

# CONSTRAINT SATISFACTION BY SURVEY PROPAGATION

A. Braunstein<sup>(1,4)</sup>, M. Mézard<sup>(2)</sup>, M. Weigt<sup>(3)</sup>, R. Zecchina<sup>(4)</sup>

<sup>1</sup>*International School for Advanced Studies (SISSA),  
via Beirut 9, 34100 Trieste, Italy*

<sup>2</sup>*Laboratoire de Physique Théorique et Modèles Statistiques,  
CNRS and Université Paris Sud,*

*Bât. 100, 91405 Orsay CEDEX, France*

<sup>3</sup>*Institute for Theoretical Physics,  
University of Göttingen, Tammannstr. 1,  
37077 Göttingen, Germany*

<sup>4</sup>*The Abdus Salam International Centre for Theoretical Physics (ICTP),  
Str. Costiera 11, 34100 Trieste, Italy*

Survey Propagation (SP) is an algorithm designed for solving typical instances of random constraint satisfiability problems. It has been successfully tested on random 3-satisfiability (3-SAT) and random  $\mathcal{G}(n, \frac{c}{n})$  graph 3-coloring (3-COL), in the *hard* region of the parameter space, relatively close to the SAT/UNSAT phase transition. Here we provide a generic formalism which applies to a wide class of discrete Constraint Satisfaction Problems.

## I. INTRODUCTION

In this paper we suggest a new theoretical framework for the so called “Survey Propagation” (SP) equations that are at the root of both the analysis and the algorithms used in ref. [1, 2, 3, 4] to solve the random 3-SAT and q-COLORING problems. In the more general context of constraint satisfaction problems we propose a slightly different way of deriving the equations which we hope can shed some light on the potentialities of the algorithms and which makes clear the differences with other well known iterative probabilistic algorithms. This line of approach, also discussed in [5] for the satisfiability problem, is developed here systematically through the addition of an extra state for the variables which allows to take care of the clustered structure of the space of solutions. Within clusters a variable can be either “frozen” to some value – that is, the variable takes always the same value for all solutions (satisfying assignments) within the cluster – or it may be “unfrozen” – that is it fluctuates from solution to solution within the cluster. As we shall discuss, scope of the SP equations is to properly describe the cluster to cluster fluctuations by associating to unfrozen variables an extra state to be added to those belonging to the original definition of the problem. The overall algorithmic strategy is iterative and decomposable in two elementary steps: First, the marginal probabilities of frozen variables are evaluated by the SP message-passing procedure; Second – the so called *decimation* step – using such information some variables are fixed and the problem is simplified. While the first step is unavoidable if one is interested in marginal probabilities, the second step is just dictated by simplicity and we expect that there could exist other ways of efficiently using the information provided by the marginals.

Throughout the paper, a detailed comparison with a similar message-passing procedure, Belief Propagation, which does not make assumptions about the structure of the solution space will also be given.

The structure of the paper is as follows. In Sec. II, we provide the general formalism, namely the definitions of Constraint Satisfaction Problems, Factor Graphs and Cavities, with concrete reference to the cases of Coloring and Satisfiability. In Sec. III, we introduce the *warnings* and the *local fields* whose histograms will provide the so called Belief Propagation equations. Finally in Sec. IV, clusters are introduced and the SP equations are derived. Explicit equations are given for both 3-COL and 3-SAT and the decimation procedure is discussed.

## II. GENERALITIES

### A. Constraint satisfaction problems

We consider a *constraint satisfaction problem* (CSP) which is defined on a set of *discrete variables*  $\vec{x} = (x_i)_{i \in I}$  with  $I = \{1, \dots, n\}$ . Each variable  $x_i$  can be in  $q$  possible states (the generalization to the case where the number of states is  $i$ -dependent is straightforward), so  $\vec{x} \in X = \{1, \dots, q\}^n$ . The vector  $\vec{x}$  is called a *configuration*. These variables are subject to a set of *constraints*  $\{C_a\}_{a \in A}$ . Each  $C_a$  depends on  $\vec{x}$  only through a subset  $(x_i)_{i \in I(a)}$  of variables. It is defined as a mapping  $C_a : \{1, \dots, q\}^{|I(a)|} \rightarrow \{0, 1\}$ , where the value  $C_a = 0$  zero corresponds to a satisfied constraint, and  $C_a = 1$  to an unsatisfied constraint. It is useful to introduce, for every  $i \in I$ , the subset  $A(i) \subset A$  of indices of

all constraints involving  $x_i$ . The index sets  $I$  and  $A$  are chosen disjoint, so that their elements uniquely determine a single variable or constraint.

We define the *cost function*

$$C[\vec{x}] = \sum_{a \in A} C_a[(x_i)_{i \in I(a)}] \quad (1)$$

which counts the number of unsatisfied constraints. Our goal is to simultaneously satisfy all constraints, i.e. to find a configuration  $\vec{s} \in X$  with  $C[\vec{s}] = 0$ . We thus introduce the subset  $S_C \subset X$  of *solutions* to our CSP instance as

$$S_C = \{\vec{s} \mid \vec{s} \in X, C[\vec{s}] = 0\} . \quad (2)$$

The algorithm aims at finding one solution  $\vec{s} \in S_C$ . We concentrate *a priori* onto instances  $C[\vec{x}]$  which possess a non-empty solution set  $S_C$ .

## B. Factor graph

We use the *factor-graph* [6] representation for a CSP:

**Definition II.1** *For any instance of the CSP problem, we define its **factor graph** as a bipartite undirected graph  $G = (V, E)$ , having two types of nodes:*

- *variable nodes  $i \in I$  and*
- *function nodes  $a \in A$ .*

*Edges connect only different node types; the edge  $(i, a)$  belongs to the graph if and only if the constraint  $C_a$  involves the variable  $x_i$ , i.e. if  $a \in A(i)$  or equivalently  $i \in I(a)$ . More formally, we define  $V = A \cup I$  and  $E = \{(i, a) \mid i \in I, a \in A(i)\} = \{(i, a) \mid a \in A, i \in I(a)\}$ .*

In figures, we always represent variable nodes by circles, whereas function nodes are drawn as squares. This notation will help to distinguish between the different origins of the two node types.

## C. Cavities

Given a CSP and its factor graph, we will use the cavity graphs obtained by removing a variable:

**Definition II.2** *Given a factor graph  $G$  and one variable node  $i \in I$ , we define the **cavity graph**  $G^{(i)}$  by deleting from  $G$  all function nodes  $a \in A(i)$  which are adjacent to  $i$ , and the edges incident to these function nodes.*

The cavity graph  $G^{(i)}$  defines a new CSP, where the cost function is

$$C^{(i)} = C - \sum_{a \in A(i)} C_a = \sum_{b \notin A(i)} C_b . \quad (3)$$

Note that in this new problem the variable  $x_i$  is isolated, it can take any value without violating a constraint. The solution set  $S^{(i)}$  for the cavity problem  $G^{(i)}$  is larger than the original one, since some constraints have been removed.

