Scuola Internazionale Superiore di Studi Avanzati - Trieste

Survey Propagation

Thesis submitted for the degree of

Doctor Philosophiæ

Candidate
Alfredo Braunstein

Supervisor
Prof. Riccardo Zecchina

SISSA – Via Beirut 2-4 – 34014 TRIESTE – ITALY

Survey Propagation

Thesis submitted for the degree of

Doctor Philosophiæ

Candidate
Alfredo Braunstein

Supervisor
Prof. Riccardo Zecchina

Contents

Acknowledgements

I thank specially Riccardo. Besides being an extraordinarily generous person, he owns one of the most creative minds I've got to know. Working with him was simultaneously challenging and pleasant.

I'm indebted with all kind souls who patiently explained to me what I know about the fascinating world of statistical physics. If I know little, I blame no one but myself (and I certainly do!): Riccardo Zecchina, Marc Mézard, Federico Ricci-Tersenghi, Michele Leone.

I also thank Vincenzo Napolano, Matteo Marsili, Silvio Franz, Brian Hayes, Giorgio Parisi, Andrea Montanari, Roberto Mulet, Martin Weigt, Yannet Interian, Marco Pretti, Demian Battaglia, Joel Chavas, Andrei Agrachev, Dimitris Achlioptas for most useful discussions.

I thank Coni (of course).

I would like to thank all these fantastic persons that became dear friends while studying here in Italy: Die & Ire, Michele, Federico, Andrea, Martin, Vincenzo, Roberto, Demian, Joel, Yannet, Massimo, Eva.

I thank all friends who shared with me the discovery of mathematics back then: Gabi, Emi, Mara, Mari, Sam & Gabi, José.

This thesis was written using exclusively open-source software (gcc, LaTeX, LyX, gnuplot among others). Thanks to the many people that generously contributed to it.

And finally, I thank Fra. Without her, nothing of this would be possible. I couldn't even breathe (and it is hard to write a thesis without breathing).

CHAPTER 1

Introduction

The k-SAT problems is historically a fundamental one in computer science, as it was among the firsts to be proved to be NP-complete (for $k \geq 3$). It can be easily stated:

DEFINITION 1.0.1. Given $k, n \in \mathbb{N}$, A k-SAT formula is a boolean map $\mathcal{F} : \{\mathrm{T}, \mathrm{F}\}^n \mapsto \{\mathrm{T}, \mathrm{F}\}$ defined by

$$\mathcal{F}(x_1, \ldots, x_n) = \bigwedge_{a \in A} C_a$$

where $C_a = \bigvee_{r=1,\ldots,k} y_{a,r}$ (C_a is called a *clause*) and $y_{a,r}$ (called a *literal*) is either $x_{i_{a,r}}$ or $\neg x_{i_{a,r}}$ for some $i_{a,r} \in \{1, \ldots, n\}$. *$A$ is some finite index set.*

Given such an $\mathcal{F}$, the $k-SAT$ problem consists in finding $\mathbf{x} \in \{\mathrm{T}, \mathrm{F}\}^n$ such that $\mathcal{F}(\mathbf{x}) = \mathrm{T}$.

DEFINITION 1.0.2. Given $n, m \in \mathbb{N}$, the random $k-$SAT problem $\mathcal{R}(k, n, m)$ is the uniform probability space of all k-SAT formulas with m clauses over a fixed set of variables $x_1, \ldots, x_n$.

The subject of random combinatorial problems and in particular random k-SAT had a fascinating development in the last decade. Used as quick benchmarks against solving algorithms, a preliminar classification shows that random formulas with few clauses are easy to solve, whereas clauses with too many are easy to prove to be unsatisfiable. It has then been proved that in fact random $k-$SAT suffers a "phase transition" when the ratio $\alpha = m/n$ of clauses to variables crosses a critical value α_c below which almost all formulas are satisfiable, and above which almost all are unsatisfiable. The precise location of α_c is still uncertain (as its independence from n), but several rigorous bounds

have been found [**2**]. Despite all advances in the area, solving typical realizations of random formulas close to α_c has shown to be remarkably elusive over a decade; and the possibility that random k-SAT near α_c was a concrete case of equivalence between worst-case computational complexity hardness and average-case hardness was present (and possibly hoped, with views in applications to cryptography). In any case, the $k-$SAT problem has been certainly used as test-bed for solving algorithms.

It was in this context that the *Survey Propagation* (SP) algorithm for k-SAT has been proposed in [**55, 56**] and has successively been generalized to other constrain satisfaction problems [**18**] and shown to achieve amazing performances in solving the random k-SAT problem [**17**] and random $q-$coloring problem [**16**] (basically solving problems several orders of magnitude bigger than was possible with known algorithms in the "hardest" region of the parameters). For a kind, non-technical introduction to SP for q-coloring, see [**36**].

Unfortunately, the fundaments behind the SP equations (the *cavity method* of statistical physics) are hidden under a number of complex unproven statements about the infinite-size limit (called the *thermodynamic* limit) of the underlying random combinatorial problem. Although most results are already well understood in the statistical physics community, most of the base propositions remain to be proved and much worse, some important mathematical definitions remain to be posed! Generally, much of the underlying technique (prominently Parisi's "replica symmetry-breaking" method) is still in an early formalization stage. Remarkably, the method has shown to give predictions in extraordinary agreement with numerical simulations, and the performance of the SP algorithm itself can be seen as a strong (although partial) numerical evidence of its accuracy. We should mention moreover some recent rigorous proofs of the fact some of the quantities computed in the solution of the cavity method provide a bound to the correct equivalent quantities in some particular cases (See [**31, 35, 82**]). Unfortunately, these proofs seem to be not constructive enough to allow an useful mathematical interpretation of the SP algorithm.

The cavity method is shown in Chapter 3 for the analysis of the q-coloring problem (a similar derivation for the k-SAT problem was given in [**55**]). In Section 3.3 the derivation of the SP equations in their original formulation in the frame of this theory are presented (in a statistical physics language). A mathematical-oriented reader may look at Chapter 3 and Section 3.3 in particular as an inspiring heuristic explanation. The SP algorithm and equations are defined again in Chapter 4 under a slightly different view (independent from the original cavity method), but with less rich interpretation. Along with SP, Chapter 4 presents a well-known algorithm called *Belief Propagation* (BP), originally introduced in the context of statistical inference and used throughly nowadays in error-correcting codes and compression in the context of mathematical information theory. We will later need this definition in the following chapter.

Although the fundaments of the cavity equations in the RSB scheme are not very well understood in mathematical terms, the SP algorithms for $k-$SAT and q-coloring are however very concrete mathematical objects, and the purpose of this work was to analyze them in mathematical terms (in the hopes of getting also some new insight about the cavity equations). In Chapter 5 we present a result that connects SP with BP, showing precisely that the SP equations are BP equations for an associated combinatorial problem. This result is twofold: on one hand allows us to automatically inherit known properties of BP (prominently, this gives an alternative algorithm based on a variational method to compute the SP fixed points) and on the other hand, it gives a well-defined mathematical interpretation of the quantities computed by SP (Section 5.1), by means of analyzing the associated combinatorial problem. This allows a further, in our opinion stimulating, interpretation of the solutions of the latter in terms of "clusters", or groups of solutions of the original problem in Section 5.4.

CHAPTER 2

Random combinatorial problems

2.1. Three important combinatorial problems

2.1.1. k-SAT. In a notation that is more amenable to algebraic manipulations,

$$\mathcal{F} : \{-1,1\}^n \mapsto \{0,1\} \tag{2.1.1}$$

with $\mathcal{F}(\sigma) \stackrel{\text{def}}{=} \prod_{a\in A} C_a$ where $C_a \stackrel{\text{def}}{=} 1 - H_a$ for the local energy term defined as

$$H_a \stackrel{\text{def}}{=} \prod_{r=1,\ldots,k} \delta(-J_{a,r}, \sigma_{i_{a,r}}) \tag{2.1.2}$$

and $J_{a,r} \in \{-1,1\}$. The problem thus consists in finding $\sigma \in \{-1,1\}^n$ such that $\mathcal{F}(\sigma) = 1$. A generalization of this problem is called max-SAT, and is defined as finding the minimum of $H \stackrel{\text{def}}{=} \sum_{a\in a} H_a$. The SAT problem described above corresponds to finding a zero of H.

2.1.2. q-Coloring and restricted q-Coloring. The q-coloring problem is a generalization of a very old problem in cartography : the one of coloring countries in a map with a predefined palette of q colors in a way such that no adjacent countries share the same color. Given a geographic map, we can build the graph of adjacency for countries $G = (V, E)$ where the set of vertices V label countries and there is an edge (v, w) whenever two countries have a common border.

The problem can be easily posed in mathematical terms in generality: given finite undirected graph $G = (V, E \subset V \times V)$, find a vector $f \in \{1,\ldots q\}^V$ such that $f_v \neq f_w$ if $(v,w) \in E$. In other therms, the problem consists in finding $\mathbf{f} \in \{1,\ldots,q\}^V$ such that $\mathcal{F} \stackrel{\text{def}}{=} \prod_{(v,w)\in E} \left(1 - H_{(v,w)}\right) = 1$ for $H_{(v,w)} = \delta_{f_v f_w}$ or equivalently that $\sum_{(v,w)\in E} H_{(v,w)} = 0$.

The original "cartography" problem above simply corresponds to *planar* graphs G.

The associated minimization problem is to find $\mathbf{f} \in \{1,\dots,q\}^V$ which minimizes $H = \sum_{e\in E} H_e$.

Given a graph G and a set of "palettes" $\{X_v\}_{v\in V}$ with

$$X_v \subset \{1,\dots,q\},$$

the *restricted* q-Coloring problem is to find an $\mathbf{f}$ like in the plain coloring problem but with the additional restriction that $f_v \in X_v$. That is, we are allowed to further restrict the colors in every node to belong to some subset of $\{1,\dots,q\}$. The importance of this generalization is that the problem of coloring already partially colored graphs belong to restricted q-Coloring.

Note that for map coloring (i.e. q coloring of planar graphs), it has been shown that every map can be colored with 4 colors (and there is an algorithm that finds such a coloring in polynomial time) while there is no such equivalent for generic graphs.

Graph coloring has been proved to be NP-complete, which is the standard measure of worst-case computational "hardness". The following is a simple fact:

LEMMA 2.1.1. *Restricted q-Coloring can be easily (polynomially) reduced to normal q-Coloring*

PROOF. Take a graph $G = (V, E)$ and a palette $\{X_v\}$ for the restricted q-Coloring problem. Build the graph $\tilde{G} = \left(\tilde{V}, \tilde{E}\right)$ as follows: $\tilde{V} = V \uplus \{v_1,\dots,v_q\}$ where the newly introduced vertices $v_1,\dots,v_q$ are labeled by the colors $\{1,\dots,q\}$. Take $\tilde{E} = E \uplus \{(v, v_r) : r \notin X_v\}$. □

2.1.3. Linear systems over finite fields. Given a finite field $F = \mathbb{GF}\left(p^k\right)$, a matrix $\mathbf{B} \in F^{m\times n}$ and a vector $\mathbf{c} \in F^m$, the problem of finding a solution of the equation

$$\mathbf{B}x = \mathbf{c} \tag{2.1.3}$$

can be solved in polynomial time by standard methods, like Gaussian elimination in time $O\left(n.m^2\right)$. Of particular interest is the case

of $F = \mathbb{GF}(2) = \mathbb{Z}_2$ the field of two elements $0, 1$. In this case, the problem will be called XOR-SAT. The sum operation on F (XOR of boolean variables) will be denoted by $\oplus$. Similarly to subsection 2.1.1, it can be posed also multiplicatively as: find $\mathbf{x} \in \{-1, 1\}^I$ satisfying $\mathcal{F} \stackrel{\text{def}}{=} \prod_{a \in A} \frac{1}{2}(1 - H_a) = 1$ where $H_a \stackrel{\text{def}}{=} J_a \prod_{j=1}^{k_a} x_{i_{a,j}}$ and $J_a \in \{-1, 1\}$ codes for the $\mathbf{c}$ vector in th e RHS of Eq. (2.1.3). When $k_a = k$ is a constant, the problem is called $k-$XOR-SAT. As usual, the associated minimization problem is to find the minimum of $\sum_a H_a$ and is called min-XOR-SAT. The min-XOR-SAT problem has been proved to be NP-hard.

If the matrix A is *sparse* (bounded number of non-zeros per row) like in the case of constant or bounded k_a above, there are methods guaranteed to find a solution in time $O(n^2)$ like the Lanczos algorithm and variants.

2.2. Computational complexity

Although not directly related to the present work, we cannot avoid making a quick note about computational complexity. Given a combinatorial (or algorithmic) problem (like graph coloring) and a specific algorithm to solve it, in many cases it is possible to compute the time the algorithm requires at worst to solve an instance of the problem, and thus giving an upper-bound to the "difficulty" of the problem. If the problem can be seen as a sequence of increasing size (as for instance finding a 4-coloring for a planar graph of size n), this can give a bound for the asymptotic behaviour: one can say that that particular problem sequence is "easier than" $O(n^2)$ for instance, meaning that there is an algorithm that will take at most time $O(n^2)$ to solve it. One typical such assertion is that a given problem is *time-polynomial,* meaning that there is a fixed polynomial $p(n)$ and an algorithm such that the problem will be solved by the problem in a time bounded by $p(n)$. The class of all time-polynomial algorithms is called P.

Proving *lower* bounds to the time *any* algorithm would need to solve a problem in the other hand (i.e. *"harder than"* bounds*) has* proved to be surprisingly difficult in general. Given two problems A and B,

such that every instance of A can be reduced (sufficiently easily) to an instance of B, one can say that B is *harder than* A, as the ability to solve every instance of B implies ability to solve every instance of A. If the reduction considered is *polynomial* in time, and B is time-polynomial, automatically so is A. So this *is* a lower bound to the complexity of B, but it is not very satisfactory as it is relative to the complexity of another problem A.

A lot of the most interesting computational problems fall in the NP class category. The NP class is, shortly, the class of problems such that there is a time-polynomial algorithm that, given an instance of the problem (e.g. a given graph) and a proposed solution (i.e. a proposed assignment of colors), can check that the proposed solution is indeed a correct solution (i.e. a good coloring). This can be viewed again as an *upper* bound to the hardness.

Complexity classes have been introduced in part to overcome the short supply of lower bounds. In 1971 S. Cook proved that the boolean satisfiability problem (SAT) is complete for the NP class. This means that every problem in the NP class can be (polynomially) reduced to SAT, or in other terms, that SAT is "harder or equal than" all other problems in NP. This can be indeed viewed as a *lower* complexity bound for the SAT problem, and in fact this is the most used strategy to declare a problem *hard:* to prove that it is complete for NP. The class of all problems complete for NP is called NPC. To prove that some problem is NP-Complete is normally not very difficult: it suffices 1) to prove that it is in NP and 2) reduce some known problem in NPC to it. Currently there is no proof at all that NPC problems are not polynomial (this is the celebrated $P \neq NP$ conjecture) but there is the strong belief that this is so. Both k-SAT and q-coloring have been proved to be NPC.

This notion of complexity hardness is clearly related to *worst case*: time for an algorithm is defined as a maximum time among all instances of the problem: for this reason this classification is often also called *worst-case* complexity.

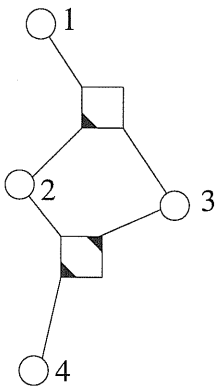

FIGURE 2.3.1. The factor graph of a 3-SAT problem corresponding to the formula $(x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_2 \vee \overline{x}_3 \vee \overline{x}_4)$. Variables are represented as circles, clauses (i.e. function nodes) as squares. A triangle-shaped decoration can indicates here that the corresponding literal is negated

Even if there is no polynomial $p(n)$ algorithm to solve all coloring instances of size n, there could be an algorithm that takes an *average* time $p(n)$ (averaged over all instances of a given size with uniform probability for instance), and this is the idea behind *average-case* complexity. Of course, a problem that is *worst-case* easy will be *average-case* easy as well, but the converse is often not true. Even if these two methods of measuring complexity are not equivalent at all, it has been often observed experimentally that taking some NPC problems (like k-SAT) and restricting it to some specific sub-ensemble (like $k-$SAT with n variables and $4.2n$ clauses), the resulting problem was very difficult to solve even in average. It is not clear (but it could be) that the study of this difficulty could prove to be useful to gain some insight about worst-case complexity as well.

2.3. An unifying formalism: factor graphs.

Al three combinatorial problems shown above in their simplest form correspond to finding an $\mathbf{x} \in X$ such that $\mathcal{F} = \prod_{a \in A} Q_a(\mathbf{x})$ is nonzero for some functions $Q_a : X \mapsto \{0,1\}$ taking each a small number of arguments. Of course, this expression for $\mathcal{F}$ is far too general to say

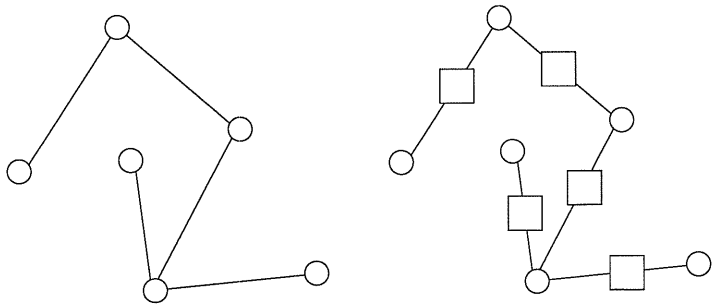

FIGURE 2.3.2. The original graph (left) and its factor graph (right) corresponding to a coloring problem

something useful about the set $\mathcal{F} = 1$, it is clearly important to know how is the topology of variable dependences goes. As it turns out, a lot can be said about the structure of $\mathcal{F} = 1$ by knowing it, at least for simple enough topologies. As a trivial example, suppose that every Q_a depends only on one variable x_i; then the set $\mathcal{F} \neq 0$ can be completely described in simple ways.

To formalize the above concepts, we will make some definitions:

For $\mathcal{F} : X \mapsto \mathbb{R}_{\geq 0}$ in the form $\mathcal{F} = \prod_{a \in A} Q_a(\mathbf{x})$, its *factor graph* is a bipartite undirected graph, with "variable" type nodes $i \in I$ and "function" type nodes $a \in A$. When possible, we will use indices a, b, c for elements of A and i, j, k for elements of I.

Bipartiteness will mean of course that edges are only allowed between nodes of the two different types. In some important cases, $Q_a(\mathbf{x}) \in \{0, 1\}$.

The space of *configurations* has been denoted by X. Every variable x_i has a finite range $x_i \in X_i$, so $X = \oplus_{i \in I} X_i$. In many important cases all X_i are equal, and $X = Y^I$.

Function nodes a neighbors to i will be denoted by the symbol $a \in i$ and variable nodes i neighbors to $a \in A$ will be denoted by $i \in a$. The symbol $i \in a \setminus j$ will mean all indices i which are neighbors of a, except index j. Symmetrically, $b \in i \setminus a$ will mean all function nodes b neighbors to variable node i except function node a. We will denote by $n_i = |\{a \in i\}|$ and $n_a = |\{i \in a\}|$ the number of neighbors (or degree) of the corresponding node.

We have seen that in many cases $\mathcal{F}$ is just a characteristic function, taking values in $\{0,1\}$ and the problem consists in finding some preimage of 1, or better some kind of description of the set $\mathcal{F}^{-1}(1)$. By means of a simple normalization term we can reinterpret this $\mathcal{F}$ as a probability measure $P = Z^{-1}\prod_{a\in A} Q_a(\mathbf{x})$ over the set X (here $Z = \sum_{\mathbf{x}\in\mathbf{X}}\prod_{a\in A} Q_a(\mathbf{x})$). Then the problem becomes to characterize this probability measure P.

In the general case we are given a probability measure in the functional form $P = Z^{-1}\prod_{a\in A} Q_a(\mathbf{x})$ where now just $Q_a : X \mapsto \mathbb{R}_{\geq 0}$ and we want to describe in some way this probability space.

2.4. Gibbs measure and Free Energy

As a reward for the generalization to $Q_a : X \mapsto \mathbb{R}_{\geq 0}$ instead of just $Q_a : X \mapsto \{0,1\}$, minimization problems can be somewhat incorporated now in the same framework: by introducing an artificial parameter $\beta \geq 0$ and defining $P_\beta = \frac{1}{Z_\beta}e^{-\beta H}$ for $H = \sum_{a\in A} H_a$ and Z_β being as usual the appropriate normalization scalar constant, called *partition function* and defined by

$$Z_\beta \stackrel{\text{def}}{=} \sum_{\mathbf{x}\in X} e^{-\beta H(\mathbf{x})}$$

We can easily see that $P_\infty \stackrel{\text{def}}{=} \lim_{\beta\to+\infty} P_\beta$ concentrates exactly on the minimum values of H; that is, by calling

$$\mathcal{S} \stackrel{\text{def}}{=} \left\{\mathbf{x} : H(\mathbf{x}) = \min_{\mathbf{y}\in\mathbf{X}} H(\mathbf{y})\right\}$$

we have that

$$P_\infty = \frac{1}{|\mathcal{S}|}\chi_{\mathcal{S}}$$

The inverse of the parameter β is called temperature T. The quantity $\frac{1}{\beta}\log Z_\beta$ is called the "free energy" and is of fundamental importance in statistical physics. Computing the scalar function $\beta \mapsto \ln Z_\beta$ often warrants knowing all sort of interesting statistical properties about P_β.

As an example, the average energy of a configuration at a given β

$$\langle H \rangle_\beta \stackrel{\text{def}}{=} \sum_{\mathbf{x} \in X} P_\beta(\mathbf{x}) H(\mathbf{x}) \tag{2.4.1}$$

can be computed simply as $\langle H \rangle_\beta = -\frac{\partial}{\partial \beta} \log Z_\beta$.

There are other uses of introducing an artificial parameter β apart from directly solving the minimization of E. As an interesting example, suppose that we are given $\mathcal{F} = \prod_{a \in A} Q_a(\mathbf{x})$ for $Q_a(\mathbf{x}) \in \{0,1\}$. Take some configuration $\mathbf{y} \in \{0,1\}$. Now for the probability measure $P_\lambda = Z_\lambda^{-1} \prod_{a \in A} Q_a(\mathbf{x}) \prod_{i \in I} e^{-\lambda \delta_{x_i y_i}}$, it is easy to check that $\lim_{\lambda \to +\infty} P_\lambda$ concentrates on the solution(s) of $\mathcal{F} = 1$ which differ in the smallest number of coordinates from $\mathbf{y}$.

2.5. Entropy

Given a probability measure $P(x)$ over a finite space X, the quantity

$$S \stackrel{\text{def}}{=} -\sum_{\mathbf{x} \in X} P(\mathbf{x}) \ln P(\mathbf{x}) \tag{2.5.1}$$

is called the entropy of P ($x \ln x$ is assumed to evaluate to 0 at $x = 0$). S is a non-negative quantity as all terms in the sum are negative or zero, and achieves a maximum value of $\log |X|$ for the uniform measure $P = \frac{1}{|X|}$. The entropy of P is 0 if only and only if P is a delta measure, $P(\mathbf{x}) = \delta(\mathbf{x}, \mathbf{y})$. In general, the entropy is a measure of how "broad" is the measure P. The following important special case may help to understand this concept.

LEMMA 2.5.1. *If $P(x)$ is uniformly concentrated on a subset N as e.g. in the three combinatorial problems of Section 2.1, then $S = \ln |N|$.*

PROOF. The hypothesis means that $P(\mathbf{x}) = \frac{1}{|N|} \chi_N(\mathbf{x})$. A direct evaluation of 2.5.1 gives the result. □

2.6. Tree factor graphs

Suppose we have a factor graph G which is singly connected, then each connected component is a tree. We will assume the graph has

only one connected component (and so is a tree), but all results here generalize trivially to singly connected graphs.

Choosing one variable node as the root, we can make the following definitions. The *depth* of a given node is its distance to the root and will be denoted by $d(\cdot)$. A node $a \in A$ (resp: $i \in I$) is said to be a child of $i \in I$ (resp: $a \in A$) if $i \in a$ and $d(a) > d(i)$ (resp: $d(a) < d(i)$). The generated partial relations will be denoted by $\geq$ and $\leq$: for two nodes v, w (type unspecified) then $v \leq w$ if every path in G from v to the root node passes through w. We will use also the relation $v < w$ if and only if $v \leq w$ and $v \neq w$. The v-subtree will denote the subgraph of nodes w such that $w < v$.

We will introduce the following notation: We will call $\mathbf{x}_a$ the tuple of variables $\{x_i\}_{i \in a}$for some function node a. By a slight abuse (which will prove to shorten the notation substantially) we will then call

$$P(\mathbf{x}_a) \stackrel{\text{def}}{=} \sum_{\{x_i\}_{i \notin a}} P(\mathbf{x})$$

and

$$P(x_i) \stackrel{\text{def}}{=} \sum_{\{x_j\}_{j \neq i}} P(\mathbf{x})$$

the *marginals* of $P(\mathbf{x})$. Given P it is very easy to obtain its marginals (though as an exponential sum), but in general the converse is much harder (or impossible). In the particular case of tree factor graphs, however, the following holds.

THEOREM 2.6.1. [74] *If the factor graph for the probability measure* $P = \frac{1}{z} \prod_{a \in A} Q_a$ *is a tree, then*

$$P(\mathbf{x}) = \prod_{a \in A} P(\mathbf{x}_a) \prod_{i \in I} P(x_i)^{1-n_i}$$

PROOF. The RHS is assumed to evaluate to 0 when $P(x_i) = 0$ (and then $P(\mathbf{x}_a) = 0$). For $\mathbf{x}$ such that $P(x_i) = 0$, the identity is immediate, because $P(\mathbf{x}) = 0$. For all other $\mathbf{x}$, it can be recovered by using $\tilde{Q}_a^\epsilon = Q_a + \epsilon$: as P is assumed to be a probability measure, there exists $\mathbf{x}$ such that $\prod_{a \in A} Q_a(\mathbf{x}) \neq 0$, and so $Z_\epsilon \not\to 0$, clearly then $\lim_{\epsilon \to 0} P^\epsilon(\mathbf{x}) = P(\mathbf{x})$ and in consequence $\lim_{\epsilon \to 0} P^\epsilon(\mathbf{x}_a) = P(\mathbf{x}_a)$ and

$\lim_{\epsilon \to 0} P^{\epsilon}(x_i) = P(x_i)$, so we can assume without losing generality that $Q_a > 0$.

We will prove by induction on the depth of the i-subtree that

$$P(\mathbf{x}|x_i) = \prod_{a \leq i} P(\mathbf{x}_a) \prod_{j \leq i} P(x_j)^{1-n_j} P(x_i)^{-1} \tag{2.6.1}$$

The initial step of the induction is trivial. Now take an arbitrary tree with root variable node r. Call T_i the subtree with root at i and $\mathbf{y} = \{x_i\}_{i \in a \in r}$. Then

$$\begin{aligned} P(\mathbf{x}|x_r) &= P(\mathbf{x}|\mathbf{y}) P(\mathbf{y}|x_r) \\ &= \prod_{i \in a \setminus r, a \in r} P(T_i|\mathbf{y}) P(\mathbf{y}|x_r) \\ &= \prod_{a \notin r} P(\mathbf{x}_a) \prod_{i \neq r} P(x_i)^{1-n_i} \prod_{i \in a} P(x_i)^{-1} P(\mathbf{y}|x_r) \end{aligned}$$

Where the last passage is due to the inductive hypothesis. Now Eq. (2.6.1) holds if one notes that

$$\begin{aligned} P(\mathbf{y}|x_r) &= \prod_{a \in r} P(\mathbf{x}_a|x_r) \\ &= \prod_{a \in r} P(\mathbf{x}_a) P(x_r)^{-1} \end{aligned}$$

Multiplying both sides of Eq. (2.6.1) by $P(x_i)$ and setting i to the root of the tree we get Theorem (2.6.1). □

This result allows us to have also an expression of the entropy in term of marginals:

THEOREM 2.6.2. *If the factor graph for the probability measure* $P = \frac{1}{z} \prod_{a \in A} Q_a$ *is a tree, then its entropy* $S = -\sum_{\mathbf{x} \in X} P(\mathbf{x}) \ln P(\mathbf{x})$ *can be computed from the marginals of* P *as*

$$S = -\sum_{a \in A} \sum_{\mathbf{x}_a} P(\mathbf{x}_a) \ln P(\mathbf{x}_a) + \sum_{i \in I} \sum_{x_i} (n_i - 1) P(x_i) \ln P(x_i) \tag{2.6.2}$$

PROOF. The proof is easy using Theorem 2.6.1.

$$
\begin{aligned}
S &= \sum_{\mathbf{x}\in X} P(\mathbf{x}) \ln \left(\prod_{a\in A} P(\mathbf{x}_a) \prod_{i\in I} P(x_i)^{1-n_i} \right) \\
&= \sum_{\mathbf{x}\in X} P(\mathbf{x}) \sum_{a\in A} \ln P(\mathbf{x}_a) + \sum_{\mathbf{x}\in X} P(\mathbf{x}) \sum_{i\in I} (1-n_i) \ln P(x_i) \\
&= \sum_{a\in A} \sum_{\mathbf{x}\in X} P(\mathbf{x}) \ln P(\mathbf{x}_a) + \sum_{i\in I} \sum_{\mathbf{x}\in X} P(\mathbf{x}) (1-n_i) \ln P(x_i) \\
&= \sum_{a\in A} \sum_{\mathbf{x_a}} \sum_{\{x_i\}_{i\notin a}} P(\mathbf{x}) \ln P(\mathbf{x}_a) + \\
&\quad + \sum_{i\in I} \sum_{x_i} \sum_{\{x_j\}_{j\neq i}} P(\mathbf{x}) (1-n_i) \ln P(x_i) \\
&= \sum_{a\in A} \sum_{\mathbf{x_a}} P(\mathbf{x}_a) \ln P(\mathbf{x}_a) + \sum_{i\in I} (1-n_i) \sum_{x_i} P(x_i) \ln P(x_i)
\end{aligned}
$$

□

This expression of the entropy is much more satisfactory than the original one, as it has typically a linear number of terms instead of an exponential one. There is a similar expression (similar easy proof but without using Theorem. 2.6.1) for the average energy, this one even valid in general graphs:

PROPOSITION 2.6.3. *For the probability measure* $P_\beta = \frac{1}{z} e^{-\beta \sum_a H_a}$, *the average energy* $\langle H \rangle_\beta = \sum_{\mathbf{x}} P_\beta(\mathbf{x}) H(\mathbf{x})$ *is given by*

$$\langle H \rangle_\beta = \sum_a \sum_{\mathbf{x}_a} P_\beta(\mathbf{x}_a) H(\mathbf{x}_a)$$

2.7. Random combinatorial ensemble and typical properties

2.7.1. 0-1 Laws and threshold values. The topic of Random Graph theory was founded by Erdös and Rényi in a famous paper [29] in 1960. They introduced two models for generating undirected random graphs over a vertex set I, with $|I| = n$: $G(n,m)$ and $G(n,p)$.

- In graph G in $G(n,p)$ each edge (among the $\frac{n(n-1)}{2}$ possible ones) belongs to G with probability p

- A graph in $G(n,m)$ is chosen uniformly between all graphs with exactly m edges

Both are probability spaces over the set of graphs with vertex set I. As there are $N = \binom{n}{2} = \frac{n(n-1)}{2}$ possible edges, this set has $2^{\frac{n(n-1)}{2}}$ elements.

