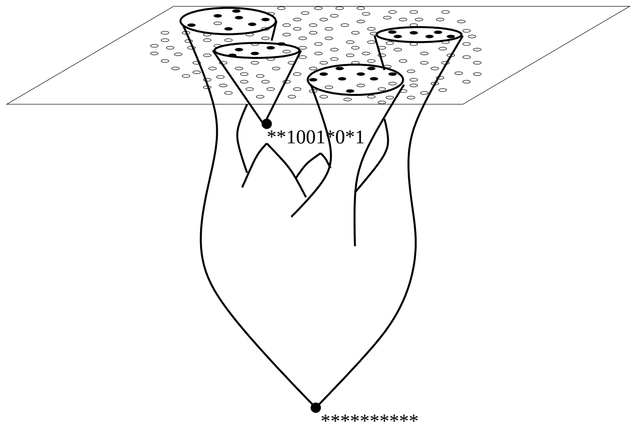


Figure 1. The set of fully assigned satisfying configurations occupy the top plane, and are arranged into clusters. Enlarging to the space of partial assignments leads to a new space with better connectiviity. Minimal elements in the partial ordering are known as cores. Each core corresponds to one or more clusters of solutions from the top plane. In this example, one of the clusters has as a core a non-trivial partial assignment, whereas the others are connected to the all-$*$ assignment.

arranged in layers depending on the number of assigned variables, so that the top layer consists of fully assigned satisfying configurations. Figure 1 provides an idealized illustration of the space of partial assignments viewed in this manner. It is argued [29, 32, 2] that for random formulas with high density of clauses, the set of fully assigned configurations are separated into disjoint clusters that cause local message-passing algorithms like belief propagation to break down (see Figure 2 for an illustration). Based on our results, the introduction of partial SAT assignments yields a *modified search space* that is far less fragmented, thereby permitting a local algorithm like belief propagation to find solutions.

We show that there is a natural partial ordering associated with this enlarged space, and we refer to minimal elements in this partial ordering as *cores*. We prove that any core is a fixed point of the pure form of survey propagation ($\rho = 1$). This fact indicates that each core represents a summary of one cluster of solutions. However, our experimental results for $k = 3$ indicate that the solutions of random formulas typically have trivial cores (i.e., the empty assignment). This observation motivates deeper study of the full family of Markov random fields for the range $0 \leq \rho \leq 1$, as well as the associated belief propagation algorithms, which we denote by SP(ρ). Accordingly, we study the lattice structure of the partial assignments, and prove a combinatorial identity that reveals how the distribution for $\rho \in (0, 1)$ can be viewed as a "smoothed" version of the MRF with $\rho = 0$. Our experimental results on the SP(ρ) algorithms indicate that they are most effective for values of ρ close to and not necessarily equal to 1. One intriguing possibility is that the effectiveness of pure survey propagation (i.e., SP(1)) may be a by-product of the fact that SP(ρ) is most effective for values of ρ less than 1, but going to 1 as n goes to infinity. The near-core assignments which are the ones of maximum weight in this case, may correspond to quasi-solutions of the cavity equations, as defined by Parisi [33]. In addition, we consider alternative sampling-based methods (e.g., Gibbs sampling) for computing marginals for the extended MRFs. We also study properties of both message-passing and Gibbs sampling that are typical over a random ensemble of k-SAT problems. We establish results that link the typical behavior of Gibbs sampling and message-passing algorithms under suitable initialization, and when applied to the extended family of MRFs with ρ sufficiently close to one.

The fact that the pure form of survey propagation (i.e., SP(1) in our notation) is a form of belief propagation was first conjectured by Braunstein et al. [6], and established independently of our work by Braunstein and Zecchina [7]. In other independent work, Aurell et al. [3] provided an alternative derivation of SP(1) that established a link to belief propagation. However, both of these papers treat only the case $\rho = 1$, and do not provide a combinatorial interpretation based on an underlying Markov random field. The results established here are a strict generalization, applying to the full range of $\rho \in [0, 1]$. Moreover, the structures intrinsic to our Markov random fields—namely cores and lattices—highlight the importance of values $\rho \neq 1$, and place the survey propagation algorithm on a combinatorial ground. As we discuss later, this combinatorial perspective has already inspired subsequent work [2] on survey propagation for satisfiability problems. Looking forward, the methodology of partial assignments may also open the door to other problems where a complicated landscape prevents local search algorithms from finding good solutions. As a concrete example, a subset of the current authors [41] have recently shown that related ideas can be leveraged to perform lossy data compression at near-optimal (Shannon limit) rates.

1.2 Organization

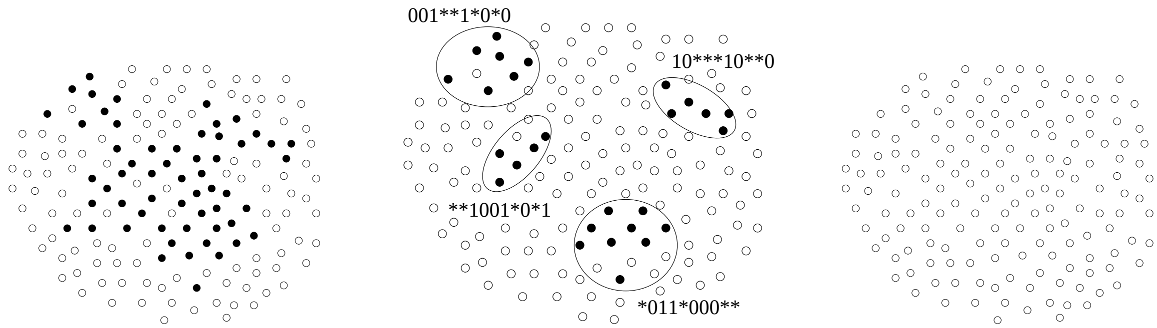
The remainder of this paper is organized in the following way:

- In Section 1.3, we provide further background on the k-SAT problem, as well as previous work on survey propagation.
- In Section 2, we introduce required notation and set up the problem more precisely.
- In Section 3, we define a family of Markov random fields (MRFs) over partial satisfiability assignments, and prove that survey propagation and related algorithms correspond to belief propagation on these MRFs.
- Section 4 is devoted to analysis of the combinatorial properties of this family of extended MRFs, as well as some experimental results on cores and Gibbs sampling.
- In Section 5, we consider properties of random ensembles of SAT formulae, and prove results that link the performance of survey propagation and Gibbs sampling to the choice of Markov random field.
- We conclude with a discussion in Section 6.

We note that many of the results reported here have been presented (without proofs or details) as an extended SODA abstract [26].

1.3 Previous work on k-SAT and survey propagation

As a classical NP complete problem [9], the k-SAT problem for $k \geq 3$ has been extensively studied. One approach is to consider ensembles of random formulas; in particular, a commonly studied ensemble is based on choosing $m = \alpha n$ clauses uniformly and with replacement from the set of all k-clauses on n variables. Clearly, a formula drawn randomly from this ensemble becomes increasingly difficult to satisfy as the clause density $\alpha > 0$ increases. There is a large body of



(a) $0 < \alpha < \alpha_d$ (b) $\alpha_d < \alpha < \alpha_c$ (c) $\alpha_c < \alpha$

Figure 2. The black dots represent satisfying assignments, and white dots unsatisfying assignments. Distance is to be interpreted as the Hamming distance between assignments. (a) For low densities the space of satisfying assignments is well connected. (b) As the density increases above α_d the space is believed to break up into an exponential number of clusters, each containing an exponential number of assignments. These clusters are separated by a "sea" of unsatisfying assignments. (c) Above α_c all assignments become unsatisfying.

work [18, 20, 8, 13, 15, 22, 1] devoted to the study of the *threshold* density where the formula becomes unsatisfiable; however, except for the case $k = 2$, the value of the threshold is currently unknown. However, non-rigorous techniques from statistical physics can be applied to yield estimates of the threshold; for instance, results from Mézard and Zecchina [31] yield a threshold estimate of $\alpha_c \approx 4.267$ for $k = 3$.

The survey propagation (SP) algorithm, as introduced by Mézard, Parisi and Zecchina [28], is an iterative message-passing technique that is able to find satisfying assignments for large instances of SAT problems at much higher densities than previous methods. The derivation of SP is based on the cavity method in conjunction with the 1-step replica summetry breaking (1-RSB) ansatz of statistical physics. We do not go into these ideas in depth here, but refer the reader to the physics literature [30, 6, 28] for further details. In brief, the main assumption is the existence of a critical value α_d for the density, smaller than the threshold density α_c, at which the structure of the space of solutions of a random formula changes. For densities below α_d the space of solutions is highly connected—in particular, it is possible to move from one solution to any other by single variable flips,[1] staying at all times in a satisfying assignment. For densities above α_d, the space of solutions breaks up into clusters, so that moving from a SAT assignment within one cluster to some other assignment within another cluster requires flipping some constant fraction of the variables simultaneously. Figure 2 illustrates how the structure of the space of solutions evolves as the density of a random formula increases. The clustering phenomenon that is believed to occur in the second phase is known in the statistical physics literature as 1-step replica symmetry breaking [30], and the estimated value for α_d in the case $k = 3$ is $\alpha_d \approx 3.921$. Within each cluster, a distinction can be made between *frozen* variables—ones that do not change their value within the cluster—and *free* variables that do change their value in the cluster. A concise description of a cluster is an assignment of $\{0, 1, *\}$ to the variables with the frozen variables taking their frozen

[1] There is no general agreement on whether assignments should be considered neighbors if they differ in only one variable, or any constant number of variables

value, and the free variables taking the joker or wild card value $*$. The original argument for the clustering assumption was the analysis of simpler satisfiability problems, such as XOR-SAT, where the existence of clusters can be demonstrated by rigorous methods [29]. In addition, if one assumes that there are no clusters, the cavity method calculation yields a value for $\alpha_c > 5$ (for $k = 3$), which is known to be wrong. More recently, Mora, Mézard and Zecchina [32] have demonstrated via rigorous methods that for $k \geq 8$ and some clause density below the unsatisfiability threshold, clusters of solutions do indeed exist.

The survey propagation (SP) algorithm is so-named, because like the belief propagation algorithm [34, 45], it entails propagating statistical information in the form of messages between nodes in the graph. In the original derivation of the updates [28, 6], the messages are interpreted as "surveys" taken over the clusters in solution space, which provide information about the fraction of clusters in which a given variable is free or frozen. However, prior to the work presented here, it was not clear how to interpret the algorithm as an instantiation of belief propagation, and thus as a method for computing (approximations) to marginal distributions in a certain Markov random field (MRF). Moreover, as discussed above, our formulation of SP in this manner provides a broader view, in which SP is one of many possible message-passing algorithms that can be applied to smoothed MRF representations of SAT problems.

2 Background and problem set-up

In this section, we begin with notation and terminology necessary to describe the k-SAT problem, and then provide a precise description of the survey propagation updates.

2.1 The k-SAT problem and factor graphs

Basic notation: Let C and V represent index sets for the clauses and variables, respectively, where $|V| = n$ and $|C| = m$. We denote elements of V using the letters i, j, k, etc., and members of C with the letters a, b, c, etc. We use x_S to denote the subset of variables $\{x_i : i \in S\}$. In the k-SAT problem, the clause indexed by $a \in C$ is specified by the pair $(V(a), J_a)$, where $V(a) \subset V$ consists of k elements, and $J_a := (J_{a,i} : i \in V(a))$ is a k-tuple of $\{0,1\}$-valued weights. The clause indexed by a is *satisfied* by the assignment x if and only if $x_{V(a)} \neq J_a$. Equivalently, letting $\delta(y, z)$ denote an indicator function for the event $\{y = z\}$, if we define the function

$$\psi_{J_a}(x) \quad := \quad 1 - \prod_{i \in V(a)} \delta(J_{a,i}, x_i), \tag{1}$$

then the clause a is satisfied by x if and only if $\psi_{J_a}(x) = 1$. The overall formula consists of the AND of all the individual clauses, and is satisfied by x if and only if $\prod_{a \in C} \psi_{J_a}(x) = 1$.

Factor graphs: A convenient graphical representation of any k-SAT problem is provided by the formalism of factor graphs (see [24] for further background). As illustrated in Figure 3, any instance of the k-SAT problem can be associated with a particular bipartite graph on the variables (denoted by circular nodes) and clauses (denoted by square nodes), where the edge (a, i) between the clause $a \in C$ and variable $i \in V$ is included in E if and only if $i \in V(a)$. Following Braunstein et al. [6], it is convenient to introduce two labellings of any given edge—namely, solid or dotted, corresponding to whether $J_{a,i}$ is equal to 0 or 1 respectively.

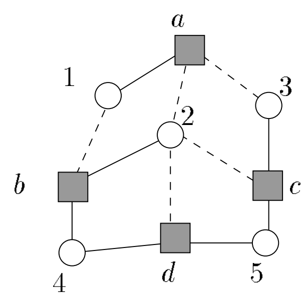


Figure 3. Factor graph representation of a 3-SAT problem on $n = 5$ variables with $m = 4$ clauses, in which circular and square nodes correspond to variables and clauses respectively. Solid and dotted edges (a, i) correspond to the weightings $J_{a,i} = 0$ and $J_{a,i} = 1$ respectively. The clause a is defined by the neighborhood set $V(a) = \{1, 2, 3\}$ and weights $J_a = (0, 1, 1)$. In traditional notation, this corresponds to the formula $(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee x_5) \wedge (\bar{x}_2 \vee x_4 \vee x_5)$.

For later use, we define (for each $i \in V$) the set $C(i) := \{a \in C \ : \ i \in V(a)\}$, corresponding to those clauses that impose constraints on variable x_i. This set of clauses can be decomposed into two disjoint subsets

$$C^-(i) := \{a \in C(i) \ : \ J_{a,i} = 1\}, \qquad C^+(i) := \{a \in C(i) \ : \ J_{a,i} = 0\}, \tag{2}$$

according to whether the clause is satisfied by $x_i = 0$ or $x_i = 1$ respectively. Moreover, for each pair $(a, i) \in E$, the set $C(i) \backslash \{a\}$ can be divided into two (disjoint) subsets, depending on whether their preferred assignment of x_i *agrees* (in which case $b \in C_a^s(i)$) or *disagrees* (in which case $b \in C_a^u(i)$) with the preferred assignment of x_i corresponding to clause a. More formally, we define

$$C_a^s(i) := \{b \in C(i) \backslash \{a\} \ : \ J_{a,i} = J_{b,i} \ \}, \qquad C_a^u(i) := \{b \in C(i) \backslash \{a\} \ : \ J_{a,i} \neq J_{b,i} \ \}. \tag{3}$$

Our focus is on random ensembles of k-SAT instances: for a given clause density $\alpha > 0$, a random instance is obtained by sampling $m = \alpha n$ clauses uniformly and with replacement from the set of all k-clauses on n variables. In terms of the factor graph representation, this procedure samples a random (n, m)-bipartite graph, in which each clause $a \in C$ has degree k.