## D. Two examples: Satisfiability and Coloring

Although the algorithm can in principle be written for arbitrary CSP, we shall present two specific examples, satisfiability and coloring.

In the satisfiability problem a constraint  $C_a$  is a clause, which is unsatisfied by only one assignment of the variables  $(x_i)_{i \in I(a)}$ . In the random 3-SAT problem each clause involves three variables ( $|I(a)| = 3$ ), the indices of which are chosen randomly with uniform distribution in  $I$ . For a given  $a$  and  $I(a)$ , there are eight different types of constraints  $C_a$ , corresponding to the combinations of possible negations of literals in one clause, see Fig. 1. In random 3-SAT the type of clauses is chosen with uniform distribution among these eight types.

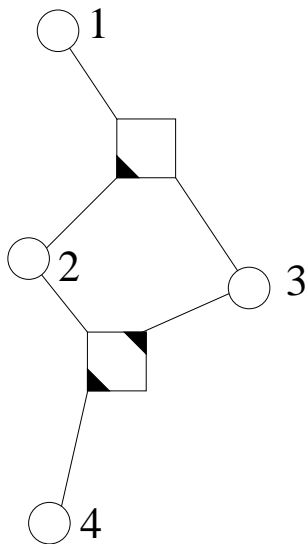


Figure 1: The factor graph of a 3-SAT problem corresponding to the simple formula:  $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$ . Variables are represented as circles, clauses (i.e. function nodes) as squares. A triangle-shaped mark indicates that the corresponding literal is negated

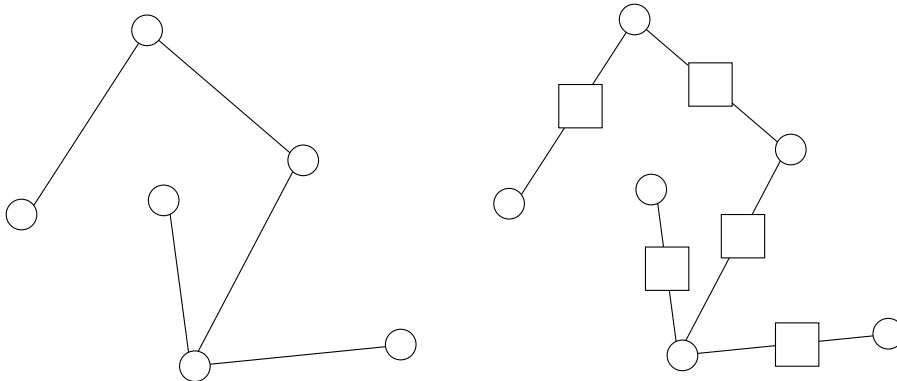


Figure 2: The original graph (left) and its factor graph (right) corresponding to a coloring problem

In the  $q$ -coloring problem one is given an original undirected graph. The problem is to color the vertices, using  $q$  colors, so that two vertices connected by an edge have different colors. There is one constraint associated with each edge of the original graph, and the factor-graph appears as a decoration of the original graph (see Fig.2), where function nodes have been added on each original edge. There is only one type of function node. In the random  $q$ -COL problem, the original graph is a random  $\mathcal{G}(n, \frac{c}{n})$  graph.

We will be particularly interested in the behavior of the algorithm for large  $n$ . Note that both  $K$ -SAT and  $q$ -COL are problems where  $|A(i)|$  have a Poisson limit distribution with finite mean when  $n \rightarrow \infty$ , i.e.  $|A(i)|$  is typically much smaller than  $n$ . Moreover, the structure of the factor graph is locally tree-like. This will guide us in the definition of the algorithm below, and it is presumably an important ingredient for the algorithm to work.

### III. BELIEF PROPAGATION

#### A. Warnings and fields

Given a CSP and a configuration  $\vec{x} \in X$ , we define the following three quantities associated with  $\vec{x}$ , cf. [2, 3]:

**Definition III.1** For a given edge  $a - i$  of the factor graph, with  $i \in I$  and  $a \in A(i)$ , we define a **warning** as the

$q$ -component vector  $\vec{u}_{a \rightarrow i}(\vec{x}) \in \{0, 1\}^q$  with components:

$$u_{a \rightarrow i}^p(\vec{x}) = C_a \left[ (x_j)_{j \in I(a)} \mid x_i \leftarrow p \right], \quad p = 1, \dots, q,$$

So  $u_{a \rightarrow i}^p(\vec{x})$  is the value of constraint  $C_a$  in the configuration obtained from  $\vec{x}$ , by substituting  $p$  in the place of  $x_i$ .

**Definition III.2** For a given edge  $a - i$  of the factor graph, with  $i \in I$  and  $a \in A(i)$ , we define a **cavity field** as the  $q$ -component vector  $\vec{h}_{i \rightarrow a}(\vec{x}) \in \{0, 1\}^q$  with components:

$$h_{i \rightarrow a}^p(\vec{x}) = \max_{b \in A(i) \setminus a} u_{b \rightarrow i}^p(\vec{x})$$

**Definition III.3** For a given node  $i \in I$ , we define the **local field** as the  $q$ -dimensional vector  $\vec{h}_i(\vec{x}) \in \{0, 1\}^q$  with components:

$$h_i^p(\vec{x}) = \max_{a \in A(i)} u_{a \rightarrow i}^p(\vec{x}).$$

The warning  $\vec{u}_{a \rightarrow i}(\vec{x})$  is understood as a *message* sent from constraint  $a$  to variable  $i$  saying:  $x_i$  cannot be in any of the states  $p$  where  $u_{a \rightarrow i}^p(\vec{x}) = 1$ , without violating constraint  $a$ . Note that we do not need the value of  $x_i$  for computing  $\vec{u}_{a \rightarrow i}$ . In fact, the warning depends explicitly only on  $(x_j)_{j \in I(a) \setminus i}$ .

The local field on variable  $i$  summarizes all warnings sent to  $i$  from the constraints, i.e.  $h_i^p(\vec{x}) \neq 0$  means that, given the values of all other variables  $(x_j), j \neq i$ , the variable  $x_i$  should not be assigned the value  $p$ , because at least one neighboring constraint would be violated.

The cavity field  $\vec{h}_{i \rightarrow a}(\vec{x})$  summarizes all warnings sent to  $i$  from the constraints different from  $a$ .