For a graph $G = (I, E)$, in the first case, the probability measure explicitly reads

$$P_{G(n,p)}(G) \stackrel{\text{def}}{=} p^{|E|}(1-p)^{N-|E|}$$

while in the second case

$$P_{G(n,m)}(G) \stackrel{\text{def}}{=} \frac{1}{\binom{N}{m}}\delta(|E| - m)$$

It turns out that for $p \sim \frac{m}{N}$ the spaces $G(n,m)$ and $G(n,p)$ are largely equivalent in many aspects, and typical properties of one space often hold automatically also for the other [**2**]. In their work, Erdös and Rényi proved also the following

THEOREM 2.7.1. [the *giant component*] *for a graph in* $G\left(n, p = \frac{c}{n}\right)$ *and any* $\epsilon > 0$,

- if $c = 1 - \epsilon$ then *almost surely* all connected components have at most one cycle and $O(\log n)$ vertices. Moreover, the number of components with cycles is $o(n)$.
- if $c = 1 + \epsilon$ then *almost surely* there is a unique connected component with many cycles and $\Omega(n)$ vertices

Where the term "almost surely" means with a probability that tends to one as $n \to \infty$.

This was the first "threshold phenomenon" result in random graphs. The structural change in a typical graph $G(n,p)$ when p crosses the value $\frac{1}{n}$ is called also "phase transition" because its similitude to the phase transition of the chilling water at $T = 0°C$: a change that microscopically looks (and is) smooth generating a macroscopic "discontinuous" global effect. Since then, this fascinating subject has received a lot of attention.

We have seen another property of graphs in Section (2.1.2), namely $q-$colorability. A graph G is is said to be $q-$colorable if there exist a $q-$coloring for G. The following theorem has been proved in [6]

THEOREM 2.7.2. *Define for $k \in \mathbb{N}$ and $q > 2$ the number*

$$g(n,q,c) \stackrel{\text{def}}{=} P_{G\left(n,\frac{c}{n}\right)}(G \text{ is } q-\text{colorable}),$$

then there exists a number $c(n,q)$ such that for any $\epsilon > 0$

(1) $\lim_{n\to\infty} g(n,q,c(n,q)-\epsilon) = 1$
(2) $\lim_{n\to\infty} g(n,q,c(n,q)+\epsilon) = 0$

Similarly for k-SAT, the following holds.

THEOREM 2.7.3. [32] *Let's denote by*

$$h(n,k,\alpha) \stackrel{\text{def}}{=} P(\text{a formula in } \mathcal{R}(k,n,m=\alpha n) \text{ is satisfiable})$$

then for every $k \geq 2$ there exists $\alpha(n,k)$ such that for any $\epsilon > 0$

- $\lim_{n\to\infty} h(n,k,\alpha(n,k)-\epsilon) = 1$
- $\lim_{n\to\infty} h(n,k,\alpha(n,k)+\epsilon) = 0$

Note that the existence of both $\lim_{n\to\infty} c(n,q)$ and $\lim_{n\to\infty} \alpha(n,k)$ for $q,k > 2$ is still unsettled, but largely believed to be true. The first of the two limits was an early conjecture by Erdös.

We will consider also the problem of finding a critical value for the k-XOR-SAT problem. The definition of the random ensemble is similar to the k-SAT one: Given a set of n $\mathbb{Z}_2$-variable indices, we consider all linear systems $B\mathbf{x} = \mathbf{c}$ with exactly $m = \gamma n$ equations over the n variables, such that each equation has exactly k terms, with uniform probability. This probability will be denoted by $P_{k,\gamma}$. In this model, the distribution of B and $\mathbf{c}$ are independent, the latter is uniform among the 2^m possible binary vectors. The probability of such a problem will then factorize as $P_{k,\gamma}(B,\mathbf{c}) = P_{k,\gamma}(B)\,2^{-m}$.

We are interested in behavior changes in the limit $n \to \infty$. In particular, we are interested in finding a critical γ_c such that

- $P_{k,\gamma}(B\mathbf{x} = \mathbf{c} \text{ is satisfiable}) \to 1$ for $\gamma < \gamma_c$
- $P_{k,\gamma}(B\mathbf{x} = \mathbf{c} \text{ is unsatisfiable}) \to 0$ for $\gamma > \gamma_c$

2.7.2. Quenched free energy. We have seen that for an energy function H, it can be useful to compute the free energy $\beta^{-1}\log Z_\beta$ where $Z_\beta = \sum_{\mathbf{x}} e^{-\beta H(x)}$ to compute certain useful properties of the probability measure (we have seen the example of the average energy).

If we are interested in computing typical properties of a random combinatorial problem, as for instance

$$P_{G\left(n,\frac{c}{n}\right)}\left(G \text{ is } k-\text{colorable}\right)$$

or

$$P\left(\text{a formula in } \mathcal{R}\left(k,n,m=\alpha n\right) \text{ is satisfiable}\right)$$

computing the following quantity can be of use:

$$F\left(\beta,n\right) \stackrel{\text{def}}{=} \frac{1}{\beta}\overline{\log Z_\beta}$$

Where the overline denotes average over the random ensemble. The quantity above is the averaged or "*quenched*" free energy. As in the non-averaged case, knowing the function $\beta \mapsto F\left(\beta,n\right)$ allows to compute all sorts of interesting quantities. For instance the β derivative equals to $-\frac{\partial}{\partial\beta}\overline{\log Z_\beta} = \overline{\langle H\rangle}$.

Remembering that $P_\beta\left(\mathbf{x}\right) = Z_\beta^{-1}e^{-\beta H(\mathbf{x})}$ and writing the expression of the entropy S_β we get

$$\begin{aligned} S_\beta &= -\sum_{\mathbf{x}} P_\beta\left(\mathbf{x}\right)\log P_\beta\left(\mathbf{x}\right) \\ &= -\sum_{\mathbf{x}} P_\beta\left(\mathbf{x}\right)\left(-\beta H\left(\mathbf{x}\right) - \log Z_\beta\right) \\ &= \beta\left\langle H\right\rangle_\beta + \log Z_\beta \end{aligned}$$

Remembering that $\left\langle H\right\rangle_\beta = \frac{\partial}{\partial\beta}\log Z_\beta$ we get the simpler expression $S_\beta = \frac{\partial}{\partial\beta}\beta\log Z_\beta$ of the entropy as a function of the free energy, and thus an expression of the average entropy as a function of the quenched free energy F.

Except from very special cases, it is much easier to compute $\log\overline{Z_\beta}$ than $\overline{\log Z_\beta}$, but the former generally typically doesn't convey as much information as the latter. Note that $\lim_{\beta\to\infty} Z_\beta = \left|\{\mathbf{x} : H\left(\mathbf{x}\right) = 0\}\right|$.

2.8. Why random problems

Shannon's 1948 paper [**79**] on the limits of reliable transmission over unreliable media, opened the field of information theory. That paper formalized the concept of information and established theoretical bounds for the maximum rate of reliable information transmission for a given *channel.* A transmission channel is simply a stochastic device that models a possibly noisy real communication channel. Examples can be a communication protocol between two endpoints like in the internet, or a defective storage device, like a hard drive. To fix ideas we mention one of the most simple possible channel models, the *binary symmetric channel* (BSC). Given a real number $0 \leq p \leq 1$, in the BSC with "error probability" p, every bit is transmitted correctly with probability $1 - p$, and inverted with probability p.

A (fixed length) *coding scheme* consists in the selection and agreement between the sender and receiver of an integer number (the *codelength)* and a subset of *codeword* vectors over the communication alphabet (that we may assume is the set $0, 1$) of size given by the codelength. Along with this codeword set there is a map that translates from source strings to be transmitted to codewords and back. The *rate* of transmission is the fraction $\log_2 (\#codewords) / codelength$: it is an indication of how much transmission are we "wasting" to ensure reliability with respect to just sending the source over the channel. Using a codeword set equal to the set of all possible words of codelength size gives a rate of 1. Only codewords are sent to the channel, so if they become corrupted in the transmission, the receiver can often know that an error has occurred just by checking that what was received is not a codeword. Error recovery (*decoding*) can be achieved by finding the nearest codeword to the received string (this is the *maximum likelihood* decoder).

Shannon proved that the channel can be characterized by a number, the *capacity,* such that reliable communication is possible for rates arbitrary close (but below) and impossible for rates above. Achieving

the communication at rates close to the capacity however, requires using increasing codelengths (tending to infinity).

Shannon's proof of the possibility to transmit information at rates arbitrary close the capacity involve the use of *random codes*, i.e a selection of an exponential (in the codelength) number of random binary vectors. Unfortunately, random codes are unuseful for real applications, as they require an exponential amount of memory and retrieval time. A natural way of reducing the memory needed to keep them, is the use of a linear subspace of $\mathbb{Z}_2^n$ as codeword set, as it suffices to keep a vector base to remember the whole subspace. This codes are called *linear codes*. Shannon's proof can be also adapted to random linear codes, proving that random linear codes are also asymptotically capacity-achieving. A linear code can be also characterized by a "parity check" matrix H having as kernel the codeword set. Checking if a given vector is a codeword can be achieved by a simple matrix-vector multiplication.

Encoding with linear codes is time polynomial, but decoding with the maximum likelihood decoder involves finding an $\mathbf{x}$ which minimizes the number of non-zero coordinates of $B\mathbf{x} - \mathbf{c}$. This problem has been proved to be worst-case NP-Complete and it seems difficult to find polynomial time algorithms for its random version.

An obvious approach around this difficulty is to restrict ourselves again to a random sub-ensemble for B and $\mathbf{c}$, for which capacity is maybe not achieved but specific algorithms perform better. The study of typical properties of the random problem is crucial to pick a good one. Several sub-ensembles and coding/decoding schemes have been developed in the last years. Low-density parity-check codes (LDPC), based on sparse matrices B, are a broadly used example of this idea.

Another unrelated application of random XOR-SAT is factorization. The most widely used integer factorization scheme (the *Number Field Sieve*) involve a final step in which a huge binary linear system must be solved.

Information theory is however not restricted to linear codes, in Appendix A we present an experimental lossy compressor based on the SP algorithm for random k-SAT.

CHAPTER 3

The statistical mechanics view: the cavity method

3.1. The energy-shift algorithm

As stated in Sec. (2.1.2), the question if a given graph $G = (I, E)$ is q-colorable can be described by the Hamiltonian

$$H = \sum_{\{i,j\} \in E} \delta(\sigma_i, \sigma_j) \tag{3.1.1}$$

where $\{\sigma_i\} \in \{1, 2, \ldots, q\}$ are "color valued" variables (Potts spins in statistical physics), and $\delta(\cdot, \cdot)$ denotes the Kronecker symbol. This Hamiltonian counts the number of edges being colored equally on both extremities, a proper coloring of the graph thus has $H = 0$. Since this Hamiltonian cannot take negative values, the combinatorial task of finding a coloring is translated to the task of finding a zero-energy ground state, i.e. to the statistical physics of the above model at zero temperature (infinite β).

Let's consider the Hamiltonian of Eq. (3.1.1) for a singly connected graph. For a variable index i we will define $H^{(i)}(\tau)$ as the minimum of the energy in the i-subtree conditioned to $\sigma_i = \tau$ (recall that the i-subtree is the subtree of descendants of i). Given a spin σ_0, we are interested in computing the variation of the energy of the system with and without the spin, i.e. the energy-shift that was generated by the addition of σ_0 to the tree. Suppose σ_0 is connected to $\sigma_1, \ldots, \sigma_k$ (belonging to otherwise disconnected subtrees):

$$H^{(0)}(\sigma_0) = \min_{\sigma_1, \ldots, \sigma_k} \{ \sum_{i=1}^{k} \delta(\sigma_i, \sigma_0) + H^{(i)}(\sigma_i) \} \tag{3.1.2}$$

$$= \sum_{i=1}^{k} \min_{\sigma_i} \{ \delta(\sigma_i, \sigma_0) + H^{(i)}(\sigma_i) \} \tag{3.1.3}$$

We define the vector $\vec{g}_i$ with coordinates $g_i^\tau \stackrel{\text{def}}{=} H^{(i)}(\tau)$ and the vector $\vec{v}_i$ with coordinates $v_i^\tau = \min_p \{\delta(p,\tau) + g_i^p\}$. Thanks to Eqs. 3.1.2, 3.1.3 the definition of v_i^τ can be coupled with $\vec{g}_0 = \sum_{i=1}^k \vec{v}_i$ to obtain a recurrence, allowing to compute the minimal energy of the (tree) system given by $\min_\tau g_{root}^\tau$.

For technical reasons we will change coordinates to energy-shifts h instead of energies g: $\vec{u}_i \stackrel{\text{def}}{=} \min_p g_i^p - \vec{v}_i$ and $\vec{h}_0 \stackrel{\text{def}}{=} \sum_{i=1}^k \vec{u}_i$. Then we have that

$$\begin{aligned} h_0^\tau &= \sum_{i=1}^k u_i^\tau \\ &= \sum_{i=1}^k \min_p g_i^p - \sum_{i=1}^k v_i^\tau \\ &= A - g_0^\tau \end{aligned} \tag{3.1.4}$$

for some $A = \sum_{i=1}^k \min_p g_i^p$ who doesn't depend on τ (A is the energy of the system before adding spin σ_0) and

$$\begin{aligned} \vec{u}_i &= \min_p g_i^p - \min_p \{\delta(p,\tau) + g_i^p\} \\ &= \min_p \{-h_i^p\} - \min_p \{\delta(p,\tau) - h_i^p\} \end{aligned}$$

So we still have a recurrence between $\vec{u}$ and $\vec{h}$. Note that the vectors $\vec{u}$ and $\vec{h}$ are equal to the corresponding $-\vec{v}$ and $-\vec{g}$ respectively except for a constant added to all coordinates. The change of coordinates is useful because $\vec{u}_i$ are very simple objects: $u_i^\tau = \hat{u}^\tau\left(\vec{h}_i\right)$ for

$$\begin{aligned} \omega\left(\vec{h}\right) &\stackrel{\text{def}}{=} \max\left(h^1, ..., h^q\right) \\ \hat{u}^\tau\left(\vec{h}\right) &\stackrel{\text{def}}{=} \max\left(h^1, ..., h^\tau - 1, ..., h^q\right) - \omega\left(\vec{h}\right) \\ &= \max\left(h^1, ..., h^\tau - 1, ..., h^q\right) - \max\left(h^1, ..., h^q\right) \end{aligned} \tag{3.1.5}$$

where we have introduced the *cavity biases* $\hat{u}\left(\vec{h}\right)$. The structure of the cavity biases is easily understood if we distinguish among two different cases:

that in step (1) we need to generate a $G\left(n, \frac{c}{n+1}\right)$ graph, but we will approximate it by $G\left(n, \frac{c}{n}\right)$ for large n.

If we want to apply the energy-shift algorithm to a random graph, we can make use of the above "graph-growing" process and then try to use mathematical induction by applying the same procedure in the previous section, calling now $H^{(0)}$ the Hamiltonian before adding node 0. The main strategy is then to find a recursion of $P_{n+1}\left(\vec{h}\right)$ as a function of $P_n\left(\vec{h}\right)$, and then use this recursion in the limit $n \to \infty$ to obtain a closed equation for the limit probabilities.

We will suddenly face a problem: i.e. that we cannot just write Eq. (3.1.2) as now variables $\sigma_1, \ldots, \sigma_k$ are coupled (mutually dependent).

With high probability (tending to one for large n) the k sites will be far from each other in the original graph (they were randomly picked): although an extensive number of loops is surely present for $c > 1$ [**29**], these loops have with high probability lengths of the order $\log n$. Based on this remark, we will just assume statistical independence of these k sites as an hypothesis and hope for the best. This is something called "one Boltzmann state" or *replica symmetry* inside the *clustering propriety* [**13**] (for a more detailed discussion of this this kind of "summoning" see [**58, 57**]). We certainly know that this is true on trees, and to mention a trivial limit, it can be easily proved that will in consequence be true with probability tending to one in random graphs for $c < 1$, thanks to Theorem. 2.7.1.

Assuming further-on the existence of a well defined thermodynamic limit of the probability distributions of local fields (i.e. $P\left(\vec{h}\right) = \lim_{n\to\infty} P_n\left(\vec{h}\right)$) (for recent rigorous studies in this direction see [**35, 31, 82**]), the distribution of the field $\vec{h}_0$ of the newly added vertex becomes the equal to those of the k neighbors. It is consequently determined by the closed expression

$$P(\vec{h}) = e^{-c}\sum_{k=0}^{\infty}\frac{c^k}{k!}\int\prod_{i=1}^{k}d^q\vec{h}_i\,P(\vec{h}_i)\,\delta(\vec{h}-\sum_{i=0}^{k}\hat{u}_i(\vec{h}_i)) \tag{3.2.1}$$

$$Q(\vec{u}) = \int d^q\vec{h}\,P(\vec{h})\,\delta(\vec{u}-\hat{u}(\vec{h})) \tag{3.2.2}$$

where we have already used that the connectivities k are distributed according to a Poissonian of mean c. The previous equations can be combined in order to have a closed form for the $Q(\vec{u})$:

$$Q(\vec{u}) = e^{-c}\sum_{k=0}^{\infty}\frac{c^k}{k!}\int\prod_{i=1}^{k}d^q\vec{u}_i Q(\vec{u}_i)\,\delta\left(\vec{u}-\hat{u}\left(\sum_{i=1}^{k}\vec{u}_i\right)\right) \tag{3.2.3}$$

From the symmetry of our model under arbitrary permutations of colors we conclude that

$$Q(\vec{e}_1) = Q(\vec{e}_2) = \cdots = Q(\vec{e}_q) = \eta \quad \text{and} \quad Q\left(\vec{0}\right) = 1-q\eta \tag{3.2.4}$$

i.e. we need a single real number η with $0 < \eta < \frac{1}{q}$ to completely specify the probability distribution function $Q(\vec{u})$. Noting that the probability $P^{(k)}\left(\vec{h}\right)$ for a site with k neighbors can be expressed by

$$P^{(k)}\left(\vec{h}\right) = \int\prod_{i=1}^{k}d^q\vec{u}_i Q(\vec{u}_i)\,\delta\left(\vec{h}-\sum_{i=1}^{k}\vec{u}_i\right) \tag{3.2.5}$$

and recalling that $\vec{u}_i \in \left\{\vec{0},\vec{e}_1,\ldots,\vec{e}_q\right\}$ it is easy to rewrite this probability distribution in a compact multinomial form

$$P^{(k)}\left(\vec{h}\right) = P^{(k)}\left(h^1,h^2,\ldots,h^q\right) \tag{3.2.6}$$

$$= \frac{k!\;\eta^{-\sum_{\tau=1}^{q}h^\tau}\left(1-q\eta\right)^{k+\sum_{\tau=1}^{q}h^\tau}}{\left(k+\sum_{\tau=1}^{q}h^\tau\right)!\prod_{\tau=1}^{q}\left(-h^\tau\right)!} \tag{3.2.7}$$

with the agreement that $1/n! = 0$ for $n < 0$. Note that h^τ belongs to $\{0,-1,\ldots,-k\}$ and that there are correlations among the different components of the cavity fields such that $P^{(k)}\left(h^1,\ldots,h^q\right) \neq \prod_{\tau=1}^{q}\mathcal{P}\left(h^\tau\right)$. Now we are ready to calculate the graph average over the

Poissonian connectivity distribution of mean c,

$$P\left(h^1,...,h^q\right) = e^{-c}\sum_k \frac{c^k}{k!}P^k\left(h^1,\ldots,h^q\right) \tag{3.2.8}$$

$$= e^{-c\eta q}\prod_{\tau=1}^{q}\frac{(c\eta)^{-h^\tau}}{(-h^\tau)!} \tag{3.2.9}$$

$$= \prod_{\tau=1}^{q}\mathcal{P}_{c\eta}(h^\tau) \tag{3.2.10}$$

It is interesting to note that after the average, correlations among the different colors disappear and P is the product of q Poissonian distributions with average $c\eta$. From Eq. (3.2.2) it is possible to derive fixed-point equation for the order parameter noting that the probability η to obtain a non-trivial cavity bias - say $\vec{e}_\tau$ - is simply given by the probability that the τ^{th} component of the local field is the non-degenerate smaller, so setting $\tau = 1$

$$\eta = \sum_{h^1=0}^{\infty}\sum_{h^2=h^1+1}^{\infty}\cdots\sum_{h^q=h^1+1}^{\infty}P\left(h^1,\ldots,h^q\right) \tag{3.2.11}$$

$$= e^{-c\eta}\sum_{n=0}^{\infty}\frac{(c\eta)^n}{n!}\left(1-\frac{\Gamma(n+1,c\eta)}{\Gamma(n+1)}\right)^{q-1} \tag{3.2.12}$$

where $\Gamma(n,x)$ is the incomplete Gamma function defined from the following useful relation

$$e^{-x}\sum_{k=n}^{\infty}\frac{x^k}{k!} = 1-\frac{\Gamma(n,x)}{\Gamma(n)} \tag{3.2.13}$$

The sum in Eq. (3.2.11) converges very fast. It is therefore easy to numerically construct a solution to this equation as a function of c. For $q > 2$ it turns out that η jumps discontinuously from zero to a finite value as shown in Fig. (3.2.1) where the order parameter η jumps at $c = 5.141$ in the case of $q = 3$.

This means that, up to $c = 5.141$ and at the level of the replica symmetric assumption, we only find the paramagnetic solution $\eta = 0$.

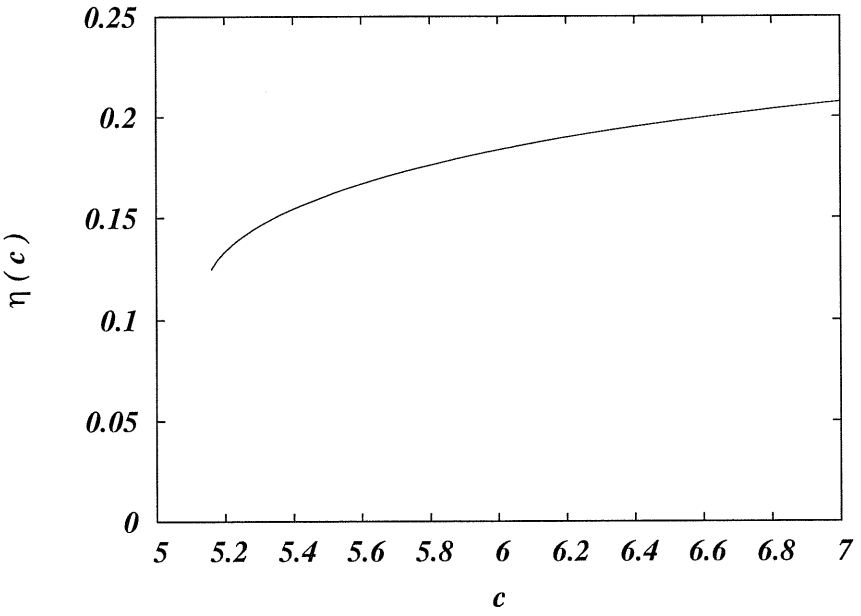

FIGURE 3.2.1. Replica-symmetric order parameter η vs. average connectivity c for $q = 3$, from Eq. (3.2.11)

3.2.1. The calculation of the energy. One can easily compute the average shift in the ground state energy when a new spin is added to the n-sites system and it is connected to k spins of the system. The energy of the original graph is given by $A = \sum_{i=1}^{k} \min_p g_i^p$ while the energy of the $n+1$-sites system is $\min_p g_0^p = A - \omega\left(\vec{h}_0\right)$ by Eq. (3.1.4), so the average energy shift is given by

$$\Delta H_1 = -\sum_{k=0}^{\infty} e^{-c} c^k / k! \int \prod_{i=1}^{k} d^q \vec{u}_i Q(\vec{u}_i)\, \omega\left(\sum_{i=1}^{k} \vec{u}_i\right) \tag{3.2.14}$$

$$= -\int d^q \vec{h}\, P(\vec{h})\, \omega(\vec{h}) \,. \tag{3.2.15}$$

One might be tempted to conclude that Eq. (3.2.15) equals the energy density of the system, at least for n large enough, but this is not true. There is a correction term due to the change in the number of links per variable in the iteration $n \to n+1$: the early approximation we made of $G\left(n, \frac{c}{n+1}\right)$ by $G\left(n, \frac{c}{n}\right)$ is slightly wrong. In fact, generating links with probability c/n in a $n+1$ system, instead of $c/(n+1)$ we

are slightly over-generating links. So, we need to calculate the average energy shift in a system when two sites - say spins σ_1 and σ_2 - are joined by an edge.

Again, the energy of the original graph is $\sum_{i=1,2} \min_p g_i^p = A_1 + A_2 - \omega(\vec{h}_1) - \omega(\vec{h}_2)$, when we join these to sites by an edge, we get an energy $\min_{\sigma_1\sigma_2} \left\{ \sum_{i=1,2} g_i^{\sigma_i} + \delta(\sigma_1, \sigma_2) \right\}$ which is equal to $A_1 + A_2 - \min_{\sigma_1,\sigma_2}(-h_1^{\sigma_1} - h_2^{\sigma_2} + \delta(\sigma_1, \sigma_2))$. The difference between the two contributions can be written as

$$\begin{aligned} \Delta H_{\rm link} &= \min_{\sigma_1}(-h_1^{\sigma_1} + \min_{\sigma_2}(-h_2^{\sigma_2} + \delta(\sigma_1,\sigma_2))) + \omega(\vec{h}_1) + \omega(\vec{h}_2) \\ &= \min_{\sigma_1}(-h_1^{\sigma_1} - u^{\sigma_1}(\vec{h}_2) - \omega(\vec{h}_2)) + \omega(\vec{h}_1) + \omega(\vec{h}_2) \\ &= -\omega(\vec{h}_1 + \hat{u}(\vec{h}_2)) + \omega(\vec{h}_1) \end{aligned} \tag{3.2.16}$$

This allows us to express the average link-energy shift as

$$\Delta H_2 = \int d^q\vec{h}_1 d^q\vec{h}_2\, P(\vec{h}_1)P(\vec{h}_2)\, \Big(\omega(\vec{h}_1) - \omega(\vec{h}_1 + \hat{u}(\vec{h}_2))\Big) \tag{3.2.17}$$

It is interesting to observe that Eqs. (3.2.15) and (3.2.17) are *model-independent*, in the sense that the actual Hamiltonian is encoded into the functions $\omega(\vec{h})$ and $\hat{u}(\vec{h})$ defined by Eq. (3.1.5).

Using Eq. (3.1.5) and (3.2.8) one shows easily that Eq. (3.2.15) reduces to:

$$\begin{aligned} \Delta H_1 &= \sum_{h^1\dots h^q} \mathcal{P}_{c\eta}(h^1)\dots\mathcal{P}_{c\eta}(h^q)\min(-h^1,-h^2,\dots,-h^q) \\ &= -\sum_{\alpha=0}^{q-1} \binom{q}{q-\alpha} \sum_{h=0}^{-\infty} h\mathcal{P}_{c\eta}(h)^{q-\alpha} \left(\sum_{g<h}^{-\infty} \mathcal{P}_{c\eta}(g)\right)^{\alpha} \end{aligned} \tag{3.2.18}$$

It is also not hard to prove that the average link-energy shift $\Delta H_2 = q\eta^2$. This result can be obtained either by direct computation of the integral, or following a simple probabilistic argument: $\Delta H_{\rm link}$ is different from zero whenever the two unlinked sites have the same color, but this happens with probability η^2 for each of the colors. Finally we have the the following equation for the energy which is equivalent to

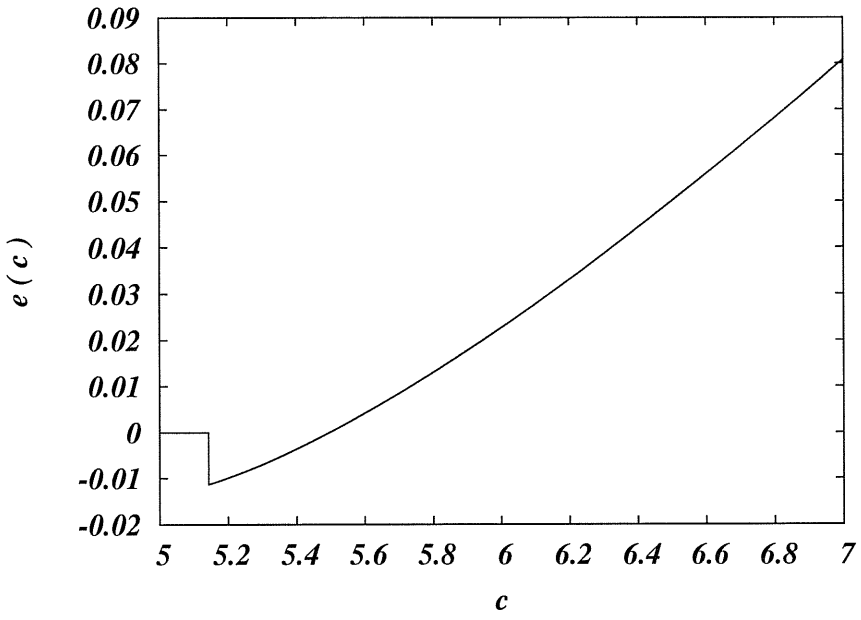

FIGURE 3.2.2. Energy density $e = H/n$ vs. average connectivity c for $q = 3$ in the RS approximation from Eq. (3.2.19)

the "replica-symmetric" approximation:

$$H = n\left(\Delta H_1 - \frac{c}{2}\Delta H_2\right) = \tag{3.2.19}$$

$$-\sum_{\alpha=0}^{q-1}\binom{q}{q-\alpha}\sum_{h=0}^{-\infty} h\mathcal{P}_{c\eta}(h)^{q-\alpha}\left(\sum_{g<h}^{-\infty}\mathcal{P}_{c\eta}(g)\right)^{\alpha} - \frac{c}{2}q\eta^2 \tag{3.2.20}$$

The behavior of the energy for $q = 3$ as a function of the average connectivity c is displayed in Fig. (3.2.2). Let us note that for average connectivity $5.141 < c < 5.497$ the energy is negative, a particularly baffling result if we consider that the Hamiltonian (3.1.1) is at least positively defined. This phenomenon is analogous to what already observed for the RS approximation in random 3-SAT **[63]**, and is a consequence of the approximation used.

3.3. 1-step RSB solution

3.3.1. Pure states. The RS results show some evident pathologies and are at odd with numerical simulations [**4, 69**] which predict a lower threshold around $c = 4.7$, and with the rigorous upper bound $c = 4.99$ [**40**]. The main assumption we have made is the statistical independence of the of the k cavity fields. Is it true that long distance among spins imply statistical independence of the cavity fields $\vec{h_i}$? This was our central hypothesis, making the RS equations closed. In general the answer is no. Still statistical physics have developed a theory to overcome this difficulty.