Markov random fields and marginalization: The k-SAT problem can also be associated with a particular distribution defined as a Markov random field. Recall that a given instance of k-SAT can be specified by the collection of clause functions $\{\psi_{J_a} \ : \ a \in C\}$, as defined in equation (1). Using these functions, let us define a probability distribution over binary sequences via

$$p(x) \ := \ \frac{1}{Z} \prod_{a \in C} \psi_{J_a}(x), \tag{4}$$

where $Z := \sum_{x \in \{0,1\}^n} \prod_{a \in C} \psi_{J_a}(x)$ is the normalization constant Note that this definition makes sense if and only if the k-SAT instance is satisfiable, in which case the distribution (4) is simply the uniform distribution over satisfying assignments.

This Markov random field representation (4) of any satisfiable formula motivates a marginalization-based approach to finding a satisfying assignment. In particular, suppose that we had an oracle that could compute exactly the marginal probability

$$p(x_i) = \sum_{\{x_j, j \in V \backslash \{i\}\}} p(x_1, x_2, \ldots, x_n), \tag{5}$$

for a particular variable x_i. Note that this marginal reveals the existence of SAT configurations with $x_i = 0$ (if $p(x_i = 0) > 0$) or $x_i = 1$ (if $p(x_i = 1) > 0$). Therefore, a SAT configuration could be obtained by a recursive marginalization-decimation procedure, consisting of computing the marginal $p(x_i)$, appropriately setting x_i (i.e., decimating), and then re-iterating the modified Markov random field.

Of course, exact marginalization is computationally intractable in general [10, 12], which motivates the use of efficient algorithms for approximate marginalization. An example of such an algorithm is what we will refer to as the "naive belief propagation algorithm". The belief propagation (BP) algorithm, described in detail in Appendix A, can be applied to a MRF of the form 4 to estimate the marginal probabilities. Even though the BP algorithm is not exact, an intuitively reasonable approach is to set the variable that has the largest bias towards a particular value, and repeat. In fact, this marginalization-decimation approach based on naive BP finds a satisfying assignment for α up to approximately 3.9 for $k = 3$; for higher α, however, the iterations for BP typically fail to converge [28, 3, 6].

2.2 Survey propagation

In contrast to the naive BP approach, a marginalization-decimation approach based on *survey propagation* appears to be effective in solving random k-SAT problems even close to threshold [28, 6]. Here we provide an explicit description of what we refer to as the SP(ρ) family of algorithms, where setting the parameter $\rho = 1$ yields the pure form of survey propagation. For any given $\rho \in [0, 1]$, the algorithm involves updating messages from clauses to variables, as well as from variables to clauses. Each clause $a \in C$ passes a real number $\eta_{a \to i} \in [0, 1]$ to each of its variable neighbors $i \in V(a)$. In the other direction, each variable $i \in V$ passes a triplet of real numbers $\Pi_{i \to a} = (\Pi^u_{i \to a}, \Pi^s_{i \to a}, \Pi^*_{i \to a})$ to each of its clause neighbors $a \in C(i)$. The precise form of the updates are given in Figure 4.

Message from clause a to variable i:

$$\eta_{a \to i} = \prod_{j \in V(a) \setminus \{i\}} \left[\frac{\Pi^u_{j \to a}}{\Pi^u_{j \to a} + \Pi^s_{j \to a} + \Pi^*_{j \to a}} \right]. \tag{6}$$

Message from variable i to clause a:

$$\Pi^u_{i \to a} = \left[1 - \rho \prod_{b \in C^u_a(i)} (1 - \eta_{b \to i}) \right] \prod_{b \in C^s_a(i)} (1 - \eta_{b \to i}). \tag{7a}$$

$$\Pi^s_{i \to a} = \left[1 - \prod_{b \in C^s_a(i)} (1 - \eta_{b \to i}) \right] \prod_{b \in C^u_a(i)} (1 - \eta_{b \to i}). \tag{7b}$$

$$\Pi^*_{i \to a} = \prod_{b \in C^s_a(i)} (1 - \eta_{b \to i}) \prod_{b \in C^u_a(i)} (1 - \eta_{b \to i}). \tag{7c}$$

Figure 4: SP(ρ) updates

We pause to make a few comments about these SP(ρ) updates:

1. Although we have omitted the time step index for simplicity, equations (6) and (7) should be interpreted as defining a recursion on (η, Π). The initial values for η are chosen randomly in the interval $(0, 1)$.

2. The idea of the ρ parameter is to provide a smooth transition from the original naive belief propagation algorithm to the survey propagation algorithm. As shown in [6], setting $\rho = 0$ yields the belief propagation updates applied to the probability distribution (4), whereas setting $\rho = 1$ yields the pure version of survey propagation.

2.2.1 Intuitive "warning" interpretation

To gain intuition for these updates, it is helpful to consider the pure SP setting of $\rho = 1$. As described by Braunstein et al. [6], the messages in this case have a natural interpretation in terms of probabilities of warnings. In particular, at time $t = 0$, suppose that the clause a sends a warning message to variable i with probability $\eta^0_{a\to i}$, and a message without a warning with probability $1 - \eta^0_{a\to i}$. After receiving all messages from clauses in $C(i)\backslash\{a\}$, variable i sends a particular symbol to clause a saying either that it can't satisfy it ("u"), that it can satisfy it ("s"), or that it is indifferent ("$*$"), depending on what messages it got from its other clauses. There are four cases:

1. If variable i receives warnings from $C^u_a(i)$ and no warnings from $C^s_a(i)$, then it cannot satisfy a and sends "u".

2. If variable i receives warnings from $C^s_a(i)$ but no warnings from $C^u_a(i)$, then it sends an "s" to indicate that it is inclined to satisfy the clause a.

3. If variable i receives no warnings from either $C^u_a(i)$ or $C^s_a(i)$, then it is indifferent and sends "$*$".

4. If variable i receives warnings from both $C^u_a(i)$ and $C^s_a(i)$, a contradiction has occurred.

The updates from clauses to variables are especially simple: in particular, any given clause sends a warning if and only if it receives "u" symbols from all of its other variables.

In this context, the real-valued messages involved in the pure SP(1) all have natural probabilistic interpretations. In particular, the message $\eta_{a\to i}$ corresponds to the probability that clause a sends a warning to variable i. The quantity $\Pi^u_{j\to a}$ can be interpreted as the probability that variable j sends the "u" symbol to clause a, and similarly for $\Pi^s_{j\to a}$ and $\Pi^*_{j\to a}$. The normalization by the sum $\Pi^u_{j\to a} + \Pi^s_{j\to a} + \Pi^*_{j\to a}$ reflects the fact that the fourth case is a failure, and hence is excluded a priori from the probability distribution

Suppose that all of the possible warning events were independent. In this case, the SP message update equations (6) and (7) would be the correct estimates for the probabilities. This independence assumption is valid on a graph without cycles, and in that case the SP updates do have a rigorous probabilistic interpretation. It is not clear if the equations have a simple interpretation in the case $\rho \neq 1$.

2.2.2 Decimation based on survey propagation

Supposing that these survey propagation updates are applied and converge, the overall conviction of a value at a given variable is computed from the incoming set of equilibrium messages as

$$\begin{aligned}
\mu_i(1) &\propto \left[1 - \rho \prod_{b \in C^+(j)} (1 - \eta_{b \to j})\right] \prod_{b \in C^-(j)} (1 - \eta_{b \to j}). \\
\mu_i(0) &\propto \left[1 - \rho \prod_{b \in C^-(j)} (1 - \eta_{b \to j})\right] \prod_{b \in C^+(j)} (1 - \eta_{b \to j}). \\
\mu_i(*) &\propto \prod_{b \in C^+(j)} (1 - \eta_{b \to j}) \prod_{b \in C^-(j)} (1 - \eta_{b \to j}).
\end{aligned}$$

To be consistent with their interpretation as (approximate) marginals, the triplet $\{\mu_i(0), \mu_i(*), \mu_i(1)\}$ at each node $i \in V$ is normalized to sum to one. We define the *bias* of a variable node as $B(i) := |\mu_i(0) - \mu_i(1)|$.

The marginalization-decimation algorithm based on survey propagation [6] consists of the following steps:

1. Run SP(1) on the SAT problem. Extract the fraction β of variables with the largest biases, and set them to their preferred values.

2. Simplify the SAT formula, and return to Step 1.

Once the maximum bias over all variables falls below a pre-specified tolerance, the Walk-SAT algorithm is applied to the formula to find the remainder of the assignment (if possible). Intuitively, the goal of the initial phases of decimation is to find a cluster; once inside the cluster, the induced problem is considered easy to solve, meaning that any "local" algorithm should perform well within a given cluster.

3 Markov random fields over partial assignments

In this section, we show how a large class of message-passing algorithms—including the SP(ρ) family as a particular case—can be recovered by applying the well-known belief propagation algorithm to a novel class of Markov random fields (MRFs) associated with any k-SAT problem. We begin by introducing the notion of a partial assignment, and then use it to define the family of MRFs over these assignments.

3.1 Partial assignments

Suppose that the variables $x = \{x_1, \ldots, x_n\}$ are allowed to take values in $\{0, 1, *\}$, which we refer to as a *partial assignment*. It will be convenient, when discussing the assignment of a variable x_i with respect to a particular clause a, to use the notation $s_{a,i} := 1 - J_{a,i}$ and $u_{a,i} := J_{a,i}$ to indicate, respectively, the values that are *satisfying* and *unsatisfying* for the clause a. With this set-up, we have the following:

Definition 1. *A partial assignment x is* invalid *for a clause a if either*

(a) all variables are unsatisfying (i.e., $x_i = u_{a,i}$ for all $i \in V(a)$), or

(b) all variables are unsatisfying except for exactly one index $j \in V(a)$, for which $x_j = *$.

Otherwise, the partial assignment x is valid for clause a, and we denote this event by $\mathrm{VAL}_a(x_{V(a)})$. We say that a partial assignment is valid *for a formula if it is valid for all of its clauses.*

The motivation for deeming case (a) invalid is clear, in that any partial assignment that does not satisfy the clause must be excluded. Note that case (b) is also invalid, since (with all other variables unsatisfying) the variable x_j is effectively forced to $s_{a,i}$, and so cannot be assigned the $*$ symbol.

For a valid partial assignment, the subset of variables that are assigned either 0 or 1 values can be divided into *constrained* and *unconstrained* variables in the following way:

Definition 2. *We say that a variable x_i is the* unique satisfying variable *for a clause if it is assigned $s_{a,i}$ whereas all other variables in the clause (i.e., the variables $\{x_j : j \in V(a)\backslash\{i\}\}$) are assigned $u_{a,j}$. A variable x_i is* constrained *by clause a if it is the unique satisfying variable.*

We let $\mathrm{CON}_{i,a}(x_{V(a)})$ denote an indicator function for the event that x_i is the unique satisfying variable in the partial assignment $x_{V(a)}$ for clause a. A variable is *unconstrained* if it has 0 or 1 value, and is not constrained. Thus for any partial assignment the variables are divided into stars, constrained and unconstrained variables. We define the three sets

$$S_*(x) := \{i \in V : x_i = *\} \quad S_c(x) := \{i \in V : x_i \ \text{constrained}\} \quad S_o(x) := \{i \in V : x_i \ \text{unconstrained}\} \tag{8}$$

of $*$, constrained and unconstrained variables respectively. Finally, we use $n_*(x)$, $n_c(x)$ and $n_o(x)$ to denote the respective sizes of these three sets.

Various probability distributions can be defined on valid partial assignments by giving different weights to stars, constrained and unconstrained variables, which we denote by ω_c, ω_* and ω_o respectively. Since only the ratio of the weights matters, we set $\omega_c = 1$, and treat ω_o and ω_* as free non-negative parameters (we generally take them in the interval $[0, 1]$). We define the weights of partial assignments in the following way: invalid assignments x have weight $W(x) = 0$, and for any valid assignment x, we set

$$W(x) := (\omega_o)^{n_o(x)} \times (\omega_*)^{n_*(x)}. \tag{9}$$

Our primary interest is the probability distribution given by $p_W(x) \propto W(x)$. In contrast to the earlier distribution p, it is important to observe that this definition is valid for any SAT problem, whether or not it is satisfiable, as long as $\omega_* \neq 0$, since the all-$*$ vector is always a valid partial assignment. Note that if $\omega_o = 1$ and $\omega_* = 0$ then the distribution $p_W(x)$ is the uniform distribution on satisfying assignments. Another interesting case that we will discuss is that of $\omega_o = 0$ and $\omega_* = 1$, which corresponds to the uniform distribution over valid partial assignments without unconstrained variables.

3.2 Associated Markov random fields

Given our set-up thus far, it is not at all obvious whether or not the distribution p_W can be decomposed as a Markov random field based on the original factor graph. Interestingly, we find that p_W does indeed have such a Markov representation for any choices of $\omega_o, \omega_* \in [0,1]$. Obtaining this representation requires the addition of another dimension to our representation, which allows us to assess whether a given variable is constrained or unconstrained. We define the *parent set* of a given variable x_i, denoted by P_i, to be the set of clauses for which x_i is the unique satisfying variable. Immediate consequences of this definition are the following:

(a) If $x_i = 0$, then we must have $P_i \subseteq C^-(i)$.

(b) If $x_i = 1$, then there must hold $P_i \subseteq C^+(i)$.

(c) The setting $x_i = *$ implies that $P_i = \emptyset$.

Note also that $P_i = \emptyset$ means that x_i cannot be constrained. For each $i \in V$, let $\mathcal{P}(i)$ be the set of all possible parent sets of variable i. Due to the restrictions imposed by our definition, P_i must be contained in either $C^+(i)$ or $C^-(i)$ but not both. Therefore, the cardinality[2] of $\mathcal{P}(i)$ is $2^{|C^-(i)|} + 2^{|C^+(i)|} - 1$.

Our extended Markov random field is defined on the Cartesian product space $\mathcal{X}_1 \times \ldots \times \mathcal{X}_n$, where $\mathcal{X}_i := \{0, 1, *\} \times \mathcal{P}(i)$. The distribution factorizes as a product of compatibility functions at the variable and clause nodes of the factor graph, which are defined as follows:

Variable compatibilities: Each variable node $i \in V$ has an associated compatibility function of the form:

$$\Psi_i(x_i, P_i) \quad := \quad \begin{cases} \omega_o & : \quad P_i = \emptyset, x_i \neq * \\ \omega_* & : \quad P_i = \emptyset, x_i = * \\ \ \ 1 & : \quad \text{for any other valid } (P_i, x_i) \end{cases} \tag{10}$$

The role of these functions is to assign weight to the partial assignments according to the number of unconstrained and star variables, as in the weighted distribution p_W.

Clause compatibilities: The compatibility functions at the clause nodes serve to ensure that only valid assignments have non-zero probability, and that the parent sets $P_{V(a)} := \{P_i \ : \ i \in V(a)\}$ are consistent with the assignments $x_{V(a)} := \{x_i \ : \ i \in V(a)\}$ in the neighborhood of a. More precisely, we require that the partial assignment $x_{V(a)}$ is valid for a (i.e., $\text{VAL}_a(x_{V(a)}) = 1$) and that for each $i \in V(a)$, exactly one of the two following conditions holds:

(a) $a \in P_i$ and x_i is constrained by a or

(b) $a \notin P_i$ and x_i is not constrained by a.