## B. Histograms

The elementary messages above are defined for an arbitrary configuration  $\vec{x}$ . We are eventually interested in knowing, for each variable  $i$ , the *histogram of local fields* for the configurations which are solutions of the CSP:

$$H_i(\vec{h}) = \frac{1}{|S_C|} \sum_{\vec{s} \in S_C} \delta_{\vec{h}, \vec{h}_i(\vec{s})}, \quad (4)$$

where the ( $q$ -dimensional) Kronecker-Delta is simply denoted by  $\delta$ . This histogram can also be interpreted as probability distributions  $H_i(\vec{h}) = \text{Prob}(\vec{h}_i(\vec{s}) = \vec{h} \mid \vec{s} \in S_C)$  of local fields for randomly chosen solutions.

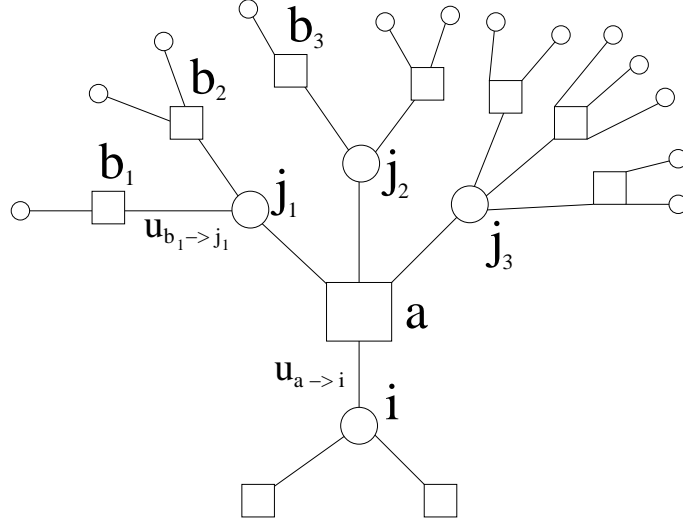
Local-field histograms contain useful information about the set  $S_C$  of solutions, which can be exploited algorithmically in order to recursively construct one solution. If, e.g., one of the field components is non-zero for all solutions  $\vec{s} \in S_C$ , this particular state is forbidden to this variable. If all but one components are non-zero, the variable is “frozen” to one specific value in all solutions, i.e. it belongs to the so-called backbone, and it can be assigned right away.

Computing  $H_i(\vec{h})$  is a difficult task, but one may compute it approximately using a message passing procedure. We first try to find a recursion relation for the related histograms of the warnings  $\vec{u}_{a \rightarrow i}(\vec{s})$  over all solutions  $\vec{s} \in S_C$  [2]. Considering Fig. 3 as an example, we note that the histogram of  $\vec{u}_{a \rightarrow i}(\vec{s})$  depends on the *joint* histogram of all the warnings  $\vec{u}_{b \rightarrow j}(\vec{s})$  sent to all variables  $j \in \{j_1, j_2, j_3\}$  “above” function node  $a$  (we call them the incoming warnings). The obvious problem is that this joint distribution is not known. If the  $\vec{u}_{b \rightarrow j}(\vec{s})$  were independent variables, we would be able to factorize the joint histogram into the product of all individual histograms of warnings  $\vec{u}_{b \rightarrow j}(\vec{s})$ , and then to obtain a recursion. But in general there is no reason for them to be independent. Moreover, they cannot even be approximately independent as there are very short paths joining variables “above” variable nodes  $j$  (the small unnamed ones in the figure) between them, variables which in turn define the  $\vec{u}_{b \rightarrow j}(\vec{s})$  messages. This is where the cavity graph is useful.

For each edge  $a - i$  of the factor graph, we define the *belief*  $B_{a \rightarrow i}(\vec{u})$  as the histogram of the warning  $\vec{u}_{a \rightarrow i}$  over the configurations  $\vec{s} \in S^{(i)}$  which are solutions of the cavity graph problem:

$$\begin{aligned} B_{a \rightarrow i}(\vec{u}) &= \frac{1}{|S^{(i)}|} \sum_{\vec{s} \in S^{(i)}} \delta_{\vec{u}, \vec{u}_{a \rightarrow i}(\vec{s})} \\ &= \text{Prob} \left( \vec{u}_{a \rightarrow i}(\vec{s}) = \vec{u} \mid \vec{s} \in S^{(i)} \right), \end{aligned} \quad (5)$$

where the second line refers again to the probabilistic interpretation:  $B_{a \rightarrow i}(\vec{u})$  describes the probability of finding a warning  $\vec{u}$  if a solution of the cavity graph is randomly selected.

Figure 3: Iteration for  $\vec{u}$  warnings.

### C. Belief propagation equations

Look again at Fig.3: If the factor graph  $G$  is a tree, vertices above  $j_1, j_2$  and  $j_3$  become disconnected if function nodes  $b_i$  are removed, and the various messages  $\vec{u}_{b \rightarrow j}$  are uncorrelated. In this case, we can thus determine the belief  $B_{a \rightarrow i}$  as a function of all the incoming beliefs (i.e. the histograms of the incoming warnings  $\{B_{b \rightarrow j}\}$  with  $j \in I(a) \setminus i$  and  $b \in A(j) \setminus a$ ), and so on recursively for the full factor graph. Standard belief propagation uses this same recursion also in more general factor graphs with loops, as a means to compute approximately the local-field histograms and the beliefs (see e.g. [7]).

In order to write the corresponding 'belief propagation' equations explicitly, we use notations similar to those of Fig.3. Given the edge  $a - i$  connecting the function node  $a$  to the variable  $i$ , we denote by  $J$  the set of indices of the variable nodes "above" the function node  $a$ , i.e.  $J = I(a) \setminus i$  (in the figure  $J = \{j_1, j_2, j_3\}$ ). For each  $j \in J$ , we denote by  $D_j = A(j) \setminus a$  the set of function node "above" the variable  $j$  (in the figure,  $D_{j_1} = \{b_1, b_2\}$ , and by  $\hat{D}$  the union of these sets:  $\hat{D} = \bigcup_{j \in J} D_j$ ). The "incoming messages", which can be warning or beliefs, are all the messages propagated on the edges  $b \rightarrow j$  with  $j \in J$  and, for each such  $j$ ,  $b \in D_j$ .