The answer coming from this theory is that the assumption holds only inside "pure states". Of course, we haven't yet defined what a pure state is, and unfortunately there is no agreement upon such a mathematical definition in the framework of the theory. But let's explain informally what a "pure state" means from our understanding of the physical point of view. Suppose we have a random Hamiltonian $H = \sum_{a \in A_n} H_a$ with $|A_n| \propto n$ over a set of configurations $\mathbf{x} \in \{-1, 1\}^n$ and we define as usual the Gibbs probability measure $P_\beta = Z^{-1} e^{-\beta H}$. It can be proved that the probability P_β is the limit probability of finding a physical system (represented as a single configuration $\mathbf{x}_t$ in a Markov chain whose graph is the hypercube with vertices in $\{-1, 1\}^n$) which obeys the dynamical transition law $P(\mathbf{x}_{t+1} | \mathbf{x}_t) = \min\left(1, e^{-\beta(H(\mathbf{x}_{t+1}) - H(\mathbf{x}_t))}\right)$. When the time t tends to infinity, the distribution of $\mathbf{x}_t$ becomes P_β, and this is in fact the utility of the Gibbs measure: be able to describe a dynamic process with a static measure.

In Markov chains the speed of convergence (relaxation or *mixing* time) to this limit probability however can be very variable, and is often strongly dependent on the size of the system: for instance the system requires at least $O\left(e^{\beta B_n}\right)$ time to cross an energy barrier of energy B_n separating the space of configurations in two. If B_n grows linearly with the size n, we will have an exponential slowdown. Physical systems are normally huge (may have about Avogadro's number of coordinates) and have normally not enough time to relaxate, and so it would seem that the observed distribution probability for $\mathbf{x}_t$ for a large (but finite)

time t on a huge physical system cannot be described satisfactorily by the static measure P_β. Is in this interplay between the limit of infinite volume and the limit of infinite time that the notion of pure states enters into scene. The "trick" is to take the limit of $n \to \infty$ *before* the time limit $t \to \infty$: in this way we are both capturing the notion that the system is not able to relaxate completely to P_β (because huge size implies terribly long time) but still maintaining the simple formalism of a static probability measure like the Gibbs measure. Is in this limit that "unbounded (with n) energy barriers" become infinite, and statistical physicists settle with a partition of the probability measure of the type $P = \sum_\alpha \omega_\alpha P_\alpha$ where $\omega_\alpha \in [0,1]$ and the supports of P_α are disjoint, and there are "infinite-energy barriers" between those supports (that is, borders are made of configurations with probabilities tending to 0). In a first approximation, they assume moreover $\omega_\alpha = \omega$ to be constant over α. The probabilities P_α are called "pure states". In the limit of $\beta \to \infty$ we don't need the concept of probability anymore and we can simply talk about "sets": we say that the set of solutions of $H = 0$ (assuming that 0 is the ground-state) separates into "clusters". The quantity $\Sigma = \frac{1}{n} \log$ (number of clusters) is called *complexity* and denotes the entropy density of clusters. This differs in general from the solution entropy since each cluster may contain as well a number of solutions.

Note that there are several obscure points in the definition of the P_α probability measures. One of the most striking ones is that this definition is stated on the limit of infinite size (where by the way still there is no good definition of P) even when the expression $P = \sum_\alpha \omega_\alpha P_\alpha$ is used at finite n. In particular, this definition is not at all satisfactory for the definition of the SP algorithm, which makes sense only at finite n. Another stunning peculiarity is that without a precise definition of these P_α it seems hard to write equations for propagating averages over them. As we will see at the end (at least in our particular case), the definition of the P_α is implicit on the equations for the propagation. We will see an alternative definition (for the case of k-SAT) in Chapter 5.

The reader may ask what does this notion of pure states, which seems mostly related to a physical phenomenon, has to do with solving combinatorial problems. The answer is in the techniques developed by physicists to address this phenomenon: the theory suggests a replacement to the presumably wrong hypothesis of un-correlation of cavity fields $\vec{h_i}$ (RS hypothesis) by a much more complex un-correlation one (1-RSB hypothesis). In other words, this partition of the probability function may be viewed as an artifice to identify a set of random variables which are indeed uncorrelated (or more likely to be so) but still give enough information for our needs. Another interesting aspect is that the analysis of the dynamical model above is still relevant to understand the behaviour of some "local search" algorithms, that reproduce a dynamics similar than of the physical model.

If not yet confused, the reader may also start to worry about things getting a bit off-hand here, as we started with a well posed problem (say find the critical threshold) and now we have a handful of not–well–determined definitions going on (some of them mutually-referenced). Please keep in mind that the cavity method is a *physical model* and not a formal proof. We will see in Chapter 5 a precise formulation of these in our particular case (k-SAT in the SAT phase) by means of interpreting the SP equations.

3.3.2. 1 RSB cavity equation . The first basic assumption we make is that inside each pure state the clustering condition holds. Under this assumption the iteration can still be applied but we have to take into account the reshuffling of energies of different states when new spins are added (pure states can have non-zero ground-state energy).

We proceed following the same steps of the previous section. Let us take the new spin σ_0 and let us connect it to k spins $\sigma_1, \ldots, \sigma_k$ in the same state α. Thanks to the hypothesis of lack of correlations inside a pure state the energy of state α for fixed value of the k spins is

$$E_\alpha^n(\sigma_1, \ldots, \sigma_k) = A_\alpha - \sum_{i=1}^{k} \sum_{\tau=1}^{q} h_{i,\alpha}^{\tau} \delta(\tau, \sigma_{i,\alpha}) \tag{3.3.1}$$

The optimization step within each pure state α runs still in close analogy to the RS computation: when we connect σ_0 to $\sigma_1, \ldots, \sigma_k$, we express the minimal energy of the $n+1$-sites graph with fixed σ_0, by minimizing the $n+1$-sites system at fixed σ_0 is thus obtained by minimizing E_α^{n+1} with respect to the k spins:

$$E_\alpha^{n+1}(\sigma_0) = A_\alpha - \sum_{i=1}^{k} \omega(\vec{h}_{i,\alpha}) - \sum_{i=1}^{k} \sum_{\tau=1}^{q} \hat{u}^\tau(\vec{h}_{i,\alpha}) \delta(\tau, \sigma_0) \tag{3.3.2}$$

This last equation shows that the local field acting on the new spin σ_0 in the state α is

$$\vec{h}_{0,\alpha} = \sum_{i=1}^{k} \hat{u}(\vec{h}_{i,\alpha}) \tag{3.3.3}$$

and that the energy shift inside a state is

$$\Delta E_\alpha = -\sum_{i=1}^{k} \omega\left(\hat{u}(\vec{h}_{i,\alpha})\right) \tag{3.3.4}$$

All the previous equation are completely equivalent to those in the RS case except for the fact that now we have a α-index labeling the different pure states. One natural question is how cavity fields and the related cavity biases are distributed for a given site among the different pure states. This leads us to the notion of *survey* **[56, 57, 58]**, *i.e.* the site dependent normalized histogram over the different states of both cavity biases and cavity fields:

$$Q_i(\vec{u}_i) = \frac{1}{\mathcal{M}} \sum_{\alpha=1}^{\mathcal{M}} \delta(\vec{u}_i - \vec{u}_{i,\alpha}) \tag{3.3.5}$$

$$P_i(\vec{h}_i) = \frac{1}{\mathcal{M}} \sum_{\alpha=1}^{\mathcal{M}} \delta(\vec{h}_i - \vec{h}_{i,\alpha}) \tag{3.3.6}$$

In close analogy with what we have already done in the RS case, the hypothesis of the existence of a well defined thermodynamic limit implies that there must exist unique functional probability distributions $\mathcal{Q}[Q(\vec{u})]$ and $\mathcal{P}[P(\vec{u})]$ for all the surveys. One may wonder how could we handle such a big functional space: Fortunately the Q-surveys are

described in terms of a single real number $0 \leq \eta \leq 1/q$, cf. Eq. (3.2.4), and a scalar function $\rho(\eta)$ is enough for specifying their distribution:
(3.3.7)

$$\mathcal{Q}[Q(\vec{u})] = \int d\eta\ \rho(\eta)\ \delta\left[Q(\vec{u}) - (1-q\eta)\delta(\vec{u}) - \eta\sum_{\tau=1}^{q}\delta(\vec{u}-\vec{e}_\tau)\right]$$

with $\delta[\cdot]$ denoting a functional Dirac distribution. Assuming that the survey of site 0 is distributed equally to those of all its k neighbors, we can write:

$$P_0(\vec{h}) = e^{-c}\sum_{k=0}^{\infty}\frac{c^k}{k!}C_k\int\prod_{i=1}^{k}d^q\vec{u}_i Q_i(\vec{u}_i)e^{y\omega(\sum_{i=1}^{k}\vec{u}_i)}\delta(\vec{h}-\sum_{i=1}^{k}\vec{u}_i) \tag{3.3.8}$$

$$Q_0(\vec{u}) = \int d^q\vec{h}P_0(\vec{h})\delta(\vec{u}-\hat{u}(\vec{h})) \tag{3.3.9}$$

Note the presence of the *reweighting factor* $\exp(y\omega(\sum_{i=1}^{k}\vec{u}_i))$ that arise after conditioning the probability distributions of the $\vec{h}$s to a given value of energy [57], the pref-actors C_k are normalization constants depending on $Q_1(\vec{u}),...,Q_k(\vec{u})$. The reweighting parameter y is a number equal to the derivative of the complexity $\Sigma(e)$ of metastable states with respect to their energy density $e = H/n$:

$$y = \frac{\partial\Sigma}{\partial e} \tag{3.3.10}$$

Eq. 3.3.8 can be explained as follows: suppose one wants to compute the distribution of $P_0(\vec{h},\epsilon)$ restricted to a specific value of the energy density $\epsilon = H/n$. This distribution depends on the distribution of the $Q_i(\vec{u},\epsilon')$ for values of ϵ' near ϵ (because adding one site changes the energy only in $\omega(\sum_{i=1}^{k}\vec{u}_i)$), and of the relative weight of these, i.e. the proportion of the number of solutions in the two energy levels. Remembering that the number of solutions at a given energy level H is $\exp(n\Sigma(H/n))$, we have (by calling $\epsilon' = \left(H+\omega(\sum_{i=1}^{k}\vec{u}_i)\right)/n$ and $\epsilon = H/n$ for shortness):

$$(3.3.11)\quad P_0^{(k)}(\vec{h},\epsilon) \propto \int \prod_{i=1}^{k} d^q\vec{u}_i Q_i(\vec{u}_i,\epsilon') \frac{e^{n\Sigma(\epsilon')}}{e^{n\Sigma(\epsilon)}} \delta(\vec{h}-\sum_{i=1}^{k}\vec{u}_i)$$
$$= \int \prod_{i=1}^{k} d^q\vec{u}_i Q_i(\vec{u}_i,\epsilon') e^{n(\Sigma(\epsilon')-\Sigma(\epsilon))} \delta(\vec{h}-\sum_{i=1}^{k}\vec{u}_i)$$

Now assuming smoothness of Σ and continuity of Q_i with respect to ϵ, the expression in the exponent converges to $\frac{\partial\Sigma}{\partial\epsilon}(\epsilon)\,\omega(\sum_{i=1}^{k}\vec{u}_i) = y\omega(\sum_{i=1}^{k}\vec{u}_i)$ and $Q_i(\vec{u}_i,\epsilon')$ converges to $Q_i(\vec{u}_i,\epsilon)$. Integrating over k we get the Equation 3.3.8 of P_0 at a fixed value of $\epsilon = H/n$. Note that y is still an unknown as we don't have the function Σ yet.

Another feature that is implicit in Eq. 3.3.11 is that every combination of "input" clusters in the n-spin system leads to a single "output" cluster in the $n+1$-spin system.

Intuitively, this reweighting factor can be understood as a penalty $e^{-y\Delta E_\alpha}$ one has to pay for positive energy shifts. Note that Eqs. (3.3.8) and (3.3.9) can be cast in the following form

$$Q_0(\vec{u}_0) = e^{-c}\sum_{k=0}^{\infty}\frac{c^k}{k!}C_k \int \prod_{i=1}^{k} d^q\vec{u}_i Q_i(\vec{u}_i) e^{y\omega(\sum_{i=1}^{k}\vec{u}_i)} \delta(\vec{u}_0 - \hat{u}(\sum_{i=1}^{k}\vec{u}_i))$$
$$(3.3.12)\quad = e^{-c}\sum_{k=0}^{\infty}\frac{c^k}{k!}C_k \int d^q\vec{h}\tilde{P}(\vec{h}) e^{y\omega(\vec{h})}\delta(\vec{u}_0 - \hat{u}(\vec{h}))$$

In the last line we have introduced the auxiliary distribution $\tilde{P}(\vec{h})$ which would result in Eq. (3.3.8) without reweighting (i.e. by setting $y = 0$). It has no direct physical meaning in this context, but it will be of great technical help in the following calculations.

Let us first concentrate on the *colorable phase*, where the ground states are proper q-colorings and have zero energy. Consequently no positive energy shifts are allowed, so this phase is characterized by $y = \infty$ (this means that $\frac{\partial\Sigma}{\partial e}(0) = +\infty$). Let us first calculate the value of the normalization constants C_k in this limit. Note that $\omega(\vec{h}) \leq 0$ for all allowed $\vec{h}$ (each component of $\vec{h}$ is non-positive as $\vec{h}$ results from a sum over $\vec{u}$s). This means that the only surviving terms in Eq. (3.3.12)

are those with zero energy shift $\omega(\vec{h}) = 0$, *i.e.* all fields must have at least one zero component, allowing for the selecting of at least one color without violating an edge. Let us first specialize to the case $q = 3$ for clarity, the generalization to arbitrary q is straightforward. Summing over $\vec{u}_0$ both sides of Eq. (3.3.12) we have:

$$\frac{1}{C_k} = \tilde{P}(0,0,0) + 3\sum_{h^1<0} \tilde{P}(h^1,0,0) + 3\sum_{h^1,h^2<0} \tilde{P}(h^1,h^2,0) \tag{3.3.13}$$

where the combinatorial factors 3 appearing in the r.h.s. are obtained by noting that $\tilde{P}(h,0,0) = \tilde{P}(0,h,0) = \tilde{P}(0,0,h)$ and that $\tilde{P}(h^1,h^2,0) = \tilde{P}(h^1,0,h^2) = \tilde{P}(0,h^1,h^2)$. Combining Eqs. (3.3.8), (3.3.9) and (3.2.4) we get

$$\begin{aligned}
\tilde{P}(0,0,0) &= \prod_{i=1}^{k}(1-3\eta_i) \\
\sum_{h^1<0} \tilde{P}(h^1,0,0) &= \prod_{i=1}^{k}(1-2\eta_i) - \tilde{P}(0,0,0) \\
&= \prod_{i=1}^{k}(1-2\eta_i) - \prod_{i=1}^{k}(1-3\eta_i) \\
\sum_{h^1,h^2<0} \tilde{P}(h^1,h^2,0) &= \prod_{i=1}^{k}(1-\eta_i) - 2\sum_{h^1<0} \tilde{P}(h^1,0,0) - \tilde{P}(0,0,0) \\
&= \prod_{i=1}^{k}(1-\eta_i) - 2\prod_{i=1}^{k}(1-2\eta_i) + \prod_{i=1}^{k}(1-3\eta_i) \qquad (3.3.14)
\end{aligned}$$

Plugging these relations into Eq. (3.3.13) we finally get

$$\frac{1}{C_k} = 3\prod_{i=1}^{k}(1-\eta_i) - 3\prod_{i=1}^{k}(1-2\eta_i) + \prod_{i=1}^{k}(1-3\eta_i) \tag{3.3.15}$$

Note also that in close analogy to the analysis that lead to Eq. (3.2.11), we can interpret (3.3.14) as the (un-normalized) probability of having the survey in site 0 *pointing* in direction $\vec{e}_3$. Therefore combining

Algorithm 1 Population dynamics algorithm for the 1RSB solution

- (i) Start with an initial population $\eta_1, ..., \eta_{\mathcal{M}}$ of size $\mathcal{M}$ which can be easily chosen to be as large as 10^6 to generate high-precision data.
- (ii) Randomly draw a number k from the Poisson distribution $e^{-c}c^k/k!$;
- (iii) Randomly select $k+1$ indices $i_0, i_1, ..., i_k$ from $\{1, ..., \mathcal{M}\}$;
- (iv) Update the population by replacing η_{i_0} by $f_d(\eta_{i_1}, ..., \eta_{i_k})$;
- (v) Go to (ii) until convergence of the algorithm is reached.

Eqs. (3.3.12) and (3.3.15) we obtain

(3.3.16)

$$\eta_0 = \hat{f}_k(\eta_1, \ldots, \eta_k) = \frac{\prod_{i=1}^k (1-\eta_i) - 2\prod_{i=1}^k (1-2\eta_i) + \prod_{i=1}^k (1-3\eta_i)}{3\prod_{i=1}^k (1-\eta_i) - 3\prod_{i=1}^k (1-2\eta_i) + \prod_{i=1}^k (1-3\eta_i)}$$

At this point we are ready to write the 1-RSB iterative equation for the Q-surveys:

$$\rho(\eta) = e^{-c} \sum_{k=0}^{\infty} \frac{c^k}{k!} \int \prod_{i=1}^{k} d\eta_i \rho(\eta_i)\ \delta(\eta - \hat{f}_k(\eta_1, \ldots, \eta_k)) \tag{3.3.17}$$

Eq. (3.3.16) can be easily generalized to an arbitrary number q of colors,

$$\hat{f}_k(\eta_1, ..., \eta_k) = \frac{\sum_{l=0}^{q-1} (-1)^l \binom{q-1}{l} \prod_{i=1}^k [1-(l+1)\eta_i]}{\sum_{l=0}^{q-1} (-1)^l \binom{q}{l+1} \prod_{i=1}^k [1-(l+1)\eta_i]} \tag{3.3.18}$$

The "self-consistency" equation (3.3.17) resembles a replica-symmetric equation and can be solved numerically using a population dynamics algorithm (Algorithm 1). Note that we are implicitly assuming independence of the η values, i.e. of the Q measures.

One obvious solution of Eqs. (3.3.17) and (3.3.18) is the paramagnetic solution $\delta(\eta)$. For small average connectivities c it is even the only one. The appearance of a non-trivial solution coincides with a clustering transition of ground states into an exponentially large number of extensively separated clusters. In spin-glass theory, this transition is called dynamical. Still, $\rho(\eta)$ will contain a non-trivial peak in $\eta = 0$ due

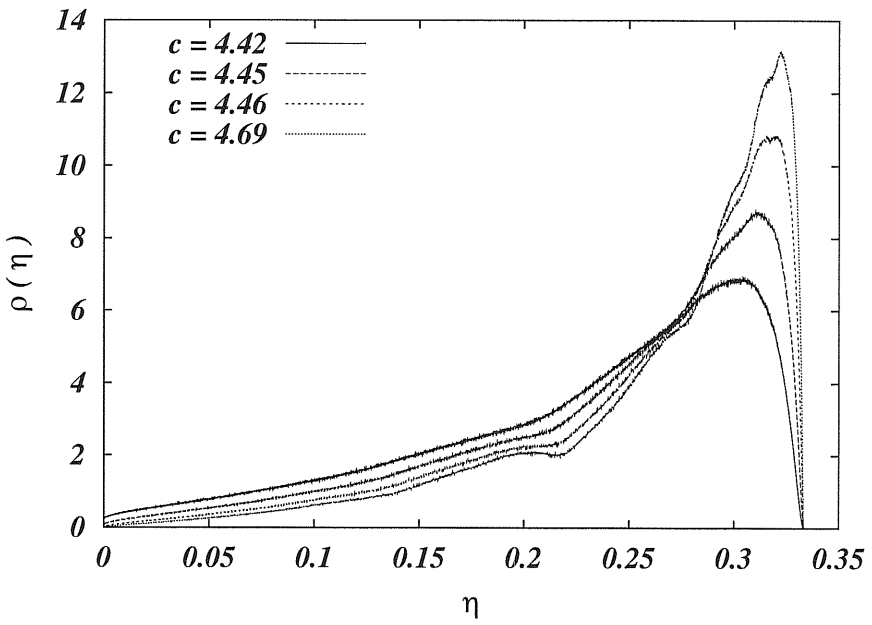

FIGURE 3.3.1. Probability distribution function $\rho(\eta)$ in the case $q = 3$ for average connectivities $4.42 < c < 4.69$. Note also that a delta peak in $\eta = 0$ is always present (but not displayed here).

to small disconnected subgraphs, dangling ends, low-connectivity vertices etc. The shape of $\rho(\eta)$ in the case $q = 3$ is displayed in Fig. (3.3.1) for connectivities c ranging from c_d to c_c.

The weight t of this peak can be computed self-consistently. Let us again consider first the case $q = 3$. Keeping in mind that for $y \to \infty$ the field $\vec{h}$ has at least one vanishing component, the only possibilities to obtain $\hat{u}(\vec{h}) = \vec{0}$ are given by $\vec{h} = \vec{0}$ or by a field $\vec{h}$ with one single non-zero component. So the probability that the cavity field acting on a given site with k neighbors equals zero is given by the sum of the probabilities that all neighboring cavity fields are zero (equal to t^k), plus the probability that exactly one cavity bias among the k is non-trivial (equal to $k(1-t)t^{k-1}$). The average over the Poissonian degree distribution leads to

$$t = e^{-c} \sum_{k=0}^{\infty} \frac{c^k}{k!} \left(t^k + k t^{k-1} (1-t) \right) = e^{-(1-t)c} \left(1 + (1-t)c \right) \tag{3.3.19}$$

Generalizing Eq. (3.3.19) to a general number q of colors easily gives

$$t = e^{-(1-t)c} \sum_{l=0}^{q-2} \frac{(1-t)^l c^l}{l!} \,. \tag{3.3.20}$$

This equation is quite interesting, since a non-trivial solution forms a necessary condition for Eq. (3.3.17) to have a non-trivial solution. In fact, this equation was first found in [**75**], the fraction of edges belonging to the q-core is given by $(1-t_{min})$ with t_{min} being the smallest positive solution of Eq. (3.3.20). Thus, we also find that the existence of an extensive q-core is necessary for a non-trivial $\rho(\eta)$, and forms a lower bound for the q-COL/UNCOL transition.

Unlike in the case of finite-connectivity p-spin-glasses or, equivalently, random XOR-SAT problems [**76, 23, 59**], the existence of a solution $t < 1$ is not sufficient for a non-trivial $\rho(\eta)$ to exist. The latter appears suddenly at the dynamical transition c_d, which can be determined to high precision using the population dynamical algorithm. This solution does not imply uncolorability, but the set of solutions is separated into an exponentially large number of clusters. The number of these clusters, or more precisely its logarithm divided by the graph size n, is called the complexity Σ and can be calculated from $\rho(\eta)$.

3.3.3. The calculation of energy and complexity. More generally, we expect also a large number of metastable states (at non-zero energy) to exist. Hereafter we will assume that they are exponentially many, $\mathcal{N}(e) \propto \exp(n\Sigma(e))$, where the complexity $\Sigma(e)$ is an intensive function (i.e. having a finite $n \to \infty$ limit) of the energy density $e = E/n$. We can introduce a thermodynamic potential $\phi(y)$ as

$$\phi(y) = -\frac{1}{yn} \ln \left(\int de \; e^{n\{-ye+\Sigma(e)\}} \right) \tag{3.3.21}$$

For large n, we calculate this integral by its saddle point:

$$\phi(y) = \min_e \left(e - \frac{1}{y}\Sigma(e) \right) = e_{sp} - \frac{1}{y}\Sigma(e_{sp}) \tag{3.3.22}$$

i.e. by calling $\hat{\phi}(y,e) \stackrel{\text{def}}{=} e - \frac{1}{y}\Sigma(e)$ we have that $\phi(y) = \hat{\phi}(y, e_{sp}(y))$. It is easily verified that the potential ϕ calculated at the saddle point energy $e_{sp}(y)$ fulfills the usual Legendre relations:

$$\partial_y \left[y\,\hat{\phi}(y, e_{sp}(y)) \right] = e_{sp} \tag{3.3.23}$$

$$y^2 \partial_y \hat{\phi}(y, e_{sp}(y)) = \Sigma(e_{sp}) \tag{3.3.24}$$

Around the saddle point the complexity can be approximated, according to Eq. (3.3.22), by

$$\Sigma(e) \simeq \Sigma(e_{sp}) + y(e - e_{sp}) = -y\phi(y) + ye \tag{3.3.25}$$

We will now consider a cavity argument: let us denote by E_n the energy of a system composed of n sites, the density of configurations is given by

$$d\mathcal{N}_n(E_n) \propto e^{-y\Phi_n(y) + yE_n} dE_n \tag{3.3.26}$$

with $\Phi_n(y)$ denoting the extensive thermodynamic potential with limit $\Phi_n(y)/n \to \phi(y)$. Now we add a spin to the system. If we consider that the total energy is $E_{n+1} = E_n + \Delta E$, we can express the density of configurations in terms of E_n and ΔE:

$$d\mathcal{N}_{n+1}(E_n, \Delta E) \propto e^{y\Phi_n(y) + y(E_n + \Delta E)}\; dE_n\; P(\Delta E)\; d\Delta E\;. \tag{3.3.27}$$

Integrating over δE we get

$$d\mathcal{N}_{n+1}(E_{n+1}) = C e^{-y\Phi_{n+1}(y) + yE_{n+1}} dE_{n+1} \tag{3.3.28}$$

$$C = \frac{1}{y} \int P(\Delta E) e^{y\Delta E} d\Delta E \equiv \frac{1}{y} \langle e^{y\Delta E} \rangle_{P(\Delta E)} \tag{3.3.29}$$

Comparing the previous equations with (3.3.26) we can deduce that

$$\Phi_{n+1}(y) = \Phi_n(y) - \frac{1}{y} \ln \langle \exp(y\Delta E) \rangle_{P(\Delta E)}\;. \tag{3.3.30}$$

In the thermodynamic limit we can thus identify

$$\phi(y) = -\frac{1}{y} \ln \langle \exp(y\Delta E) \rangle_{P(\Delta E)} \tag{3.3.31}$$

In close analogy with what we have already done in the RS case, and using Eq. (3.3.4), we can compute ϕ as a *site* contribution plus a *link* contribution in the 1-RSB scenario:

- *Site Addition*

$$\exp\left(-y\Delta\phi_1\right) = \int \prod_{j=1}^{k} d^q\vec{u}_{i_j} Q_{i_j}(\vec{u}_{i_j}) \exp(y\omega(\sum_{j=1}^{k} \vec{u}_{i_j})) = \frac{1}{C_k} \tag{3.3.32}$$

- *Link Addition*

$$\begin{aligned}
\exp\left(-y\Delta\phi_2\right) &= \int \prod_{j=1}^{k} d^q\vec{u}_{i_j} Q_{i_j}(\vec{u}_{i_j}) \exp(-y\omega(\vec{h}_{i_1}) + y\omega(\vec{h}_{i_1} + \hat{u}(\vec{h}_{i_2}))) \\
&= \int d^q\vec{h} P_{i_1}(\vec{h}) d^q\vec{u} Q_{i_2}(\vec{u}) \exp[-y\left(\omega(\vec{h}) - \omega(\vec{h}+\vec{u})\right)] \\
&= 1 + q\eta_{i_1}\eta_{i_2}(e^{-y} - 1) \qquad (3.3.33)
\end{aligned}$$

Note that in the limit $y \to 0$ and assuming $P_i = P$ for each site, we obtain the RS expressions. Once the functional distributions $\mathcal{Q}[Q(\vec{u})]$ and $\mathcal{P}[P(\vec{h})]$ are known we can eventually average the energy shifts $\Delta\phi_1$, $\Delta\phi_2$ in the usual linear combination:

$$\phi(y) = \overline{\Delta\phi_1} - \frac{c}{2}\overline{\Delta\phi_2} \tag{3.3.34}$$

where the over-lines denote the average over both disorder and functional distributions. One finally finds

$$\phi(y) = -\frac{1}{y}\sum_{k=1}^{\infty} e^{-c}\frac{c^k}{k!}\left(A_k + B_k\right)$$

where

$$A_k = \int \prod_{i=0}^{k} \mathcal{D}Q_i \mathcal{Q}[Q_i] \ln(\int \prod_{i=0}^{k} d^q\vec{u}_i Q_i(\vec{u}_i)\ \exp(y\omega(\sum_{i=1}^{k}\vec{u}_i)))$$

and

$$B_k = \frac{c}{2y}\int \prod_{i=1}^{2} \mathcal{D}P_i \mathcal{P}[P_i] \ln(\int \prod_{i=1}^{2} d^q\vec{h}_i P_i(\vec{h}_i) \exp(y\omega(\vec{h}_1) - y\omega(\vec{h}_1 + \hat{u}(\vec{h}_2))))$$

In the limit $y \to \infty$ these relations can be written in a more explicit form. Let us consider first the term $\Delta\phi_1$ in Eq. (3.3.32). Referring to Eq. (3.3.15) it easy to see that:

$$\lim_{y\to\infty} e^{-y\Delta\phi_1} = \sum_{l=0}^{q-1}(-1)^l \binom{q}{l+1} \prod_{i=1}^{k}[1-(l+1)\eta_i] \tag{3.3.35}$$

such that

$$\lim_{y\to\infty} -y\overline{\Delta\phi_1} = \sum_{k=1}^{\infty} e^{-c}\frac{c^k}{k!}D_k \tag{3.3.36}$$

for

$$D_k = \int \prod_{i=1}^{k} d\rho(\eta_i) \ln(\sum_{l=0}^{q-1}(-1)^l \binom{q}{l+1} \prod_{i=1}^{k}[1-(l+1)\eta_i])$$

In order to compute the average link contribution $\overline{\Delta\phi_2(y)}$ we need to evaluate the large y limit of Eq. (3.3.33) which gives:

$$\lim_{y\to\infty} -y\overline{\Delta\Phi_2(y)} = \int d\rho(\eta_1)d\rho(\eta_2)\ln(1-q\eta_1\eta_2) \tag{3.3.37}$$

This equation has a nice probabilistic interpretation complementary to that used in the derivation of ΔE_2 in the RS case. In fact the integrand of (3.3.33) is different from zero for $y \to \infty$ only when both sites i_1 and i_2 have a different color, and this happens with probability $(1-q\eta_{i_1}\eta_{i_2})$ (note that $q\eta_{i_1}\eta_{i_2}$ is the probability that the two sites have same color). It is now clear from Eq. (3.3.22) that taking the $y \to \infty$ of $-y\Phi(y)$ gives us the complexity in the COL region where $e = 0$:

$$\begin{aligned}\Sigma \;=\; & \sum_{k=1}^{\infty} e^{-c}\frac{c^k}{k!}\int \prod_{i=1}^{k} d\rho(\eta_i) \ln(\sum_{l=0}^{q-1}(-1)^l \binom{q}{l+1} \prod_{i=1}^{k}[1-(l+1)\eta_i]) \\ & -\frac{c}{2}\int d\eta_1\rho(\eta_1)d\eta_2\rho(\eta_2)\ln(1-q\eta_1\eta_2)\end{aligned} \tag{3.3.38}$$

3.3.4. Results. The previous analysis results for the q-coloring problem in the existence of a dynamic transition, characterized by the sudden appearance of an exponential number of clusters that disconnect the solutions of the problem. This is represented in figure 3.3.2 for $q = 3$ and 4, where the complexity is plotted as a function of the graph

connectivity. Note, that at a certain value average connectivity $c = c_d$ the complexity abruptly jumps from zero to a positive value. Then it decreases with growing c and disappears at c_q where the number of solutions become zero. It is not possible any more to find a zero-energy ground state for the system, i.e. the graph becomes uncolorable with q colors, and its chromatic number grows by one.