The following compatibility function corresponds to an indicator function for the intersection of these events:

$$\Psi_a\big(x_{V(a)}, P_{V(a)}\big) := \text{VAL}_a(x_{V(a)}) \times \prod_{i \in V(a)} \delta\big(\text{Ind}[a \in P_i], \ \ \text{CON}_{a,i}(x_{V(a)})\big). \tag{11}$$

[2] Note that it is necessary to subtract one so as not to count the empty set twice.

We now form a Markov random field over partial assignments and parent sets by taking the product of variable (10) and clause (11) compatibility functions

$$p_{gen}(x, P) \propto \prod_{i \in V} \Psi_i(x_i, P_i) \quad \prod_{a \in C} \Psi_a\big(x_{V_a}, P_{V(a)}\big). \tag{12}$$

With these definitions, some straightforward calculations show that $p_{gen} = p_W$.

3.3 Survey propagation as an instance of belief propagation

We now consider the form of the belief propagation (BP) updates as applied to the MRF p_{gen} defined by equation (12). We refer the reader to Section A for the definition of the BP algorithm on a general factor graph. The main result of this section is to establish that the SP(ρ) family of algorithms are equivalent to belief propagation as applied to p_{gen} with suitable choices of the weights ω_o and ω_*. In the interests of readability, most of the technical lemmas will be presented in the appendix.

We begin by introducing some notation necessary to describe the BP updates on the extended MRF. The BP message from clause a to variable i, denoted by $M_{a\to i}(\cdot)$, is a vector of length $|\mathcal{X}_i| = 3 \times |\mathcal{P}(i)|$. Fortunately, due to symmetries in the variable and clause compatibilities defined in equations (10) and (11), it turns out that the clause-to-variable message can be parameterized by only three numbers, $\{M^u_{a\to i}, M^s_{a\to i}, M^*_{a\to i}\}$, as follows:

$$M_{a\to i}(x_i, P_i) \;=\; \begin{cases} M^s_{a\to i} & \text{if } x_i = s_{a,i},\ P_i = S \cup \{a\} \text{ for some } S \subseteq C^s_a(i), \\ M^u_{a\to i} & \text{if } x_i = u_{a,i},\ P_i \subseteq C^u_a(i), \\ M^*_{a\to i} & \text{if } x_i = s_{a,i},\ P_i \subseteq C^s_a(i) \text{ or } x_i = *\ , P_i = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

where $M^s_{a\to i}$, $M^u_{a\to i}$ and $M^*_{a\to i}$ are elements of $[0, 1]$.

Now turning to messages from variables to clauses, it is convenient to introduce the notation $P_i = S \cup \{a\}$ as a shorthand for the event

$$a \in P_i \quad \text{and} \quad S = P_i \backslash \{a\} \subseteq C^s_a(i),$$

where it is understood that S could be empty. In Appendix B, we show that the variable-to-clause message $M_{i\to a}$ is fully specified by values for pairs (x_i, P_i) of six general types:

$$\{(s_{a,i}, S \cup \{a\}),\ \ (s_{a,i}, \emptyset \neq P_i \subseteq C^s_a(i)),\ \ (u_{a,i}, \emptyset \neq P_i \subseteq C^u_a(i)),\ \ (s_{a,i}, \emptyset),\ \ (u_{a,i}, \emptyset),\ \ (*, \emptyset)\}.$$

The BP updates themselves are most compactly expressed in terms of particular linear combinations of such basic messages, defined in the following way:

$$R^s_{i\to a} \;:=\; \sum_{S \subseteq C^s_a(i)} M_{i\to a}(s_{a,i}, S \cup \{a\}) \tag{14a}$$

$$R^u_{i\to a} \;:=\; \sum_{P_i \subseteq C^u_a(i)} M_{i\to a}(u_{a,i},\ P_i) \tag{14b}$$

$$R^*_{i\to a} \;:=\; \sum_{P_i \subseteq C^s_a(i)} M_{i\to a}(s_{a,i}, P_i) + M_{i\to a}(*, \emptyset). \tag{14c}$$

Note that $R^s_{i\to a}$ is associated with the event that x_i is the unique satisfying variable for clause a; $R^u_{i\to a}$ with the event that x_i does not satisfy a; and $R^*_{i\to a}$ with the event that x_i is neither unsatisfying nor uniquely satisfying (i.e., either $x_i = *$, or $x_i = s_{a,i}$ but is not the only variable that satisfies a).

With this terminology, the BP algorithm on the extended MRF can be expressed in terms of the following recursions on the triplets $(M^s_{a\to i}, M^u_{a\to i}, M^*_{a\to i})$ and $(R^s_{i\to a}, R^u_{i\to a}, R^*_{i\to a})$:

BP updates on extended MRF:

Messages from clause a to variable i

$$
\begin{aligned}
M^s_{a\to i} &= \prod_{j\in C(a)\backslash\{i\}} R^u_{j\to a}\\
M^u_{a\to i} &= \prod_{j\in C(a)\backslash\{i\}} (R^u_{j\to a} + R^*_{j\to a}) + \sum_{k\in C(a)\backslash\{i\}} (R^s_{k\to a} - R^*_{k\to a}) \prod_{j\in C(a)\backslash\{i,k\}} R^u_{j\to a} - \prod_{j\in C(a)\backslash\{i\}} R^u_{j\to a}\\
M^*_{a\to i} &= \prod_{j\in C(a)\backslash\{i\}} (R^u_{j\to a} + R^*_{j\to a}) - \prod_{j\in C(a)\backslash\{i\}} R^u_{j\to a}.
\end{aligned}
$$

Messages from variable i to clause a:

$$
\begin{aligned}
R^s_{i\to a} &= \prod_{b\in C^u_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^s_a(i)} (M^s_{b\to i} + M^*_{b\to i})\Big]\\
R^u_{i\to a} &= \prod_{b\in C^s_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^u_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o)\prod_{b\in V^u_a(i)} M^*_{b\to i}\Big]\\
R^*_{i\to a} &= \prod_{b\in C^u_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^s_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o)\prod_{b\in C^s_a(i)} M^*_{b\to i}\Big] + \omega_* \prod_{b\in C^s_a(i)\cup C^u_a(i)} M^*_{b\to i}.
\end{aligned}
$$

We provide a detailed derivation of these BP equations on the extended MRF in Appendix B. Since the messages are interpreted as probabilities, we only need their ratio, and we can normalize them to any constant. At any iteration, approximations to the local marginals at each variable node $i \in V$ are given by (up to a normalization constant):

$$
\begin{aligned}
F_i(0) &\propto \prod_{b\in C^+(i)} M^u_{b\to i}\Big[\prod_{b\in C^-(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o)\prod_{b\in C^-(i)} M^*_{b\to i}\Big]\\
F_i(1) &\propto \prod_{b\in C^-(i)} M^u_{b\to i}\Big[\prod_{b\in C^+(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o)\prod_{b\in C^+(i)} M^*_{b\to i}\Big]\\
F_i(*) &\propto \omega_* \prod_{b\in C(i)} M^*_{b\to i}
\end{aligned}
$$

The following theorem establishes that the SP(ρ) family of algorithms is equivalent to belief propagation on the extended MRF:

Theorem 3. *For all $\omega_* \in [0,1]$, the BP updates on the extended (ω_*, ω_o)-MRF p_{gen} are equivalent to the* SP(ω_*) *family of algorithms under the following restrictions:*

(a) the constraint $\omega_o + \omega_* = 1$ is imposed, and

(b) all messages are initialized such that $M^u_{a\to i} = M^*_{a\to i}$ for every edge (a,i).

Proof. Under the constraint $\omega_o + \omega_* = 1$, if we initialize $M^u_{a\to i} = M^*_{a\to i}$ on every edge, then there holds $R^s_{i\to a} = R^*_{i\to a}$ and consequently $M^u_{a\to i} = M^*_{a\to i}$ remains true at the next iteration. Initializing the parameters in this way and imposing the normalization $M^u_{a\to i} + M^*_{a\to i} = 1$ leads to the following recurrence equations:

$$M^s_{a\to i} = \frac{\prod_{j\in C(a)\setminus\{i\}} R^u_{j\to a}}{\prod_{j\in C(a)\setminus\{i\}} (R^*_{j\to a} + R^u_{j\to a})}$$

where:

$$\begin{aligned} R^u_{i\to a} &= \prod_{b\in C^s_a(i)} (1 - M^*_{b\to i}) \Big[1 - \omega_* \prod_{b\in C^u_a(i)} (1 - M^*_{b\to i})\Big] \\ R^*_{i\to a} &= \prod_{b\in C^u_a(i)} (1 - M^s_{b\to i}). \end{aligned}$$

These updates are equivalent to SP(ω_*) by setting $\eta_{a\to i} = M^s_{a\to i}$, $\Pi^u_{i\to a} = R^u_{i\to a}$, and $\Pi^s_{i\to a} + \Pi^*_{i\to a} = R^*_{i\to a}$. □

Remarks:

1. Theorem 3 is a generalization of the result of Braunstein and Zecchina [7], who showed that SP(1) is equivalent to belief propagation on a certain MRF.

2. The essence of Theorem 3 is that the pure survey propagation algorithm, as well as all the ρ-variants thereof, are all equivalent to belief propagation on our extended MRF with suitable parameter choices. This equivalence is important for a number of reasons:

   (a) Belief propagation is a widely-used algorithm for computing approximations to marginal distributions in general Markov random fields [45, 24]. It also has a variational interpretation as an iterative method for attempting to solve a non-convex optimization problem based on the Bethe approximation [45]. Among other consequences, this variational interpretation leads to other algorithms that also solve the Bethe problem, but unlike belief propagation, are guaranteed to converge [43, 46, 40].

   (b) Given the link between SP and extended MRFs, it is natural to study combinatorial and probabilistic properties of the latter objects. In Section 4, we show how so-called "cores" arise as fixed points of SP(1), and we prove a weight-preserving identity that shows how the extended MRF for general ρ is a "smoothed" version of the naive MRF.

   (c) Finally, since BP (and hence SP) is computing approximate marginals for the MRF, it is natural to study other ways of computing marginals and examine if these lead to an effective way for solving random k-SAT problems. We begin this study in Section 4.5.

3. The initial messages have very small influence on the behavior of the algorithm, and they are typically chosen to be uniform random variables in $(0,1)$. In practice, for $\omega_o + \omega_* = 1$ if we start with different values for $M^u_{a\to i}$ and $M^*_{a\to i}$ they soon converge to become equal.

4. If we restrict our attention to 3-SAT, the equations have simpler form. In particular for a clause a on x_i, x_j, x_k, the messages to variable node i are:

$$\begin{aligned} M^*_{a\to i} &= R^u_{j\to a} R^u_{k\to a} \\ M^u_{a\to i} &= R^*_{j\to a} R^*_{k\to a} + R^s_{j\to a} R^u_{k\to a} + R^u_{j\to a} R^s_{k\to a} \\ M^*_{a\to i} &= R^*_{j\to a} R^*_{k\to a} + R^*_{j\to a} R^u_{k\to a} + R^u_{j\to a} R^*_{k\to a}. \end{aligned}$$

4 Combinatorial properties

This section is devoted to an investigation of the combinatorial properties associated with the family of extended Markov random fields defined in the previous section. We begin by defining an acyclic directed graph on all valid partial assignments. Of particular interest are the minimal elements in the resulting partial ordering. We refer to these as *cores*.

4.1 Directed graph and partial ordering

The vertex set of the directed graph G consists of all valid partial assignments. The edge set is defined in the following way: for a given pair of valid partial assignments x and y, the graph includes a directed edge from x to y if there exists an index $i \in V$ such that (i) $x_j = y_j$ for all $j \neq i$; and (ii) $y_i = *$ and $x_i \neq y_i$. We label the edge between x and y with the index i, corresponding to the fact that y is obtained from x by adding one extra $*$ in position i.

This directed graph G has a number of properties:

(a) Valid partial assignments can be separated into different levels based on their number of star variables. In particular, assignment x is in level $n_*(x)$. Thus, every edge is from an assignment in level $l - 1$ to one in level l, where l is at most n.

(b) The out-degree of any valid partial assignment x is exactly equal to its number of unconstrained variables $n_o(x)$.

(c) It is an acyclic graph so that its structure defines a partial ordering; in particular, we write $y < x$ if there is a directed path in G from x to y. Notice that all directed paths from x to y are labeled by indices in the set $T = \{i \in V : x_i \neq y_i = *\}$, and only the order in which they appear is different.

Given the partial ordering defined by G, it is natural to consider elements that are minimal in this partial ordering. For any valid partial assignment x and a subset $S \subseteq V$, let $\gamma_S(x)$ be the minimal $y < x$, such that the path from x to y is labeled only by indices in S. In particular $\gamma_V(x)$ is a minimal assignment in the order. It is easy to show that there always exists a unique $\gamma_S(x)$.

Proposition 4. *For any valid assignment x and $S \subseteq V$, there is a unique minimal $y < x$ such that the path from x to y is labeled only by indices in S. Furthermore $S_o(y) \cap S = \emptyset$ and $S_*(y) = S_*(x) \cup T$, where $T \subseteq S$ is the set of labels on any path from x to y.*

Proof. To prove the second assertion in the proposition statement for a minimal y, suppose there exists $i \in S \cap S_o(y)$. Then there must be be an outgoing edge from y labeled by an element in S, which contradicts the assumed minimality of y. The equivalence $S_*(y) = S_*(x) \cup T$ follows directly from the definition of G and its edge labels.

To establish the uniqueness statement, suppose that there are two minimal such assignments y_1 and y_2, and the paths from x to y_1 and y_2 are labeled by sets of indices $T_1, T_2 \subseteq S$ respectively. If $T_1 = T_2$ then $y_1 = y_2$, so let us assume that T_1 and T_2 are distinct. Without loss of generality, we may take $T_1 \backslash T_2 \neq \emptyset$. Consider a particular path from x to y_1, with labels $t_1, t_2, \ldots t_r$, where $r = |T_1|$. Let t_i be the first label such that $t_i \notin T_2$. Then its corresponding variable is unconstrained when the variables indexed by $\{t_1, \ldots t_{i-1}\} \cup S_*(x) \subseteq T_2 \cup S_*(x)$ are assigned $*$, therefore it is unconstrained in y_2. This implies that there exists an edge out of y_2 that is labeled by $t_i \in S$, which contradicts the assumption that y_2 is minimal. □

We define a *core assignment* to be a valid partial assignment $y \in \{0, 1, *\}^n$ that contains no unconstrained variables. We say that a core assignment y is *non-trivial* if $n_*(y) < n$, so that it has at least one constrained $\{0, 1\}$ variable. Under this definition, it follows that for any partial assignment x, the associated minimal element $\gamma_V(x)$ is a core assignment.

Given a valid ordinary assignment $z \in \{0, 1\}^n$, an interesting object is the subgraph of partial assignments that lie below it in the partial ordering. It can be seen that any pair of elements in this subgraph have both a unique maximal element and a unique minimal element, so that any such subgraph is a lattice [38].

In examples shown in Figure 5, only a subset of the partial assignments is shown, since even for small formulas the space of partial assignments is quite large. For the first formula all satisfying assignments have a trivial core. For the second one, on the other hand, there are assignments with non-trivial cores.