Let us first consider a set of incoming warnings  $\{\vec{u}_{b \rightarrow j}\}$ . This warning set may or may not be "extensible" to a configuration  $(s_j)_{j \in J}$  satisfying all constraints  $(C_b)_{b \in \hat{D}}$ . One can easily go through a bureaucratic procedure to evaluate all configurations  $(s_j)_{j \in J}$  compatible with the warning set. First compute the cavity fields III.3 component-wise:  $h_{j \rightarrow a}^p = \max_{b \in D_j} (u_{b \rightarrow j}^p)$ . For each  $j \in J$ , the allowed values of  $s_j$  are those such that  $h_{j \rightarrow a}^{s_j} = 0$ . We denote by  $T(\{\vec{h}_{j \rightarrow a}\}) \subset \{1, \dots, q\}^{|J|}$  the set of allowed configurations of the  $s_j$  variables:

$$T(\{\vec{h}_{j \rightarrow a}\}) = \{(s_j)_{j \in I(a) \setminus i} \mid h_{j \rightarrow a}^{s_j} = 0, \forall j \in I(a) \setminus i\} \quad (6)$$

For each  $(s_j)$  in  $T(\{\vec{h}_{j \rightarrow a}\})$ , one can determine the output warning  $\vec{u}_{a \rightarrow i}$  using definition III.1.

This procedure can be embedded into the probabilistic description of solutions on the cavity graph. For doing so, we assume that incoming warnings are *independent*. Following the steps above, one first calculates from the incoming beliefs the distributions of cavity fields

$$H_{j \rightarrow a}(\vec{h}) = \sum_{\{\vec{u}_{b \rightarrow j}\}_{b \in D_j}} \delta_{\vec{h}, \vec{h}_{j \rightarrow a}} \prod_{b \in D_j} B_{b \rightarrow j}(\vec{u}_{b \rightarrow j}). \quad (7)$$

The new distribution of warnings  $\vec{u}_{a \rightarrow i}$  is now given by an average over cavity fields,

$$B_{a \rightarrow i}(\vec{u}) = \mathcal{Z}^{-1} \sum_{\{\vec{h}_{j \rightarrow a}\}_{j \in J}} \left[ \sum_{\{\vec{s} \in T(\{\vec{h}_{j \rightarrow a}\})\}} \delta_{\vec{u}, \vec{u}_{a \rightarrow i}(\vec{s})} \right] \prod_{j \in J} B_{j \rightarrow a}(\vec{h}_{j \rightarrow a}) \quad (8)$$

The prefactor  $\mathcal{Z}^{-1}$  is a normalization constant. Note that each cavity-field configuration  $\{\vec{h}_{j \rightarrow a}\}$  is contributing  $|T(\{\vec{h}_{j \rightarrow a}\})|$  terms. As a byproduct, contradictory messages automatically do not contribute anything to (8).

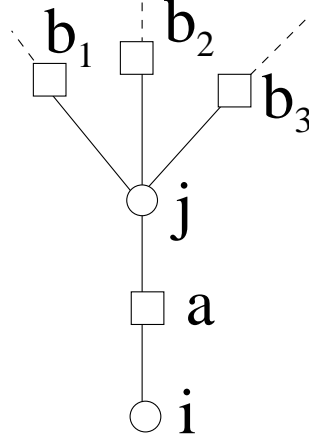


Figure 4: An example of the coloring problem. This part of the factor graph is the one necessary to compute the messages (warning and belief) passed from the function node  $a$  to the variable node  $i$ .

The BP equations (7,8) are equivalent to the so called sum-product (or belief network, or Bayesian network) equations [8, 9]. One can try to solve them by iteration, starting from some randomly chosen beliefs, and updating  $B_{a \rightarrow i}$  sequentially on randomly chosen  $a-i$  edges. In some cases, the iteration converges, independently of the scheme of updating, to a unique solution. When the belief propagation equations converge, one can use the obtained beliefs in order to estimate the histogram of local fields, using:

$$H_j(\vec{h}) \simeq \sum_{\{\vec{u}_{b \rightarrow j}\}_{b \in A(j)}} \delta_{\vec{h}, \vec{h}_{j \rightarrow a}} \prod_{b \in A(j)} B_{b \rightarrow j}(\vec{u}_{b \rightarrow j}) . \quad (9)$$

and this histogram can be used for decimation.

#### D. An example of Belief Propagation: 3-COL

For the sake of clarity, let us work out BP on a simple example of the 3-COL problem ( $q = 3$ ), see Fig. 4. Since function nodes are connected to two variable nodes only (constraints are edges in the original graph), there is only one variable node  $j$  above function node  $a$ . For a given configuration of incoming warnings  $\{\vec{u}_{b \rightarrow j}\}$ , we can make a table of allowed states  $s_j$ , and for each of them, we can compute the outgoing warning  $\vec{u}_{a \rightarrow i}(s_j)$ . Note that possible warnings are  $(1, 0, 0)$ ,  $(0, 1, 0)$ ,  $(0, 0, 1)$ , since a function node can only forbid one color (which is given by the state of the other variable connected to the function node).

- Suppose that  $\vec{u}_{b_1 \rightarrow j} = (1, 0, 0)$ ,  $\vec{u}_{b_2 \rightarrow j} = (0, 1, 0)$ , and  $\vec{u}_{b_3 \rightarrow j} = (0, 0, 1)$ . Then  $h_{j \rightarrow a} = (1, 1, 1)$ , we find a contradictory message. No satisfiable configuration exists for  $s_j$ . According to the procedure given above, this configuration does not contribute.
- Suppose that incoming messages are  $\vec{u}_{b_1 \rightarrow j} = \vec{u}_{b_2 \rightarrow j} = (1, 0, 0)$ , and  $\vec{u}_{b_3 \rightarrow j} = (0, 1, 0)$ . Then  $h_{j \rightarrow a} = (1, 1, 0)$ , and the only possible coloring state for  $j$  is  $s_j = 3$ . For this configuration, we thus have only one possible outgoing warning,  $\vec{u}_{a \rightarrow i} = (0, 0, 1)$ .
- If  $\vec{u}_{b_1 \rightarrow j} = \vec{u}_{b_2 \rightarrow j} = \vec{u}_{b_3 \rightarrow j} = (1, 0, 0)$ , then  $h_{j \rightarrow a} = (1, 0, 0)$ , and there are two possible colors for  $s_j$ , namely states 2 and 3. For the first one we have  $u_{a \rightarrow i} = (0, 1, 0)$ , and for the second one  $u_{a \rightarrow i} = (0, 0, 1)$ . Both contribute with equal weight to  $B_{a \rightarrow i}$ .
- All other configurations are simple color permutations of the three cases mentioned above, and are handled analogously.