In the following table, we present the results for $q = 3, 4$ and 5, for the dynamical transition we show the corresponding values of c_d and the complexity $\Sigma(c_d)$. For the q-COL/UNCOL transition, only the critical connectivity c_q is given ($\Sigma(c_q) = 0$).

q	c_d	$\Sigma(c_d)$	c_q
3	4.42	0.0223	4.69
4	8.27	0.0553	8.90
5	12.67	0.0794	13.69

In Fig. (3.3.4) we display the average complexity Σ as a function of the energy density e in the 1-RSB approximation. Recently Montanari and Ricci showed in **[67]** that in the p-spin spherical spin glass the 1-RSB scheme is incorrect above a certain critical energy density e_G, where this solution become unstable and a FRSB calculation would be required. It is possible that such a phenomenon might happen also in this case.

The dynamical transition is not only characterized by a sudden clustering of ground states, at the same point an exponential number of meta-stable states of positive energy appears **[58]**. Such states (besides algorithm-dependent entropic barriers which may exist even below c_d) are expected to act as traps for local search algorithms causing an exponential slowing down of the search process. Well known examples of search processes that are overwhelmed by the presence of excited states are simulated annealing or greedy algorithms based on local information.

To test this prediction, we have applied several of the best available solvers for Coloring and SAT problems available in the net **[78, 24]**. After some preliminary simulations we observed that the best results

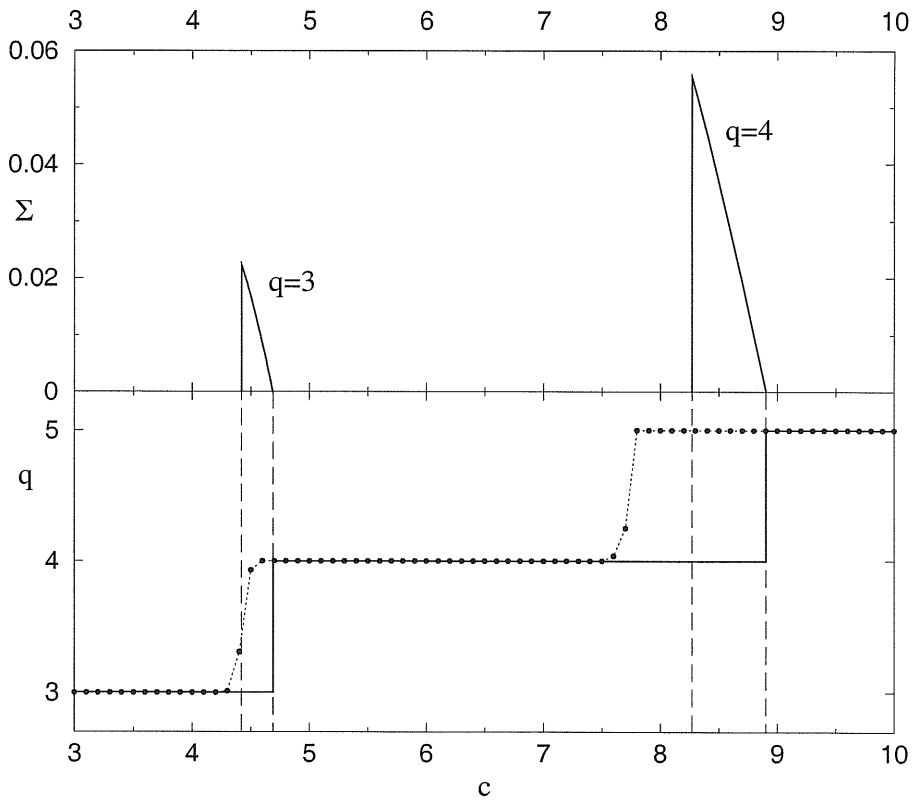

FIGURE 3.3.2. Top: Complexity $\Sigma(c)$ vs. average connectivity for $q = 3$ and $q = 4$. Non-zero complexity appears discontinuously at the dynamical threshold c_d, and goes down continuously to zero at the q-COL/UNCOL transition. The curves are calculated using the population-dynamical solution for $\rho(\eta)$ with population size $\mathcal{M} = 10^6$.
Bottom: The full line shows the chromatic number of large random graphs vs. their connectivity c. The symbols give results of *smallk* for $N = 10^3$, each averaged over 100 samples.

could be obtained with the *smallk* program [**24**] and concentrated our efforts on it. The simulation results, as shown in the lower half of Fig. 3.3.2, were obtained in the following way: First a random graph ($N = 10^3$) was generated and we tried to color it with a small number of colors (here $q = 3$). If, after some cutoff time (we probed with

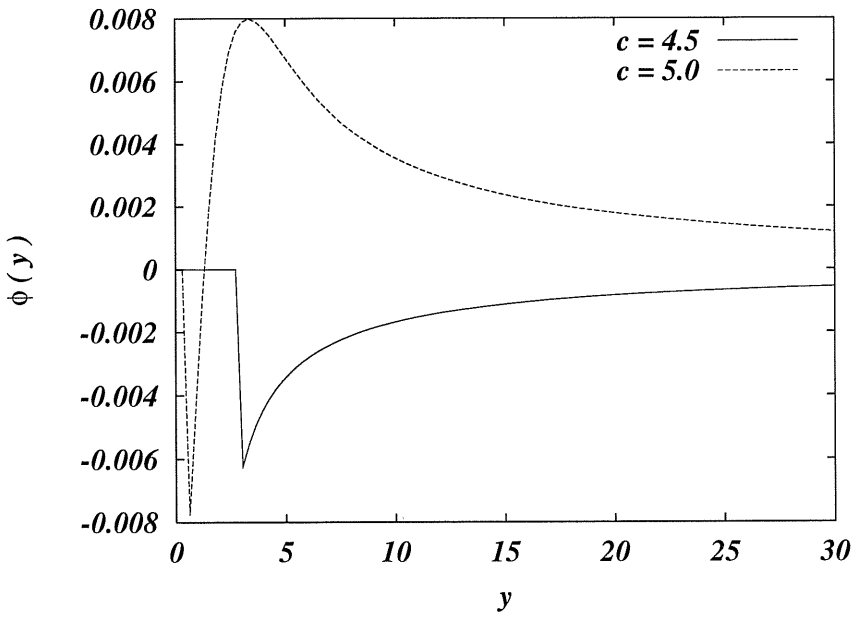

FIGURE 3.3.3. Average thermodynamic potential $\phi(y)$ vs. y in the HARDCOL phase ($c = 4.5$) and in the UNCOL phase ($c = 5.0$). Note that $\phi(y)$ above the paramagnetic region ($\phi = 0$) is a monotonously increasing function of y in the first case, while it displays a maximum at finite y in the second one.

10 seconds, 1 minute and 2 minutes without substantial changes), the graph was not colored, we stop and tried to color it with larger q. For each connectivity we averaged over 100 samples. As it can be clearly seen, the algorithm fails with q colors slightly below the dynamical transition, confirming our expectations. In Sec. (3.3.6) we explain how the cavity approach helps to design an algorithm being able to deal also with this problem.

3.3.5. The large-q asymptotic. From Eqs. (3.3.17) and (3.3.18) one can easily deduce the large-q asymptotics of $\rho(\eta)$. For average connectivities $c >> q$ (the threshold c_q is expected to scale like $\mathcal{O}(q \ln q)$), f_k is dominated by the $l = 0$-contributions in the numerator and in the denominator, leading to $\rho(\eta) = \delta(\eta - 1/q)$ in leading order. Plugging this result into the Eq. (3.3.38) one can easily calculate the

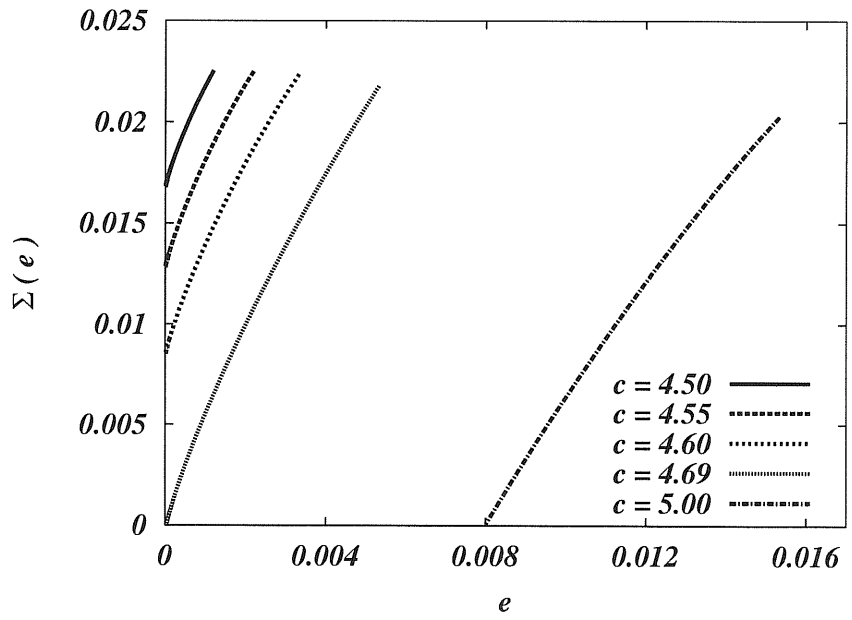

FIGURE 3.3.4. Average complexity Σ as a function of the energy density e for various average connectivities c. In this figure we only display the *physical* branches (see text).

COL/UNCOL threshold c_q by setting the complexity to zero. Taking care only of the dominant contribution we find

$$c_q = 2q \ln q + o(q \ln q) \ . \tag{3.3.39}$$

This result coincides with the exact asymptotics found by Luczak [44]. Note, however, that the same dominant term can also be obtained from the vanishing of the replica-symmetric (paramagnetic) entropy $s(c)$ which is expected to be exact up to the COL/UNCOL transition. This means that, for $q \to \infty$, the threshold entropy goes to zero. This behavior could already be conjectured from the above table where the threshold entropies are given for small q.

3.3.6. Working with single graph instances: Survey propagation. Up to now we have proposed an analytical model to the coloring problem averaged over the set of Erdös-Rényi graphs at a given average connectivity. In this way we derived the q-dependent threshold

connectivities of c_q at which the graph becomes almost surely uncolorable with q colors, i.e. the location of the COL/UNCOL transition. We have also shown the existence of another threshold value c_d above which a clustering phenomenon takes place in the space of solutions.

However, one of the relevant consequences of this cavity approach is that it can be naturally implemented to study single case instances, i.e. specific non-random graphs which have to have a locally tree-like structure to fulfill the conditions of the cavity approach. In the average-case analysis at each step of the iteration, we selected *randomly* k sites from the $\mathcal{M}$ possible ones to be used in Eq. (3.3.16), and we substituted another randomly chosen entry η_0 from the $\mathcal{M}$ possible entries. From here on, we will assume that the iteration procedure used above is also valid for single instances – with one significant change: For the generation of survey for one vertex (or edge) we have to use its actual neighbors, the connections between sites are fixed once for ever by the specific graph under consideration.

3.3.7. The survey propagation algorithm. The SP algorithm resembles formally the sum-product algorithm [**43**]. In the latter, to each vertex arrive u-messages from $k-1$ neighbors, then this messages are transformed (become h-fields) and sent as a new message through the link to the descendant k neighbors. So, at each time step, in the links of the graphs you will have messages traveling, like in a communication network. The survey propagation algorithm (SP), works with the same principle. The basic difference is that now the messages are replaced by $u-$ *surveys* of the messages (i.e. by probability distributions of messages). SP is defined for one given value of the reweighting parameter y that must be optimized to minimize the "free energy" of the system. To each edge $\{i,j\}$ of the graph we associate two u-surveys $Q_{i\to j}(\vec{u})$ and $Q_{j\to i}(\vec{u})$ of messages traveling in the two possible directions. The algorithm self-consistently determines these surveys by a message passing procedure to be described below, and

Algorithm 2 Survey propagation for coloring

(1) Select a graph $G = (V, E)$.
(2) All the $Q_{i\to j}(\vec{u})$ with $\{i,j\} \in E$ are randomly initialized.
(3) We sequentially consider all sites i and randomly update the links $\{i,j\}$ to all neighbors j in the following way:
   (a) For each neighbor j of i we calculate:

$$P_{i|j}(\vec{h}) = C_{i|j} \int \prod_{k\in i\setminus j} d^q\vec{u}_k Q_{k\to i}(\vec{u}_k)\delta(\vec{h} - \sum_{k\in i\setminus j} \vec{u}_k) \exp(y\,\omega(\sum_{k\in i\setminus j} \vec{u}_k))$$

   where with the symbol $V(i)$ denotes all neighbors of i. The prefactor $C_{i|j}$ is chosen such that $P_{i|j}$ is properly normalized to one.
   (b) From $P_{i|j}(\vec{h})$ we derive the new u-surveys of all edges $\{i,j\}$:

$$Q_{i\to j}(\vec{u}) = \int d^q\vec{h}\; P_{i|j}(\vec{h})\delta\left(\vec{u} - \hat{u}(\vec{h})\right)$$

(4) The iteration step 3. is repeated until convergence is reached.

finds consequently all the thermodynamic properties of the model defined on the specific graph. We describe in Algorithm 2 how SP works in practice for the 3-coloring problem.

It was already shown in [58] that the free energy of the system may be written as:

$$\phi(y) = \frac{1}{n}\left[\sum_{\{i,j\}\in E} \phi^{link}_{i,j}(y) - \sum_i (n_i - 1)\phi^{node}_i(y)\right] \tag{3.3.40}$$

where n_i is the connectivity of the vertex i, and $\phi^{link}_{i,j}(y)$ and $\phi^{node}_i(y)$ represent the contributions of links and vertices which are given by:

(3.3.41)

$$\phi^{link}_{i,j}(y) = -\frac{1}{y}\ln\Big(\int d^q\vec{h} P_{i|j}(\vec{h})\; d^q\vec{u} Q_{j\to i}(\vec{u})\;\exp\{-y[\omega(\vec{h}) - \omega(\vec{h}+\vec{u})]\}\Big)$$

and

$$\phi^{node}_i(y) = -\frac{1}{y}\ln\Big(\int \prod_{k\in i} d^q\vec{u}_k Q_{k\to i}(\vec{u}_k)\;\exp\{y\omega\left(\sum_{i=1}^{k} \vec{u}_i\right)\}\Big)\,. \tag{3.3.42}$$

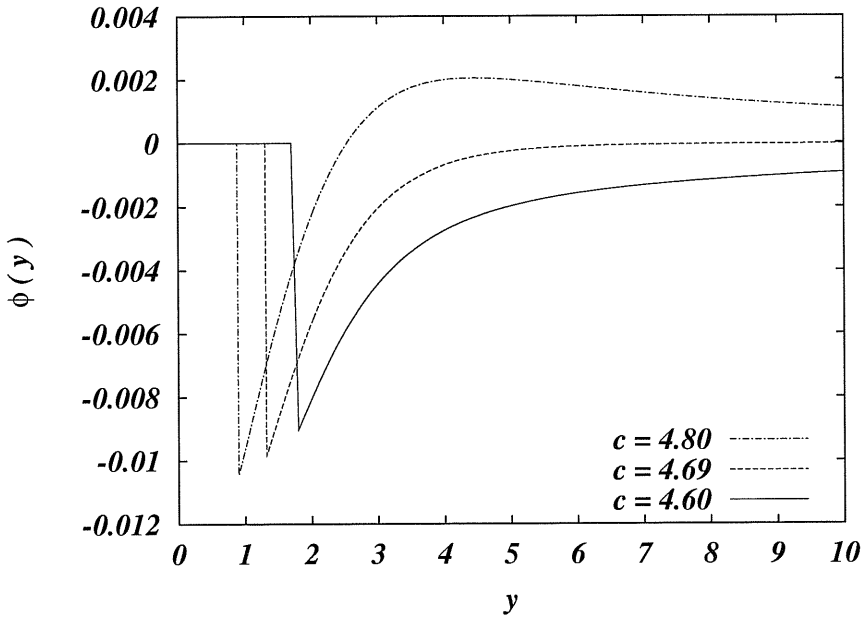

FIGURE 3.3.5. Free energies ϕ as a function of y for three given samples of $N = 10000$ of connectivities $c = 4.60, 4.69, 4.80$.

Repeating the above procedure for various values of y, Eqs. (3.3.41) and (3.3.42) do not only provide the values of $\phi(y)$, but also $\Sigma(y) = -y^2 \partial\phi(y)/\partial y$ and the energy density $e(y) = \partial(y\Phi(y))/\partial y$ of states. The parametric plot of $\Sigma(y)$ versus $\epsilon(y)$ gives the complexity of states as a function of their energy. For example, Fig. (3.3.5) shows the free energy $\phi(y)$ of single graphs with $n = 10000$ vertices as a function of y for three different values of the average connectivity c.

We observe that for high enough connectivities the maximum of $\phi(y)$ is located at finite values of y. While decreasing c, the location of the maximum grows and approaches $y \to \infty$ at the coloring threshold. From these curves and by means of numerical derivatives, we may also calculate the complexity and energy. Fig. (3.3.6) shows the two branches obtained in the parametric plot of $\Sigma(y)$ vs. $e(y)$ for various connectivities c. While the physical meaning of the upper branches is not clear [**57**] we wanted to stress that they interpolate between the RS solution and the maximum complexity point.

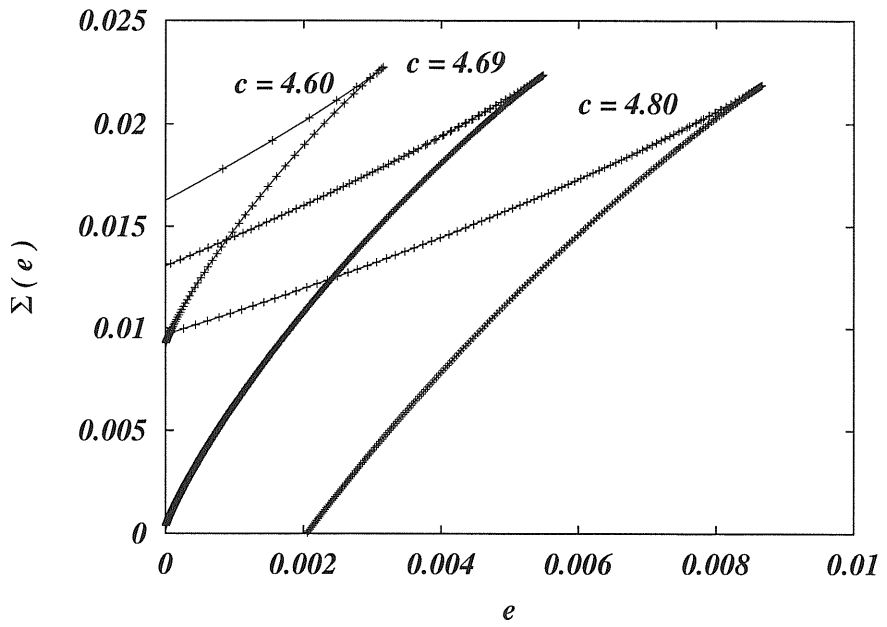

FIGURE 3.3.6. Complexity Σ as a function of ϵ for three given samples of random graph with average connectivities $c = 4.60, 4.69, 4.80$ and $n = 10000$ sites. At odd with Fig. (3.3.4) here we display both physical and unphysical branches.

From the previous figure we may extract two characteristic values of the energy: The first one, is associated with the minimal number $e_0 n$ of miscolored edges in the graph, i.e. it gives the *ground state energy* of the instance. The value of $e_g s$ is determined as the positive point where the lower branch of the complexity curve intersects the energy axis, or it equals zero if $\Sigma(e = 0) > 0$ on the lower branch.

The other relevant energy value is the *threshold energy* e_{th}. It is determined by the point where the complexity reaches its maximum. It is therefore the point where e.g. simulated annealing gets stuck. The same remark of Sec. (3.3.4) holds here: this calculation should be probably improved along the line of [**67**] in order to take into account the FRSB instability at higher energy density as in the case of the p-spin spherical model.

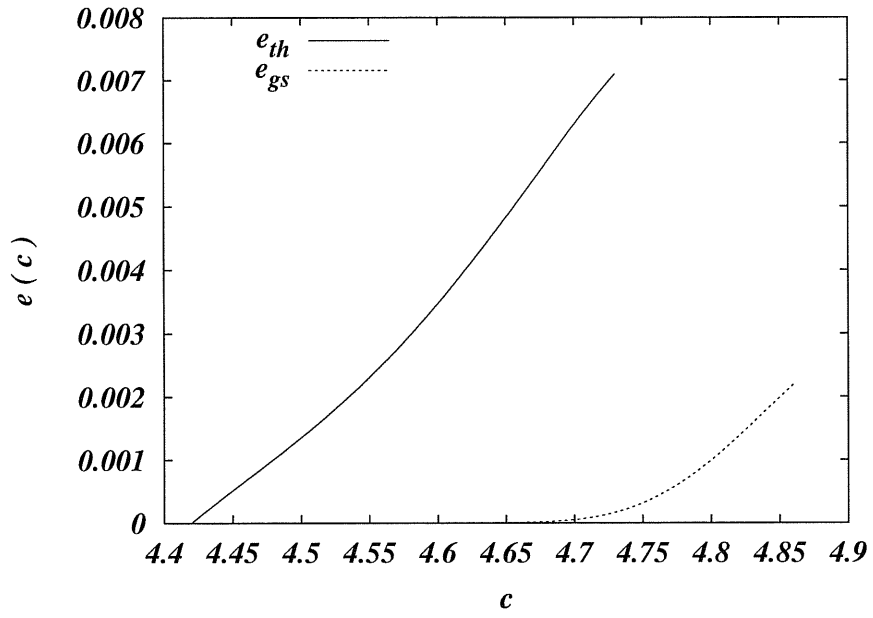

FIGURE 3.3.7. Density of miscolored links e_{gs} vs. average connectivity of the graph c (lower dotted curve) and threshold energy density e_{th} vs. c (upper continuous curve) in the 1-RSB approximation.

From the practical side this is of course not the way to determine this values, it is much more desirable to look for the value of y at which $\phi(y)$ becomes maximal, cf. Eq. (3.3.24). Fig. (3.3.7), shows a plot of these two energies as a function of the connectivities obtained using this single-instance algorithm.

Of course, the exact meaning of the numerical values of these quantities is an open question. In principle they were defined for infinite systems, whereas our single-instance algorithm works for systems of finite sizes n. We expect that the numerical values give good approximations once we look to large values of n, where e.g. the scales dividing distances of solutions inside one state from those between states are well separated. A more detailed discussion about this may be found in [**54, 58**].

3.3.8. A polynomial algorithm to color graphs. The Survey Propagation described above was very useful for the design of an efficient algorithm to find a solution of randomly generated k-SAT formulas [**56, 58**] in the "hard" but satisfiable phase. Here we will show that, with small modifications, the same idea can be extended to the q-coloring problem.

The relevant idea in this algorithm is to fix spins which are strongly biased toward (or away from) one color. Therefore, we have to first determine the distributions of local magnetic fields in the system using SP, and select those which have the strongest bias. Once these are fixed, the problem is reduced. We can rerun SP on the reduced instance, new spins may be biased and fixed. The procedure will be repeated until only paramagnetic spins remain. At this point SP cannot help any more, but surprisingly the decimated coloring problem becomes "easy". Using any reasonable local solver known in the literature, we can proceed to construct a proper coloring.

In the case of q-coloring the subject is technically slightly more complex than in k-SAT, since spins can be biased in q different directions and it is hard to decide what do we mean precisely by biased. In addition, by fixing the color of one vertex, all its neighbors have to have different colors, i.e. they are left with $q-1$ colors. In the reduction process the problem, initially being a pure q-coloring problem, becomes a list coloring problem where each vertex has an own list of allowed colors. In this way the permutation symmetry of colors is broken, which requires a modification of the SP given above to non-symmetric surveys.

In order to keep the presentation as simple as possible we concentrate our efforts on the 3-coloring problem and hence, from now on, all the discussion will be associated for the case $q = 3$. The extension of the results to higher q is, however, straightforward although exponential in q.

As mentioned above, the first things we should do are a generalization of SP to non-color-symmetric situations, and to correctly define a biased spin. Let us start first noting that equation (3.3.7) may be

written as

$$\mathcal{Q}[Q(\vec{u})] = \int d^q\vec{\eta}\ \rho(\vec{\eta})\ \delta\left[Q(\vec{u}) - \eta^0\delta(\vec{u}) - \sum_{\tau=1}^{q}\eta^\tau\delta(\vec{u}-\vec{e}_\tau)\right] \tag{3.3.43}$$

where we simply avoid to consider the color symmetry of the problem, and where we introduce $\eta^0 = (1 - \sum_{\tau=1}^{3}\eta^\tau)$. Then, following the same lines of reasoning that lead from Eq. (3.3.7) to Eq. (3.3.16) we may deduce the following update of the surveys in the limit $y \to \infty$:

$$\eta^r_{i\to j} = \frac{\prod_{k\in i\setminus j}(1-\eta^r_{k\to i}) - \sum_{p\neq r}\prod_{k\in i\setminus j}(\eta^0_{k\to i}+\eta^p_{k\to i}) + \prod_{k\in i\setminus j}\eta^0_{k\to i}}{\sum_p\prod_{k\in i\setminus j}(1-\eta^p_{k\to i}) - \sum_p\prod_{k\in i\setminus j}(\eta^0_{k\to i}+\eta^p_{k\to i}) + \prod_{k\in i\setminus j}\eta^0_{k\to i}} \tag{3.3.44}$$

for $r \in \{1,2,3\}$. The value of $\eta^0_{i\to j}$ can be calculated by imposing the normalization condition. Using this update rule instead of the one proposed in the above version of SP, we directly work with a reweighting parameter $y = \infty$ which forbids any positive energy changes and thus characterizes proper colorings.

Having η^τ_i, for all the sites of the graph, we have to define the site dependent color polarizations

$$\Pi^r_i = \frac{\prod_{j\in i}(1-\eta^r_{j\to i}) - \sum_{p\neq r}\prod_{j\in i}(\eta^0_{j\to i}+\eta^p_{j\to i}) + \prod_{j\in i}\eta^0_{j\to i}}{\sum_{p=1,2,3}\prod_{j\in i}(1-\eta^p_{j\to i}) - \sum_{p=1,2,3}\prod_{j\in i}(\eta^0_{j\to i}+\eta^p_{j\to i}) + \prod_{k\in i}\eta^0_{j\to i}} \tag{3.3.45}$$

for $r = 1,2,3$. This equation is analogous to Eq. (3.3.44) but the products are extended to all neighbors. The polarization Π^r_i is the probability that vertex i is fixed to color r in a randomly selected cluster of solutions. Vertices which may change their color within one cluster are characterized by $\Pi^0_i = (1 - \sum_{r=1}^{3}\Pi^r_i)$.

3.4. Solution of k-XORSAT by the cavity method

In a derivation completely similar than of Sections 3.2, 3.3.1 we give below the 1-RSB solution obtained by the cavity method for the random 3−XOR-SAT problem.

Binary variables allow to represent $u \in \{-1, 0, 1\}$ and $h \in \mathbb{Z}$, and the equation for their distributions are

$$\begin{aligned} Q(u) &= \sum_{J=\pm 1} \sum_{i,j=0}^{\infty} p(i)\, p(j) \int dh P^{(i)}(h)\, dh' P^{(j)}(h')\, \delta(u - u_J(h, h')) \\ P^{(i)}(h) &\propto \int \prod_{j=1}^{i} du_j Q(u_j) \delta(h - \sum_{j=1}^{i} u_j) \exp(-y(\sum_{j=1}^{i} |u_j| - |\sum_{j=1}^{i} u_j|)) \end{aligned}$$

where

$$\begin{aligned} p(i) &= \frac{e^{-3\gamma}(3\gamma)^i}{i!} \\ u_J(h, h') &= sign(Jhh') \quad (sign(0) = 0) \end{aligned}$$

Numerical simulations of the population dynamics equivalent to Algorithm 1 show that cavity biases spontaneously peak in only two categories, $Q(u) \equiv \delta(u)$ and $Q(u) \equiv \frac{\delta(u-1)+\delta(u+1)}{2}$, allowing to parametrize the probability distribution $\mathcal{Q}[Q]$ with only one parameter t corresponding to the probability of the "trivial" $Q(u) \equiv \delta(u)$. A direct inspection of the equations show that obtaining a non-trivial $Q(u)$ is only possible if the $P(h)$ input terns are in term non-trivial. These in term are trivial only when all input $Q(u)$ are trivial, giving the additional constraint (for $k = 3$ here for simplicity)

$$\begin{aligned} 1 - t &= \sum_{i,j=0}^{\infty} (3\gamma)^i (3\gamma)^j \left(1 - t^j\right)\left(1 - t^j\right) \\ &= \left(1 - e^{-3\gamma(1-t)}\right)^2 \end{aligned}$$

For $\gamma < \gamma_d \sim 0.818469$ this equation has only one solution at $t = 1$ (the system is a paramagnet), whereas at $\gamma = \gamma_d$ there appears a non-trivial one.

By the change of variables $\lambda = 3\gamma(1 - t)$, it is possible to write zero-energy ($y = \infty$) complexity as a function of λ

$$\Sigma(\lambda) = \log(2) \left[1 - \frac{\lambda}{3} - e^{-\lambda}\left(1 + \frac{2}{3}\lambda\right)\right] \tag{3.4.1}$$

where λ satisfies the equation

$$\lambda = 3\gamma\left(1 - e^{-\lambda}\right)^2 \tag{3.4.2}$$

The identity $\Sigma = 0$ gives the numerical threshold of $\gamma_c \sim 0.917935$.

3.5. Alternative solution for k-XORSAT

We will reproduce hereafter a schematic description of the exact computation of the critical value for the k-XORSAT problem, to be found in **[59, 23]**. This is one of the few problems that have been completely analyzed rigorously, and the result is in complete agreement with the results given by the cavity approach.

We are interested in computing critical properties of random linear systems

$$B\mathbf{x} = \mathbf{c} \tag{3.5.1}$$

where B is a random binary matrix with distribution $P = P_{k,\gamma}$ and $\mathbf{c}$ is a random uniform binary vector.

3.5.1. First moment. By calling $N = N_{B,\mathbf{c}} \stackrel{\text{def}}{=} |\{\mathbf{x} : B\mathbf{x} = \mathbf{c}\}|$ the number of solutions to this system, the condition of this system to be satisfiable could just be restated as $N \geq 1$. Calling $\mathbb{E}$ the average over the random ensemble and using the fact that $N \geq \delta_{N\geq 1}$ because $N \in \mathbb{N}_0$, we have

$$P(N \geq 1) \leq \mathbb{E}(N) \tag{3.5.2}$$

This is called *first moment method.* Computing $\mathbb{E}(N)$ is fortunately very easy: $N_{B,c} = \sum_{\mathbf{x}} \delta_{B\mathbf{x}=c}$ so

$$\begin{aligned}
\sum_{\mathbf{c}} \sum_{B} P(B)\, 2^{-m} N_{B,c} &= 2^{-m} \sum_{B} P(B) \sum_{\mathbf{x}} \sum_{\mathbf{c}} \delta_{B\mathbf{x}=c} \\
&= 2^{-m} \sum_{B} P(B)\, 2^n \\
&= 2^{n-m}
\end{aligned}$$

For $m = \gamma n$ and $\gamma > 1$ we get that $P_{k,\gamma}(N \geq 1) \overset{n\to\infty}{\to} 0$, i.e. $N = 0$ *almost surely.* That is, the critical value γ_c for satisfiability (if it exists) must be ≤ 1. This is in fact very intuitive: for $\gamma > 1$, there will be *at least* $m - n$ equations linearly dependent on the others, so the value of their correspondent coordinate of $\mathbf{c}$ will be fixed by the others coordinates of $\mathbf{c}$ they depend on: then with probability $1 - 2^{-(m-n)}$ the assigned value of these coordinates will be wrong and the problem will not have any solution.