4.2 Pure survey propagation as a peeling procedure

As a particular case of Theorem 3, setting $\omega_* = 1$ and $\omega_o = 0$ yields the extended MRF that underlies the SP(1) algorithm. In this case, the only valid assignments with positive weight are those without any unconstrained variables—namely, core assignments. Thus, the distribution p_W for $(\omega_o, \omega_*) = (0, 1)$ is simply uniform over the core assignments. The following result connects fixed points of SP(1) to core assignments:

Proposition 5. *For a valid assignment x, let* SP(1) *be initialized by:*

$$\Pi^u_{i \to a} = \delta(x_i, u_{a,i}), \quad \Pi^s_{i \to a} = \delta(x_i, s_{a,i}), \quad \Pi^*_{i \to a} = 0.$$

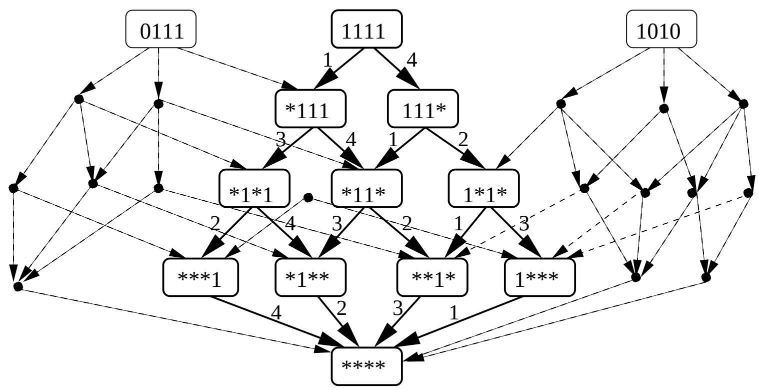
Then within a finite number of steps, the algorithm converges and the output fields are

$$\mu_i(b) = \delta(y_i, b),$$

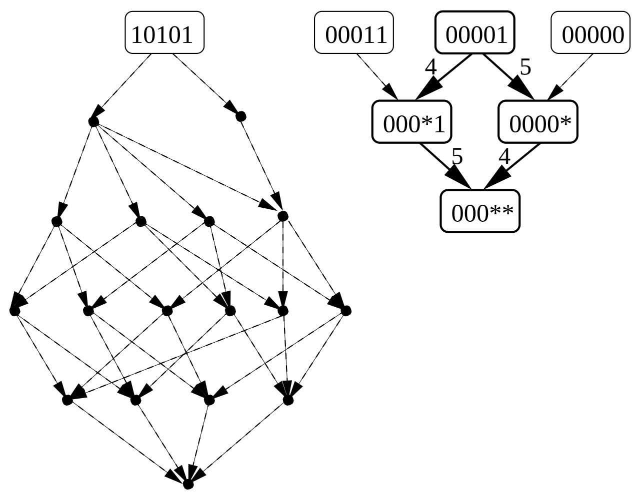
where $y = \gamma_V(x)$ and $b \in \{0, 1, *\}$.

Proof. We say that a variable i belongs to the core if $y_i \neq *$. We say that a clause a belongs to the core if all the variables in the clause belong to the core. We first show by induction that

I. If a and i belong to the core and y_i is not the unique satisfying variable for a then $\Pi^u_{i \to a} = \delta(x_i, u_{a,i})$ and $\Pi^s_{i \to a} = \delta(x_i, s_{a,i})$, and



(a)



(b)

Figure 5. Portion of the directed graph on partial assignments for two different formulas: (a) $(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. highlighted is the lattice below the satisfying assignment $z = (1, 1, 1, 1)$, whose core is trivial (i.e., $\gamma_V(z) = (*, *, *, *)$). (b) $(\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee x_1) \wedge (x_2 \vee \bar{x}_3 \vee x_5) \wedge (x_1 \vee x_5 \vee \bar{x}_4)$. the satisfying assignment $z = (0, 0, 0, 0, 1)$ has the non-trivial core $\gamma_V(z) = (0, 0, 0, *, *)$. For the same formula there are other satisfying assignments, e.g. $(1, 0, 1, 0, 1)$ which have a trivial core.

II. If a and i belong to the core and y_i is the unique satisfying variable for a then $\eta_{a\to i} = 1$.

Clearly, property I holds at time 0. Therefore, it suffices to prove that if property I holds at time t then so does II. and that if property II holds at time t then property I holds at time $t+1$.

Suppose that property I holds at time t. Let a and i belong to the core such that y_i is the unique satisfying variable of the clause a. By the induction hypothesis for all $j \in V(a) \setminus \{i\}$ it holds that $\Pi^u_{j\to a} = \delta(x_j, u_{a,j}) = 1$. This implies that $\eta_{a\to i} = 1$ as needed.

Suppose that property II holds at time t. Let a and i belong to the core such that y_i is not unique satisfying for a. By the assumption, it follows that there exists b which belongs to the core such that y_i is the unique satisfying variable for b. This implies by the induction hypothesis that $\eta_{b\to i} = 1$. It is now easy to see that at update $t+1$: $\Pi^u_{i\to a} = \delta(x_i, u_{a,i})$ and $\Pi^s_{i\to a} = \delta(x_i, s_{a,i})$. Note that the claim above implies that for all times t and all i such that $y_i \neq *$, it holds that $\mu_i(b) = \delta(y_i, b)$.

Let $i_1, i_2, \ldots, i_s$ be a "peeling-path" from x to y. In other words, the variable i_1 is not uniquely satisfying any clause. Once, this variable is set to $*$, the variable i_2 is not uniquely satisfying any clause etc. We claim that for all $1 \leq t \leq s$, for all updates after time t and for all clauses a such that $i_t \in V(a)$ it holds that $\eta_{a\to i_t} = 0$. The proof follows easily by induction on t. This in turn implies that if for all updates after time t $\mu_{i_t}(b) = \delta(y_i, *)$, from which the result follows. □

Thus, SP(1), when suitably initialized, simply strips the valid assignment x down to its core $\gamma_V(x)$. Moreover, Proposition 5, in conjunction with Theorem 3, leads to viewing the pure form of survey propagation SP(1) as performing an approximate marginalization over cores. Therefore, our results raise the crucial question: do cores exist for random formulas? Motivated by this perspective, Achlioptas and Ricci-Tersenghi [2] has answered this question affirmatively for k-SAT with $k \geq 9$. In Section 5, we show that cores, if they exist, must be "large" in a suitable sense (see Proposition 8). In the following section, we explore the case $k = 3$ via experiments on large instances.

4.3 Peeling experiments

We have performed a large number of the following experiments:

1. starting with a satisfying assignment x, change a random one of its unconstrained variables to $*$,

2. repeat until there are no unconstrained variables.

This procedure, which we refer to as "peeling", is equivalent to taking a random path from x in G, by choosing at each step a random outgoing edge. Any such path terminates at the core $\gamma_V(x)$. It is interesting to examine at each step of this process the number of unconstrained variables (equivalently, the number of outgoing edges in the graph G). For $k = 3$ SAT problems, Figure 6 shows the results of such experiments for $n = 100,000$, and using different values of α. The plotted curves are the evolution of the number of unconstrained variables as the number of $*$'s increases. We note that for $n = 100$ and α close to the threshold, satisfying assignments often correspond to core assignments; a similar observation was also made by Braunstein and Zecchina [7]. In contrast, for larger n, this correspondence is rarely the case. Rather, the generated curves suggest that $\gamma_V(x)$ is almost always the all-$*$ assignment, and moreover that for high density α, there is a critical level

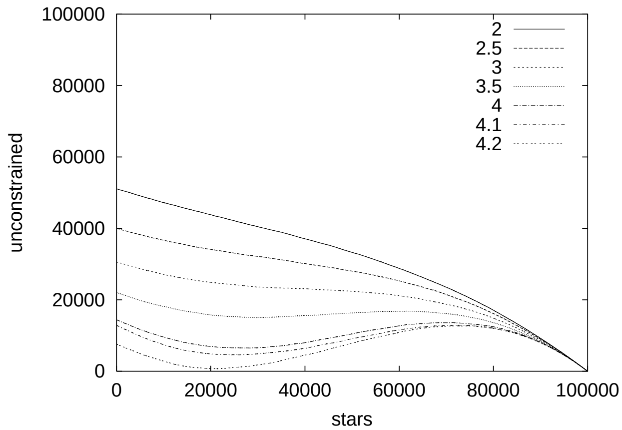


Figure 6. Evolution of the number of unconstrained variables in the peeling process: start with a satisfying assignment, change a random unconstrained variable to $*$ and repeat. Plotted is the result of an experiment for $n = 100,000$, for random formulas with $k = 3$ and $\alpha = \{2, 2.5, 3, 3.5, 4, 4.1, 4.2\}$. In particular, core assignments are on the x-axis, and satisfying assignments are on the y-axis.

in G where the out-degrees are very low. Increasing α results in failure of the algorithm itself, rather than in the formation of real core assignments.

For $k = 2$, the event that there is a path in G from a satisfying assignment to the all-$*$ assignment has a very natural interpretation. In particular, it is equivalent to the event that the *pure-literal rule* succeeds in finding an assignment. The pure-literal rule [36] is an algorithm consisting of the following steps: assign 1 to a variable if it only appears positively in a clause, and 0 if it only appears negatively in a clause, reduce the formula, and repeat the procedure. It is straightforward to check that the sequence of variables given by the labels on any path from the all-$*$ assignment to a satisfying assignment can be identified with a sequence of steps of the pure-literal type. Furthermore, it is known [36] that there is a phase transition for the event that the pure-literal rule succeeds at $\alpha = 1$.

Interestingly, as mentioned earlier, for $k \geq 9$ there are values for $\alpha < \alpha_c$ such that this peeling procedure provably results in a non-trivial core assignment with high probability, according to [2]. The fact that we do not observe core assignments for $k = 3$, and yet the algorithm is successful, means that an alternative explanation is required. Accordingly, we propose studying the behavior of SP(ρ) for $\rho \in (0, 1)$. Our experimental results, consistent with similar reports from Kirkpatrick [23], show that SP(ρ) tends to be most effective in solving k-SAT for values of $\rho < 1$. If so, the good behavior of SP(1) may well follow from the similarity of SP(1) updates to SP(ρ) updates for $\rho \approx 1$. To further explore this issue, the effects of varying the weight distribution (ω_o, ω_*), and consequently the parameter ρ, are discussed in the following section.

4.4 Weight distribution and smoothing

One of the benefits of our analysis is that it suggests a large pool of algorithms to be investigated. One option is to vary the values of ω_o and ω_*. A "good" setting of these parameters should place significant weight on precisely those valid assignments that can be extended to satisfying assignments. At the same time, the parameter setting clearly affects the level of connectivity in the

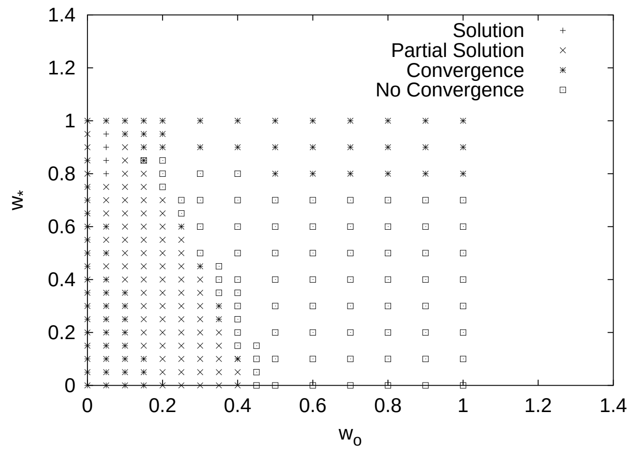


Figure 7. Performance of BP for different choices of (ω_o, ω_*) as applied to a particular randomly chosen formula with $n = 10000$, $k = 3$, $\alpha = 4.2$. Four distinct cases can be distinguished: (i) BP converges and the decimation steps yields a complete solution, (ii) BP converges and the decimation steps yield a partial solution, completed by using Walk-SAT, (iii) BP converges, but the decimation steps don't lead to a solution, and (iv) BP does not converge.

space of valid assignments. Connectivity most likely affects the performance of belief propagation, as well as any other algorithm that we may apply to compute marginals or sample from the distribution.

Figure 7(a) shows the performance of belief propagation on the extended MRF for different values of (ω_o, ω_*), and applied to particular random formula with $n = 10,000$, $k = 3$ and $\alpha = 4.2$. The most successful pairs in this case were $(0.05, 0.95)$, $(0.05, 0.9)$, $(0.05, 0.85)$, and $(0.05, 0.8)$. For these settings of the parameters the decimation steps reached a solution, so a call to WalkSAT was not needed. For weights satisfying $\omega_o + \omega_* > 1$, the behavior is very predictable: although the algorithm converges, the choices that it makes in the decimation steps lead to a contradiction. Note that there is a sharp transition in algorithm behavior as the weights cross the line $\omega_o + \omega_* = 1$, which is representative of the more general behavior.

The following result provides some justification for the excellent performance in the regime $\omega_o + \omega_* \leq 1$.

Theorem 6. *If $\omega_o + \omega_* = 1$, then $\sum_{y \leq x} W(y) = \omega_*^{n_*(x)}$ for any valid assignment x. If $\omega_o + \omega_* < 1$, then $\sum_{y \leq x} W(y) \geq (\omega_*)^{n_*(x)}$ for any valid assignment x.*

It should be noted that Theorem 6 has a very natural interpretation in terms of a "smoothing" operation. In particular, the (ω_o, ω_*)-MRF may be regarded as a smoothed version of the uniform distribution over satisfying assignments, in which the uniform weight assigned to each satisfying assignment is spread over the lattice associated with it.[3]

The remainder of this section is devoted to the proof of Theorem 6.

Proof. We start with the case $\omega_o + \omega_* = 1$. Let A denote the set of partial assignments z such that $z_j \in \{x_j, *\}$ for all $j \in V$. We refer to these as the set of assignments consistent with x. Let

[3]Note, however, that any partial assignment that belongs to two or more lattices is assigned a weight only once. Otherwise, the transformation would be a convolution operation in a strict sense.

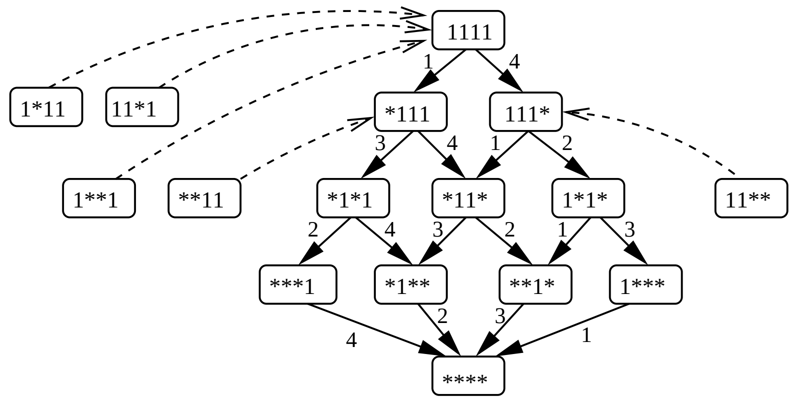


Figure 8. The directed graph G and the map σ for the formula $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_2} \vee \bar{x_3} \vee x_4)$ and the satisfying assignment $(0, 0, 1, 0)$. The solid arrows denote edges in G and the dashed arrows denote σ.