From Eqs. (7,8), we can easily deduce the equation giving the probability distribution  $B_{a \rightarrow i}$  in terms of all distributions  $\{B_{b_l \rightarrow j}; l = 1, 2, 3\}$ . Parameterizing  $B_{a \rightarrow i}$  according to the three possible messages as

$$B_{a \rightarrow i}(\vec{u}) = \eta_{a \rightarrow i}^1 \delta_{\vec{u}, (1, 0, 0)} + \eta_{a \rightarrow i}^2 \delta_{\vec{u}, (0, 1, 0)} + \eta_{a \rightarrow i}^3 \delta_{\vec{u}, (0, 0, 1)} , \quad (10)$$

we find

$$\eta_{a \rightarrow i}^p = \frac{\prod_{l=1}^3 (1 - \eta_{b_l \rightarrow j}^p)}{\sum_{r=1}^q \prod_{l=1}^3 (1 - \eta_{b_l \rightarrow j}^r)} \quad (11)$$

This expression can be easily understood:  $\eta_{a \rightarrow i}^p$  equals the probability that color  $p$  is forbidden for node  $i$ , which means that node  $j$  has already taken this color,  $s_j = p$ . Now, node  $j$  can take color  $p$  if and only if it is not forbidden by any incoming warning: The numerator in Eq. (11) simply calculates the probability that none of the incoming messages forbids color  $p$ , the denominator guarantees normalization. Note that configurations in which all variables above  $j$  take the same color  $r$ , are counted twice, namely in the expressions for both  $p \neq r$ . According to the general discussion given above, this is correct because we have *two* new configurations for  $s_j$ , and two corresponding messages  $\vec{u}_{a \rightarrow i}$  can be sent.

Due to the symmetry between colors, a possible solution would be  $\eta_{a \rightarrow i}^p = 1/q$  for all edges  $(i, a) \in E$  and all colors  $p$ . Note, however, that the main intention is to use these equations in a recursive coloring algorithm. This means, some variable nodes may already be assigned a color before, which explicitly breaks the symmetry. Still, Eq. (11) is valid.

## IV. SURVEY PROPAGATION

### A. Clustering

Pitifully, the Belief Propagation dynamics is known not to converge for the random version of many combinatorial problems (including again 3-SAT and  $q$ -COL) in the region of the parameters near the SAT/UNSAT threshold. Recently, using tools from statistical physics, it has been possible to reach some understanding of what happens in the solution space of these problems around this threshold [1, 3, 10]. Well below the threshold, i.e. where the number  $|A|/n$  of constraints per variable is relatively small, a generic problem has exponentially many solutions, which tend to form one giant cluster: For any two solutions, it is possible to find a connecting path via other solutions that requires short steps only (each pair of consecutive assignments in the path is close together in Hamming distance).

Close to the critical threshold, however, the solution space breaks up into many smaller clusters. Solutions in separate clusters are generally far apart. In addition, the cost function  $C[\vec{x}]$  has exponentially many local minima, separated from each other by large cost “barriers”. These local cost minima are exponentially more numerous than the solution clusters, and they act as traps for local search algorithms.

According to the statistical-physics analysis (which considers the infinite size limit,  $n \rightarrow \infty$ ), there exist exponentially many widely separated clusters of solutions. Within one such cluster of solutions, we may identify two types of variables: those which are frozen in one single state, for all configurations belonging to the cluster, and those – unfrozen – which fluctuate from solution to solution inside the cluster. Note that also the variables which are frozen within one solution cluster may change their state when we go to another cluster, there they may even be unfrozen. While in general the above distinction can only provide an approximate description of clusters, it appears from numerical experiments that in many hard random CSP, like  $K$ -SAT or  $q$ -COL, such type of approximation is already rather accurate.

### B. The joker states

Survey propagation turns out [2] to be able to deal with this clustering phenomenon for large (finite) sizes  $n$ . Although the original derivation uses sophisticated statistical physics ideas, one can also develop it more directly in algorithmic terms. The main idea is that we do not work any more with individual solutions  $\vec{s} \in S_C$ , but with complete clusters of solutions. As already said, some variables may be frozen inside a cluster, so they retain one single value  $s_j \in \{1, \dots, q\}$  in our description. Other variables may take several values in the cluster. For handling these variables, one can introduce an additional *joker state* which we denote by a “ $\star$ ”. An even finer description, useful for general CSP, uses varieties of joker states, describing the set of values which are allowed for the variable, so that  $s_j \in \mathcal{P}$ , where  $\mathcal{P}$  is the ensemble built from all subsets of  $\{1, \dots, q\}$ . To each cluster one should associate exactly one generalized value of each variable. One can then generalize the constraint to this enlarged space and work out the corresponding belief propagation equations. The resulting equations are the survey propagation equations.

We shall not develop in more details this ‘derivation’, since it does not give any rigorous construction, but we will directly write the equations themselves, in terms of the original variables  $\vec{x} \in \{1, \dots, q\}^n$ , and then analyze them.

### C. Generalized messages

We first need to define the generalizations of the warnings, the cavity-fields and the local field used in survey propagation. In order to lighten the notation and presentation, we shall drop the 'generalized', and use the same notation for generalized warning as we used for warnings in the BP section. The reader should remember that in SP all the messages are 'generalized' messages.

For a given CSP, we define the generalized warnings:

**Definition IV.1** For a given edge  $a - i$  of the factor graph, with  $i \in I$  and  $a \in A(i)$ , let  $S$  be a given set of possible values for the variables  $(x_j)_{j \in I(a) \setminus i}$  which are 'above'  $a$ . We define the **warning** as the  $q$ -component vector  $\vec{u}_{a \rightarrow i}(\vec{x}) \in \{0, 1\}^q$  with components:

$$\hat{u}_{a \rightarrow i}^p(S) = \min_{(x_j) \in S} C_a [(x_j)_{j \in I(a) \setminus i} \mid x_i \leftarrow p] \quad , \quad p = 1, \dots, q,$$

(The generalized warning was called cavity-bias in [3]). Note that the set of possible warnings is enlarged in SP: For the example of 3-COL the *null message*  $(0, 0, 0)$  is added to  $(0, 0, 1)$ ,  $(0, 1, 0)$  and  $(1, 0, 0)$ . As we have discussed before, the non-null messages are sent if the node "above" a function node is assigned a fixed color in the solution cluster. Correspondingly, the new message is sent if this vertex is not fixed to a single color, i.e. if it is in the joker state.