3.5.2. Second moment. For an opposite bound, we can use the Cauchy-Schwartz inequality $\|N\|_1 = \|N\delta_{N\geq 1}\|_1 \leq \|N\|_2 \|\delta_{N\geq 1}\|_2$ to obtain

$$\frac{\mathbb{E}(N)^2}{\mathbb{E}(N^2)} \leq P(N \geq 1) \tag{3.5.3}$$

This is an application of the *second moment method.* The computation of $\mathbb{E}(N^2)$ is more difficult:

$$\begin{aligned} N_{B,c}^2 &= \sum_{\mathbf{x}} \sum_{\mathbf{y}} \delta_{B\mathbf{x}=c} \delta_{B\mathbf{y}=c} \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{y}} \delta_{B\mathbf{x}=c} \delta_{B(\mathbf{x}-\mathbf{y})=0} \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{z}} \delta_{B\mathbf{x}=c} \delta_{B\mathbf{z}=0} \end{aligned}$$

so:

$$\begin{aligned} \mathbb{E}(N^2) &= 2^{-m} \sum_{B} P(B) \sum_{\mathbf{z}} \delta_{B\mathbf{z}=0} \sum_{\mathbf{x}} \sum_{\mathbf{c}} \delta_{B\mathbf{x}=c} \\ &= 2^{n-m} \sum_{B} P(B) \sum_{\mathbf{z}} \delta_{B\mathbf{z}=0} \qquad (3.5.4) \\ &= 2^{n-m} \sum_{\mathbf{z}} \sum_{B} P(B)\, \delta_{B\mathbf{z}=0} \qquad (3.5.5) \end{aligned}$$

Note that the value of $\mathbb{E}(N)$ doesn't depend at all on the probability distribution $P(B)$ and the computation of $\mathbb{E}(N^2)$ up to Eq. (3.5.5) is independent of the explicit form of $P(B)$. The last term of Eq. (3.5.4) is the average of the number of solutions N_B in homogeneous systems

$B\mathbf{x} = 0$ in the probability space of matrices B with measure $P(B)$. After exchanging the order of the sums, it suffices to count (the probability space is uniform) how many such matrices annihilate a given vector $\mathbf{z}$, and this number $\sum_B \delta_{B\mathbf{z}=0}$ depends only on the number of zeroes of $\mathbf{z}$. Even if a combinatorial expression can be easily obtained for this number for the original distribution of matrices, it seems hard to compute the asymptotic behaviour for such expression. Instead, the combinatorics can be computed also for the uniform probability measure $P(B)$ of matrices with n variables, m equations and no degree-1 nor degree-0 variables (i.e. each variable participates in two or more equations). For the resulting the asymptotic behaviour can be computed (it is a fairly hard computation in [**28**]), and lead to $2^{n(1-\gamma)}$, so the RHS of Eq. (3.5.4) goes asymptotically as $4^{n(1-\gamma)}$. For $\gamma < 1$ then the LHS of Eq. (3.5.3) goes to 1, resulting in $P(N \geq 1) \overset{n\to\infty}{\to} 1$, i.e. the system has almost surely solutions.

Given a random variable R for which $\frac{\mathbb{E}^2 R}{\mathbb{E}R^2} \overset{n\to\infty}{\longrightarrow} 1$ we say that R is "self-averaging". This case of N being self-averaging is a very special situation: the number of solutions of random problems is rarely self-averaging, as the statistics are generally dominated by large deviations tails (improbable systems with huge number of solutions). In this case this is a special property of XOR-SAT with the additional condition $d_0 = d_1 = 0$. Unfortunately our original graph has an asymptotic Poisson distribution of connectivities $d_\ell = \frac{(k\gamma)^\ell e^{-k\gamma}}{\ell!}$ i.e., the condition is not fulfilled.

3.5.3. Leaf removal. Nodes with degree 0 and 1 however affect the number N in a very simple and predictable way. We could "purge" nodes of degree 1 in a preliminar stage with a simple "leaf-removal" algorithm: in each stage a degree 1 node is erased along with its equation (removing this node does not affect the value of N, as the variable is completely determined by the values of the other variables in the removed equation) and the process restarts recursively in the resulting graph.

Fortunately, the probability space $P(B)$ evolves under this algorithm in simple way and can be analyzed with a standard method (See for instance **[3]** for a detailed derivation using this method). Define $D_\ell(T)$ as the number of variables of degree ℓ for a particular graph at step T in this process. It is relatively easy to prove that conditioned to the values of $D_\ell(T)$ the resulting system is uniformly distributed. We then will have

$$\begin{aligned}(3.5.6)\qquad &\mathbb{E}\left(D_\ell(T+1) - D_\ell(T) \mid \{D_i(T)\}\right) = \\ &= \delta_{\ell,0} - \delta_{\ell,1} + (k-1)\left(e_{\ell+1}(T) - e_\ell(T)\right)\end{aligned}$$

where $\mathbb{E}$ denotes expected value due to the random choice of the leaf *and* of the randomness of the ensemble, and e_ℓ is the distribution of degrees for each of the other participant nodes in the removed equation. This distribution is proportional to ℓD_ℓ and so by normalization $e_\ell(T) = \frac{1}{3(m-T)}\ell D_\ell(T)$.

A theorem of Wormald (See Thm. 3.5.1 below) ensures that in the limit $n \to \infty$ the densities $d_\ell = D_\ell/n$ become self-averaging and almost surely equal to the solution of the following ODE system of the degree densities d_ℓ as a function of the reduced time $t = T/n$ derived from Eq. (3.5.6) :

$$\frac{\partial d_\ell(t)}{\partial t} = \delta_{\ell,0} - \delta_{\ell,1} + \frac{(k-1)}{k} \cdot \frac{(\ell+1)\, d_{\ell+1}(t) - \ell d_\ell(t)}{\gamma - t}$$

with initial condition given by the original Poisson distribution,

$$d_\ell(0) = \frac{(k\gamma)^\ell e^{-k\gamma}}{\ell!}$$

Note that strictly speaking, we need the system to be finite dimensional to be able to apply Thm. 3.5.1, but the argument can be made strict by grouping together all d_ℓ for $\ell \geq 2$ as in **[28]**.

The solution of this system reads

$$\begin{aligned}
\lambda(t) &\stackrel{\text{def}}{=} k\left[\gamma(\gamma-t)^2\right]^{\frac{1}{3}} \\
d_\ell(t) &= e^{-\lambda(t)}\frac{\lambda(t)^\ell}{\ell!} \quad \ell \geq 2 \\
d_1(t) &= \lambda(t)\left[e^{-\lambda(t)} - 1 + \left(\frac{\lambda(t)}{3\gamma}\right)^{\frac{1}{2}}\right] \\
d_0(t) &= 1 - \sum_{\ell=1}^{\infty} d_\ell(t)
\end{aligned}$$

In particular for $\ell \geq 2$ the densities coincide with a Poissonian at all times. The dynamic process ends when there are no more degree 1 variable nodes, i.e. at a time $t_f(\gamma)$ defined by the first zero of the condition $d_1 = 0$:

$$e^{-\lambda(t_f)} - 1 + \left(\frac{\lambda(t_f)}{3\gamma}\right)^{\frac{1}{2}} = 0$$

This defines the parameter $\lambda_f(\gamma) = \lambda(t_f(\gamma))$ of the resulting final truncated Poissonian degree densities $d_\ell(t_f(\gamma)) = e^{-\lambda_f(\gamma)}\frac{\lambda_f(\gamma)^\ell}{\ell!}$ for $\ell \geq 2$.

At time t_f, we know that the number of equations of the final system m_f will be $(\gamma - t_f)n$ as we remove just one equation per step. If we disregard nodes of degree 0 in the final system, the number of remaining nodes n_f can be computed as

$$\begin{aligned}
n_f &= n\sum_{\ell=1}^{\infty} d_\ell(t_f) \\
&= n\left[1 - (1+\lambda(t_f))\,e^{-\lambda(t_f)}\right]
\end{aligned}$$

The resulting system has the needed $0,1$-truncated Poissonian degrees densities and an effective $\gamma_f = n_f/m_f$ given by

$$\gamma_f = \frac{1}{\gamma - t_f}\left[1 - (1+\lambda(t_f))\,e^{-\lambda(t_f)}\right]$$

and can be analyzed with the results of the last two sections; with the outcome that the original system is almost surely satisfiable if $\gamma_f(\gamma) <$

1 and almost surely unsatisfiable if $\gamma_f(\gamma) > 1$. The condition $\gamma_f(\gamma_c) = 1$ can be solved numerically, giving a value of $\gamma_c \sim 0.918$. Note that conditions $\gamma_f(\gamma_c) = 1$ and $\Sigma(\gamma_c) = 0$ in Eqs. 3.4.1, 3.4.2 are identical.

THEOREM 3.5.1. [Wormald [85]] Consider a sequence $\{D_i(T)\}_{1\leq i\leq k}$ for $T \in \mathbb{N}_0$ of real random variables such that for $|D_i| \leq Bn$ for a constant B. Let $\mathbf{H}(T)$ be the history of this sequence for $T' \leq T$. Let $f_i : \mathbb{R}_{>0} \times \mathbb{R}^k \to \mathbb{R}$ be Lipschitz functions. Let D be an open domain containing the intersection of $\mathbb{R}_{>0} \times \mathbb{R}^k$ with a neighborhood of $\{(0, d_1, \ldots, d_k) : P(D_i(0) = nd_i) \neq 0 \text{ for some } n\}$ and suppose that for some $m = m(n)$ and for all i and uniformly over all $T < m$ the following two conditions hold:

(1) $\mathbb{E}(D_i(T+1) - D_i(T) | \mathbf{H}(T)) = f_i\left(\frac{T}{n}, \frac{D_1(T)}{n}, \ldots \frac{D_k(T)}{n}\right) + o(1)$
(2) $P\left[|Y_i(T+1) - Y_i(T)| > n^{1/5} | \mathbf{H}(T)\right] = o(n^{-3})$

Then almost surely, $D_i(T) = d_i(T/n) \cdot n + o(n)$ uniformly for $0 \leq T \leq \min\{\sigma n, m\}$ where $\mathbf{d} : [0, \sigma) :\mapsto \mathbb{R}^k$ is the maximally-defined unique solution of the ODE system *inside* domain D

$$\frac{\partial d_i}{\partial t} = f_i(t, d_1, \ldots d_k) \quad 1 \leq i \leq k$$

with initial conditions $d_i(0) = D_i(0)/n$.

3.5.4. Clusters. The method above allows a division of the variable nodes in a formula into two types:

- The core, formed by spins remaining after the leaf removal process
- The non-core, formed of the spins which where removed in the process

Solutions in the core can be thought as *seeds* to a full solution: once the variables in the core have been fixed, the remaining problem has a very simple structure, and all compatible solutions can be recovered in linear time. Moreover, the following has been proved in [59, 23]. Almost surely:

- Hamming distance between two different solutions in the core is $\Omega(n)$.

- Two solutions with the same seed can be joined by a path of solutions with jumps of Hamming distance $O(1)$.

This has an immediate effect in the space of solutions of the original problem: solutions "clusterize", i.e. the set of all solution can be divided into smaller subsets with solutions close to each other inside.

CHAPTER 4

Propagation algorithms for random sparse problems

4.1. Warning propagation

As a sort of warm-up we will re-derive the energy shift-algorithm of Section 3.1 for general problems but with binary variables $x_i \in \{-1, 1\}$, as this will help us to interpret the SP equations for k-SAT later. We will assume $H = \sum_a H_a$ for $H_a \geq 0$, and we are looking for the minimum of H.

On a tree, every variable node i has one parent and (possibly) many children, except for the root node that has no parent. Remember that the *i-subtree* of a variable node i the tree of all descendants of i.

Given a variable node i, call H_i^σ the minimum energy configuration in its subtree, given that $x_i = \sigma$. Let's compute all H_i^σ values from the leaves to the root. Once $H_j^{\pm 1}$ for all j children a, for all children a of i have been computed, the computation of H_i^σ is easy:

$$H_i^\sigma = \min_{\{\sigma_j\}} \sum_{a \text{ child of } i} \left[H_a(\{\sigma_j\}, \sigma) + \sum_{j \text{ child of } a} H_j^{\sigma_j} \right] \tag{4.1.1}$$

Once H_{root} has been computed, $H_{root}^{\sigma_0} = \min\left(H_{root}^{\pm 1}\right)$ will be the minimum achievable energy for the formula and one can go up in the tree, fixing the children variables to the ones that realize the minimum (or just chose one in case of degeneracy of the minimum), and so on iteratively up to the leaves.

Suppose now that we are just interested in a configuration of zero energy (or rather, in knowing if such a configuration exists). Then we can simplify the above algorithm even further, and propagate just $\{1, 0, -1\}$ instead of the energies. Define the following quantity:

$$F_i = -sign\left(H_i^+ - H_i^-\right)$$

(where $sign\,(0) = 0$). If only one of H_i^+, H_i^- is non-zero, then the value of F_i gives the forced value for spin σ_i. If both are zero, then $F_i = 0$ and the spin is free to take any value. If both are non-zero then we know that there is no configuration of zero energy, so we can just give up.

We can easily see that the value of F_i depends just on the values of F_j for j children of node i: by defining

$$F_i^\sigma = \min\left\{ \sum_{a \text{ child of } i} H_a(\{\sigma_j\}, \sigma) : \sigma_j F_j \geq 0 \right\}$$

we will have that $F_i = -sign\left(F_i^+ - F_i^-\right)$. If at some point, both F_i^+ and F_i^- are greater than zero, then the algorithm must stop, and we know that there is no solution to $H = 0$. This algorithm, when applied to general graphs, was called warning propagation in [**17**].

4.2. Belief propagation

Belief propagation (BP) is an algorithm designed to compute some partial information about the probability measure $P(\mathbf{x})$. Precisely, it aims at computing the marginals $P(x_i)$ and $P(\mathbf{x}_a)$. Its definition is very simple and it is exact on a tree, or in a general factor graph if some (generally hard to prove) ad-hoc hypothesis are assumed.

4.2.1. Definition. Given a factorized probability distribution $P \propto \prod_{a \in A} G_a$ (the symbol $\propto$ means as usual that the normalization constant is missing) with a factor graph G, consider the probability distribution

$$P^{(i)}\left(\mathbf{s}^{(i)}\right) \propto \prod_{a \notin i} Q_a\left(\mathbf{s}^{(i)}\right)$$

corresponding to the factor graph $G^{(i)}$ *where variable node* i *and all factor nodes* $a \in i$ *were removed.* We will add the (i) exponent to all mathematical objects corresponding to the graph $G^{(i)}$, i.e. disregarding variable node i and factor nodes $a \in i$. Suppose then that for a given

variable node i, the joint distribution $P^{(i)}\left(\{s_j\}_{j\in a, a\in i}\right)$ is known. Then it's easy to compute the marginal $P(s_i)$:

$$\begin{aligned} P(s_i) &= \sum_{\{s_j\}_{j\neq i}} P(\mathbf{s}) \\ &\propto \sum_{\{s_j\}_{j\neq i}} P^{(i)}\left(\mathbf{s}^{(i)}\right) \prod_{a\in i} Q_a(\mathbf{s}) \\ &\propto \sum_{\{s_j\}_{j\in a\setminus i, a\in i}} \sum_{\{s_j\}_{j\notin a, a\in i}} P^{(i)}\left(\mathbf{s}^{(i)}\right) \prod_{a\in i} Q_a(\mathbf{s}) \\ &\propto \sum_{\{s_j\}_{j\in a\setminus i, a\in i}} P^{(i)}\left(\{s_j\}_{j\in a\setminus i, a\in i}\right) \prod_{a\in i} Q_a(\mathbf{s}) \end{aligned} \tag{4.2.1}$$

One would like to find a *closed* equation relating the same type of quantities on the left and right side; then one can hope to establish some kind of recursion. Eq. (4.2.1) is not closed for two main reasons: the marginal $P^{(i)}\left(\{s_j\}_{j\in a\setminus i, a\in i}\right)$ in the RHS is (a) a much more complex object than the simple marginal $P(s_i)$ in the LHS, and (b) corresponds to a different graph, in which a "hole" has been carved.

In the case of a simply connected factor graph G, in $G^{(i)}$ all nodes $j \in a \setminus i, a \in i$ belong to different connected components, and one has that the marginal $P^{(i)}\left(\{s_j\}_{j\in a\setminus i, a\in i}\right)$ factorizes into $\prod_{j\in a\setminus i, a\in i} P^{(i)}(s_j)$; we are a step forward and (a) is solved and we get

$$P(s_i) \propto \sum_{\{s_j\}_{j\in a\setminus i, a\in i}} \prod_{a\in i} Q_a(\mathbf{s}) \prod_{j\in a\setminus i} P^{(i)}(s_j) \tag{4.2.2}$$

In some non-simply connected graphs, the factorization may still be a reasonable approximation as once the variable node (i), which was the direct connection between two of the variables is removed, their correlation will become typically smaller. As in Section 3.1, this un-correlation hypothesis is central.

Note that for a simply connected G, $P^{(i)}(s_j) = P^{(a)}(s_j)$ for $i, j \in a$, where $P^{(a)}$ corresponds to the graph $G^{(a)}$ in which just factor node a

has been eliminated (the extra elimination of node i and other $b \in i$ is irrelevant because they belong to another connected component once a is removed). Equation 4.2.1 becomes then

$$P(s_i) \propto \sum_{\{s_j\}_{j \in a \setminus i, a \in i}} \prod_{a \in i} Q_a(\mathbf{s}) \prod_{j \in a \setminus i} P^{(a)}(s_j) \tag{4.2.3}$$

Even if (b) has still to be addressed, Eq. (4.2.3) will still be later useful in its own. Point (b) is easy to take care of: we will now correct the left-hand side and instead of computing $P(s_i)$ we will compute a RHS-like quantity. We can remove a function node $a \in i$ from both sides of Eq. (4.2.3), i.e.:

$$P^{(a)}(s_i) \propto \sum_{\{s_j\}_{j \in b \setminus i, b \in i \setminus a}} \prod_{b \in i \setminus a} Q_a(\mathbf{s}) \prod_{j \in b \setminus i} P^{(b)}(s_j) \tag{4.2.4}$$

$$\propto \prod_{b \in i \setminus a} \sum_{\{s_j\}_{j \in b \setminus i}} Q_a(\mathbf{s}) \prod_{j \in b \setminus i} P^{(b)}(s_j) \tag{4.2.5}$$

It is interesting to identify the quantity inside the outer product:

$$E^{(b)}(Q_b | s_i) \stackrel{\text{def}}{=} \sum_{\{s_j\}_{j \in b \setminus i}} Q_b(\mathbf{s}) \prod_{j \in b \setminus i} P^{(b)}(s_j) \tag{4.2.6}$$

is the expected value of Q_b given the value of s_i weighted in $G^{(b)}$. When Q_b is a constrain characteristic function, $Q_b(\mathbf{s}) \in \{0, 1\}$ then $E^{(b)}(Q_b | s_i)$ is the probability of this constrain to be satisfied, given the value of s_i (of course, always on the graph without constrain b).

These are the belief propagation equations. We then substitute $P^{(a)}(s_i)$ by unknowns probabilities called *messages* $\mu_{i \to a}$ and $E^{(b)}(Q_b | s_i)$ by $\mu_{a \to i}(s_i)$ in Eq. (4.2.4) and Eq. (4.2.6) we get

$$\mu_{b \to i}(s_i) \stackrel{\text{def}}{=} \sum_{\{s_j\}_{j \in b \setminus i}} Q_b(\mathbf{s}) \prod_{j \in b \setminus i} \mu_{j \to b}(s_j) \tag{4.2.7}$$

$$L^{bp}_{i \to a}(\mu)(s_i) \stackrel{\text{def}}{=} \frac{1}{z} \prod_{b \in i \setminus a} \mu_{b \to i}(s_i) \tag{4.2.8}$$

but for $L^{bp}_{i\to a}(\mu) = \mu_{i\to a}$ (z is the appropriate normalization scalar). The messages $\mu_{i\to a}$ are probability measures over the finite space of values $s_i \in X_i$ can be parametrized by a normalized vector in $(\mathbb{R}_{\geq 0})^{|X_i|}$; messages $\mu_{a\to i}$ are not normalized and thus can be parametrized by just a vector in $(\mathbb{R}_{\geq 0})^{|X_i|}$. The vector $\{\mu_{i\to a}\}_{a\in i, i\in I}$ of all messages from variable nodes to function nodes will be denoted by μ. Grouping all the functions $L^{bp}_{i\to a}$ in one single vector function we define the map L by:

$$\left\{L^{bp}(\mu)\right\}_{a\to i} \stackrel{\text{def}}{=} L_{a\to i}(\mu)$$

We can now search for a fixed point of the form $L^{bp}(\mu) = \mu$. In the case of tree factor graphs, we have shown that the correct marginals $P^{(a)}(s_i)$ form such a fixed point.

The BP procedure consists in choosing some random initial condition μ_0 and numerically compute

$$\mu^* = \lim_{n\to\infty} \left(L^{bp}\right)^{(n)}(\mu_0)$$

where the (n) exponent above means composition. Numerically, this amounts to compute the sequence inside the limit, defined recursively by $\mu_{t+1} \stackrel{\text{def}}{=} L^{bp}(\mu_t)$ for $t = 0, \ldots, n$ until the value $\|\mu_{t+1} - \mu_t\|_\infty$ becomes smaller than a predetermined small constant ϵ.

Once the fixed point μ^* of L has been computed, their corresponding complete marginals approximations (called beliefs) b can be computed also: $b(s_i)$ can be recovered using Equation 4.2.3, and $b(\mathbf{s}_a)$ can be computed similarly

$$b(s_i) \propto \prod_{a\in i} \mu_{a\to i}(s_i) \tag{4.2.9}$$

$$b(\mathbf{s}_a) \propto \sum_{\{s_i\}_{i\in a}} Q_a(\mathbf{s}) \prod_{i\in a} \mu_{i\to a}(s_i) \tag{4.2.10}$$

If the fixed point μ^* equals to the correct vector of "cavity" partial marginals $\left\{P^{(a)}(s_i)\right\}$ then the beliefs $b(s_i)$, $b(\mathbf{s}_a)$ will be equal to the exact complete marginals $P(s_i)$ and $P(\mathbf{s}_a)$ respectively. We have seen however that convergence to the correct fixed point is only ensured for

tree factor graphs. In the next section we will briefly show explicitly the BP equations in the particular case of graph coloring and restricted graph coloring.

We wish to make a brief comparison between the BP iteration in Eq. (4.2.2) and the energy-shift or warning propagation iteration in Eq. (4.1.1) on a tree. When they can be compared, i.e. when we are addressing the problem of finding a zero ground-state in BP (i.e., when we apply BP to the problem of computing $P \propto \prod_a (1 - H_a)$ for $H_a(\cdot) \in \{0, 1\}$), we can easily see that the warning propagation equations are in a sense a simplification of the BP ones: if one looks simply at whether $P(s_i) \neq 0$ in Eq. (4.2.2), we see that this depends only on the status of $P^{(a)}(s_j) \neq 0$ for neighbors s_j. This "discretization" results in Eq. (4.1.1).

4.2.2. BP for coloring. In the case of the coloring problem, given $G = (I, A)$ and a coloring configuration $\mathbf{c} \in \{1, \dots q\}^I$, we recall that $E_{(i,j)} = \delta_{c_i c_j}$ and so $E = \sum_{(i,j)\in A} \delta_{c_i c_j}$ and the Boltzmann measure is $P = \prod_{(i,j)\in A} e^{-\beta \delta_{c_i c_j}}$.

The BP equations reduce to

$$\mu_{i\to j}(c_i) \propto \prod_{(ki)\in A\setminus(ji)} \sum_{c_k} e^{-\beta\delta_{c_k c_i}} \mu_{k\to i}(c_k) \tag{4.2.11}$$

The equation for $\beta = +\infty$ can be simplified a bit further: $e^{-\beta\delta_{c_k c_i}} = 1 - \delta_{c_k c_i}$, so

$$\begin{aligned} \mu_{i\to j}(c_i) &\propto \prod_{(ki)\in A\setminus(ji)} \sum_{c_k} (1 - \delta_{c_k c_i})\, \mu_{k\to i}(c_k) \\ &\propto \prod_{(ki)\in A\setminus(ji)} \sum_{c_k \neq c_i} \mu_{k\to i}(c_k) \end{aligned} \tag{4.2.12}$$

Which has a simple interpretation: the probability that node i takes a given color c_i is proportional to the probability that this color is not already taken by one of the neighbors of node i (note that "proportional" is still needed, as the latters are not mutually exclusive). The

missing normalization term amounts to

$$\sum_{p} \prod_{(ki)\in A\setminus(ji)} \sum_{r\neq p} \mu_{k\to i}(r) \tag{4.2.13}$$

where $p, r \in \{1, \ldots, q\}$.

Note that this simplified expression of the BP equations may have singular points, i.e. the normalization term Eq. (4.2.13) may be 0. One can obtain a well defined expression by taking correctly the limit of $\beta \to \infty$ in Eq. 4.2.11, the resulting expression will be well-defined for all μ, but unfortunately will be discontinuous in the old singular points. In many cases this second expression is not wanted anyway, as a solution corresponding to those singular points corresponds to $H > 0$, i.e. an uncorrect coloring.

The reader should note that the Equations 4.2.11, 4.2.12 are color-symmetric, i.e. if all input probabilities μ in the right-hand side are uniform among colors, then clearly the output must be also uniform. Then $\mu_{k\to i} = \left(\frac{1}{q}, \ldots, \frac{1}{q}\right)$ for all $(k, i) \in A$ is a fixed point of the equations, giving marginal beliefs $b(c_i) = \frac{1}{q}$ for all $i \in I$ and $c_i \in \{1, \ldots, q\}$. In fact, this is the correct solution! For the coloring problem in any graph there is a color symmetry, every marginal cannot be anything else than uniform among colors. Of course, this doesn't help us at all in solving the coloring problem: to this extent one expects to be able to compute also marginals for a partially colored graph. Fortunately, it is very easy to generalize the BP equations to the *restricted coloring* problem, as we just have to restrict the range of c_i variables:

$$\mu_{i\to j}(c_i) \propto \chi_{\{c_i\in Q_i\}}(c_i) \prod_{(ki)\in A\setminus(ji)} \sum_{c_k\in Q_k} e^{-\beta\delta_{c_k c_i}} \mu_{k\to i}(c_k)$$

and of course the correspondent $\beta = +\infty$ version is

$$\mu_{i\to j}(c_i) \propto \chi_{\{c_i\in Q_i\}}(c_i) \prod_{(ki)\in A\setminus(ji)} \sum_{c_k\in Q_k\setminus c_i} \mu_{k\to i}(c_k)$$

4.2.3. Some known properties of BP for trees. The situation for trees is somewhat optimal:

THEOREM 4.2.1. *The BP equations are exact for trees.*

PROOF. We have already proved this in Section 4.2.1. □

COROLLARY 4.2.2. *There exist at least one fixed point for the BP equations in a tree*

PROOF. The correct marginals satisfy the equations and so are a fixed point. □

THEOREM 4.2.3. *For any tree, there is vector μ^* such that the BP equations converge exactly to μ^* from any initial condition in a finite number of steps.*

PROOF. Thanks to the special tree topology, we can separate messages $\mu_{a\to i}$ in two types: "upstream" ones (from leaves to the root), with $a < i$ and "downstream" ones (from the root to the leaves) with $a > i$. Let's look first at upstream messages:

We label nodes by a "distance to the leaves" function $e : I \to \mathbb{N}$ in this way: start with $k = 1$ and for all the leaves i put $e(i) = k$. Then remove all leaves from the tree, increase k by one and repeat. The procedure stops when the remaining subgraph is void and all nodes have been labeled. It is easy to see that e is an increasing function: if $i < j$ then $e(i) < e(j)$. This procedure is similar to the "leaf removal algorithm" used in other contexts.

It is easy to see from Eq. (4.2.4) that "upstream" messages only depend on children "upstream" messages. It is trivial then to prove by induction the following: after a number of steps k greater than n the value $\mu^k_{i\to a}$ for $a > i$ and $e(i) < n$ will remain constant (and this constant is independent of the initial condition).

So after a number of steps k greater to the depth of the tree D, all upstream $\mu^k_{i\to a}$ remain constant. Now we can repeat a similar argument but going downstream, using the "depth" d instead of e. Using the fact that downstream messages depend on upstream messages and parent downstream messages only, we can easily prove by induction that after a number of steps k greater than $D + n$ all downstream messages $\mu^k_{i\to a}$ for $a < i$ and $d(i) < n$ will remain constant. Putting together both results, we get that after a number of $2D$ steps or greater, all involved

quantities will remain constant (and the constant is independent from the initial condition). □

Proposition 4.2.3 has an easy but interesting consequence:

COROLLARY 4.2.4. *For tree factor graphs, the BP fixed point is unique.*

PROOF. Trivial because of Theorem 4.2.3. □

4.2.4. Known results for general graphs. Most results in this section come from [88].

In the following, we will assume $\mathcal{F} = \prod_{a \in A} Q_a$. with $Q_a : X \mapsto \mathbb{R}_{>0}$ for reasons that will be clear later. Constrains of this type will be called "soft" constrains. As always, $P = \frac{1}{Z} \prod_{a \in A} Q_a$.

THEOREM 4.2.5. *Let b be a probability measure approximation to P. Then $b \equiv P$ if and only if b achieves the minimum of the functional G:*

$$G(b) \stackrel{\text{def}}{=} \sum_{\mathbf{x}} b(\mathbf{x}) \ln b(\mathbf{x}) - \sum_{\mathbf{x}} b(\mathbf{x}) \ln \mathcal{F}(\mathbf{x})$$

Moreover, this minimum value is $-\ln Z$.

PROOF. This is a standard variational argument: note first that

$$G(b) = \sum_{\mathbf{x}} b(\mathbf{x}) \ln b(\mathbf{x}) - \sum_{\mathbf{x}} b(\mathbf{x}) \ln P(x) - \ln Z \tag{4.2.14}$$

and that G is continuous in its range and C^∞ for $b > 0$. Taking $t \in [0,1]$ and the curve of interpolating measures

$$c_t(\mathbf{x}) = (1-t) P(\mathbf{x}) + t b(\mathbf{x})$$

we have trivially that $c_t > 0$ for $t < 1$. Consider $f : [0,1] \mapsto \mathbb{R}$, $f \in C^\infty[0,1) \cap C^0[0,1]$ defined by $f(t) = G(c_t)$. It is easy to check that $f(0) = -\ln Z$, $f(1) = G(b)$. Computing derivatives we get $f'(0) = 0$ and $f''(t) \geq 0$ for $t \in [0,1)$and so f is increasing and the result holds. □

The functional G in equation 4.2.14 is called Gibbs free energy in [88]. It is composed of two terms: the first one is the negative of the entropy of b (see Section (2.5)), and the second is the b-average of $-\ln \mathcal{F}$. When $\mathcal{F} = e^{-\beta E}$, the second term is then the average of the energy multiplied by β (see Eq. 2.4.1). Theorem (4.2.5) implies that the probability P is the one that simultaneously minimizes its average energy and maximizes its entropy; one should keep this in mind when proposing approximating candidates for P. Moreover, the fact that the correct P is the one that minimizes G shows one way to obtain approximations b for it: propose a simple functional form approximation b and then fix all parameters by minimizing G.