$B = \{y : y \leq x\}$ be the set of valid assignments that are reachable from x. Notice that all $y \in B$ are valid and consistent with x, but not every valid assignment in A is reachable from x. We will let $S_*(z)$ denote the set of variables assigned $*$ both for valid and invalid assignments z.

We define a map between all assignments consistent with x and the set of reachable ones. Let $\sigma : A \to B$ be defined as

$$\sigma(z) := \gamma_{S_*(z)}(x).$$

Notice that if $y \in B$ then $\sigma(y) = y$. The map is, of course, many-to-one. We define what we'll show is the reverse map. For $y \in B$ let

$$\tau(y) := \{z \in A : S_*(z) = S_*(y) \cup T, T \subseteq S_c(y)\}.$$

Lemma 7. *For any $y \in B$ and $z \in A$, $z \in \tau(y)$ if and only if $\sigma(z) = y$.*

Proof. Let $z \in \tau(y)$ so that $S_*(z) = S_*(y) \cup T$ for some $T \subseteq S_c(y)$. $\sigma(z) = \gamma_{S_*(z)}(x)$ is the minimal valid assignment such that the path from x to it is labeled only by elements in $S_*(z)$. We'll show that y satisfies these properties, and therefore by proposition 4, $y = \sigma(z)$. Any path from x to y (which exists since $y \in B$) is labeled by $S_*(y) \backslash S_*(x) \subseteq S_*(z)$. Furthermore, for every $i \in S_*(z)$, $i \notin S_o(y)$ so there is no outgoing edge from y labeled by an element in $S_*(z)$. Therefore y is minimal.

Let $y = \sigma(z) = \gamma_{S_*(z)}(x)$. By proposition 4 there is no $i \in S_*(z)$ such that $i \in S_o(y)$. Therefore $S_*(z) \subseteq S_*(y) \cup S_c(y)$. Further we have that $S_*(y) \subseteq S_*(z) \cup S_*(x) = S_*(z)$, therefore $S_*(z) = S_*(y) \cup T$ for some $T \subseteq S_c(y)$. Hence $z \in \tau(y)$. $\square$

For a set of partial assignments X let $W(X) = \sum_{x \in X} W(x)$. Let $W^{\emptyset}(z) = (\omega_*)^{n_*(z)} \times (\omega_o)^{n - n_*(z)}$, denote the weight of any partial assignment, if the formula had no clauses. For such a formula all partial assignments are valid. Observe that if we restrict our attention to the assignments that are consistent with x,

$$\begin{aligned} W^{\emptyset}(A) &= \sum_{z\in A} W^{\emptyset}(z) \\ &= \sum_{S\subseteq V\setminus S_*(x)} (\omega_*)^{|S_*(x)|+|S|} \times (\omega_o)^{n-|S_*(x)|-|S|} \\ &= (\omega_*)^{|S_*(x)|} \times (\omega_* + \omega_o)^{n-|S_*(x)|} \\ &= (\omega_*)^{n_*(x)} \end{aligned}$$

We show that when clauses are added to the formula, the total weight under x is preserved as long as x is still valid. In particular when an assignment z that is consistent with x becomes invalid, it passes its weight to an assignment that is still valid, namely $\sigma(z)$, which has fewer $*$ variables than z.

$$\begin{aligned} W(y) &= (\omega_*)^{n_*(y)} \times (\omega_o)^{n_o(y)} \times 1^{n_c(y)} \\ &= (\omega_*)^{n_*(y)} \times (\omega_o)^{n_o(y)} \times (\omega_* + \omega_o)^{n_c(y)} \qquad (17) \\ &= \sum_{T\subseteq S_c(y)} (\omega_*)^{n_*(y)+|T|} \times (\omega_o)^{n_o(y)+n_c(y)-|T|} \\ &= \sum_{T\subseteq S_c(y)} W^{\emptyset}(z : S_*(z) = S_*(y)\cup T) \\ &= W^{\emptyset}(\{z : S_*(z) = S_*(y)\cup T, T\subseteq S_c(y)\}) \\ &= W^{\emptyset}(\tau(y)). \end{aligned}$$

Finally, we have:

$$\sum_{y\le x} W(y) = \sum_{y\le x} W^{\emptyset}(\tau(y)) = W^{\emptyset}(A) = (\omega_*)^{n_*(x)}$$

where we used the fact that the sets $\tau(y)$ for $y \in B$ partition A by lemma 7.

The proof of the case $\omega_o + \omega_* < 1$ is similar except that equation (17) becomes an inequality:

$$W(y) = (\omega_o)^{n_o(y)} \times (\omega_*)^{n_*(y)} \times 1^{n_c(y)} \ge \sum_{T\subseteq S_c(S)} W^{\emptyset}(\tau(y)).$$

When an assignment z that is consistent with x becomes invalid, it passes more than its own weight to $\sigma(z)$. □

4.5 Gibbs sampling

Based on our experiments, the algorithm SP(ρ) is very effective for appropriate choices of the parameter ρ. The link provided by Theorem 6 suggests that the distribution p_W, for which SP(ρ)—as an instantiation of belief propagation on the extended MRF—is computing approximate marginals, must posses good "smoothness" properties. One expected consequence of such "smoothness" is that algorithms other than BP should also be effective in computing approximate marginals. Interestingly, rigorous conditions that imply (rapid) convergence of BP [39]—namely, uniqueness of Gibbs measures on the computation tree—are quite similar to conditions implying rapid convergence of

Gibbs samplers, which are often expressed in terms of "uniqueness", "strong spatial mixing", and "extremality" (see, for example [27, 4]).

In this section, we explore the application of sampling methods to the extended MRF as a means of computing unbiased stochastic approximations to the marginal distributions, and hence biases at each variable. More specifically, we implemented a Gibbs sampler for the family of extended MRFs

SAT α	Gibbs ρ			
	0.4	0.5	0.7	0.9
4.2	**0.0493**	0.1401	0.3143	0.4255
4.1	**0.0297**	0.1142	0.3015	0.4046
4.0	0.0874	**0.0416**	0.2765	0.3873
3.8	0.4230	0.4554	0.1767	**0.0737**
3.6	0.4032	0.4149	0.1993	**0.0582**
3.4	0.4090	0.4010	0.2234	**0.0821**

(a) Comparison to SP(0.95)

SAT α	Gibbs ρ			
	0.4	0.5	0.7	0.9
4.2	**0.0440**	0.1462	0.3166	0.4304
4.1	0.0632	**0.0373**	0.2896	0.4119
4.0	**0.0404**	0.0666	0.2755	0.3984
3.8	0.1073	**0.0651**	0.2172	0.3576
3.6	0.1014	**0.0922**	0.1620	0.3087
3.4	0.3716	0.3629	0.1948	**0.0220**

(b) Comparison to SP(0.9)

SAT α	Gibbs ρ			
	0.4	0.5	0.7	0.9
4.2	SP fails	SP fails	SP fails	SP fails
4.1	**0.0230**	0.0985	0.3236	0.4341
4.0	0.0493	**0.0079**	0.3273	0.4309
3.8	0.0531	**0.0194**	0.2860	0.4104
3.6	0.0980	**0.0445**	0.2412	0.3887
3.4	0.0365	**0.0356**	0.1301	0.3869

(c) Comparison to SP(0.7)

SAT α	Gibbs ρ			
	0.4	0.5	0.7	0.9
4.2	SP fails	SP fails	SP fails	SP fails
4.1	**0.1925**	0.2873	0.3989	0.4665
4.0	**0.0483**	0.1092	0.2986	0.4179
3.8	0.0924	**0.0372**	0.3235	0.4323
3.6	**0.0184**	0.0304	0.2192	0.4009
3.4	0.0323	**0.0255**	0.0718	0.3613

(d) Comparison to SP(0.5)

Figure 9. Comparison of $SP(\beta)$ pseudomarginals for $\beta \in \{0.95, 0.9, 0.7, 0.5\}$ to marginals estimated by Gibbs sampling on weighted MRFs with $\rho \in \{0.4, 0.5, 0.7, 0.9\}$ for the range of SAT problems $\alpha \in \{4.2, 4.1, 4.0.3.8, 3.6, 3.4\}$. Each entry in each table shows the average ℓ_1 error between the biases computed from the $SP(\beta)$ pseudomarginals compared to the biases computed from Gibbs sampling applied to $MRF(\rho)$. Calculations were based on top 50 most biased nodes on a problem of size $n = 1000$. The bold entry within each row (corresponding to a fixed α) indicates the $MRF(\rho)$ that yields the smallest ℓ_1 error in comparison to the SP biases.

developed in Section 3. The Gibbs sampler performs a random walk over the configuration space of the extended MRF—that is, on the space of partial valid assignments. Each step of the random walk entails picking a variable x_i uniformly at random, and updating it randomly to a new value $b \in \{0, 1, *\}$ according to the conditional probability $p_W(x_i = b | (x_j : j \neq i))$. By the construction of our extended MRF (see equation (12)), this conditional probability is an (explicit) function of the variables x_j and x_i appear together in a clause, and of the variables x_k such that x_k and x_j appear together in a clause, where x_j and x_i appear together in a clause.

It is of interest to compare the approximate marginals computed by the SP(β) family of algorithms (to which we refer as *pseudomarginals*) to the (stochastic) estimates computed by Gibbs sampler. Given the manner in which the SP pseudomarginals are used in the decimation procedure, the most natural comparison is between the biases $\mu_i(0) - \mu_i(1)$ provided by the $SP(\beta)$ algorithm, and the biases $\tau_i(0) - \tau_i(1)$ associated with the Gibbs sampler (where τ_i are the approximate marginals obtained from Gibbs sampling on the extended MRF with parameter ρ (denoted MRF(ρ)). The results of such comparisons for the SP parameter $\beta \in \{0.95, 0.9, 0.7, 0.5\}$ and the Gibbs sampling parameter $\rho \in \{0.4, 0.5, 0.7, 0.9\}$ are shown in Figure 9. Comparisons are made for each pair (β, ρ) in these sets, and over a range of clause densities $\alpha \in \{4.2, 4.1, 4.0.3.8, 3.6, 3.4\}$. For fairly dense formulas (e.g., $\alpha \geq 4.0$), the general trend is that the SP(β) biases with larger β agree most closely with the Gibbs biases with ρ relatively smaller (i.e., $\rho < \beta$). For lower clause densities (e.g., $\alpha = 3.4$), the agreement between the SP(β) and Gibbs biases on $MRF(\rho)$ when $\beta = \rho$ is substantially closer.

5 Expansion arguments for random formulas

This section is devoted to the study of properties of the MRF on random formulas. We will use simple random graph arguments in order to obtain typical properties of cores, as well as the behavior of Gibbs sampling or message-passing algorithms applied to the MRF associated with a randomly chosen formula. Throughout this section, we denote p_W^{ϕ} to denote the MRF distribution for a fixed formula ϕ. Otherwise, we write $\mathbb{P}^{n,m}$ for the uniform measure on k-sat formulas with n variables and m clauses, and $\mathbb{P}^{n,\alpha}$ for the uniform measure on k-sat formulas with n variables and $m = \alpha n$ clauses. We often drop n, m, and/or α when they are clear from the context. Finally, we use E_W^{ϕ}, $\mathbb{E}^{n,m}$ and $\mathbb{E}^{n,\alpha}$ to denote expectations with respect to the distributions p_W^{ϕ}, $\mathbb{P}^{n,m}$ and $\mathbb{P}^{n,\alpha}$ respectively.

5.1 Size of cores

We first prove a result that establishes that cores, if they exist, are typically at least a certain linear fraction $c(\alpha, k)$ of the total number n of variables.

Proposition 8. *Let ϕ be a random k-sat formula with $m = \alpha n$ clauses where $k \geq 3$. Then for all positive integers C it holds that*

$$\mathbb{P}^{n,\alpha}[\ \phi \text{ has a core with } C \text{ clauses }\ ] \ \leq\ \left(\frac{e^2 \alpha C^{k-2}}{n^{k-2}}\right)^C, \tag{18}$$

Consequently, if we define $c(\alpha, k) := (\alpha e^2)^{-1/(k-2)}$, then with $\mathbb{P}^{n,\alpha}$-probability tending to one as $n \to +\infty$, there are no cores of size strictly less than $c(\alpha, k)\, n$.

Proof. Suppose that the formula ϕ has a core with C clauses. Note that the variables in these clauses all lie in some set of at most C variables. Thus the probability that a core with C clauses exist is bounded by the probability that there is a set of C clauses all whose variables lie in some set of size $\leq C$. This probability is bounded by

$$\binom{m}{C}\binom{n}{C}\left(\frac{C}{n}\right)^{Ck},$$

which can be upper bounded by

$$\left(\frac{em}{C}\right)^C \left(\frac{en}{C}\right)^C \left(\frac{C}{n}\right)^{Ck} = \left(\frac{e^2 \alpha C^{k-2}}{n^{k-2}}\right)^C,$$

as needed. $\square$

5.2 (Meta)-stability of the all $*$ assignment for small ρ

By definition, the extended MRF for $\rho = 1$ assigns positive mass to the all-$*$ vector. Moreover, Proposition 8 implies that the size of cores (when they exist) is typically linear in n. It follows that the state space of the MRF for $\rho = 1$ typically satisfies one of the following properties:

- Either the state space is trivial, meaning that it contains only the all $*$ state, or
- The state space is disconnected with respect to all random walks based on updating a small linear fraction of the coordinates in each step.

The goal of this section is to establish that a similar phenomenon persists when ρ is close to 1 (i.e., when $1-\rho$ is small).

We begin by introducing some notions from the analysis of the mixing properties of Markov chains. Let T be a reversible chain with respect to a measure p on a state space Ω. For sets $A, B \subset \Omega$, write

$$q_T(A,B) = \sum_{x\in A, y\in B} p(x) T_{x\to y} = \sum_{x\in A, y\in B} p(y) T_{y\to x}.$$

The *conductance* of the chain T is defined as

$$c(T) = \inf_{S\subset\Omega}\{\frac{q_T(S,S^c)}{p(S)(1-p(S))}\}.$$

It is well-known that $c(T)/2$ is an upper bound on the spectral gap of the chain T and that $2/c(T)$ is a lower bound on the mixing time of the chain. We note moreover that the definition of T implies that for every two sets A, B it holds that $q_T(A,B) \le \min\{p(A), p(B)\}$.

Definition 9. *Consider a probability measure p on a space Ω of strings of length n. Let T be a Markov chain on Ω. The radius of T denoted by $r(T)$ is defined by*

$$r(T) := \sup\{d_H(x,y) : T_{x,y} > 0\}, \tag{19}$$

where d_H is the Hamming distance. We let the radius r-conductance of p denote by $c(r,p)$ be

$$c(r,p) := \sup\{c(T) : \ T \text{ is reversible with respect to } p \text{ and } r(T) \le r\}. \tag{20}$$

Now returning to the random k-SAT problem, we write p_ρ for the measure $p_W = p_W^\phi$ with $\omega_* = \rho$ and $\omega_o = 1-\rho$.