Based on these warnings, we define local and cavity fields according to Def. III.3, with the argument (a single configuration) replaced by a set  $S$  of configurations:

$$\begin{aligned} h_i^p(S) &= \max_{a \in A(i)} u_{a \rightarrow i}^p(S) \quad , \\ h_{j \rightarrow a}^p(S) &= \max_{b \in A(j) \setminus a} u_{b \rightarrow j}^p(S) \quad . \end{aligned} \quad (12)$$

### D. Histograms

Histograms of warnings and fields are now defined as sums over clusters. The histogram of local fields is given by

$$H_i(\vec{h}) = \frac{1}{n_{cl}} \sum_{\alpha=1}^{n_{cl}} \delta_{\vec{h}, \vec{h}_i(S_C^\alpha)} \quad . \quad (13)$$

The histogram of the generalized warning on an edge  $a - i$  is now called the survey, denoted  $Q_{a \rightarrow i}(\vec{u})$ . It is defined in terms of the clusters of solutions for the cavity graph where  $i$  has been taken away. Calling  $S_C^{\alpha, (i)}$  the corresponding clusters, and  $n_{cl}^{(i)}$  their numbers, one defines:

$$Q_{a \rightarrow i}(\vec{u}) = \frac{1}{n_{cl}^{(i)}} \sum_{\alpha=1}^{n_{cl}^{(i)}} \delta_{\vec{u}, \vec{u}_{a \rightarrow i}(S_C^{\alpha, (i)})} \quad . \quad (14)$$

### E. Survey propagation equations

Based on these definitions, one can easily guess the generalized recurrence equations for the (approximate) probabilities  $Q_{a \rightarrow i}(\vec{u})$  that implement the solutions in this enlarged configuration space. These SP equations lead to a small, yet fundamental, modification of the BP equations. The basic assumption is again, that incoming warnings are independent. But contradictory messages have to be explicitly forbidden. Having Fig. 3 in mind, and using the same notations as in sect. III C, we use the incoming surveys (i.e. the set of surveys:  $\{Q_{b \rightarrow j}\}$  with  $j \in I(a) \setminus i$  and  $b \in A(j) \setminus a$ ) to calculate the cavity-field distributions as in (7):

$$H_{j \rightarrow a}(\vec{h}) = \sum_{\{\vec{u}_{b \rightarrow j}\}_{b \in D_j}} \delta_{\vec{h}, \vec{h}_{j \rightarrow a}} \prod_{b \in A(j) \setminus a} Q_{b \rightarrow j}(\vec{u}_{b \rightarrow j}) \quad . \quad (15)$$

Remember that these fields may lead to contradictions, if and only if  $\vec{h}_{j \rightarrow a} = (1, 1, \dots, 1)$  for at least one  $j$ . We therefore introduce the set of all non-contradictory cavity field configurations,

$$\mathcal{M}_{a \rightarrow i} = \left\{ \{\vec{h}_{j \rightarrow a}\}_{j \in I(a) \setminus i} \mid \forall j : \vec{h}_{j \rightarrow a} \in \{0, 1\}^q, \vec{h}_{j \rightarrow a} \neq (1, \dots, 1) \right\} \quad . \quad (16)$$



Then, for an element of  $\mathcal{M}_{a \rightarrow i}$  we define again

$$T(\{\vec{h}_{j \rightarrow a}\}) = \{(s_j)_{j \in I(a) \setminus i} \mid h_{j \rightarrow a}^{s_j} = 0, \forall j \in I(a) \setminus i\}, \quad (17)$$

the set of allowed configuration for the variable nodes above function node  $a$ . Now the difference to BP enters: All elements of  $T(\{\vec{h}_{j \rightarrow a}\})$  naturally belong to the same cluster, i.e. they give rise to a *single* outgoing warning! The new warning is thus computed on the set of allowed configurations, it is given by  $\vec{u}_{a \rightarrow i}(T(\{\vec{h}_{j \rightarrow a}\}))$ . Its distribution follows immediately,

$$Q_{a \rightarrow i}(\vec{u}) = \tilde{Z}^{-1} \sum_{\{\vec{h}_{j \rightarrow a}\} \in \mathcal{M}_{a \rightarrow i}} \delta_{\vec{u}, \vec{u}_{a \rightarrow i}(T(\{\vec{h}_{j \rightarrow a}\}))} \prod_{j \in I(a) \setminus i} H_{j \rightarrow a}(\vec{h}_{j \rightarrow a}). \quad (18)$$

The equations (15,18) are the SP equations. Note that Eq. (18) produces a dramatic change in the iteration of the probabilities with respect to the BP Eq. (8): Every allowed cavity-field configuration contributes only one term to the sum. Note also that contradictory messages have to be excluded *explicitly* by summing only over  $\mathcal{M}_{a \rightarrow i}$ . In belief propagation, for each configuration of input messages one takes the full collection of possible outputs, thereby introducing a bifurcation mechanism (which may easily become unstable). On the contrary, in SP the presence of multiple outputs is collapsed into the null message (which may even not be present in the belief propagation formalism as it happens for the coloring problem). A variable which receives a message having at least two zero components will be “unfrozen” in the corresponding cluster.

The SP equations (15-18) provide a closed set of equation for the surveys. Practically, this recurrence defines a map

$$\Lambda : \{Q_{a \rightarrow i}^{old}\}_{a \in A(i)}^{i=1, \dots, n} \mapsto \{Q_{a \rightarrow i}^{new}\}_{a \in A(i)}^{i=1, \dots, n} \quad (19)$$

and we are looking for a fixed point of this map, that will be obtained numerically by starting with some (random) initial  $\{Q_{a \rightarrow i}^0\}$  and applying  $\Lambda$  iteratively:

$$\{Q_{a \rightarrow i}^{SP}\} = \lim_{m \rightarrow \infty} \overbrace{\Lambda \circ \dots \circ \Lambda}^{m \text{ times}} \{Q_{a \rightarrow i}^0\} \quad (20)$$

Such a fixed point will be called a “self-consistent” set of surveys.

### F. An example of Survey Propagation: 3-COL

For the 3-COL example, because of the additional null message, the warning distribution now reads [11]

$$Q_{a \rightarrow i}^{SP}(\vec{u}) = \eta_{a \rightarrow i}^* \delta_{\vec{u}, (0,0,0)} + \eta_{a \rightarrow i}^1 \delta_{\vec{u}, (1,0,0)} + \eta_{a \rightarrow i}^2 \delta_{\vec{u}, (0,1,0)} + \eta_{a \rightarrow i}^3 \delta_{\vec{u}, (0,0,1)}, \quad (21)$$

and the SP equations corresponding to fig.4 are given by

$$\eta_{a \rightarrow i}^p = \frac{\prod_{l=1}^3 (1 - \eta_{b_l \rightarrow j}^p) - \sum_{r \neq p} \prod_{l=1}^3 (\eta_{b_l \rightarrow j}^* + \eta_{b_l \rightarrow j}^r) + \prod_{l=1}^3 \eta_{b_l \rightarrow j}^*}{\sum_{r=1}^3 \prod_{l=1}^3 (1 - \eta_{b_l \rightarrow j}^r) - \sum_{r=1}^3 \prod_{l=1}^3 (\eta_{b_l \rightarrow j}^* + \eta_{b_l \rightarrow j}^r) + \prod_{l=1}^3 \eta_{b_l \rightarrow j}^*}, \quad (22)$$

for  $p \in \{1, 2, 3\}$ . Then  $\eta_{a \rightarrow 1}^*$  can be computed by normalization, i.e:

$$\eta_{a \rightarrow i}^* = 1 - (\eta_{a \rightarrow i}^1 + \eta_{a \rightarrow i}^2 + \eta_{a \rightarrow i}^3) \quad (23)$$