The following result however is based on a slightly different approach: minimize an approximation to G. Let's define the following approximation to G, called *Bethe free energy:*

$$F_{Bethe} \stackrel{\text{def}}{=} S + U \tag{4.2.15}$$

where

$$S = -\sum_a \sum_{\mathbf{x}_a} b(\mathbf{x}_a) \ln b(\mathbf{x}_a) + (n_i - 1) \sum_i \sum_{x_i} b(x_i) \ln b(x_i)$$

and

$$U = -\sum_a \sum_{\mathbf{x_a}} b(\mathbf{x}_a) \ln Q_a(\mathbf{x}_a)$$

Note that S and U are given by Eq. (2.6.2) and Prop. (2.6.3), both of them evaluated on a set of independent variables $b(\mathbf{x}_a)$ and $b(x_i)$.

THEOREM 4.2.6. *An interior local stationary point of the Bethe free energy, subject to the normalization conditions* $b(x_i) = \sum_{\{x_j\}_{j \in a \setminus i}} b(\mathbf{x_a})$ and $\sum_{\mathbf{x}_a} b(\mathbf{x}_a) = 1$ are the computed beliefs *Eqs. (4.2.9),(4.2.10)* of a fixed point of the BP equations.

PROOF. We add to Eq. (4.2.15) the Lagrange multipliers $\nu_{a\to i}(x_i)$, γ_a and γ_i ensuring normalization, giving the terms

$$M = \sum_i \sum_{x_i} \nu_{a\to i}(x_i) \cdot \left(b(x_i) - \sum_{\{x_j\}_{j\in a\setminus i}} b(\mathbf{x_a}) \right)$$

$N = \sum_a \gamma_a \cdot \left(1 - \sum_{x_a} b(\mathbf{x}_a)\right)$ and $L = \sum_i \gamma_i \left(1 - \sum_{x_i} b(x_i)\right)$. Then we analyze the condition

$$\nabla (S + U + M + N + L) = 0$$

Conditions $\frac{\partial}{\partial \nu_{a\to i}(x_i)} = 0$, $\frac{\partial}{\partial \gamma_a} = 0$ and $\frac{\partial}{\partial \gamma_i} = 0$ give of course the normalization constraints. Then $\frac{\partial}{\partial b(\mathbf{x}_a)} F_{Bethe} = \ln Q_a(\mathbf{x}_a) - \ln b(\mathbf{x}_a) - 1 - \sum_{i\in a} \nu_{a\to i} - \gamma_a$ so the condition $\frac{\partial}{\partial b(\mathbf{x}_a)} = 0$ gives the equation

$$b(\mathbf{x}_a) \propto Q_a(\mathbf{x}_a) \prod_{i\in a} e^{\nu_{a\to i}} \tag{4.2.16}$$

The condition $\frac{\partial}{\partial b(x_i)} = 0$ for the the last derivative

$$\frac{\partial}{\partial b(x_i)} F_{Bethe} = (n_i - 1)(\ln b(x_i) + 1) + \sum_{a\in i} \nu_{a\to i}(x_i) + \gamma_i$$

gives

$$b(x_i) \propto \prod_{a\in i} (e^{\nu_{a\to i}})^{\frac{1}{1-n_i}} \tag{4.2.17}$$

Eq. 4.2.16, 4.2.17 plus the normalization constrains are equivalent to the BP equations for $\mu_{a\to i} = e^{\nu_{a\to i}}$. □

This proof is reversible. Reciprocally,

THEOREM 4.2.7. *If a vector μ is a fixed point of the BP equations such that the beliefs $b(x_i)$, $b(\mathbf{x}_a)$ are positive then they form a local (interior) stationary point of the Bethe Free energy (subject to the normalization constrains).*

This result suggest an alternative way of obtaining BP fixed points: find local stationary points of F_{Bethe} just by minimizing it directly, for instance by a gradient descent method.

It can be easily seen that if $\mathcal{F} > 0$ then local minima of F_{Bethe} must be interior points. This has been analyzed in [84], and the following stronger fact is proved (by an approximate-convexity argument):

THEOREM 4.2.8. *If $\mathcal{F} > 0$, there is only one minimum of F_{Bethe} subject to the normalization constrains, and this minimum is interior (located strictly inside the bounds $0 < b(x_i), b(\mathbf{x_a}) < 1$)*

PROOF. See [84]. □

This theorem in particular proves that there exists at least one fixed point of the BP equations when $\mathcal{F} > 0$, and proves also that the gradient descent method is guaranteed to converge.

THEOREM 4.2.9. *For $\mathcal{F} > 0$, there exists at least one fixed point of the BP equations.*

PROOF. An alternative proof to the one above is by Brower's fixed point theorem. For $\mathcal{F} > 0$ the Equations 4.2.2 define a continuous function $L : \oplus_{b \in B} M(X_b) \mapsto \oplus_{b \in B} M(X_b)$ where $B = \{a \to i : a \in A, i \in I\}$, X_b is a finite set and $M(X_b)$ is the space of probability measures over X_b. As $M(X_b)$ is homeomorphic to the power of the closed unit ball $D^{|X_b|-1}$, then $\oplus_{b \in B} M(X_b)$ is homeomorphic to D^α for some $\alpha \in \mathbb{N}$, and L has a fixed point by Brower's theorem. □

The case when $\mathcal{F}$ is allowed to take value 0 is still most interesting, and unfortunately there is no proof of existence of the fixed point.

The two conjectures can be found in [88].

CONJECTURE 4.2.10. *BP fixed points with some beliefs equal to zero are local stationary points of the Bethe free energy subject to all normalization constrains* and *subject to the condition that those beliefs are zero*

CONJECTURE 4.2.11. *Local minima of the Bethe free energy with some b equal to zero subject to the condition that those b are zero and normalization constraints are BP fixed points.*

However, although certainly desirable, we don't think that a proof of existence is of particular urgency, because a proof of "accuracy" of the equations for $\mathcal{F} > 0$ will give a provable method to approximate P by means of the approximate measures P_β.

4.3. Survey propagation

The survey propagation algorithm is in a sense a mixture of an application of the discrete warning propagation (WP) algorithm of Section 4.1 to general graphs with the BP algorithm. The underlying intuition is that in random combinatorial problems in the hard region, the solution space breaks into many separated subgroups "clusters" of solutions. If we can restrict ourselves to some cluster, the substructure is simple enough that WP would be able to handle it, but still WP cannot deal with the existence of different clusters and falls into contradictions. In some sense, SP is a BP algorithm that tries to ignore fluctuations coming from near solutions, i.e. inside clusters.

4.3.1. SP for k-SAT. For simplicity we will deal first with the k-SAT problem. Remember that $Q_a(\mathbf{s}) \in \{0,1\}$. Consider the BP equations (4.2.4) in this equivalent form:

$$P^{(a)}(s_i) \propto \sum_{\{s_j\}_{j\in b\setminus i, b\in i\setminus a}} \prod_{b\in i\setminus a} Q_b(\mathbf{s}) \prod_{j\in b\setminus i} P^{(b)}(s_j)$$

For every configuration of $\{s_j\}_{j\in b\setminus i}$ two things can happened: either s_i is forced to take one value by clause b (that is, the opposite value of s_i is canceled by $Q_b(\mathbf{s})$) or it's not. Let us code these outcomes with a function

(4.3.1)

$$\hat{u}_{b\to i}(\{s_j\}_{j\in b\setminus i}) \stackrel{\text{def}}{=} \begin{cases} -1 & \text{if } s_i \text{ is forced to take the value } -1 \text{ by } Q_b \\ 0 & \text{if } s_i \text{ is not forced by } Q_b \\ 1 & \text{if } s_i \text{ is forced to take the value 1 by } Q_b \end{cases}$$

An algebraic expression of $\hat{u}$ for k-SAT can be easily obtained:

$$\hat{u}_{b\to i}(\{s_j\}_{j\in b\setminus i}) = J_{b,i} \prod_{j\in b\setminus i} \delta(s_j; -J_{b,j}) \tag{4.3.2}$$

When considering all $b \in i \setminus a$ and all variables $\{s_j\}_{j \in b \setminus i, b \in i \setminus a}$, three things can happen for s_i: it can be forced to take one value by some clauses and free for the others, it can be free for all clauses, or it can be forced to take two contradictory values for two different clauses. As side note, it is easy to see that BP equations for binary variables depend only on this outcome and not directly on the s_j values.

The last of the tree outcomes is of course undesirable, and such configurations $\{s_j\}_{j \in b \setminus i, b \in i \setminus a}$ must be filtered out. For the other outcomes, one can define a $\{-1, 0, 1\}$-valued function $\hat{h}$ as

(4.3.3)

$$\hat{h}_{i \to a}(\{s_j\}_{j \in b \setminus i, b \in i \setminus a}) \stackrel{\text{def}}{=} \begin{cases} -1 & \text{if } s_i \text{ is forced to take the value } -1 \\ 0 & \text{if } s_i \text{ is not forced} \\ 1 & \text{if } s_i \text{ is forced to take the value } 1 \end{cases}$$

Note that $\hat{h}$ can be expressed as a function of the $\hat{u}$ as

$$\hat{h}_{i \to a} = sign\left(\sum_{b \in i \setminus a} \hat{u}\right)$$

and the condition to be non-contradictory is

$$\chi\left(\{\hat{u}_{b \to i}\}_{b \in i \setminus a}\right) = 1$$

for the function χ defined as

$$\chi\left(\{u_b\}\right) \stackrel{\text{def}}{=} \delta\left(|\sum_b u_b| - \sum_b |u_b|\right)$$

The idea is to switch now to the propagation of statistics of three-state variables h that reflect the $\hat{h}$ outcome for non-contradictory configurations:

$$P^{(a)}(h_i) \propto \sum_{\{h_j\}} \delta(h_i - \hat{h}_{i \to a}) \prod_{b \in i \setminus a} \chi\left(\{\hat{u}_{b \to i}\}_{b \in i \setminus a}\right) \prod_{j \in b \setminus i} P^{(b)}(h_j) \tag{4.3.4}$$

where the sum runs over $\{h_j\}_{j \in b \setminus i, b \in i \setminus a}$ and the arguments of the functions $\hat{h} = \hat{h}_{i \to a}(\{h_j\}_{j \in b \setminus i, b \in i \setminus a})$ and $\hat{u} = \hat{u}_{b \to i}(\{h_j\}_{j \in b \setminus i})$ have been elided for shortness.

Of course one has still to define what Eqs. 4.3.1, 4.3.3 mean when some of the s_j are 0. In the case of k-SAT this will be very simple: $\hat{u}$ will be zero whenever any of the s_j is 0 (coinciding with the expression already given in Eq. 4.3.2). These are the SP equations for $k-$SAT. They can be split as a coupled system

$$Q_{b\to i}(u) = \sum_{\{h_j\}_{j\in b\setminus i}} \delta\left(u - \hat{u}_{b\to i}\right) \prod_{j\in b\setminus i} P_{j\to b}(h_j)$$

$$P_{i\to a}(h) \propto \sum_{\{u_b\}_{b\in i\setminus a}} \delta(h - \mathrm{sign}(\sum_{b\in i\setminus a} u_b))\chi\left(\{u_b\}\right) \prod_{b\in i\setminus a} Q_{b\to i}(u_b)$$

Where we have introduced the intermediate probability measures $Q_{b\to i}(u)$ and have denoted $P^{(a)}(h_i)$ by $P_{i\to a}$. This equations simplify substantially: if for every $j \in a \setminus i$ we parametrize

$$\begin{aligned} Q_{b\to i}(u) &= \sum_{\sigma=-1,0,1} \eta^{\sigma}_{b\to i}\delta(u-\sigma) \\ P_{i\to a}(h) &\propto \sum_{\sigma=-1,0,1} \Pi^{\sigma}_{i\to a}\delta(h-\sigma) \end{aligned}$$

Then the equation for $P_{i\to a}$ reduce to

$$\Pi^{\pm}_{j\to a} = \prod_{b\in j\setminus a}\left(1-\eta^{\mp}_{b\to j}\right) - \rho\prod_{b\in j\setminus a}\eta^{0}_{b\to j} \tag{4.3.5}$$

$$\Pi^{0}_{j\to a} = \rho\prod_{b\in j\setminus a}\eta^{0}_{b\to j} \tag{4.3.6}$$

where the new introduced parameter ρ must be set to 1. It is interesting to note that the same equations for $\rho = 0$ are equivalent to the BP equations for SAT. Intermediate values of the interpolation parameter ρ give a mixed SP/BP algorithm that has experimentally found to be useful in some cases (See Appendix A).

Then the equations become closed with the equation for $Q_{a\to i}$,

$$\eta_{a\to i}^{J_{a,i}} = \prod_{j\in a\setminus i}\left[\frac{\Pi_{j\to a}^{-J_{a,j}}}{\Pi_{j\to a}^{+}+\Pi_{j\to a}^{-}+\Pi_{j\to a}^{0}}\right] \tag{4.3.7}$$

$$\eta_{a\to i}^{0} = 1-\eta_{a\to i}^{J_{a,i}} \tag{4.3.8}$$

$$\eta_{a\to i}^{-J_{a,i}} = 0 \tag{4.3.9}$$

For the computation of the "local fields" $P(h_i)$ (i.e. for the complete graph) we have a similar equation to the one of $P_{i\to a}$, except that we take into account all neighbors:

$$\hat{\Pi}_j^{\pm} \stackrel{\text{def}}{=} \prod_{b\in j}\left(1-\eta_{b\to j}^{\mp}\right)-\rho\prod_{b\in j}\eta_{b\to j}^{0} \tag{4.3.10}$$

$$\hat{\Pi}_j^{0} \stackrel{\text{def}}{=} \rho\prod_{b\in j}\eta_{b\to j}^{0} \tag{4.3.11}$$

then for $h_i\in\{-1,0,1\}$ the complete local fields can be computed by normalizing $\hat{\Pi}$ as

$$P(h_i) = \frac{\hat{\Pi}_{j\to a}^{h_i}}{\hat{\Pi}_{j\to a}^{+}+\hat{\Pi}_{j\to a}^{-}+\hat{\Pi}_{j\to a}^{0}} \tag{4.3.12}$$

Note that when there are no 1-clauses, these equations have always a trivial fixed point, namely $\eta_{a\to j}^{s}=\delta(s)$ for all $j\in b$ and $b\in A$.

Another thing which is useful to note for a practical implementation of SP for $k-$SAT is that as by Eq. (4.3.9) $\eta_{a\to i}^{-J_{a,i}}$ is always zero (a clause cannot force a variable to *not* satisfy it) and $\eta_{a\to i}^{0}$ can be recovered by normalization, it is sufficient with only one parameter to describe the probabilities $Q_{a\to i}$. That is, $Q_{a\to i}(u)=\nu_{a\to i}\delta(u-J_{a,i})+(1-\nu_{a\to i})\delta(u)$ for $\nu_{a\to i}=\eta_{a\to i}^{J_{a,i}}$.

4.3.1.1. *Complexity.* The *complexity* (or logarithm of the number of "clusters") of the SP solution for $k-$SAT, Eq.(25-27) in[**17**], obtained with a computation at all similar to Section 3.3.3, has the following form

$$\Sigma \stackrel{\text{def}}{=} \sum_{a\in A}\Sigma_a-\sum_{i\in I}(n_i-1)\Sigma_i \tag{4.3.13}$$

where

$$\Sigma_a \stackrel{\text{def}}{=} \log\left[\prod_{j\in a}\left(\Pi^{+}_{j\to a}+\Pi^{0}_{j\to a}+\Pi^{-}_{j\to a}\right)-\prod_{j\in a}\Pi^{-J_{a,j}}_{j\to a}\right]$$

$$\Sigma_i \stackrel{\text{def}}{=} \log\left[\hat{\Pi}^{+}_{i}+\hat{\Pi}^{0}_{i}+\hat{\Pi}^{-}_{i}\right]$$

4.3.2. SP for coloring. A similar reasoning can be done to derive the SP equations for coloring used in [**16**]. Given a state of all neighbors $\{c_{j\to i}\}_{j\in i\setminus k}$ in the graph $G^{(k)}$ one wants determine which of the following three conditions holds:

(1) Variable c_i is *forced* to take some specific value p in order to satisfy all neighboring constrains
(2) Variable c_i is *free* to chose between two or more values
(3) The input configuration is contradictory, there exists no possible value of c_i which would satisfy all neighboring constrains simultaneously.

Let's call C the set $\{c_{j\to i} : j \in i \setminus k\}$. Clearly variable c_i will be forced to take value p (case 1) if and only if $C = \{1,\dots,q\}\setminus\{p\}$; the configuration $\{c_{j\to i}\}_{j\in i\setminus k}$ will be contradictory (case 3) if $C = \{1,\dots,q\}$ and the variable will be free (case 2) in all remaining cases.

The states of $c_{i\to k}$ will be coded by $q+1$ values $\{0,1,\dots,q\}$ where 0 will mean that the variable is *free*. If we assume that *free* neighbors do not force any constrain on $c_{i\to k}$, the conditions for case (1) above becomes

$$\{1,\dots,q\}\setminus p \subset C \subset \{0,1,\dots,q\}\setminus\{p\} \tag{4.3.14}$$

and the one for case (3) become

$$\{1,\dots,q\} \subset C \tag{4.3.15}$$

We just need now to write the equations for the probabilities of the above outcomes as always assuming independence between $\{c_{j\to i}\}_{j\in i}$. We will parametrize the probabilities as follows $\eta^{p}_{j\to i} \stackrel{\text{def}}{=} P\left(c_{j\to i}=p\right)$.

To fix ideas, let's restrict ourselves to the case of 3-coloring. For $p = 1, 2, 3$, after a quick examination of Eqs. 4.3.14, 4.3.15 we get (the r subindex run also over $1, 2, 3$):

(4.3.16)

$$\eta^p_{i\to j} = \frac{\prod_{k\in i\setminus j}(1-\eta^p_{k\to i}) - \sum_{r\neq p}\prod_{k\in i\setminus j}(\eta^0_{k\to i}+\eta^r_{k\to i}) + \prod_{k\in i\setminus j}\eta^0_{k\to i}}{\sum_r\prod_{k\in i\setminus j}(1-\eta^r_{k\to i}) - \sum_r\prod_{k\in i\setminus j}(\eta^0_{k\to i}+\eta^r_{k\to i}) + \prod_{k\in i\setminus j}\eta^0_{k\to i}}$$

which is identical to Eq. (3.3.44). This equation can be easily explained: first term in the numerator is the condition that no neighbor takes color p, but the cases in which more than one color is not taken must be substracted (two last terms) as they leave the variable free. The denominator is simply the inclusion-exclusion principle to compute the probability of non-contradiction as the probability of the non-disjoint union of the sets

$$\cup_p \{ \text{ color } p \text{ has not been taken}\}$$

Then $\eta^0_{i\to j}$ can be recovered by normalization, i.e.

$$\eta^0_{i\to j} = 1 - \sum_{p\in\{1,2,3\}} \eta^p_{i\to j}$$

To compute the "complete" local fields, the equation is completely similar, but we just consider *all* neighbor nodes of i. For $p = 1, 2, 3$ we get

(4.3.17)

$$\Pi^p_i = \frac{\prod_{k\in i}(1-\eta^p_{k\to i}) - \sum_{r\neq p}\prod_{k\in i}(\eta^0_{k\to i}+\eta^r_{k\to i}) + \prod_{k\in i}\eta^0_{k\to i}}{\sum_r\prod_{k\in i}(1-\eta^r_{k\to i}) - \sum_r\prod_{k\in i}(\eta^0_{k\to i}+\eta^r_{k\to i}) + \prod_{k\in i}\eta^0_{k\to i}}$$

Equations for general q can be computed by noting that for independent random variables $c_1, \dots c_k \in D$ the following relation holds:

$$\begin{aligned} P(\{c_1,\dots,c_k\} = D) &= P(\{c_1,\dots,c_{k-1}\} = D) + \\ &\quad + \sum_{d\in D} P(\{c_1,\dots,c_{k-1}\} = D\setminus\{d\})\, P(c_k = d) \end{aligned}$$

This recurrence allows one to write the SP equations for any q. Of course these equations need a computation time that scales exponentially with q, unlike BP equations for coloring (Cf. Eq. (4.2.12))

which can be computed in a time which is linear in q. In all cases the computational time is linear in k.

4.3.3. General SP equations. Consider again the BP equations Eq. (4.2.4) in this equivalent form:

$$P^{(a)}\left(s_i\right) \propto \sum_{\{s_j\}_{j\in b\setminus i, b\in i\setminus a}} \prod_{b\in i\setminus a} Q_b\left(\mathbf{s}\right) \prod_{j\in b\setminus i} P^{(b)}\left(s_j\right)$$

For every configuration of $\{s_j\}_{j\in b\setminus i}$ there is a subset of available values for s_i. Let us code these outcomes with a binary vector $\mathbf{u}_{b\to i}$ defined as $\left(u^q_{b\to i}\right)_{q\in X} \stackrel{\text{def}}{=} Q_b\left(\{s_j\}_{j\in b\setminus i}, q\right)$. The non-zero coordinates of $\mathbf{u}_{b\to i}$ are the values for s_i that satisfy constrain a.

We can then compute what are the possible values of s_i that satisfy all constrain $b \in i\setminus a$ simultaneously, and we can code this with a binary vector $\mathbf{h}_{i\to a}$ that will be defined as $\mathbf{h}_{i\to a} = H\left(\{\mathbf{u}^q_{b\to i}\}_{b\in i\setminus a}\right)$ with

$$H\left(\{\mathbf{u}_{b\to i}\}_{b\in i\setminus a}\right)^q \stackrel{\text{def}}{=} \prod_{b\in i\setminus a} \mathbf{u}^q_{b\to i}$$

There are several possibles outcomes for $\mathbf{h}_{i\to a}$, among which:

- $\mathbf{h}_{i\to a} = \delta_t$ means that the only possible value for s_i is t, that is, variable s_i is "forced" to take value t
- $\mathbf{h}_{i\to a}$=$(0,\ldots,0)$ means that every value of s_i will violate some constrain $b \in a \setminus i$ (these outcome must be eliminated, as they don't correspond to solutions).

The idea of the SP equations is to propagate the distribution of the $\mathbf{h}$ values instead of the $\mathbf{s_i}$ ones. To this extent we have to define $\mathbf{u}_{b\to i}$ as a function of $\{\mathbf{h}_{j\to b}\}_{j\in b\setminus i}$ instead of as a function of $\{s_j\}_{j\in b\setminus i}$: we will define it as $\mathbf{u}_{b\to i} = U\left(\{h_{j\to b}\}_{j\in b\setminus i}\right)$ with

$$U\left(\{h_{j\to b}\}_{j\in b\setminus i}\right)^q \stackrel{\text{def}}{=} \max_{\left\{\{s_j\}_{j\in b\setminus i} : h^{s_j}_{j\to b}=1\right\}} Q_b\left(\{s_j\}_{j\in b\setminus i}, q\right) \tag{4.3.18}$$

Note that Eq.(4.3.18) reduces to the old definition $Q_b\left(\{s_j\}_{j\in b\setminus i}, q\right)$ when $\mathbf{h}_{j\to b} = \delta_{s_j}$. Now the propagation of the distribution of $\mathbf{h}$ messages will become this pair of coupled equations:

$$P\left(\mathbf{h}_{i\to a}\right) \propto \sum_{\{\mathbf{u}_{b\to i}\}_{b\in i\setminus a}} \left(1-\delta_{\mathbf{h}_{i\to a},(0,\ldots,0)}\right)\delta_{\mathbf{h}_{i\to a},H} \prod_{b\in i\setminus a} P\left(\mathbf{u}_{\mathbf{j}\to\mathbf{b}}\right) \tag{4.3.19}$$

and

$$P\left(\mathbf{u}_{a\to i}\right) \propto \sum_{\{\mathbf{h}_{j\to a}\}_{j\in a\setminus i}} \delta_{\mathbf{u}_{a\to i},U} \prod_{j\in a\setminus i} P\left(\mathbf{h}_{\mathbf{j}\to\mathbf{a}}\right) \tag{4.3.20}$$

Where the arguments of H and U have been elided for shortness. These are the general SP equations in [**18**]. When all variables are binary, possible (non contradictory) h values are $(1,0)$,$(0,1)$ and $(1,1)$, which can be coded as -1,1 and 0 as in Section 4.1. Eq. (4.3.20) can be computed extensively for a general constrain Q_b in a number of operations which is bounded by $3^{|\{b:b\in i\}|}$. If $|\{b : b \in i\}|$is small, this is computationally feasible. This has been implemented in a fairly flexible program and is publicly available at [**81**].

4.3.4. A short note about BP and SP equations. It is easy to see that BP and SP equations are formally very similar: both sets of equations can be described in a general way as follows: given the set $X = \oplus_{i\in I} X_i$ of configurations, consider the vector field $Y_i = \mathbb{R}^{X_i}$ of vectors with $|X_i|$ real coordinates with indices in X_i, the space $Y_i^+ = \{y \in Y_i : y_p \geq 0, y \neq 0\}$, and the space $V = \oplus_{a\in A} \oplus_{i\in I} Y_i$ with coordinates $\mu_{a\to i} \in Y_i$. A multilinear function $L : V \mapsto V$ is a function which is linear in each coordinate of V. Both BP and SP equations can be cast as the following system of equations:

$$\begin{aligned} (L\mathbf{v})_{a\to i} &= \lambda_{a\to i} v_{a\to i} \;\; a \in A \; i \in I \\ \lambda_{a\to i} &\in \mathbb{R}_{>0} \\ v_{a\to i} &\in Y_i^+ \end{aligned}$$

for a multilinear L.

Of course, it is doubtful that something useful can be said in this level of generality, as the whole complexity of the problem is hidden in the structure of L.

4.3.5. Numerical results: SID on random k-SAT . Once that we know how to compute all local fields $P(h_i)$ for a given formula, 4.3.12, we still need some work to build a problem-solving algorithm. This is for what the SID (survey inspired decimation) algorithm was designed. The idea is very simple: pick a variable index i and fix it to some value s_i such that $P(s_i) \neq 0$. Then recalculate all Ps for the reduced formula, and go on iteratively. If at each step the computed values P are assumed to be the exact marginals, then the reduced subproblem will have at least one solution (otherwise $P(s_i)$ would be 0). Naturally, considering that the SP equations are only approximate, one would chose the variable with the biggest $P(s_i)$ and fix it to the direction of that maximum. This indeed has shown to give very good results, which are exposed below.

Algorithm 3 SID algorithm for k-SAT

INPUT: The factor graph of a Boolean formula in conjunctive normal form. A maximal number of iterations t_{max} and a precision ϵ used in SP

OUTPUT: One assignment which satisfies all clauses, or 'SP UNCONVERGED', or "MAYBE UNSAT"

(1) Random initial condition for the surveys

(2) Run SP. **If SP does not converge**, return 'SP UNCONVERGED' and stop. **If SP converges**, use the fixed-point surveys $\eta^*_{a\to i}$ in order to:

(3) Decimate:

  (a) **If non-trivial surveys** ($\{\eta^*(s_i)\} \neq \{\delta(s_i)\}$) **are found**, then:

    (i) Evaluate, for each variable node i, the marginal $\left\{W_i^{h_i} \stackrel{\text{def}}{=} P^*(h_i)\right\}$ for $h_i \in \{-1, 0, 1\}$ defined by Eqs. (4.3.10)-(4.3.11)

    (ii) Fix the variable with the largest $|W_i^+ - W_i^-|$ to the value $x_i = 1$ if $W_i^+ > W_i^-$, to the value $x_i = -1$ if $W_i^+ < W_i^-$. Clean the graph, which means: remove the clauses satisfied by this fixing, reduce the clauses that involve the fixed variable with opposite literal, update the number of unfixed variables.

  (b) **If all surveys are trivial** ($\{\eta^*\} = \{\delta\}$) , then output the simplified sub-formula and run on it a local search process (e.g. walksat).

(4) If the problem is solved completely then output "SAT" and stop. If no contradiction is found then continue the decimation process on the smaller problem (**go to 1.**) else (if a contradiction is reached) stop with "MAYBE UNSAT"

In step 4. an alternative is to go to 2 to continue on the smaller subproblem (i.e. to not reinitialize all surveys). In practice this is much faster and seems to give the same results.

Algorithm 4 SP algorithm for $k-$SAT

INPUT: a set of biases $\{\eta^0\}$, a precision number ϵ and a maximum number of iterations m

OUTPUT: an ϵ-quasi fixed point $\{\eta^*\}$for the SP equations or UN-CONVERGE

Apply Eq. (4.3.5)-(4.3.9) to build the iteratively sequence $\{\eta^t\}$ for $t \in \mathbb{N}$, until $\|\eta^{t+1} - \eta^t\|_\infty < \epsilon$ or $t > m$. In the first case, return $\{\eta^*\} = \{\eta^t\}$. In the second, return UN-CONVERGE

The SID Algorithm for k-SAT is exposed in Algorithm 3.

Once the system reaches the paramagnetic state (all surveys are trivial), the following alternative to calling a local search algorithm has been experimentally been found. Each time the trivial state is reached, slightly decrease the BP/SP interpolation parameter ρ of Eq. (4.3.5) and set the remaining messages to a random initial condition; then go on with the decimation. This seems to allow to reduce the formula completely, without calling any external algorithm. This result seems to suggest that further study of the interpolating algorithm for $0 < \rho < 1$ may be fruitful.

We have experimented SP and SID on single instances of the random 3-SAT problem with many variables, up to $n \sim 10^7$. In this section we summarize these experiments and their results.

Instances of the 3-SAT problem were generated with the pseudo random number generator "Algorithm B" on p.32 of Knuth [**42**]. However we found that results are stable with respect to changes in the random number generators. Formulas are generated by choosing k-tuples of variable indices at random (with no repetitions) and by negating variables with probability 0.5.

We first discuss the behavior of the SP algorithm itself. We have use a precision parameter $\epsilon = 10^{-3}$ (smaller values didn't seem to increase performance significantly). Depending on the range of α, we have found the following behaviors, for large enough n:

- For $\alpha < \alpha_d \sim 3.9$, SP converges toward the set of trivial messages $\eta_{a\to i} = 0$, for all $a - i$ edges. All variables are under-constrained.
- For $3.9 < \alpha < 4.3$, SP converges to a unique fixed-point set of non-trivial messages, independently from the initial conditions, where a large fraction of the messages $\eta_{a\to i}$ are in $]0, 1[$.

Notice that, for 'small' values of n, around $n = 1000$, one often finds some instances in which SP does not converge. But the probability of convergence, at a given $\alpha < 4.3$, increases with n. This is exemplified by the following quantitative measure of the performance of the SID algorithm (which uses SP). We have solved several instances of the random 3-SAT problem, for various values of α and n, using the SID algorithm in which we fix at each step the fraction fn_t of variables with largest $\left|W_i^+ - W_i^-\right|$. Table 1 gives in each case the fraction of samples which are solved by SID, in a single run of decimation (without any restart). The algorithm fails when, either SP does not converge, or the simplified sub-formula found by SID is not solved by walksat (this last situation was found only in very few cases). The performance of SID improves when n increases and when f decreases. Notice that for $n = 10^5$ we solve all the 50 randomly generated instances at $\alpha = 4.24$. For larger values of α the algorithm often fails. Notice that in such cases it does not give any information on whether the instance is UNSAT. Some failures may be due to UNSAT instances, others are just real failures of the SID for SAT instances.