Proposition 10. *Consider a randomly chosen k-SAT formula with density α. Then there exists a $\rho_0 \in (0,1)$ such that if $\rho > \rho_0$ then $\mathbb{P}^n[\phi \in A_n \cup B_n] \to 1$ as $n \to +\infty$ where A_n and B_n are the following events:*

(I) A_n consists of all the formulas ϕ satisfying $p_\rho^\phi[n - n_*(x) \le 2\sqrt{(1-\rho)}\, n] \ge 1 - \exp(-\Omega(n))$.

(II) B_n consists of all the formulas ϕ for which the measure p_ρ^ϕ satisfies $c(\sqrt{(1-\rho)}\, n, p_\rho) \le \exp(-\Omega(n))$.

Proof. We let δ be a small positive number to be determined, and set $1-\rho = \delta^2$. As it suffices to work with ratios of probabilities, we use the unnormalized weight $W^\phi(x)$ instead of $p_W^\phi(x)$.

The proof requires the following:

Lemma 11. *Let d be an integer satisfying $\delta n \leq d \leq 2\delta n$. For δ sufficiently small, it holds that with $\mathbb{P}^n$ probability going to 1 as $n \to \infty$*

$$\frac{\sum_{d=\delta n}^{2\delta n} W^\phi[n - n_* = d]}{\rho^{3n}} = \exp(-\Omega(n)). \tag{21}$$

Proof. See Appendix C.1. □

To establish the proposition, it suffices to show that for any formula ϕ for which equation (21) of Lemma 11 is valid, then one of either condition (I) or condition (II) must hold.

(i) First suppose that $W^\phi[n - n_*(x) > 2\delta n] \leq \rho^{3n/2}$. In this case, condition (I) in the statement of the proposition follows immediately.

(ii) Otherwise, we may take $W^\phi[n - n_*(x) > 2\delta n] \geq \rho^{3n/2}$. In this case, we can apply the conductance bound in order to bound the gap of any operator with radius $\leq \delta n$. Take the set A to be all x with $n - n_*(x) < \delta n$ and B be the set of all x with $\delta n \leq n - n_*(x) \leq 2\delta n$. Let T be any Markov chain with radius δn that is reversible with respect to p_W. Then we have $q_T(A, A^c) = q_T(A, B) \leq p(B)$. In addition, it holds that $W^\phi[n - n_*(x) < \delta n] \geq \rho^n$ (since if x is the all-$*$ assignment, we have $W^\phi(x) = \rho^n$); moreover, if we take n sufficiently large, then we have $W^\phi[\delta n \leq n - n_*(x) \leq 2\delta n] \leq \rho^{3n}$ by Lemma 11. Combining these inequalities, we obtain that the conductance of T is bounded above by

$$\begin{aligned}
\frac{q(A, A^c)}{p(A)p(A^c)} &\leq \frac{p(B)}{p(A)p(A^c)} \\
&\leq \frac{W^\phi[\delta n \leq n - n_*(x) \leq 2\delta n]}{W^\phi[n - n_*(x) < \delta n] W^\phi[n - n_*(x) > 2\delta n]} \\
&\leq \frac{\rho^{3n}}{\rho^n \ \rho^{\frac{3n}{2}}} = \rho^{n/2},
\end{aligned}$$

which implies condition (II).

□

5.3 Message-passing algorithms on random ensembles

The analysis of the preceding section demonstrated that for values of ρ close to 1, any random sampling technique based on local moves (e.g., Gibbs sampling), if started at the all $*$ assignment, will take exponentially long to get to an assignment with more than a negligible fraction of non-$*$. This section is devoted to establishing an analogous claim for the belief propagation updates on the extended Markov random fields. More precisely, we prove that if ρ is sufficiently close to 1, then running belief propagation with initial messages that place most of their mass on on $*$ will result assignments that also place most of the mass on $*$.

This result is proved in the "density-evolution" setting [e.g., 35] (i.e., the number of iterations is taken to be less than the girth of the graph, so that cycles have no effect). More formally, we establish the following:

Theorem 12. *For every formula density $\alpha > 0$, arbitrary scalars $\epsilon'' > 0$ and $\delta > 0$, there exists $\rho' < 1$, $\epsilon' \in (0, \epsilon'')$ and $\gamma > 0$ such that for all $\rho \in (\rho', 1]$ and $\epsilon \in (0, \epsilon')$, the algorithm $SP(\rho)$ satisfies the following condition.*

Consider a random formula ϕ, a random clause b and a random variable i that belongs to the clause b. Then with probability at least $1 - \delta$, if $SP(\rho)$ is initialized with all messages $\eta^0_{a\to j} < \epsilon$, then the inequality $\eta^t_{b\to i} < \epsilon'$ holds for all iterations $t = 0, 1, \ldots, \gamma \log n$.

The first step of the proof is to compare the SP iterations to simpler "sum-product" iterations.

Lemma 13. *For any $\rho \in [0, 1]$, the $SP(\rho)$ iterations satisfy the inequality:*

$$\eta^{t+1}_{a\to i} \quad \leq \quad \prod_{j\in V(a)\backslash\{i\}} \min\left(1, (1-\rho) + \rho \sum_{b\in C(j)\backslash\{a\}} \eta^t_{b\to j}\right)$$

Proof. See Appendix C.2. □

Since our goal is to bound the messages $\eta^{t+1}_{a\to i}$, Lemma 13 allows us to analyze the simpler message-passing algorithm with updates specified by:

$$\eta^{t+1}_{a\to i} = \prod_{j\in V(a)\backslash\{i\}} \min\left(1, (1-\rho) + \rho \sum_{b\in C(j)\backslash\{a\}} \eta^t_{b\to j}\right). \tag{22}$$

The next step is to bound the probability of "short-cycles" in the computation tree corresponding to the message-passing updates specified in equation (22). More formally, given a formula ϕ, we define a directed graph $G(\phi) = (V, E)$, in which the vertex set V consists of messages $\eta_{a\to i}$. The edge set E includes the edge $\eta_{a\to i} \to \eta_{b\to j}$ belongs to E if and only if $j \in V(a)\backslash\{i\}$ and $b \in C^u_a(i)$. In words, the graph $G(\phi)$ includes an edge between the $\eta_{a\to i}$ and $\eta_{b\to j}$ if the latter is involved in the update of $\eta_{a\to i}$ specified in equation (22).

Lemma 14. *Let $G(\phi)$ be the random graph generated by choosing a formula ϕ uniformly at random with αn clauses and n variables. Let v be a vertex of $G(\phi)$ chosen uniformly at random. For all clause densities $\alpha > 0$, there exists $\gamma > 0$ such that with probability $1 - o(1)$, the vertex v does not belong to any directed cycle of length smaller than $\gamma \log n$ in $G(\phi)$.*

Proof. The proof is based on standard arguments from random graph theory [e.g., 21]. □

Our analysis of the the recursion (22) on the computation tree in based on an edge exposure technique that generates a neighborhood of a vertex v in the graph $G(\phi)$ for a random ϕ. More specifically, pick a clause a and a variable i in a at random. Now for each variable $j \in V(a)\backslash\{i\}$, expose all clauses b containing j (but not any other of the variables appearing so far). Then for each such b, we look at all variables $k \in V(b)\backslash\{j\}$, and so on. We consider the effect of repeating this exposure procedure over $t = \gamma \log n$ steps. When the vertex $\eta_{a\to i}$ does not belong to cycles shorter than t in $G(\phi)$, such an analysis yields a bound on $\eta^t_{a\to i}$.

Note that each clause can expose at most $k-1$ variables. Recall that we generate the formula ϕ by choosing each of the $N_c = 2^k\binom{n}{k}$ clauses with probability $\alpha n/N_c$. The distribution of the number of clauses exposed for each variable is thus dominated by $\mathrm{Bin}(M_c, \alpha n/N_c)$ where $M_c = 2^k\binom{n}{k-1}$. An equivalent description of this process is the following: each vertex $v = \eta_{a\to i}$ exposes X_v neighbors

$\eta_{b\to j}$, where the distribution of the collection $\{X_v\}$ is dominated by a collection $\{Y_v\}$ of i.i.d. random variables. Moreover, the Y's are jointly distributed as the sum of $k-1$ i.i.d. $\mathrm{Bin}(M_c, \alpha n/N_c)$ variables.

The proof requires the following lemma on branching processes.

Lemma 15. *Consider a branching process where each vertex gives birth to Y children. Assume further that the branching process is stopped after m levels and let $K > 0$ be given.*

The notion of a good *vertex is defined inductively as follows. All vertices at level m are good. A vertex at level $m-1$ is good if it has ℓ children and $\ell \le K$. By induction for $s \ge 2$ we call a vertex at level $m-s$ good if v has ℓ children $v_1, \ldots, v_\ell$ with $\ell \le K$ and*

(a) Either all of $v_1, \ldots, v_\ell$ have at most K children, of which all are good; or

(b) all of $v_1, \ldots, v_\ell$ have at most K children, of which all but one are good.

Denote by $p(m, K)$ the probability that the root of the branching process is good. Then

$$\inf_{0\le m<\infty} p(m,K) = 1 - \exp(-\Omega(K)).$$

Proof. See Appendix C.3. ∎

We are now equipped to complete the proof of Theorem 12. Using Lemma 14, first choose $\gamma = \gamma(\alpha)$ such that a random vertex in $G(\phi)$ does not belong to cycles shorter than $\gamma \log n$ with probability $1 - o(1)$. Next use Lemma 15 to choose K such that the probability $\inf_{0\le m<\infty} p(m,K)$ that the root of the branching process is good is at least $1 - \delta/2$.

Next we define a pair of functions θ and ζ (each mapping $R \times R$ to the real line) in the following way:

$$\theta(\epsilon, \rho) := ((1-\rho) + K\rho\epsilon), \qquad\qquad \zeta(\epsilon,\rho) := \theta\left(\theta(\epsilon,\rho),\rho\right) \times \theta\left(\theta(\epsilon,\rho)^2,\rho\right).$$

Setting $\epsilon' := \min(\epsilon'', \frac{1}{2K^3})$, observe that $\theta(\epsilon', 1) = K\epsilon'$ and therefore $\theta^2(\epsilon', 1) \le \frac{\epsilon'}{4}$ and

$$\zeta(\epsilon', 1) = \theta(K\epsilon', 1)\theta((K\epsilon')^2, 1) = (K^2\epsilon')(K^4\epsilon'^2) = K^6\epsilon'^3 \le \frac{\epsilon'}{4}.$$

It now follows by continuity that there exists $\rho' < 1$ such that for all $1 \ge \rho \ge \rho'$ it holds that

$$\theta^2(\epsilon', \rho) \le \tfrac{\epsilon'}{2}, \quad \zeta(\epsilon', \rho) \le \tfrac{\epsilon'}{2}. \tag{23}$$

We claim that the statement of the theorem holds with the choices of γ, ϵ' and ρ' above. Indeed, choose a formula ϕ with density α at random and let $v = \eta_{a\to i}$ be a random vertex of $G(\phi)$. With probability at least $1-\delta/2$, the vertex v does not belong to any cycle shorter than $t = \gamma \log n$.

Since v does not belong to any such cycle, the first t levels of the computation tree of v may be obtained by the exposure process defined above. We will then compare the computation tree to an exposure process where each variable gives birth to exactly $Bin(M_c, \alpha n/N_c)$ clauses. Since the messages are generated according to (22), any bound derived on the values of non-$*$ messages for the larger tree implies the same bound for the real computation tree.

We now claim that if v is a good vertex on that tree, then the message at v after t iterations—namely, $\eta^t_{a\to i}$—is at most ϵ'. Since a vertex of the tree is good with probability $1-\delta/2$, proving this claim will establish the theorem.

We prove this claim by induction on s, where $m-s$ is the level of w. For $s=0$, the claim follows immediately from the initialization of the messages. For $s=1$, observe that equation (22) implies that if $w=\eta_{b\to j}$ is good at level $m-1$, then

$$\eta_{b\to j} \leq \theta^{k-1}(\rho,\epsilon) \leq \theta^2(\rho,\epsilon') \leq \frac{\epsilon'}{2}.$$

For the general induction step, assume that $w=\eta_{b\to j}$ at level $m-s$ is good and $s\geq 2$. There are two cases to consider:

(i) w has all its grand children good. In this case we repeat the argument above twice to obtain $\eta_{b\to j} \leq \epsilon'$.

(ii) Exactly one of $w=\eta_{b\to j}$ grand children is not good. Let $y'=\eta_{d'\to \ell'}$ denote the grand-child and $y=\eta_{d\to\ell}$ denote y parent. Then by equation (22):

$$\eta_{d\to\ell} \leq (1-\rho)+K\rho\epsilon' = \theta(\epsilon',\rho).$$

Using (13) again yields

$$\begin{aligned}\eta_{d\to\ell} &\leq ((1-\rho)+K\rho\theta(\epsilon',\rho))((1-\rho)+K\rho\theta^2(\epsilon',\rho))^{k-2}\\ &\leq ((1-\rho)+K\rho\theta(\epsilon',\rho))((1-\rho)+K\rho\theta^2(\epsilon',\rho)) = \zeta(\epsilon',\rho) \leq \epsilon'/2,\end{aligned}$$

which completes the proof.

6 Conclusion

The survey propagation algorithm, recently introduced by Mézard, Parisi and Zecchina [28] for solving random instances of k-SAT problems, has sparked a great deal of excitement and research in both the statistical physics and computer science communities [e.g., 6, 5, 7, 3, 2, 32, 33, 41]. This paper provides a new interpretation of the survey propagation algorithm—namely, as an instance of the well-known belief propagation algorithm but as applied to a novel probability distribution over the partial satisfiability assignments associated with a k-SAT formula. The perspective of this paper reveals the combinatorial structure that underlies survey propagation algorithm, and we established various results on the form of these structures and the behavior of message-passing algorithms, both for fixed instances and over random ensembles.

The current work suggests various questions and open issues for further research. As we described, associated with any k-SAT problem is a large family of Markov random fields over partial assignments, as specified by the parameter ρ (or more generally, the parameters ω_o and ω_*). Further analysis of survey propagation and its generalizations requires a deeper understanding of the following two questions. First, for what parameter choices do the marginals of the associated Markov random field *yield useful information* about the structure of satisfiability assignments? Second, for what parameter choices do efficient message-passing algorithms like belief propagation *yield accurate approximations* to these marginals? Our results show that the success of SP-like algorithms depends on a delicate balance between these two factors. (For instance, the marginals of the uniform distribution over SAT assignments clearly contain useful information, but belief propagation fails to yield good approximations for sufficiently large clause densities.) More generally, these questions fall in a broader collection of issues, all related to a deeper understanding of satisfiability problems and especially the relationship between finite satisfiability problems and their asymptotic analysis. Given the fundamental role that satisfiability plays in diverse branches of computer science, further progress on these issues is of broad interest.

7 Acknowledgments

We would like to thank Dimitris Achlioptas, Federico Ardila, Andrea Montanari, Mark Mézard, Giorgio Parisi and Alistair Sinclair for helpful discussions.