The interpretation of this equation is again straightforward, for simplicity we explain it just for color 1: Now  $\eta_{a \rightarrow i}^1$  is given by the probability that  $s_j$  is forced to take value 1, i.e. by the probability that the cavity field equals  $\vec{h}_{j \rightarrow a} = (0, 1, 1)$ , conditioned to non-contradictory fields. The numerator calculates the unconditioned probability: The first term excludes  $h_{j \rightarrow a}^1$  which would forbid color 1 to  $s_j$ . The second term takes out fields  $(0, h^2, 0)$  and  $(0, 0, h^3)$  where  $s_j$  would be allowed to also take at least one other color. The last term adds again the probability of field  $(0, 0, 0)$  which was subtracted twice in the second term. The denominator realizes the conditioning to non-contradictory fields, i.e. it gives the probability that  $\vec{h}_{j \rightarrow a} \neq (1, 1, 1)$ . The counting of possible cases follows again the inclusion-exclusion principle: In the first term, we count fields that have a zero component in color  $r$ , summed over  $r$ . We have to subtract

the double countings due to fields having two zero-components, and we have to add once the field  $(0, 0, 0)$  which was added three times in the first term, and subtracted three times in the second one.

Note that the symmetry between colors leads immediately to a solution  $\eta_{a \rightarrow i}^* = 1$  for all edges  $(i, a)$  of the factor graph, i.e. only null-messages are sent. This is, however, not the correct solution in the clustered regime, the color symmetry is not valid at the level of solution clusters. In fact, the appearance of a non-trivial solution for the  $\eta_{a \rightarrow i}^p$  marks the onset of clustering.

### G. The $K$ -SAT case

In the SAT case,  $q = 2$ , so possible  $\vec{u}$  messages are  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$  and  $(1, 1)$ . As any clause can be satisfied by any given variable (by choosing its value according with the negation of its corresponding literal), the  $(1, 1)$  message will never show up. Moreover, for a given  $a \rightarrow i$ , which of  $(1, 0)$  or  $(0, 1)$  can appear on  $\vec{u}_{a \rightarrow i}$  will be completely determined by the sign of the corresponding literal. So we can parameterize distributions  $Q_{a \rightarrow i}$  with only one real number  $\eta_{a \rightarrow i}$  being the probability of the nontrivial  $\vec{u}_{a \rightarrow i}$  message  $((1, 0) \text{ or } (0, 1))$ . The probability of  $(0, 0)$  will simply be  $1 - \eta_{a \rightarrow i}$ . The corresponding equations – which have been written and implemented in [2, 12] – read in the case of 3-sat:

$$\eta_{a \rightarrow i} = \prod_{j \in I(a) \setminus i} \left[ \frac{\Pi_{j \rightarrow a}^u}{\Pi_{j \rightarrow a}^u + \Pi_{j \rightarrow a}^s + \Pi_{j \rightarrow a}^0} \right], \quad (24)$$

where:

$$\begin{aligned} \Pi_{j \rightarrow a}^u &= \left[ 1 - \prod_{b \in A_a^u(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in A_a^s(j)} (1 - \eta_{b \rightarrow j}) \\ \Pi_{j \rightarrow a}^s &= \left[ 1 - \prod_{b \in A_a^s(j)} (1 - \eta_{b \rightarrow j}) \right] \prod_{b \in A_a^u(j)} (1 - \eta_{b \rightarrow j}) \\ \Pi_{j \rightarrow a}^0 &= \prod_{b \in A(j) \setminus a} (1 - \eta_{b \rightarrow j}) \end{aligned} \quad (25)$$

$A_a^u(j), A_a^s(j)$  are the two sets in which  $A(j)$  is decomposed ( $A(j) = A_a^u(j) \cup A_a^s(j)$ ) where the indexes  $s$  (resp:  $u$ ) refer to the neighbors  $b$  for which the literals  $(b, j)$  and  $(a, j)$  agree (resp: disagree). This separation corresponds to the distinction of which neighbors contribute to make variable  $j$  satisfy or not-satisfy the clause  $a$ .

For example, the product  $\prod_{b \in A_a^s(j)} (1 - \eta_{b \rightarrow j})$  gives the probability that no nontrivial message arrives on  $j$  from the function nodes  $b \in A_a^s(j)$  (empty products are set to 1 by definition).

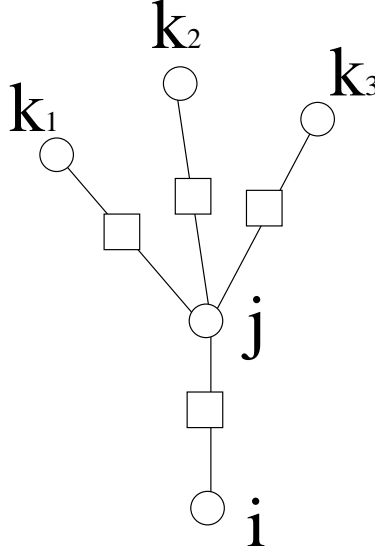
### H. The $q$ -COL case

We have already discussed in detail the 3-COL problem, a general number  $q$  of colors can be handled analogously. Messages  $\vec{u}_{a \rightarrow i}$  are elements of  $\{0, 1\}^q$  forbidding the colors which have 1 in their corresponding coordinate. We can immediately see that the possible types of  $\vec{u}_{a \rightarrow i}$  message are  $(0, \dots, 0, 1, 0, \dots, 0)$  (a 1 on the color taken by the variable in the other end of the link), plus the additional null message  $(0, \dots, 0)$  (if the neighboring variable is in the joker state). So we can parameterize  $Q_{a \rightarrow i}$  by only  $q$  real numbers.

Looking to Figure 5, suppose variable  $i$  has color  $p$  forbidden (i.e. the message  $\vec{u}_{a \rightarrow i}$  has a 1 on component  $p$ ). This implies that on this configuration, variable  $j$  is forced to take color  $p$ , that is  $\vec{h}_{j \rightarrow a}$  is of the form  $(1, \dots, 1, 0, 1, \dots, 1)$ , with a single 0 in the  $p$ -th position (“freezing” type). For all other  $h_{j \rightarrow a}$  the variable  $j$  is in the joker state and the output message will be  $(0, \dots, 0)$ .