As far as computational cost is concerned, we have found that the convergence time (number of iterations) of the SP algorithm basically does not grow with n (a growth like $\log n$, which could be expected from the geometrical properties of the factor graph, is not excluded). Therefore the process of computing all the SP messages $\eta^*_{a\to i}$ takes $\theta(n)$, or maybe $\theta(n \log n)$, operations. If SID fixes at each step only one variable, it will thus converge in $\theta(n^2 \log n)$ operations. When we fix a fraction of variables at a time, we get a further reduction of the cost to $O\left(n (\log n)^2\right)$ (the second ln comes from sorting the biases).

$n = 2.5 \cdot 10^4$

	4.21	4.22	4.23	4.24
4%	86%	66%	28%	8%
2%	100%	86%	50%	22%
1%		94%	78%	32%
0.5%		98%	88%	50%
0.25%		100%	90%	60%
0.125%			94%	60%
$< t >$	1369	2428	4635	7843

$n = 5.0 \cdot 10^4$

4%	98%	84%	52%	22%
2%	100%	98%	86%	48%
1%		100%	94%	64%
0.5%			98%	66%
0.25%			100%	78%
0.125%				84%
$< t >$	1238	1751	3411	8607

$n = 1.0 \cdot 10^5$

4%	100%	100%	72%	22%
2%			100%	68%
1%				88%
0.5%				92%
0.25%				92%
0.125%				100%
$< t >$	1204	1557	2573	7461

TABLE 1. obtained by solving with a single decimation run of the SID algorithm 50 random instances of 3-SAT for different sizes for each values of α. Samples were tried to solve by fixing variables in blocks $\frac{f}{100}n_t$, with f was taken in the progression $f = 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}$ and n_t being the number of unfixed variables, stopping if the formula was solved. The maximal number of iteration was taken equal to 10^3 and the precision for convergence was taken equal to 10^{-3}. The table shows the fraction of instances which were solved by SID (first column), the fraction of variables which remained in the simplified instance when all surveys are trivial, and the average computer time requested for solving an instance in every case (on a 2.4 GHz PC). The last row of each table indicates the average number of SP iterations per formula along the whole decimation process.

A very basic yet complete version of the code which is intended to serve only for the study on random k-SAT instances is available at the web site [81]. A generalization of the algorithm to other binary variable problems is also available at the same site.

4.3.6. Numerical results: SID on random q-coloring. Once the polarizations computed in Eq. (4.3.17) are known, many strategies can be adopted for coloring the graph. We believe that the simplest and most intuitive one is the following:

- (i) If one spin is very biased to one color, fix that spin and remove it from the graph. Forbid this color to all neighbors.
- (ii) If the bias of one spin toward some color is very low, forbid that color.

Forbidding a color c to a node i implies rewriting Eq. (4.3.16) using only two colors for that particular node. This can be achieved simply by taking Eq. (4.3.16) and (4.3.17) but setting $\eta^c_{i\to k} = 0$ and $\eta^c_{k\to i} = 1$ for all $k \in i$. Similarly, fixing a node i to a certain color c can be achieved in the same equations by fixing $\eta^c_{i\to k} = 1$ and $\eta^c_{k\to i} = 0$ for all $k \in i$.

In practice, we put a cutoff for the value of the bias to be used for the previous criteria. We use rule (i) every time a bias toward some color is greater than $C_U = 0.8$ and rule (ii) if the bias was lower than $C_L = 0.15$. There is no special reason for selecting specifically these values, but we found numerically a fast convergence to solvable paramagnetic problem instances. It could be useful to make a systematic analysis for improving this choice, and also to discuss other selection rules. However, this is not the objective of the present work. We have observed that the algorithm works substantially better than every other algorithm for random q-coloring known to us, even without extensive parameter optimization. Summarizing the discussion above, the result is shown in Algorithm 5.

An interesting point about the algorithm described above is the fact that we can fix a certain fraction of spins in every algorithmic step,

Algorithm 5 SID Algorithm for coloring

(1) Take the original graph and run SP in its infinite-y version defined by Eq. (4.3.16).
(2) Calculate the biases of all spins according to (4.3.17).
(3) Select spins whose bias to one color is larger than C_U, fix and remove these spins from the graph. Forbid the color to all neighbors.
(4) Select spins whose bias to one color is lower than C_L and forbid that color to these spins.
(5) Take all spins where just one color is allowed, fix these spins, and remove then from the graph. Forbid the fixed color to all neighbors.
(6) If the the graph is not completely paramagnetic: rerun SP and go to 2.
(7) Run any smart program that solves the coloring sub-problem.
    Actually, we did not find any free program in the web which was able to easily handle large graphs for the coloring problem. The best we could find was the *smallk*-program by Culberson [**24**], but even in the easy region it exploded in memory for graphs with sizes larger than $n = 2000$. So step 7 above was changed into:
    (a) Transform the resulting graph into a satisfiability problem.
    (b) Run walk-SAT [**78**] on this satisfiability problem.

without rerunning SP every time. This drastically reduces the computational time. How many spins we may fix, depends in a non-trivial way on the system size and on the distance from the COL/UNCOL transition.

Figure 4.3.1 shows the success rate of our algorithm in 3-coloring random graphs in the hard region $c \in [4.42, 4.69]$. From left to right the sample sizes increase: $n = 4*10^3$, $8*10^3$, $16*10^3$, $32*10^3$ and $64*10^3$. In all the cases we fixed the 0.5 percent of the spins in every iteration step. Note that keeping this value fix we find a clear improvement of the algorithm for sizes going from $n = 4*10^3$, $8*10^3$ to $n = 16*10^3$ the performance is roughly the same for larger graphs suggesting that we should reduce the fraction of spins to fix. However, note that even

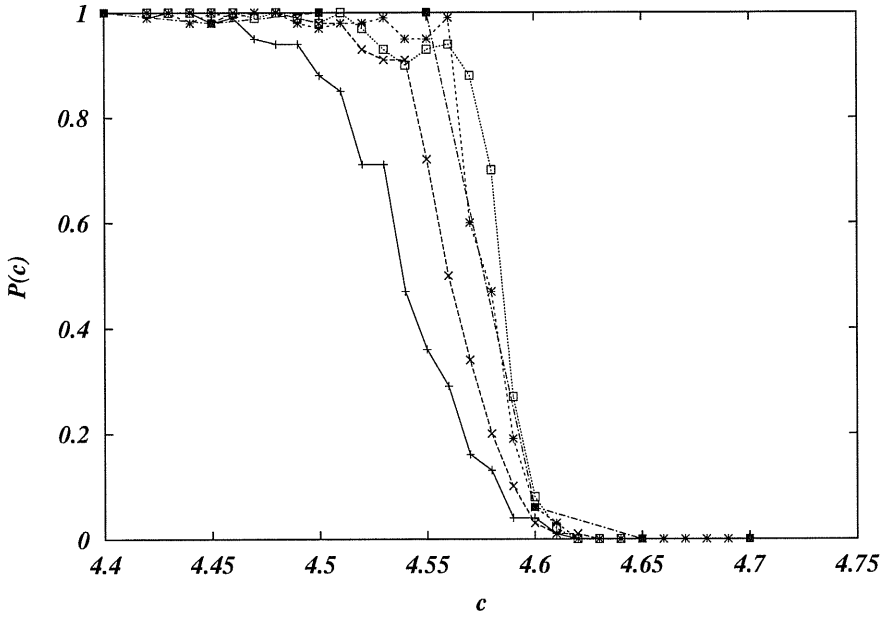

FIGURE 4.3.1. Probability of coloring a graph using our algorithm for different graph sizes. From left to right $n = 4 * 10^3$, $8 * 10^3$ $16 * 10^3$, $32 * 10^3$ and $64 * 10^3$.

within these conditions the algorithm works quite well in the hardest region of the parameters.

Note, that strong "finite-size effects" are present, in fact the algorithm doesn't behave very well for small graph sizes. Two reasons may explain this: First there are short loops that disappear in the thermodynamic limit, second there could be some shift in the location of the COL/UNCOL transition toward higher connectivities for larger graphs. This point should be clarified in a forthcoming work.

Another relevant feature of the curve is the following: The closer our graph is to the critical point, the smaller is also the fraction of spins we may fix in one algorithmic step. However, extrapolating the results, the worst (or best) we can do is to fix only one single spin at a time. This would change the complexity of our algorithm from $n \ln n$ (resulting from sorting spins with respect to their biases) to n^2, i.e. the algorithm remains polynomial.

CHAPTER 5

Microscopic interpretation of the SP equations

This section is about the SP equations for the 3-SAT problem[1]. In this chapter we will show how the SP equations for a 3-SAT formula $\mathcal{F}$ can be reinterpreted as BP equations of an associated problem with a certain formula $\mathcal{G}$.

In SP, we want to represent the condition for a variable of "not being forced" to take any specific value (or *unfrozen*) in a given solution configuration, and to this end we consider the configuration space of $3-$value variables $s_i \in \{-1, *, 1, \}$ instead of just $\sigma_i \in \{-1, 1\}$. We observe that C_a with the expression defined in Eq. (2.1.2) can be evaluated also in extended variables: it behaves as if variables with the $*$ value could be chosen to the best of -1 or 1 and thus satisfy the "clause". That is, if any variable in C_a is $*$ then the clause is satisfied (value 1), independently of the sign of the corresponding literal. This gives the name "joker state" to the value $*$. For a configuration $\mathbf{s}^{(i,x)}$ such that $s_i^{(i,x)} = x$ and $s_j^{(i,x)} = s_j$ for $j \neq i$ call

$$C_a^{i,x}(\mathbf{s}) = C_a\left(s^{(i,x)}\right) \tag{5.0.21}$$

and introduce the constraint over $\{-1, *, 1\}^I$ configurations given by

$$V_i = \delta_{s_i,*} \prod_{a \in i} C_a^{i,-1} C_a^{i,1} + \sum_{\sigma=\pm 1} \delta_{s_i,\sigma} \prod_{a \in i} C_a^{i,\sigma} \left(1 - \prod_{a \in i} C_a^{i,-\sigma}\right) \tag{5.0.22}$$

What does the constraint V_i enforces? If the variable s_i is ± 1 then V_i will be 1 (second term in the above equation) if and only if a) all surrounding clauses C_a are satisfied and b) by flipping s_i to ∓ 1 one would violate at least one of them. If on the contrary, $s_i = *$ then V_i will be 1 (first term) if and only if by changing s_i to any of ± 1

[1]Although we believe that most of it could be generalized to other problems

all surrounding clauses C_a would be satisfied. Note that in any case, clauses C_a have all value 1 for V_i to be 1.[2]

The LEC formula derived from $\mathcal{F}$ will be defined as

$$\mathcal{G} = \prod_i V_i \tag{5.0.23}$$

Note that V_i depends only on $(s_j)_{j\in a, a\in i}$ and therefore preserves the "locality" of the structure, if any, of the original formula. A solution of the LEC problem is a configuration $\mathbf{s} = (s_i)_{i\in I} \in \{-1, *, 1\}^n$ such that $\mathcal{G}(\mathbf{s}) = 1$. As a particular case, a solution $\mathcal{G}(\mathbf{s}) = 1$ such that $s_i \in \{\pm 1\}$ is also a solution of $\mathcal{F}$.

One could think as the the factor graph of the LEC has having $|I|$ additional function nodes (the A_i terms enforcing the joker condition) that extend over second neighbors (inset (b) in Fig. 5.0.1).

By inspecting Eq.(5.0.23) we notice a first problem, namely that we have lost the locally tree-likeness of the original graph. There are interactions terms between every (ordered) pair of neighbors variable nodes $i, j \in a$ (in the original graph), and thus for instance every such pair shares two constraints V_i, V_j (making an effective 2-loop). This introduces an obvious problem for implementing BP over this factor graph, and moreover would make difficult to compare both sets of equations, as the underlying geometry is now different. Fortunately, there is an easy (but unfortunately notationally somewhat involved) way out. We will group together neighbor variables, effectively performing a sort of duality transformation over the graph. We describe the procedure explicitly below (Note that this is a particularly simple case of a Kikuchi or "generalized belief propagation"-type approximation).

---

[2]For a generalization to the non-zero energy regime, one would have to separate the constraint V_i into two pieces: terms H_a that were present in the original formula (that can be violated), and terms

$$A_i = \delta_{s_i,*}\delta_{H_i^{-1}=H_i^1} + \sum_{\sigma=\pm 1} \delta_{s_i,\sigma}\delta_{H_i^\sigma < H_i^{-\sigma}}$$

for $H_i^\sigma = \sum_{a\in i}\left(1 - C_a^{i,\sigma}\right)$ enforcing the "joker condition" (that cannot be violated). In the case of the zero energy regime this separation is not needed and we have the more compact form of the definition of V_i.

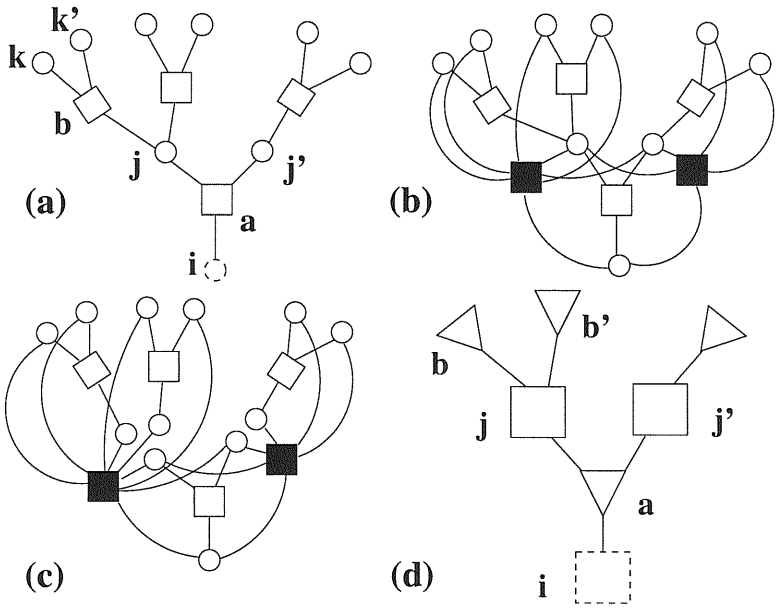

FIGURE 5.0.1. (a) Portion of the original factor graphs, (b) LEC graph with 3-state variables and additional constraints A_i (black nodes) (c) duality transformation (d) dual graph

We will define: **(i.)** $|A|$ multi state variables each one corresponding to a tuple $t_a = \left\{t_a^{(i)}\right\}_{i \in a}$ $(t_a^{(i)} \in \{-1, *, 1\})$ and "centered" on a clauses and have (uniform) connectivity n_a ((c) in Fig.5.0.1), and **(ii.)** $|I|$ function nodes χ_i^{dbp} depending on $T_i \equiv \{t_a\}_{a \in i}$ and enforcing both the joker state condition of Eq. 5.0.22 as well as identifying the values of the single variables $t_a^{(i)}$ shared by different tuples $a \in i$ ((d) in Fig.5.0.1). An explicit expression of $\chi_i^{dbp}(T_i)$ (conf. Eq.(5.0.22)) is

$$\chi_i^{dbp} = \sum_{\{s_i\}} \left(\prod_{a \in i} \delta_{t_a^{(i)}, s_i} \right) \left(\delta_{s_i,*} \prod_{a \in i} C_a^{i,-1} C_a^{i,1} + \right. \\ \left. + \sum_{\sigma = \pm 1} \delta_{s_i,\sigma} \prod_{a \in i} C_a^{i,\sigma} \left(1 - \prod_{a \in i} C_a^{i,-\sigma} \right) \right) \tag{5.0.24}$$

We shall refer to the BP equations over the dual graph as *Dual BP* (DBP).

5.1. Local equilibrium equations

5.1.1. SP equations as BP equations over the dual graph. Basic SP and DBP iterations can be thought of as transformations in the space of probability distributions of respectively the signs $h_i = \{-1, 0, 1\}$ of the effective fields acting on the single spin variables and of the tuples $t_a = \{-1, *, 1\}^{n_a}$ in the dual graph. In the cavity notation the quantities that are iterated refer to a graph in which a given node and all its neighbor nodes are temporarily eliminated (see Fig. 5.0.1 (a) and (d)) and all quantities are labeled by oriented indices of the type $a \to i$ or $i \to a$ where the node on the right of the arrow is the one eliminated. Therefore the equations describe a local transformation of some input probability distributions into an output distribution in which a characteristic function χ eliminates contributions from those combinations of input and output fields or variables that violate some kind of local constraints. Explicitly we have:

DBP equations:

$$P^{dbp}_{a\to i}(t_a) \propto \sum_{\{t_b\}} \prod_{j\in a\setminus i} \chi^{dbp}_j(t_a, \{t_b\}) \prod_{b\in j\setminus a} P^{dbp}_{b\to j}(t_b) \tag{5.1.1}$$

where χ^{dbp}_j was given in Eq. (5.0.24).

SP equations: Eq. (4.3.4) can be written as

$$P^{sp}_{j\to a}(h_j) \propto \sum_{\{h_k\}} \chi^{sp}_{j\to a}(h_j, \{h_k\}) \prod_{b\in j\setminus a} \prod_{k\in b\setminus j} P^{sp}_{k\to b}(h_k) \tag{5.1.2}$$

where

$$\chi^{sp}_{j\to a} \stackrel{\text{def}}{=} \delta_{h_j,*} \prod_{b\in j\setminus a} C^{j,1}_b C^{j,-1}_b + \sum_{\sigma=\pm 1} \delta_{h_j,\sigma} \prod_{b\in j\setminus a} C^{j,\sigma}_b \left(1 - \prod_{b\in j\setminus a} C^{j,-\sigma}_b\right)$$

C_b clauses are here evaluated in $\left((h_k)_{k\in b\setminus j}, h_j\right)$.

In order to show the connection between the above equations it is convenient to introduce an auxiliary transformation τ of a similar type:

τ transformation:

$$P^{\tau}_{a\to i}(t_a) \propto \sum_{\{h_j\}} \prod_{j\in a\setminus i} \chi^{\tau}_{j\to a}(t_a, h_j)\, P_{j\to a}(h_j) \tag{5.1.3}$$

for

$$\begin{aligned} \chi^{\tau}_{j\to a} &\stackrel{\text{def}}{=} \sum_{\sigma=\pm 1} C_a \delta_{h_j,\sigma}\delta_{t_a^{(j)},\sigma} + \\ &+\delta_{h_j,*}[\delta_{t_a^{(j)},*} C_a^{j,-1} C_a^{j,1} + \sum_{\sigma=\pm 1} \delta_{t_a^{(j)},\sigma} C_a^{j,\sigma}\left(1 - C_a^{j,-\sigma}\right)] \end{aligned} \tag{5.1.4}$$

C_a terms are evaluated here in t_a.

We will drop now the argument dependence of the measures $P^{sp}_{j\to a}$, $P^{dbp}_{a\to i}$ and $P^{\tau}_{j\to a}$ and make instead explicit the dependence on the input probability measures $\{P_{k\to b}\}, \{P_{b\to j}\}, \{P_{j\to a}\}$ respectively.

THEOREM 5.1.1. *The connection between DBP and SP can be written as follows:*

$$P^{dbp}_{a\to i}(\{P^{\tau}_{k\to b}\}) \equiv P^{\tau}_{a\to i}\left(\left\{P^{sp}_{j\to a}\right\}\right) \tag{5.1.5}$$

where both sides of the (functional) identity in turn depend on some arbitrary set of probability distributions $\{P_k(h_k)\}$ where $k \in b \setminus j$ for $b \in j \setminus a$ and finally $j \in a \setminus i$. In short,

$$P^{dbp} \circ P^{\tau} \equiv P^{\tau} \circ P^{sp} \tag{5.1.6}$$

In order to check the validity of the above identity we observe that a direct inspection of the composition shows that it is true if for every $j \in a \setminus i$ the following condition among the characteristic functions holds:

$$\sum_{\{h_j\}} \chi^{\tau}_{j\to a}\chi^{sp}_{j\to a} = \sum_{\{t_b\}} \chi^{dbp}_j \prod_{b\in j\setminus a}\prod_{k\in b\setminus j} \chi^{\tau}_{k\to b} \tag{5.1.7}$$

It is not surprising that we can reduce the problem to a "discrete" identity. Remembering that, modulo a normalization factor, all equations are multilinear, it is clear that it suffices to prove the identity over (a

product of) vectors of a linear base. A natural base is given by delta measures, and this result in the identity of Eq. (5.1.7).

In Section 5.3 we display the proof that this identity holds and, as a consequence, that also identity Eq. (5.1.6) is valid. Eq. (5.1.6) in turn implies trivially that

THEOREM 5.1.2. *The following identity holds*

$$\left(P^{dbp}\right)^{(k)} \circ P^{\tau} \equiv P^{\tau} \circ \left(P^{sp}\right)^{(k)} \tag{5.1.8}$$

where the (k) exponent means composition. This in turn implies that we have a direct step-by-step connection between the elementary quantities used in the DBP equations and those used in the SP equations: convergence is obtained simultaneously and Eq. (5.1.6) holds for the respective fixed points (which will be called conjugate fixed points). It is straightforward to compute from the DBP equations the marginals beliefs $P_i^{dbp}(s_i)$ of the single variables as a marginalization of $P_a^{dbp}(t_a)$ for some $a \in i$ with respect to all other variables in the clause, (on a fixed point, it doesn't matter which $a \in i$ one chooses). One finds that the marginals predicted by DBP are in one to one correspondence with the local fields given by SP, that is $P_i^{dbp}(s_i = -1, *, 1)$ coincides respectively with $P_i^{sp}(H_i = -1, 0, 1)$.

5.2. Entropy and complexity

The equivalence between the DBP marginals and the SP local field probability distributions gives also a nice interpretation for the complexity Σ. In fact one may show that, if respectively evaluated in conjugate fixed points:

THEOREM 5.2.1. *The Bethe approximation to the entropy on the dual graph, S^{dbp}, coincides with the logarithm of the number of clusters of solutions predicted by SP, the so-called complexity Σ.*

On general grounds the Bethe approximation to the entropy of a problem is exact if correlations among cavity variables can be neglected (i.e. the global joint probability distribution takes a factorized

form). This is certainly true over tree graphs and it is conjectured to be true in some cases for locally tree-like random graphs in the limit of large size (one informal explanation is that distance between cavity variables diverges with probability tending to one). Factorization of marginal probabilities over our dual factor graph amounts at writing $P(\{t_a\}) = \prod_{i\in I} P_i^{dbp}(T_i) \prod_{a\in A} P_a^{dbp}(t_a)^{1-n_a}$ where $P_i^{dbp}(T_i)$ is the joint probability distribution of the triples connected to node i ($T_i \equiv \{t_b\}_{b\in I}$) and $P_a^{dbp}(t_a)$ is the single triple marginal. Under this condition the entropy in Formula (2.6.2) reads

$$S = -\sum_i \sum_{\{T_i\}} P_i^{dbp}(T_i) \log P_i^{dbp}(T_i) + \tag{5.2.1}$$

$$+ \sum_a (n_a - 1) \sum_{\{t_a\}} P_a^{dbp}(t_a) \log P_a^{dbp}(t_a) \tag{5.2.2}$$

Showing $S = \Sigma$ is a straightforward but long calculation. It requires to express the entropy in terms of the cavity fields given by SP exploiting both Eq.(5.1.6) and the fixed point conditions. One finds (this boring proof is delayed to Section 5.2.1):

$$S = \sum_i \log c_i - \sum_a (n_a - 1) \log c_a - \sum_i \sum_{a\in i} \log D_{a\to i} \tag{5.2.3}$$

where the three normalization constants are defined by

$$c_i = \sum_{\{T_i\}} \prod_{a\in i} P_{a\to i}(t_a)\, \chi_i(T_i) \tag{5.2.4}$$

$$c_a = \sum_{t_a} \sum_{\{h_j\}} \prod_{j\in a} P_{j\to a}(h_j)\, \chi^{\tau}_{j\to a}(h_j, t_a) \tag{5.2.5}$$

$$D_{a\to i} = \sum_{t_a} \sum_{\{h_j\}} \prod_{j\in a\setminus i} P_{j\to a}(h_j)\, \chi^{\tau}_{j\to a}(h_j, t_a) \tag{5.2.6}$$

These constants are not independent and the explicit expressions of the first two are sufficient for writing S in terms of SP quantities:

$$
\begin{aligned}
(5.2.7)\ c_a &= \sum_{\{h_j\}} \prod_{j\in a} P_{j\to a}(h_j) \sum_{\{t_a\}} \prod_{j\in a} \chi^{\tau}_{j\to a}(h_j, t_a) \\
(5.2.8) &= 1 - \sum_{\{h_j\}} \prod_{j\in a} P_{j\to a}(h_j) \left(1 - \sum_{\{t_a\}} \prod_{j\in a} \chi^{\tau}_{j\to a}(h_j, t_a)\right) \\
(5.2.9) &= 1 - \prod_{j\in a} P_{j\to a}(J_{a,j}) \\
(5.2.10) &= 1 - \prod_{j\in a} \frac{\Pi^{u}_{j\to a}}{\left(\Pi^{s}_{j\to a} + \Pi^{0}_{j\to a} + \Pi^{u}_{j\to a}\right)}
\end{aligned}
$$

where we have borrowed the notation of Eq.(18) in[**17**]. For computing c_i we first notice that

$$
(5.2.11) \qquad P_{a\to i}(t_a) = D_{a\to i} \sum_{\{h_j\}_{j\in a\setminus i}} \chi^{\tau}_{j\to a}(t_a, h_j) \prod_{j\in a\setminus i} P_{j\to a}(h_j)
$$

so that Eq. (5.2.4) reads

$$
\begin{aligned}
c_i &= \prod_{a\in i} D_{a\to i} \sum_{\{H_i\}} \sum_{\{T_i\}} \chi_i(T_i) \prod_{a} \prod_{j\in a\setminus i} \chi^{\tau}_{j\to a}(t_a, h_j)\, P_{j\to a}(h_j) \\
&= \prod_{a\in i} D_{a\to i} \sum_{\{H_i\}} \chi^{sp}_i(H_i) \prod_{a} \prod_{j\in a\setminus i} P_{j\to a}(h_j) \\
(5.2.12) &= \prod_{a\in i} D_{a\to i} \left(\hat{\Pi}^{+}_i + \hat{\Pi}^{0}_i + \hat{\Pi}^{-}_i\right)
\end{aligned}
$$

in the notations of Eq. (21) in[**17**]. Finally, plugging these expressions into Eq.(5.2.3) and calling

$$
\begin{aligned}
w_i &= \hat{\Pi}^{+}_i + \hat{\Pi}^{0}_i + \hat{\Pi}^{-}_i \\
x_{i\to a} &= \Pi^{s}_{j\to a} + \Pi^{0}_{j\to a} + \Pi^{u}_{j\to a} \\
(5.2.13) \qquad y_{i\to a} &= \Pi^{u}_{j\to a}
\end{aligned}
$$

we get from Eq. (5.2.3)

$$
(5.2.14) \qquad S = \sum_i \log w_i - (n_a - 1) \sum_a \log \left(1 - \prod_{j\in a} \frac{y_{i\to a}}{x_{i\to a}}\right)
$$

In this expression, w_i represents the probability the local field acting on the spin variable i does not produce a contradiction and $1 - \frac{y_{i\to a}}{x_{i\to a}}$ is the probability that the cavity fields satisfy clause a.

We recall that the expression of the SP complexity Σ defined in Eq. (4.3.13) is

$$\Sigma = \sum_i (1-n_i)\log w_i + \sum_a \log\left(\prod_{i\in a} x_{i\to a} - \prod_{i\in a} y_{i\to a}\right)$$

$$= \sum_i \log w_i - \sum_a \sum_{i\in a} \log w_i + \sum_a \log\left(\prod_{i\in a} x_{i\to a} - \prod_{i\in a} y_{i\to a}\right) \tag{5.2.15}$$

Despite their different look, it turns out that Eq. (5.2.14) and Eq. (5.2.15) are identical if evaluated in a fixed point of the SP equations. Their difference $\Sigma - S$ is

$$\sum_a \left\{ -\sum_{i\in a} \log w_i + n_a \log\left(1 - \prod_{i\in a} \frac{y_{i\to a}}{x_{i\to a}}\right) - \sum_{i\in a} \log x_{i\to a} \right\} \tag{5.2.16}$$

is zero since in the fixed point every term inside the curly brackets vanishes: using Eq.(17) in[**17**] we have that $\eta_{a\to i} = \prod_{j\in a\setminus i} \frac{y_{j\to a}}{x_{j\to a}}$, i.e. $\prod_{j\in a} \frac{y_{i\to a}}{x_{i\to a}} = \eta_{a\to i}\frac{y_{i\to a}}{x_{i\to a}}$ for every $i \in a$ and hence

$$n_a \log\left(1 - \prod_{j\in a} \frac{y_{i\to a}}{x_{i\to a}}\right) = \sum_{j\in a} \log\left(1 - \eta_{a\to i}\frac{y_{i\to a}}{x_{i\to a}}\right) \tag{5.2.17}$$

A simple calculation shows that $w_i = x_{a\to i} - \eta_{a\to i} y_{a\to i}$ for every $a \in i$ and therefore we get $\Sigma = S$ as desired.

We will now proceed to prove Eq. (5.2.3):

For simplicity of notation, in what follows we will write $P_a(t_a)$, $P_i(T_i)$, $P_{a\to i}(t_a)$ and $\chi_i(T_i)$ in place of $P_a^{dbp}(t_a)$, $P_{a\to i}^{dbp}(t_a)$, $P_i^{dbp}(T_i)$ and $\chi_i^{dbp}(T_i)$ respectively and $P_{i\to a}(h_i)$ in place of $P_{i\to a}^{sp}(h_i)$.