A Belief propagation on a generic factor graph

Given a subset $S \subseteq \{1, 2, \ldots, n\}$, we define $x_S := \{x_i \mid i \in S\}$. Consider a probability distribution on n variables $x_1, x_2, \ldots, x_n$, that can be factorized as

$$p(x_1, x_2, \ldots, x_n) = \frac{1}{Z} \prod_{i=1}^{n} \psi_i(x_i) \prod_{a \in C} \psi_a(x_{V(a)}), \tag{24}$$

where for each $a \in C$ the set $V(a)$ is a subset of $\{1, 2, \ldots n\}$; and $\psi_i(x_i)$ and $\psi_a(x_{V(a)})$ are non-negative real functions, referred to as compatibility functions, and

$$Z := \sum_{x} \Big[\prod_{i=1}^{n} \psi_i(x_i) \prod_{a \in C} \psi_a(x_{V(a)}) \Big] \tag{25}$$

is the normalization constant or partition function. A factor graph representation of this probability distribution is a bipartite graph with vertices V corresponding to the variables, called *variable nodes*, and vertices C corresponding to the sets $V(a)$ and called *function nodes*. There is an edge between a variable node i and function node a if and only if $i \in V(a)$. We write also $a \in C(i)$ if $i \in V(a)$.

Suppose that we wish to compute the marginal probability of a single variable i for such a distribution, as defined in equation (5). The belief propagation or sum-product algorithm [24] is an efficient algorithm for computing the marginal probability distribution of each variable, assuming that the factor graph is acyclic. The essential idea is to use the distributive property of the sum and product operations to compute independent terms for each subtree recursively. These recursions can be cast as a message-passing algorithm, in which adjacent nodes on the factor graph exchange intermediate values. Let each node only have access to its corresponding compatibility function. As soon as a node has received messages from all neighbors below it, it can send a message up the tree containing the term in the computation corresponding to it. In particular, let the vectors $M_{i \to a}$ denote the message passed by variable node i to function node a; similarly, the quantity $M_{a \to i}$ denotes the message that function node a passes to variable node i.

The messages from function to variables are updated in the following way:

$$M_{a \to i}(x_i) \;\; \propto \sum_{x_{V(a) \setminus \{i\}}} \Big[\psi_a(x_{V(a)}) \prod_{j \in V(a) \setminus \{i\}} M_{j \to a}(x_j) \Big]. \tag{26}$$

In the other direction, the messages from variable nodes to function nodes are updated as follows

$$M_{i \to a}(x_i) \;\; \propto \;\; \psi_i(x_i) \prod_{b \in C(i) \setminus \{a\}} M_{b \to i}(x_i). \tag{27}$$

It is straightforward to show that for a factor graph without cycles, these updates will converge after a finite number of iterations. Upon convergence, the local marginal distributions at variable nodes and function nodes can be computed, using the message fixed point $\hat{M}$, as follows:

$$F_i(x_i) \propto \psi_i(x_i) \prod_{b \in C(i)} \hat{M}_{b \to i}(x_i) \tag{28a}$$

$$F_a(x_{V(a)}) \propto \psi_a(x_{V(a)}) \prod_{j \in V(a)} \hat{M}_{j \to a}(x_j). \tag{28b}$$

The same updates, when applied to a graph with cycles, are no longer exact due to presence of cycles. An exact algorithm will generally require exponential time. For certain problems, including error-control coding, applying belief propagation to a graph with cycles gives excellent results. Since there are no leaves on graphs with cycles, usually the algorithm is initialized by sending random messages on all edges, and is run until the messages converge to some fixed value [24].

B Derivation of BP updates on the extended MRF

B.1 Messages from variables to clauses

We first focus on the update of messages from variables to clauses. Recall that we use the notation $P_i = S \cup \{a\}$ as a shorthand for the event

$$a \in P_i \quad \text{and} \quad S = P_i \backslash \{a\} \subseteq C_a^s(i),$$

where it is understood that S could be empty.

Lemma 16 (Variable to clause messages). *The variable to clause message vector $M_{i \to a}$ is fully specified by values for pairs (x_i, P_i) of the form:*

$$\{(s_{a,i}, S \cup \{a\}),\ (s_{a,i}, \emptyset \neq P_i \subseteq C_a^s(i)),\ (u_{a,i}, \emptyset \neq P_i \subseteq C_a^u(i)),\ (s_{a,i}, \emptyset),\ (u_{a,i}, \emptyset),\ (*, \emptyset)\}.$$

Specifically, the updates for these five pairs take the following form:

$$M_{i \to a}(s_{a,i},\ P_i = S \cup \{a\}) = \prod_{b \in S} M^s_{b \to i} \prod_{b \in C_a^s(i) \backslash S} M^*_{b \to i} \prod_{b \in C_a^u(i)} M^u_{b \to i} \tag{29a}$$

$$M_{i \to a}(s_{a,i},\ \emptyset \neq P_i \subseteq C_a^s(i)) = \prod_{b \in P_i} M^s_{b \to i} \prod_{b \in C_a^s(i) \backslash P_i} M^*_{b \to i} \prod_{b \in C_a^u(i)} M^u_{b \to i} \tag{29b}$$

$$M_{i \to a}(u_{a,i},\ \emptyset \neq P_i \subseteq C_a^u(i)) = \prod_{b \in P_i} M^s_{b \to i} \prod_{b \in C_a^u(i) \backslash P_i} M^*_{b \to i} \prod_{b \in C_a^s(i)} M^u_{b \to i} \tag{29c}$$

$$M_{i \to a}(s_{a,i},\ P_i = \emptyset) = \omega_o \prod_{b \in C_a^s(i)} M^*_{b \to i} \prod_{b \in C_a^u(i)} M^u_{b \to i} \tag{29d}$$

$$M_{i \to a}(u_{a,i},\ P_i = \emptyset) = \omega_o \prod_{b \in C_a^u(i)} M^*_{b \to i} \prod_{b \in C_a^s(i)} M^u_{b \to i} \tag{29e}$$

$$M_{i \to a}(*, P_i = \emptyset) = \omega_* \prod_{b \in C(i) \backslash \{a\}} M^*_{b \to i}. \tag{29f}$$

Proof. The form of these updates follows immediately from the definition (10) of the variable compatibilities in the extended MRF, and the BP message update (27). $\square$

B.2 Forms of R quantities

In this section, we compute the specific forms of the linear sums of messages defined in equation (14). First, we use the definition (14a) and Lemma 16 to compute the form of $R^s_{i\to a}$:

$$\begin{aligned}
R^s_{i\to a} &:= \sum_{S\subseteq C^s_a(i)} M_{i\to a}(s_{a,i}, P_i = S\cup\{a\}) \\
&= \sum_{S\subseteq C^s_a(i)} \prod_{b\in S} M^s_{b\to i} \prod_{b\in C^s_a(i)\backslash S} M^*_{b\to i} \prod_{b\in C^u_a(i)} M^u_{b\to i} \\
&= \prod_{b\in C^u_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^s_a(i)} (M^s_{b\to i} + M^*_{b\to i})\Big].
\end{aligned}$$

Similarly, the definition (14b) and Lemma 16 allows us compute the following form of $R^u_{i\to a}$:

$$\begin{aligned}
R^u_{i\to a} &= \sum_{S\subseteq C^u_a(i)} M_{i\to a}(u_{a,i}, P_i = S) \\
&= \sum_{S\subseteq C^u_a(i), S\neq\emptyset} \prod_{b\in S} M^s_{b\to i} \prod_{b\in C^u_a(i)\backslash S} M^*_{b\to i} \prod_{b\in C^s_a(i)} M^u_{b\to i} + \omega_o \prod_{b\in C^u_a(i)} M^*_{b\to i} \prod_{b\in C^s_a(i)} M^u_{b\to i} \\
&= \prod_{b\in C^s_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^u_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^u_a(i)} M^*_{b\to i}\Big].
\end{aligned}$$

Finally, we compute $R^*_{i\to a}$ using the definition (14c) and Lemma 16:

$$\begin{aligned}
R^*_{i\to a} &= \Big[\sum_{S\subseteq C^s_a(i)} M_{i\to a}(s_{a,i}, P_i = S)\Big] + M_{i\to a}(*, P_i = \emptyset) \\
&= \Big[\sum_{S\subseteq C^s_a(i), S\neq\emptyset} \prod_{b\in S} M^s_{b\to i} \prod_{b\in C^s_a(i)\backslash S} M^*_{b\to i} \prod_{b\in C^u_a(i)} M^u_{b\to i}\Big] + \omega_o \prod_{b\in C^s_a(i)} M^*_{b\to i} \prod_{b\in C^u_a(()i)} M^u_{b\to i} \\
&\qquad + \omega_* \prod_{b\in C^s_a(i)} M^*_{b\to i} \prod_{b\in C^u_a(i)} M^*_{b\to i} \\
&= \prod_{b\in C^u_a(i)} M^u_{b\to i}\Big[\prod_{b\in C^s_a(i)} (M^s_{b\to i} + M^*_{b\to i}) - (1-\omega_o) \prod_{b\in C^s_a(i)} M^*_{b\to i}\Big] + \omega_* \prod_{b\in C^s_a(i)\cup C^u_a(i)} M^*_{b\to i}.
\end{aligned}$$

B.3 Clause to variable updates

In this section, we derive the form of the clause to variable updates.

Lemma 17 (Clause to variable messages). *The updates of messages from clauses to variables in the extended MRF take the following form:*

$$M^s_{a\to i} = \prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a} \tag{30a}$$

$$M^u_{a\to i} = \prod_{j\in V(a)\backslash\{i\}} (R^u_{j\to a} + R^*_{j\to a}) + \sum_{k\in V(a)\backslash\{i\}} (R^s_{k\to a} - R^*_{k\to a}) \prod_{j\in V(a)\backslash\{i,k\}} R^u_{j\to a} - \prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a} \tag{30b}$$

$$M^*_{a\to i} = \prod_{j\in V(a)\backslash\{i\}} (R^u_{j\to a} + R^*_{j\to a}) - \prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a}. \tag{30c}$$

Proof. (i) We begin by proving equation (30a). When $x_i = s_{a,i}$ and $P_i = S \cup \{a\}$ for some $S \subseteq C_a^s(i)$, then the only possible assignment for the other variables at nodes in $V(a) \backslash \{i\}$ is $x_j = u_{a,j}$ and $P_j \subseteq C_a^u(j)$. Accordingly, using the BP update equation (26), we obtain the following update for $M_{a \to i}^s = M_{a \to i}(s_{a,i}, P_i = S \cup \{a\})$:

$$\begin{aligned} M_{a \to i}^s &= \prod_{j \in V(a) \backslash \{i\}} \sum_{P_j \subseteq C_a^u(j)} M_{j \to a}(u_{a,j}, P_j) \\ &= \prod_{j \in V(a) \backslash \{i\}} R_{j \to a}^u. \end{aligned}$$

(ii) Next we prove equation (30c). In the case $x_i = *$ and $P_i = \emptyset$, the only restriction on the other variables $\{x_j : j \in V(a) \backslash \{i\}\}$ is that they are not all unsatisfying. The weight assigned to the event that they are all unsatisfying is

$$\begin{aligned} \sum_{\left\{S_j \subseteq C_a^u(j) \,:\, j \in V(a) \backslash \{i\}\right\}} \prod_{j \in V(a) \backslash \{i\}} M_{j \to a}(u_{a,j}, S_j) &= \prod_{j \in V(a) \backslash \{i\}} \Big[\sum_{S_j \subseteq C_a^u(j)} M_{j \to a}(u_{a,j}, S_j) \Big] \\ &= \prod_{j \in V(a) \backslash \{i\}} R_{j \to a}^u. \end{aligned} \tag{31}$$

On the other hand, the weight assigned to the event that each is either unsatisfying, satisfying or $*$ can be calculated as follows. Consider a partition $J^u \cup J^s \cup J^*$ of the set $V(a) \backslash \{i\}$, where J^u, J^s and J^* corresponds to the subsets of unsatisfying, satisfying and $*$ assignments respectively. The weight $W(J^u, J^s, J^*)$ associated with this partition takes the form

$$\sum_{\left\{S_j \subseteq C_a^u(j) \,:\, j \in J^u\right\}} \sum_{\left\{S_j \subseteq C_a^s(j) \,:\, j \in J^s\right\}} \prod_{j \in J^u} M_{j \to a}(u_{a,j}, S_j) \prod_{j \in J^s} M_{j \to a}(s_{a,j}, S_j) \prod_{j \in J^*} M_{j \to a}(*, \emptyset).$$

Simplifying by distributing the sum and product leads to

$$\begin{aligned} W(J^u, J^s, J^*) &= \prod_{j \in J^u} \Big[\sum_{S_j \subseteq C_a^u(j)} M_{j \to a}(u_{a,j}, S_j) \Big] \prod_{j \in J^s} \Big[\sum_{S_j \subseteq C_a^s(j)} M_{j \to a}(s_{a,j}, S_j) \Big] \prod_{j \in J^*} M_{j \to a}(*, \emptyset) \\ &= \prod_{j \in J^u} R_{j \to a}^u \prod_{j \in J^s} \left[R_{j \to a}^* - M_{j \to a}(*, \emptyset) \right] \prod_{j \in J^*} M_{j \to a}(*, \emptyset), \end{aligned}$$

where we have used the definitions of $R_{j \to a}^u$ and $R_{j \to a}^*$ from Section B.2. Now summing $W(J^u, J^s, J^*)$ over all partitions $J^u \cup J^s \cup J^*$ of $V(a) \backslash \{i\}$ yields

$$\begin{aligned} & \sum_{J^u \cup J^s \cup J^*} W(J^u, J^s, J^*) \\ &= \sum_{J^u \subseteq V(a) \backslash \{i\}} \prod_{j \in J^u} R_{j \to a}^u \sum_{J^s \cup J^* = V(a) \backslash \{J^u \cup i\}} \Big\{ \prod_{j \in J^s} \left[R_{j \to a}^* - M_{j \to a}(*, \emptyset) \right] \prod_{j \in J^*} M_{j \to a}(*, \emptyset \Big\} \\ &= \sum_{J^u \subseteq V(a) \backslash \{i\}} \prod_{j \in J^u} R_{j \to a}^u \prod_{j \in V(a) \backslash \{J^u \cup i\}} R_{j \to a}^* \\ &= \prod_{j \in V(a) \backslash \{i\}} \left[R_{j \to a}^u + R_{j \to a}^* \right], \end{aligned} \tag{32}$$

where we have used the binomial identity twice. Overall, equations (31) and (32) together yield that

$$M^*_{a\to i} \;=\; \prod_{j\in V(a)\backslash\{i\}} \left[R^u_{j\to a} + R^*_{j\to a}\right] - \prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a},$$

which establishes equation (30c).

(iii) Finally, turning to equation (30b), for $x_i = u_{a,i}$ and $P_i \subseteq C^u_a(i)$, there are only two possibilities for the values of $x_{V(a)\backslash\{i\}}$:

(a) either there is one satisfying variable and everything else is unsatisfying, or

(b) there are at least two variables that are satisfying or $*$.

We first calculate the weight $W(A)$ assigned to possibility (a), again using the BP update equation (26):

$$\begin{aligned}
W(A) &= \sum_{k\in V(a)\backslash\{i\}} \sum_{S^k\subseteq C^s_a(k)} M_{k\to a}(s_{a,k}, S^k \cup \{a\}) \prod_{j\in V(a)\backslash\{i,k\}} \sum_{S^j\subseteq C^u_a(j)} M_{j\to a}(u_{j,a}, S^j) \\
&= \sum_{k\in V(a)\backslash\{i\}} R^s_{k\to a} \prod_{j\in V(a)\backslash\{i,k\}} R^u_{j\to a},
\end{aligned} \tag{33}$$

where we have used the definitions of $R^s_{k\to a}$ and $R^u_{k\to a}$ from Section B.2.