### I. Decimation

Once the convergence is reached in Eq. (20) (we stop when  $\max_{a \in A, i \in I(a)} |Q_{a \rightarrow i}^{\text{old}} - Q_{a \rightarrow i}^{\text{new}}|$  becomes small enough), we can use the information computed so far to find a solution to the original problem [2, 3]. We can easily

Figure 5: Iteration for  $q$ -COL

compute (approximately) the local-field distributions  $\{H_i\}_{i \in I}$  introduced in Eq. (13) by considering all neighboring function nodes, and forbidding contradictory messages (remember that in the cavity graph we have deleted the constraints containing variable  $i$ , whereas in  $H_i$  we have to restrict the sum to messages being “extensible” to solutions of the complete problem):

$$H_i(\vec{h}) = \mathcal{Z}'^{-1} \sum_{\{\vec{u}_{a \rightarrow i}\}_{a \in A(i)}} \left(1 - \delta_{\vec{h}, (1, \dots, 1)}\right) \delta_{\vec{h}, \vec{h}_i} \prod_{a \in A(i)} Q_{a \rightarrow i}(\vec{u}_{a \rightarrow i}), \quad (26)$$

with  $\vec{h}_i$  determined according to Eq. (12).

The value  $H_i((1, \dots, 1, 0, 1, \dots, 1))$ , with a single zero entry at component  $p$ , gives now the probability of a variable  $i$  to be frozen to a certain value  $p$ . A simple decimation procedure can be implemented: Select the most frozen variable and fix it to its most frozen value, then simplify the problem: Certain constraints may already be satisfied independently of the values of other participating variables, and can be deleted from the problem instance. Other constraints may now immediately fix single variables to one state (unit-clause resolution). Reconverge the warning distributions on the smaller subproblem.

The decimation algorithm can have three types of behaviors:

1. The algorithm is able to solve the problem fixing all, or almost all variables (some variables may be still unfixed even if the problem is already solved).
2. The surveys converge at some stage to the trivial solution concentrated on null messages,  $Q_{a \rightarrow i}(u) = \delta_{h, (0, \dots, 0)}$  for all  $(i, a) \in E$ . In this case SP has nothing more to offer. Luckily, these problems are generally under-constrained and then easy to solve by other means. Note that, for  $q$ -COL, the trivial solution exists always. In numerical experiments, we found that in case of existence of another solution, the latter was the correct one. In this case it is therefore reasonable to restart the iteration of the SP equations starting from a new random initial condition, even if a trivial solution was found once. Only if no non-trivial solution can be found after several restarts, the subproblem is passed to a different solver.
3. The SP algorithm does not converge at some stage, even if the initial problem was satisfiable.

On large random instances of 3-SAT [1, 2, 3, 5] or  $q$ -COL [11], in the hard sat region, but not too close to the satisfiability threshold, numerical experiments show that the algorithm behaves as in case 2).

The generated subproblems turn out to be very simple to solve by other conventional heuristics, e.g. WALKSAT [13] or unmodified belief propagation.

Case 3) happens in general very close to the SAT/UNSAT transition. It is not yet clear if this problem appears due to the existence of finite loops in the original problem (which make the SP equations to be only approximate), due to the simple decimation heuristic which fixes always the most frozen variable, or due to some problems which go beyond the validity of the SP equation itself.

## V. WHAT'S NEXT

We would like to remark two possible directions of research, among all those that may follow from the presented algorithm. One is to formalize rigorously the notions suggested in Section IV, allowing for some well defined definitions of the clusters, and a corresponding derivation of the SP equations.

Another one, of big computational relevance, is to generalize SP, which was presented in its purest form, to deal with correlations between warnings that arise from local problem structures like small loops in the factor graph, cf. [7] for similar generalizations of BP. A second possible generalization would include diverse structures of the space of solutions, e.g. in a sense of clusters of solution clusters etc. In the language of statistical physics, this would include “more than one step of replica-symmetry breaking”.

After completing this work, we learned from G. Parisi that he has reached a similar conclusion on the interpretation of SP in the colouring problem through the addition of an extra state for the variables [14].

**Acknowledgment:** It is a pleasure to thank R. Mulet, A. Pagnani, and F. Ricci-Tersenghi for numerous discussions. MM and MW acknowledge the hospitality of the ICTP Trieste where a part of this work was done. This work has been supported in part through the EC ‘STIPCO’ network, grant No HPRN-CT-2002-00319.

- 
- [1] *Analytic and algorithmic solutions to random satisfiability problems*, M. Mézard, G. Parisi, and R. Zecchina, Science 297, 812 (2002) (Scienceexpress published on-line 27-June-2002; 10.1126/science.1073287).
  - [2] *Survey propagation: An algorithm for satisfiability*, A. Braunstein, M. Mezard, and R. Zecchina, preprint URL: <http://lanl.arXiv.org/cs.CC/0212002>.
  - [3] *Random K-satisfiability: from an analytic solution to a new efficient algorithm*, M. Mézard and R. Zecchina, Phys.Rev. E **66** 056126 (2002).
  - [4] *Coloring Random Graphs*, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **89**, 268701 (2002).
  - [5] *On the survey-propagation equations for the random K-satisfiability problem*, G. Parisi, preprint URL: <http://lanl.arXiv.org/cs.CC/0212009>.
  - [6] *Factor Graphs and the Sum-Product Algorithm*, F.R. Kschischang, B.J. Frey, H.-A. Loeliger, *IEEE Trans. Infor. Theory* **47**, 498 (2002).
  - [7] *Generalized Belief Propagation* J.S. Yedidia, W.T. Freeman and Y. Weiss, in *Advances in Neural Information Processing Systems 13* eds. T.K. Leen, T.G. Dietterich, and V. Tresp, pp. 689-695, (Cambridge, MA, MIT Press, 2001).
  - [8] *Low-Density Parity-Check Codes*, R. G. Gallager (Cambridge, MA, MIT Press, 1963).
  - [9] *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, 2nd ed. (San Francisco, Morgan Kaufmann, 1988).
  - [10] *The cavity method at zero temperature* M. Mézard and G. Parisi, J. Stat. Phys. **111**, 1 (2003).
  - [11] *Polynomial iterative algorithms for coloring and analyzing random graphs*, A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. E **68**, 036702 (2003).
  - [12] [www.ictp.trieste.it/~zecchina/SP](http://www.ictp.trieste.it/~zecchina/SP)
  - [13] *Local search strategies for satisfiability testing*, B. Selman, H. Kautz, B. Cohen, in: *Proceedings of DIMACS*, p. 661 (1993).
  - [14] G. Parisi, in preparation.