We now calculate the weight $W(B)$ assigned to possibility (b) in the following way. From our calculations in part (ii), we found that the weight assigned to the event that each variable is either unsatisfying, satisfying or $*$ is $\prod_{j\in V(a)\backslash\{i\}} \left[R^u_{j\to a} + R^*_{j\to a}\right]$. The weight $W(B)$ is given by subtracting from this quantity the weight assigned to the event that there are *not* at least two $*$ or satisfying assignments. This event can be decomposed into the disjoint events that either all assignments are unsatisfying (with weight $\prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a}$ from part (ii)); or that exactly one variable is $*$ or satisfying. The weight corresponding to this second possibility is

$$\begin{aligned}
&\sum_{k\in V(a)\backslash\{i\}} \left[M_{k\to a}(*, \emptyset) + \sum_{S^k\subseteq C^s_a(k)} M_{k\to a}(s_{k,a}, S^k)\right] \prod_{j\in V(a)\backslash\{i,k\}} \sum_{S^j\subseteq C^u_j(a)} M_{j\to a}(u_{j,a}, S^j) \\
&\qquad = \sum_{k\in V(a)\backslash\{i\}} R^*_{k\to a} \prod_{j\in V(a)\backslash\{i,k\}} R^u_{j\to a}.
\end{aligned}$$

Combining our calculations so far we have

$$W(B) \;=\; \prod_{j\in V(a)\backslash\{i\}} \left[R^u_{j\to a} + R^*_{j\to a}\right] - \sum_{k\in V(a)\backslash\{i\}} R^*_{k\to a} \prod_{j\in V(a)\backslash\{i,k\}} R^u_{j\to a} - \prod_{j\in V(a)\backslash\{i\}} R^u_{j\to a}. \tag{34}$$

Finally, summing together the forms of $W(A)$ and $W(B)$ from equations (33) and (34) respectively, and then factoring yields the desired equation (30b). □

C Proofs for random formulae

C.1 Proof of Lemma 11

In order to prove (21), it suffices by the Markov inequality to show that for every integer d in the interval $[\delta n, 2\delta n]$, it holds that

$$\frac{\mathbb{E}^n[W^\phi[n - n_* = d]]}{\rho^{3n}} = \exp(-\Omega(n)). \tag{35}$$

To establish (35), consider a fixed set of d variables. The average W-weight assigned to the event that this set of size d constitutes all the non-star variables is bounded by

$$\rho^{n-d} \sum_{r=0}^{d} (1-\rho)^{d-r} \binom{d}{r} \binom{\alpha n}{r} (d/n)^{kr},$$

where r represents the number of constrained variables. We obtain this bound by the following reasoning. First, the $n - d$ variables assigned $*$ all receive weight ρ. Otherwise, if r out of the remaining d variables are constrained, there must be r clauses chosen from a total of αn, and each such clause must have all of its k variables chosen from within the set of d non-star variables.

Consequently, the total probability of having d non-star variables is bounded by

$$\begin{aligned}
\rho^{n-d}\binom{n}{d} \sum_{r=0}^{d} (1-\rho)^{d-r} \binom{d}{r} \binom{\alpha n}{r} \left(\frac{d}{n}\right)^{kr} &\leq \rho^{n-d} \left(\frac{en}{d}\right)^d \sum_{r=0}^{d} (1-\rho)^{d-r} \left(\frac{ed}{r}\right)^r \left(\frac{\alpha e n}{r}\right)^r \left(\frac{d}{n}\right)^{kr} \\
&= \rho^{n-d} \left(\frac{(1-\rho)en}{d}\right)^d \sum_{r=0}^{d} \left(\frac{e^2 d^{k+1}\alpha}{r^2(1-\rho)n^{k-1}}\right)^r,
\end{aligned}$$

Recalling that $1 - \rho = \delta^2$ and $d \in [\delta n,\, 2\delta n]$, we obtain that the last expression is at most

$$\begin{aligned}
\rho^{n-2\delta n} \left(\frac{\delta^2 en}{\delta n}\right)^d \sum_{r=0}^{2\delta n} \left(\frac{e^2(2\,\delta n)^{k+1}\alpha}{r^2\delta^2 n^{k-1}}\right)^r &= \rho^{n-2\delta n}(\delta e)^d \sum_{r=0}^{2\delta n} \left(\frac{e^2 2^{k+1}\delta^{k-1}n^2\alpha}{r^2}\right)^r \\
&\leq \rho^{n-2}(\delta e)^{\delta n} \sum_{r=0}^{2\delta n} \left(\frac{2^{k+1}\alpha\delta^{k-1}n^2 e^2}{r^2}\right)^r,
\end{aligned}$$

where the final inequality is valid when $\delta e < 1$. A straightforward calculation yields that the function $g(r) := \left(\frac{2^{k+1}\alpha\delta^{k-1}e^2n^2}{r^2}\right)^r$ is maximized at $r^* = \sqrt{2^{k+1}\alpha\delta^{k-1}}n$ and the associated value is $g(r^*) = e^{2r^*}$. Consequently, the sum above is bounded by

$$\begin{aligned}
2\delta n\rho^{n-2\delta n}(\delta e)^{\delta n} e^{2r^*} &= 2\delta n\rho^{n-2\delta n}\left[\delta \exp\left(1 + \frac{2r^*}{\delta n}\right)\right]^{\delta n} \\
&= 2\delta n\rho^{n-2\delta n}\left[\delta \exp\left(1 + \sqrt{2^{k+3}\alpha\delta^{k-3}}\right)\right]^{\delta n} \\
&\leq 2\delta n\rho^{n-2\delta n}\left[\delta \exp\left(1 + \sqrt{2^{k+3}\alpha}\right)\right]^{\delta n}.
\end{aligned}$$

This expression is exponentially smaller than ρ^{3n} for large n if

$$\left[\delta \exp\left(1+\sqrt{2^{k+3}\alpha}\right)\right]^{\delta} < \rho^3 = (1-\delta^2)^3. \tag{36}$$

Inequality (36) holds for sufficiently small $\delta > 0$, which establishes the lemma.

C.2 Proof of Lemma 13

It will be useful to denote $\prod_{b\in C_a^s(i)}(1-\eta_{b\to i})$ by $P_s(i)$ and $\prod_{b\in C_a^u(i)}(1-\eta_{b\to i})$ by $P_u(j)$. With this notation, the j'th term in (6) is given by

$$\begin{aligned}\frac{\Pi^u_{j\to a}}{\Pi^u_{j\to a}+\Pi^s_{j\to a}+\Pi^*_{j\to a}} &= \frac{(1-\rho P_u(j))P_s(j)}{(1-\rho P_u(j))P_s(j)+(1-P_s(j))P_u(j)+P_s(j)P_u(j)}\\ &= \frac{(1-\rho P_u(j))P_s(j)}{P_s(j)+P_u(j)-\rho P_s(j)P_u(j)} \leq 1-\rho P_u(j).\end{aligned}$$

We therefore conclude that

$$\eta_{a\to i} \leq \prod_{j\in V(a)\setminus\{i\}} (1-\rho P_u(j)).$$

On the other hand, we have $P_u(j) = \prod_{b\in C_a^u(i)}(1-\eta_{b\to i}) \geq \max\left(0, 1-\sum_{b\in C_a^u(i)}\eta_{b\to i}\right)$, so that

$$1-\rho P_u(j) \leq \min\left(1, (1-\rho)+\rho\sum_{b\in C_a^u(i)}\eta_{b\to i}\right).$$

This yields the bound $\eta^{t+1}_{a\to i} \leq \prod_{j\in V(a)\setminus\{i\}}\min\left(1,(1-\rho)+\rho\sum_{b\in C_a^u(i)}\eta^t_{b\to j}\right)$, from which equation (22) follows.

C.3 Proof of Lemma 15

We start by estimating the probability that a vertex is bad by induction. Let g_K denote the probability that v has more than K children, or that one of v's children has more than K children. Clearly,

$$g_K \leq (K+1)\mathbb{P}[Y \geq K] \leq (K+1)(k-1)\mathbb{P}[\mathrm{Bin}(M_c, \frac{\alpha n}{N_c}) \geq \frac{K}{k-1}] \leq \exp(-\Omega(K)). \tag{37}$$

Write $q(m,K) = 1-p(m,K)$ and note that $q(0,K)=0$ and $q(1,K)\leq g_K$. By induction, A vertex can be bad for two reasons: it has two many descendants in the two levels below it, or it has 2 bad descendant in the two levels below it. We may thus bound the probability of a vertex being bad as

$$q(s,K) \leq g_K + \mathbb{P}[\mathrm{Bin}(K^2, q(s-2,K)) \geq 2]. \tag{38}$$

Note also that

$$\mathbb{P}[\mathrm{Bin}(K^2, q(s-2,K)) \geq 2] \leq K^4 q(s-2,K)^2. \tag{39}$$

Combining (38) and (39) yields

$$q(s,K) \leq g_K + K^4 q(s-2,K)^2. \tag{40}$$

By (37) when K is sufficiently large $K^4(2g_K)^2 < g_K$. Thus when K is sufficiently large, it follows from equation (40) that

$$q(s, K) \leq 2g_K$$

for all s. Finally when K is sufficiently large $p(s, K) \geq 1 - 2g_K$ for all s and $1 - 2g_K \geq 1 - \exp(-\Omega(K))$ as needed.

References

[1] D. Achiloptas and Y. Peres. The threshold for random k-SAT is $2^k 2 \log 2 - o(k)$. In *Proceedings of FOCS*, pages 223–231, 2003.

[2] D. Achlioptas and F. Ricci-Tersenghi. Clustering in random k-sat. Manuscript, 2005.

[3] E. Aurell, U. Gordon, and S. Kirkpatrick. Comparing beliefs, surveys, and random walks. In *Neural Information Processing Systems*, January 2005.

[4] N. Berger, C. Kenyon, E. Mossel, and Y. Peres. Glauber dynamics on trees and hyperbolic graphs. To appear. Extended abstract by Kenyon, Mossel and Peres appeared in proceeding of 42nd STOC, 2004.

[5] A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina. Constraint satisfaction by survey propagation. Technical report, 2003. Preprint at URL:http://lanl.arXiv.org/cond-mat/0212451.

[6] A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. Technical report, 2003. Preprint at URL:http://lanl.arXiv.org/cs.CC/0212002.

[7] A. Braunstein and R. Zecchina. Survey propagation as local equilibrium equations. Technical report, 2004. Preprint at URL:http://lanl.arXiv.org/cond-mat/0312483.

[8] V. Chvatal and B. Reed. Mick gets some (the odds are on his side). In *Proceedings of 33rd FOCS*, Pittsburgh Pennsylvania, 1992.

[9] S. Cook. The complexity of theorem-proving procedures. In *Proceeding of 3rd STOC*, page 151, 1971.

[10] G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[11] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation. In *European Conference on Computer Vision*, 2002.

[12] P. Dagum and M. Luby. Approximate probabilistic reasoning in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[13] W. F. de la Vega. On random 2-sat. Unpublished manuscript., 1992.

[14] R. Dechter. *Constraint processing*. Morgan Kaufmann, Palo Alto, CA, 2003.

[15] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-sat formulae and the satisfiability threshold. In *Proceedings of 11'th SODA*, pages 126–127, 2000.

[16] U. Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the 34th STOC*, 2002.

[17] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Intl. J. Computer Vision*, 40(1):25–47, 2000.

[18] E. Friedgut. Neccesary and sufficient conditions for sharp threhsolds of graph properties and the k-problem. *J. Amer. Math. Soc.*, 12:1017–1054, 1999.

[19] R. G. Gallager. *Low-density parity check codes*. MIT Press, Cambridge, MA, 1963.

[20] A. Goerdt. A remark on random 2-sat. *J. Computer System and Sciences*, 53:469–486, 1996.

[21] S. Janson, T. Łuczak, and A. Rucinski. *Random graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[22] A. Kaporis, L. M. Kirousis, and E. G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. In *Proceedings of 10'th Annual European Symposium on Algorithm*, pages 574–585, 2000.

[23] S. Kirkpatrick. On survey propagation. Personal communication, 2004.

[24] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, 47:498–519, February 2001.

[25] L. Levin. Average case complete problems. *SIAM Jour. Comput.*, 15, 1986.

[26] E. Maneva, E. Mossel, and M. J. Wainwright. A new look at survey propagation and its generalizations. In *Proceedings of Symposium on Discrete Algorithms*, pages –, 2005.

[27] F. Martinelli. Lectures on Glauber dynamics for discrete spin models. In *Lectures on probability theory and statistics (Saint-Flour, 1997)*, volume 1717 of *Lecture Notes in Math.*, pages 93–191. Springer, Berlin, 1999.

[28] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297, 812, 2002. (Scienceexpress published on-line 27-June-2002; 10.1126/science.1073287).

[29] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina. Two solutions to diluted p-spin models and xorsat problems. *J. Stat. Phys.*, 111:505, 2003.

[30] M. Mézard and R. Zecchina. Random k-satisfiability: from an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66, 2002.

[31] R. Monasson and R. Zecchina. Statistical mechanics of the random k-satisfiability model. *Phys. Rev. E*, 3:1357–1370, 1997.

[32] T. Mora, M. Mézard, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 2005. In press.

[33] G. Parisi. On local equilibrium equations for clustering states. Technical report, 2002. Preprint at URL:http://lanl.arXiv.org/cs.CC/0212047.

[34] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, Palo Alto, CA, 1988.

[35] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Info. Theory*, 47:599–618, February 2001.

[36] J. Rosenthal, J. Plotkin, and J. Franco. The probability of pure literals. *Journal of Computational Logic*, 9:501–513, 1999.

[37] B. Selman, H. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In D. S. Johnson and M. A. Trick, editors, *Cliques, coloring, and satisfiability : second DIMACS implementation challenge, October 11-13, 1993*, Providence, RI, 1996. American Mathematical Society.

[38] R. P. Stanley. *Enumerative combinatorics*, volume 1. Cambridge University Press, Cambridge, UK, 1997.

[39] S. Tatikonda and M. I. Jordan. Loopy belief propagation and Gibbs measures. In *Uncertainty in Artificial Intelligence (UAI), Proceedings of the Eighteenth Conference*, 2002.

[40] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational methods. Technical report, UC Berkeley, Department of Statitics, No. 649, 2003. Preprint at URL:http://www.eecs.berkeley.edu/~wainwrig/Papers/WaiJorVariational03.ps.

[41] M. J. Wainwright and E. Maneva. Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes. In *Proceedings of IEEE International Symposium on Information Theory*, 2005.

[42] J. Wang. Average case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity theory retrospective*, volume II. Springer, 1997.

[43] M. Welling and Y. Teh. Belief optimization: A stable alternative to loopy belief propagation. In *Uncertainty in Artificial Intelligence*, July 2001.

[44] J. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. Info. Theory*, 51(7):2282–2312, July 2005.

[45] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*, chapter 8. Science and Technology books, 2003.

[46] A. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.