5.2.1. Expression of the Bethe entropy in terms of SP quantitites. To compute the entropy (5.2.1) we first need

$$\begin{aligned} P_a(t_a) &= c_a^{-1} \sum_{\{h_i\}} \prod_{i \in a} P_{i \to a}(h_i) \prod_{i \in a} \chi^{\tau}_{i \to a}(t_a, h_i) \\ &= c_a^{-1} \prod_{i \in a} \sum_{\{h_i\}} P_{i \to a}(h_i) \, \chi^{\tau}_{i \to a}(t_a, h_i) \end{aligned}$$

Thus calling

$$f_{a \to i} = \sum_{\{h_i\}} P_{i \to a}(h_i) \, \chi^{\tau}_{i \to a}(t_a, h_i) \tag{5.2.18}$$

we have that

$$\begin{aligned} \sum_{\{t_a\}} P_a(t_a) \log P_a(t_a) &= -c_a^{-1} \log c_a + \sum_{\{t_a\}} P_a(t_a) \sum_{i \in a} \log f_{a \to i} \\ &= -c_a^{-1} \log c_a + \sum_{i \in a} \sum_{\{t_a\}} P_a(t_a) \log f_{a \to i} \end{aligned} \tag{5.2.19}$$

Writing $\omega_{a \to i} = \sum_{\{t_a\}} P_a(t_a) \log f_{a \to i}$ we get

$$\begin{aligned} \sum_a (n_a - 1) \sum_{i \in a} \omega_{a \to i} &= \sum_i \sum_{a \in i} \sum_{j \in a \setminus i} \omega_{a \to j} \\ &= \sum_i \sum_{a \in i} \sum_{j \in a \setminus i} \sum_{\{t_a\}} P_a(t_a) \log f_{a \to j} \\ &= \sum_i \sum_{a \in i} \sum_{\{t_a\}} P_a(t_a) \prod_{j \in a \setminus i} \log f_{a \to j} \\ &= \sum_i \sum_{a \in i} \sum_{\{t_a\}} \sum_{\{t_b\}_{b \in i \setminus a}} P_i(T_i) \prod_{j \in a \setminus i} \log f_{a \to j} \\ &= \sum_i \sum_{a \in i} \sum_{\{T_i\}} P_i(T_i) \log \prod_{j \in a \setminus i} f_{a \to j} \end{aligned} \tag{5.2.20}$$

The term inside the logarithm above reads

$$\begin{aligned} \prod_{j \in a \setminus i} f_{a \to j} &= \sum_{\{h_j\}} \prod_{j \in a \setminus i} \chi^{sp}_{j \to a}(t_a, h_j) \prod_{j \in a \setminus i} P_{j \to a}(h_j) \qquad (5.2.21) \\ &= \frac{1}{D_{a \to i}} P_{a \to i}(t_a) \qquad (5.2.22) \end{aligned}$$

where $D_{a\to i}$ is an appropriate normalization constant. Going back to Eq.(5.2.20), we have

$$\sum_a (n_a - 1) \sum_{i\in a} \omega_{a\to i} = \tag{5.2.23}$$

$$= -\sum_i \sum_{a\in i} \log D_{a\to i} + \sum_i \sum_{a\in i} \sum_{\{T_i\}} P_i(T_i) \log P_{a\to i}(t_a)$$

The second term in the right-hand side equals

$$\begin{aligned}
\sum_i \sum_{\{T_i\}} P_i(T_i) \log \prod_{a\in i} P_{a\to i}(t_a) &= \sum_i \sum_{\{T_i\}} P_i(T_i) \log \chi_i(T_i) \prod_{a\in i} P_{a\to i}(t_a) \\
&= \sum_i \sum_{\{T_i\}} P_i(T_i) \log Q_i(T_i) \\
&= \sum_i \sum_{\{T_i\}} P_i(T_i) \log P_i(T_i) + \\
&\quad + \sum_i \sum_{\{T_i\}} P_i(T_i) \log c_i \qquad (5.2.24)
\end{aligned}$$

where in the second step above $\chi_i(T_i)$ has been artificially multiplied inside the logarithm (we can do it because there is a $P_i(T_i)$ outside) and $P_i(T_i) = \frac{1}{c_i} Q_i(T_i)$. Eqs. (5.2.23), (5.2.24) give:

$$\begin{aligned}
(5.2.25) \quad \sum_a (n_a - 1) \sum_{i\in a} \omega_{a\to i} &= -\sum_i \sum_{a\in i} \log D_{a\to i} + \\
&\quad + \sum_i \sum_{\{T_i\}} P_i(T_i) \log P_i(T_i) + \sum_i \log c_i
\end{aligned}$$

Going back to the first expression of the entropy Eq.(5.2.1), and using Eq.(5.2.19) and Eq.(5.2.25) we get:

$$\begin{aligned} S &= -\sum_i \sum_{\{T_i\}} P_i(T_i) \log P_i(T_i) + \\ &+ \sum_a (n_a - 1) \sum_{\{t_a\}} P_a(t_a) \log P_a(t_a) \qquad (5.2.26) \\ &= \sum_i \log c_i - \sum_i \sum_{\{T_i\}} P_i(T_i) \log Q_i(T_i) + \\ &+ \sum_a (n_a - 1) \sum_{\{t_a\}} P_a(t_a) \log P_a(t_a) \qquad (5.2.27) \\ &= \sum_i \log c_i - \sum_a (n_a - 1) \log c_a - \sum_i \sum_{a \in i} \log D_{a \to i} \qquad (5.2.28) \end{aligned}$$

where the constants are defined in Eqs. (5.2.4-5.2.6).

5.3. Proof of equivalence

For the LHS of Eq.(5.1.7) we have:

If $h_j = \sigma \in \{\pm 1\}$ then

$$\chi^{\tau}_{j \to a} = C_a \delta_{t_a^{(j)}, \sigma} \qquad (5.3.1)$$

$$\chi^{sp}_{j \to a} = \prod_{b \in j \backslash a} C_b \left(1 - \prod_{b \in j \backslash a} C_b^{j, -\sigma} \right) \qquad (5.3.2)$$

If $h_j = *$ then

$$\chi^{\tau}_{j \to a} = \delta_{t_a^{(j)}, *} C_a^{j,-1} C_a^{j,1} + \sum_{\sigma = \pm 1} \delta_{t_a^{(j)}, \sigma} C_a^{j,\sigma} \left(1 - C_a^{j,-\sigma} \right) \qquad (5.3.3)$$

$$\chi^{sp}_{j \to a} = \prod_{b \in j \backslash a} C_b^{j,-1} C_b^{j,1}. \qquad (5.3.4)$$

Summing up both products and regrouping the LHS of Eq.(5.1.7) reads:

$$\sum_{\sigma = \pm 1} \delta_{t_a^{(j)}, \sigma} \prod_{b \in j} C_b^{j,\sigma} \left(1 - \prod_{b \in j} C_b^{j,-\sigma} \right) + \delta_{t_a^{(j)}, *} \prod_{b \in j} C_b^{j,-1} C_b^{j,1} \qquad (5.3.5)$$

where C_b for $b \in j \setminus a$ is evaluated here in $\left(\{h_k\}_{k\in b\setminus j}, t_a^{(j)}\right)$ and C_a is evaluated in t_a.
For the RHS of Eq.(5.1.7) we first notice that as the χ_j^{dbp} term includes $\prod_{a\in j}\delta_{t_a^{(j)},s_j}$ we will simply replace all occurrences of $t_b^{(j)}$ and s_j variables by $t_a^{(j)}$ and drop the outer sum and the product term itself. For instance, the sum over $\{t_b\}_{b\in j}$ thus reduces to a sum over $\left\{\left\{t_b^{(k)}\right\}_{k\in b\setminus j}, \left\{t_a^{(j)}\right\}\right\}$.
Let's evaluate the RHS of Eq.(5.1.7) on the three possible values of $t_a^{(j)}$:
If $t_a^{(j)} = *$ then by Eq.(5.0.24) $\chi_j^{dbp} = \prod_{b\in j} C_b^{j,-1}C_b^{j,1}$. Moreover, just by looking at its definition Eq.(5.1.4), one finds that in $\chi^{\tau}_{k\to b}$ all C terms are equal to 1 since their j coordinate $t_b^{(j)} = t_a^{(j)}$ is $*$. Then $\chi^{\tau}_{k\to b} = \delta_{t_b^{(k)},h_k}$ and the RHS of Eq.(5.1.7) becomes

$$C_a^{j,-1}C_a^{j,1}\prod_{b\in j\setminus a} C_b^{j,-1}C_b^{j,1}\prod_{k\in b\setminus j}\delta_{t_b^{(k)},h_k} \tag{5.3.6}$$

which is exactly the term in Eq.(5.3.5) corresponding to $t_a^{(j)} = *$ (remember that C_b clauses here are evaluated in t_b).
If $t_a^{(j)} = \sigma \in \{\pm 1\}$ then it is convenient to break χ_j^{dbp} in two addenda:

$$\prod_{b\in j} C_b - \prod_{b\in j} C_b C_b^{j,-\sigma} \tag{5.3.7}$$

so that the RHS of Eq.(5.1.7) becomes

$$C_a\prod_{b\in j\setminus a}\left(\sum_{\{t_b\}} C_b\prod_{k\in b\setminus j}\chi^{\tau}_{k\to b}\right) - C_aC_a^{j,-\sigma}\prod_{b\in j\setminus a}\left(\sum_{\{t_b\}} C_bC_b^{j,-\sigma}\prod_{k\in b\setminus j}\chi^{\tau}_{k\to b}\right)$$

Finally, both sums can be computed explicitly and the result is again exactly the corresponding term in Eq.(5.3.5). This ends the proof of the identity Eq.(5.1.6).

As a short note, adding an interpolating parameter ρ to Eq. 5.0.22 as

$$V_i = (1-\rho)\,\delta_{s_i,*}\prod_{a\in i} C_a^{i,-1}C_a^{i,1} + \sum_{\sigma=\pm 1}\delta_{s_i,\sigma}\prod_{a\in i} C_a^{i,\sigma}\left(1-\rho\prod_{a\in i} C_a^{i,-\sigma}\right)$$

Gives the original $k-$SAT combinatorial problem for $\rho = 1$ (in the dual graph) and the LEC energy of Eq. 5.0.22 for $\rho = 0$. Corresponding interpolating propagation equations where shown in Eqs. 4.3.5, 4.3.6.

5.4. Clustering and whitening

In this section we will try to interpret what solutions of combinatorial problem defined by Eq.(5.0.23) mean in term of clusters (or groups) of solutions of the original problem defined by Eq.(2.1.1).

We will first define a distance between two configurations $\mathbf{s}, \mathbf{t} \in \{1, *, -1\}$.

$$H(\mathbf{s}, \mathbf{t}) \stackrel{\text{def}}{=} |\{i : s_i \neq t_i\}|$$

Clearly H reduces to the usual Hamming distance when $\mathbf{s}, \mathbf{t} \in \{-1, 1\}^n$. We can now define a natural way of grouping the solutions of $\mathcal{F} = 1$.

DEFINITION 5.4.1. [exact clustering] Given $\mathbf{s} \in \{-1, 1\}^n$, the *connected component* $c(\mathbf{s}) \subset \{-1, 1\}^n$ is the equivalence class of $\mathbf{s}$ in the set $\mathcal{F} = 1$ for the equivalence relation generated by "$\mathbf{s} \overset{c}{\sim} \mathbf{t}$ if $H(\mathbf{s}, \mathbf{t}) = 1$".

That is, $c(\mathbf{s})$ is the set of all $\mathbf{t} \in \{-1, 1\}^n$ such that there is a path $\mathbf{s} = \mathbf{s_0}, \ldots, \mathbf{s_k} = \mathbf{t}$ with $H(\mathbf{s}_t, \mathbf{s}_{t+1}) = 1$ and $\mathbf{s}_t \in \{\mathcal{F} = 1\}$. The connected component is the most natural way of "clustering" or grouping configurations. A practical method to obtain $c(\mathbf{t})$ is by a "breath search": build the sequence of sets $C_t \subset \{-1, 1\}^n$ with $C_0 = \{\mathbf{t}\}$, and $C_{t+1} = C_t \cup D_t$ where $D_t = \cup_{\mathbf{s} \in C_t \setminus C_{t-1}} N(\mathbf{s})$ and $N(\mathbf{s}) = \{\mathbf{r} : \mathcal{F}(\mathbf{r}) = 1 \wedge H(\mathbf{s}, \mathbf{r}) = 1\}$. That is, at each step we add all nearest neighbors to the solutions already gathered. The process stops when $C_{t+1} = C_t$, and then $C_t = c(t)$.

We will now define the following partial ordering relation over three-state configurations: if $\mathbf{s}, \mathbf{t} \in \{-1, *, 1\}^n$ we say that $\mathbf{s} \leq \mathbf{t}$ if and only if $t_i \neq s_i$ implies that $t_i = *$. For instance, $(0, 1) \leq (0, *)$ and $(1, 1, 1) \leq (1, *, *)$ but $(0, 1) \not\leq (1, *)$. Then for $\mathbf{s}, \mathbf{t} \in \{-1, *, 1\}^n$ we will also say that $\mathbf{s}$ is *contained* in $\mathbf{t}$ if $\mathbf{s} \leq \mathbf{t}$. In this sense, "clustering" will mean, starting with some set $S \subset \{\pm 1\}^n$ of solutions of the original combinatorial problem, to find some set $T \subset \{1, *, -1\}^n$ such that every

$s \in S$ is contained in some $t \in T$. Of course, one would like to do so in some maximal way, but satisfying some kind of separation between different clusters. We will now introduce one such methods.

Consider first the following (non-metric) "distance"

$$D(\mathbf{s},\mathbf{t}) \stackrel{\text{def}}{=} |\{i : s_i \neq t_i, s_i \neq *, t_i \neq *\}|$$

Note that if we associate to a configuration $\mathbf{t} \in \{-1,*,1\}^n$ the set

$$U(\mathbf{t}) \stackrel{\text{def}}{=} \{\mathbf{s} \in \{-1,1\}^n : s_i = t_i \text{ if } t_i \neq *\}$$

then $D(\mathbf{s},\mathbf{t}) = H(U(\mathbf{s}), U(\mathbf{t}))$ where H is the hamming distance in its normal extension to sets, i.e.

$$H(A,B) \stackrel{\text{def}}{=} \min\{H(a,b) : a \in A, b \in B\}$$

Moreover, the relation $\mathbf{s} \leq \mathbf{t}$ is equivalent to the relation $U(\mathbf{s}) \subset U(\mathbf{t})$.

DEFINITION 5.4.2. [clustering by hypercubes] Given $\mathbf{s}$ such that $\mathcal{F}(\mathbf{s}) = 1$, the *hypercubic hull* $h(\mathbf{s}) \in \{-1,*,1\}^n$ of $\mathbf{s}$ for $\mathcal{F}$ is defined as the unique configuration satisfying simultaneously

(1) $\mathbf{s} \leq h(\mathbf{s})$
(2) for $\mathbf{t}$ such that $\mathcal{F}(\mathbf{t}) = 1$, if $\mathbf{t} \not\leq h(\mathbf{s})$ then $D(\mathbf{t}, h(\mathbf{s})) > 1$
(3) $h(\mathbf{s}) \leq \mathbf{t}$ for all $\mathbf{t}$ satisfying 1 and 2.

A method to obtain $h(\mathbf{s})$ is as follows (and a proof that such $h(\mathbf{s})$ exists): build the finite sequence $\mathbf{t} = \mathbf{t}^{(0)}, \ldots, \mathbf{t}^{(k)}$ in the following way: take a solution $\mathbf{s}^{(i)}$ such that $D\left(\mathbf{t}^{(i)}, \mathbf{s}^{(i)}\right) = 1$. Then define $\mathbf{t}^{(i+1)}$ as equal to $\mathbf{t}^{(i)}$ except in the coordinate j where $t_j^{(i)} \neq s_j^{(i)}$ but $t_j^{(i)} \neq *$ in which we define $t_j^{(i+1)} = *$. Iterate until no such $\mathbf{s}^{(i)}$ is found.

Clearly 1 and 2 hold for the last element in the sequence $\mathbf{t}^{(k)}$. Take $\mathbf{r}$ also verifying 1 and 2 and suppose there is an i such that $\mathbf{t}^{(i)} \not\leq \mathbf{r}$, and take the first such i. There is a coordinate j such that $t_j^{(i)} = *$ and $r_j = \sigma$, $t_j^{(i-1)} = \sigma$ and $s_j^{(i-1)} = -\sigma$ for some $\sigma = \pm 1$. Now $\mathbf{t}^{(i-1)} \leq \mathbf{r}$ so $D\left(\mathbf{r}, \mathbf{s}^{(i-1)}\right) \leq D\left(\mathbf{t}^{(i-1)}, \mathbf{s}^{(i-1)}\right) = 1$, so by condition 1 for $\mathbf{r}$ we have that $\mathbf{s}^{(i-1)} \leq \mathbf{r}$ but this is clearly absurd in view of coordinate j.

The sequence $\mathbf{t}^{(i)}$ may depend on the order in which the comparisons are made but clearly the final element $\mathbf{t}^{(k)} = h(\mathbf{s})$ is unique, thanks to property 3.

DEFINITION 5.4.3. Equivalence classes under the equivalence relation $\mathbf{s} \overset{h}{\sim} \mathbf{t}$ if and only if $h(\mathbf{s}) = h(\mathbf{t})$ will be called "hypercubic hull clustering"

LEMMA 5.4.4. *If* $\mathbf{s}, \mathbf{t}$ *belong to the same connected component of* $\mathcal{F} = 1$, *then* $h(\mathbf{s}) = h(\mathbf{t})$.

PROOF. We can assume that $H(\mathbf{s}, \mathbf{t}) = 1$. Then $D(h(\mathbf{s}), \mathbf{t}) \leq 1$ and so $\mathbf{t} \leq h(\mathbf{s})$ by definition of $h(\mathbf{s})$. Then $h(\mathbf{s})$satisfies the first two conditions of the definition of $h(\mathbf{t})$, so $h(\mathbf{t}) \leq h(\mathbf{s})$. By symmetry, $h(\mathbf{s}) = h(\mathbf{t})$. □

By complete enumeration of solutions, we have numerically verified that the two above methods are almost equivalent for random formulas (see Figure 5.4.4).

One trivial observation about the set $\mathcal{G} = 1$ is that solutions are (weakly) separated, in the sense of the following proposition:

LEMMA 5.4.5. $H(\mathbf{s}, \mathbf{t}) > 1$ *if* $\mathcal{G}(\mathbf{s}) = \mathcal{G}(\mathbf{t}) = 1$ *and* $\mathbf{s} \neq \mathbf{t}$.

PROOF. Just by looking at Eq. (5.0.22), it is easy to see that the terms $A = \prod_{a \in i} C_a^{i,-1} C_a^{i,1}$, $B = \prod_{a \in i} C_a^{i,-1} (1 - C_a^{i,1})$ and $C = \prod_{a \in i} (1 - C_a^{i,-1}) C_a^{i,1}$ cannot be pairwise simultaneously 1 (i.e. they have pairwise disjoint supports), and so the only the possible value for variable s_i is determined by which one of the three is. □

DEFINITION 5.4.6. [clustering by whitening] Given $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = 1$, the *whitening* $w(\mathbf{x}) \in \{1, *, -1\}^n$ is the unique configuration such that $\mathcal{G}(w(\mathbf{x})) = 1$, $\mathbf{x} \leq w(\mathbf{x})$ and $w(\mathbf{x})$ has the minimal number of $*$.

We will prove that such configuration exists by building it: suppose that $\mathcal{G}(\mathbf{x}) = 0$, Choose a V_i such that $V_i = 0$. It can be easily seen that by replacing x_i by $*$, then V_i becomes 1 (because as $\mathcal{F}(\mathbf{x}) = 1$, we will

have that the term $\prod_{a\in i} C_a^{i,-1} C_a^{i,1}$ in Eq. (5.0.22 is equal to 1). Then we pick another violated constrain $V_i = 0$ and repeat the process, until $\mathcal{G} = 1$. As $V_i = 0$ will continue to be 0 in the procedure, exactly until we switch x_i to $*$, the result of the procedure is independent of the order of the picked V_i and is exactly $w(\mathbf{x})$. Note that two configurations $\mathbf{x}, \mathbf{y}$ at Hamming distance $H(\mathbf{x}, \mathbf{y}) = 1$ will have $w(\mathbf{x}) = w(\mathbf{y})$ and so every solution in a fixed connected component of the solution space will end up inside the same "cluster". An example of the whitening procedure for some set of solutions is depicted in Figure 5.4.1. Numerical experiments for pseudo-random formulas show the average logarithm of the number of solutions of the whitening in Figure 5.4.2.

This procedure has been already used under the same name of *whitening* in the context of graph coloring by G. Parisi [**72**].

LEMMA 5.4.7. *For* $\mathbf{s}$ *such that* $\mathcal{F}(\mathbf{s}) = 1$, $\mathbf{s} \leq h(\mathbf{s}) \leq w(\mathbf{s})$

PROOF. The first inequality is in the definition of $h(\mathbf{s})$. Suppose now there exist $\mathbf{t}$ such that $\mathcal{F}(\mathbf{t}) = 1$ and $D(\mathbf{t}, w(\mathbf{s})) = 1$. Take the coordinate i such that $t_i = \sigma$ and $w(\mathbf{s})_i = -\sigma$. But then the term $\prod_{a\in i} C_a^{i,-\sigma} C_a^{i,\sigma}$ in $V_i(w(\mathbf{s}))$ must be 1, and so $w(\mathbf{s})_i = *$, an absurd. So for every $\mathbf{t}$ such that $\mathcal{F}(\mathbf{t}) = 1$ and $D(\mathbf{t}, w(\mathbf{s})) \leq 1$ we have that $D(\mathbf{t}, w(\mathbf{s})) = 0$, i.e. $\mathbf{t} \leq w(\mathbf{s})$. We just proved that $w(\mathbf{s})$ verifies the second condition of the definition of $h(\mathbf{s})$. As also $\mathbf{s} \leq w(\mathbf{s})$, we have that $w(\mathbf{s})$ satisfy also the first one, so by the minimality of $\mathbf{h}$ we get $h(\mathbf{s}) \leq w(\mathbf{s})$ □

LEMMA 5.4.8. *If* $s \leq w(\mathbf{t})$, *then* $w(\mathbf{s}) \leq w(\mathbf{t})$.

PROOF. $w(\mathbf{t})$ verifies conditions 1-2 of the definition of $w(\mathbf{s})$, so $w(\mathbf{s}) \leq w(\mathbf{t})$ □

LEMMA 5.4.9. *If* $h(\mathbf{s}) = h(\mathbf{t})$ *then* $w(\mathbf{s}) = w(\mathbf{t})$

PROOF. We have that $\mathbf{s} \leq h(\mathbf{s}) = h(\mathbf{t}) \leq w(\mathbf{t})$, so $w(\mathbf{s}) \leq w(\mathbf{t})$. By symmetry, $w(\mathbf{t}) \leq w(\mathbf{s})$. □

DEFINITION 5.4.10. The equivalence relation $\mathbf{s} \overset{\mathrm{w}}{\sim} \mathbf{t}$ iif $w(\mathbf{s}) = w(\mathbf{t})$ will define a third type of clustering.

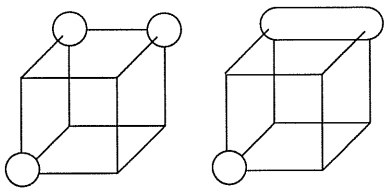

FIGURE 5.4.1. The hypercubic hull clustering procedure (and coinciding whitening) from left to right: the original sets of solutions $\{(-1,-1,-1),(1,1,-1)(1,1,1)\}$ and the set of clusters $\{(-1,-1,-1),(1,1,*)\}$ in the first (and final) step.

We have just shown three fairly natural different clusterization methods (exact clusterization, hypercubic hull, and whitening), with increasingly "loose" or large clusters. Trying to build from scratch a local Hamiltonian to capture the outcomes of the whitening procedure applied to the solutions of some SAT formula leads naturally to Eq. (5.0.23).

The reader should note however that the presented definition of clustering is far from perfect in the worst case: there is a number of systematic errors produced by the whitening. For instance, in Figure 5.4.3 we can see one cluster claiming an incorrectly large volume. And there is of course also another problem: unfortunately, there is no warranty that there are no other solutions of $\mathcal{G} = 1$ than the ones of the whitening. Spurious ground states (i.e. configurations that are not extensible to real solutions) do exist, however they turn out to be always unstable fixed points of SP, that is UNSAT configurations which seems to be disregarded by SP marginals. While such a result may be expected to hold for tree-like graphs, it is somewhat surprising to observe it numerically on small, loopy, random factor graphs. The robustness of such result calls for a finite N probabilistic analysis which would represent another building brick for the rigorous analysis of SP (of course, small ad-hoc counterexamples on improbable formulas can be relatively easily constructed).

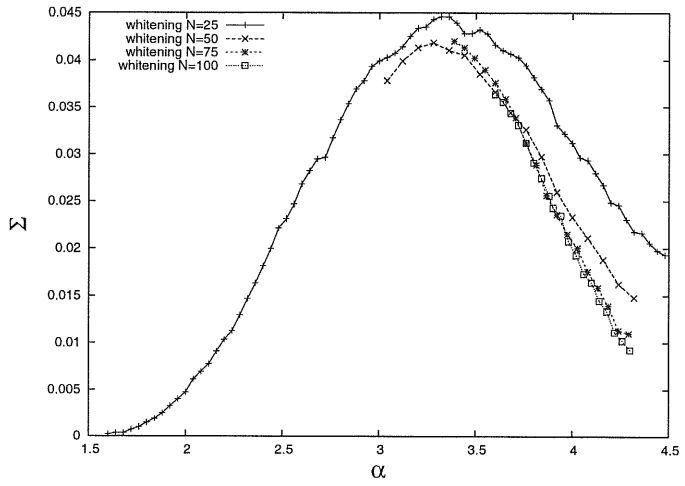

FIGURE 5.4.2. Number of whitening clusters averaged over 50 satisfiable pseudo random formulas with $n = 100$

Numerical work is being done to ascertain a quantification of these two types of "errors" [**19**]. A preliminar result shows a notable agreement even for small sizes; for 50 pseudo-random formulas at $n = 75$, exact enumeration of the above definitions of clustering is presented in Figure 5.4.4. We mention a related fact, that exact enumerations in all analyzed cases (on a large number of small random 3-sat formulas) showed that all the zero energy configurations of $\mathcal{G}$ which are stable under SP iterations under small perturbations when evaluated as pure measures can be extended to real solution of the original problem, giving good perspectives to further analysis.

5.5. Clustering in tree factor graphs

The argument turns out to be similar to the one given in an analogous "tutorial" appendix in ref. [**10**] for the Vertex Cover problem.

We will show that on a satisfiable tree with "boundary conditions", there is only one connected component of $\mathcal{F} = 1$, and also one unique solution of $\mathcal{G} = 1$. We will define a boundary condition as a set of 1-clauses attached to some of the leaves (forcing them to take one specific value), so the equivalent "boundary conditions" for $\mathcal{G}$ are automatically defined.

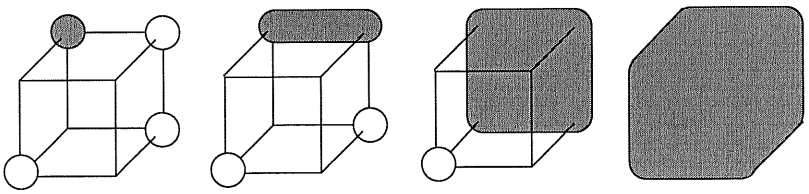

FIGURE 5.4.3. A systematic "error" of the hypercubic hull and then of the whitening $w((1,1,-1))$ (the dark solution in the left). From left to right: the original sets of solutions $\{(1,1,-1),(1,1,1),(1,-1,1),(-1,-1,-1)\}$ and first step $(1,1,-1)$, second step $(1,1,*)$, third step $(1,*,*)$ and final step $(*,*,*)$.

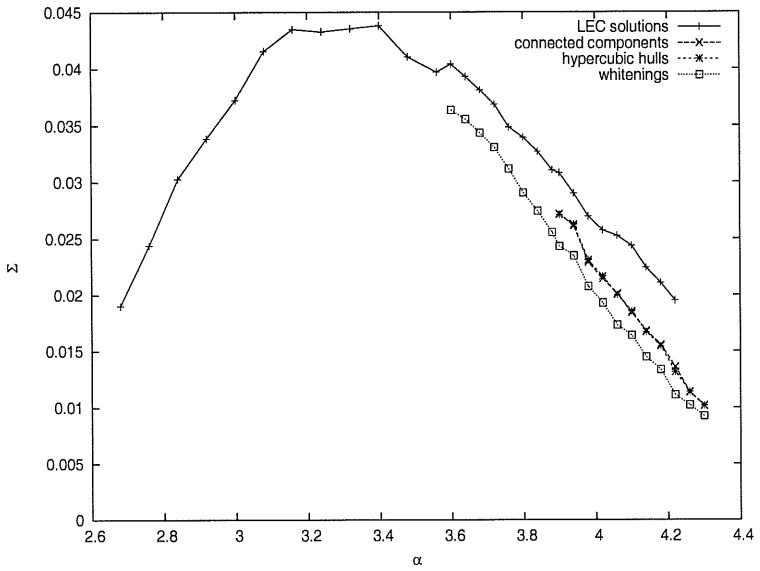

FIGURE 5.4.4. $\Sigma = \frac{1}{n}\log N_c$ (where N_c is the number of clusters in all four different definitions) averaged over 50 satisfiable pseudo random formulas of size $n = 100$ versus the clause density $\alpha = m/n$. Curves were continued up to where was computationally feasible (due to the exponential proliferation of solutions). Although not conclusive, the plot shows a remarkable agreement between all four methods

We will first build a reference solution $\mathbf{x}$, and then show that every solution of $\mathcal{F} = 1$ is connected to it. $\mathbf{x}$ will be built from the leaves to the root. Suppose the variables are labeled in an ordering that respects

distances to the root, such that the first ones are the leaves and the last one is the root. In such an ordering, the children (resp. parent) of i are neighbors with labels $j < i$ (resp. $j > i$). We will fix x_i iteratively: once x_j for $j < i$ are fixed, all children of j are fixed; then for x_j there are two possibilities: either its children force it to take a specific value, or they don't. In the first case we chose x_i to take the forced value; in the second one we chose the value that satisfy the parent clause. Now we can show that $\mathbf{x}$ is connected with every other solution $\mathbf{s}$ (and thus every two solution are connected). It is easy to see that the configurations $\mathbf{y}^{(k)}$ defined by $\mathbf{y}_j^{(k)} = s_j$ if $j < k$ and $\mathbf{y}_j^{(k)} = x_j$ if $j \geq k$ form a path of configurations connecting $\mathbf{x}$ and $\mathbf{s}$. Clearly $\mathbf{y}^{(1)} = \mathbf{x}$ and $\mathbf{y}^{(n)} = \mathbf{s}$. Also they are all solutions, since if $\mathbf{y}^{(k)}$ is a solution, then clearly $\mathbf{y}^{(k+1)}$ is also a solution: if they are different it is because $\mathbf{y}_{k+1}^{(k+1)}$ has been chosen to satisfy the parent clause (and it was not forced from children in s and thus neither in $y^{(k+1)}$).

We can now look for solutions of $\mathcal{G} = 1$ on a satisfiable tree with boundary conditions. Let's start with a free-boundary tree with 2 and 3-clauses: it is easy to see that the solution with all $*$ assignments has $\mathcal{G} = 1$. It is also clearly unique: suppose that there is a solution with some variable set to $\sigma \neq *$. Then there is forcefully one of its neighboring clauses in which the two (or one) remaining variables are fixed in order to not satisfy the clause. Repeating again the argument recursively for one of them, we can get a never-ending path of fixed variables in the tree. But as a trees have no loops and our graph is finite, this is a contradiction.

There is also exactly one such solutions for a satisfiable tree with with boundary conditions. Note that V_i constraints on the i variables with assigned boundary values are automatically satisfied if we assign them to the forced value. We will build it explicitly using the so-called unit clause propagation (UCP). The UCP procedure consists in removing (in this case starting from the boundary) every fixed variable by (a) removing all clauses satisfied by the variable and (b) removing the variable from all clauses in which it appears without satisfying the clause (if the original tree is satisfiable, no 0-clause can appear

in this erasure step). Then every possibly appearing 1-clause is taken and its variable fixed in order to satisfy the clause, and the procedure starts again from the beginning until no more 1-clauses show up. The resulting graph is boundary-free and with no 1-clauses.

The promised solution will be built by taking all variables fixed by UCP with their assigned value, and by assigning the value $*$ to the remaining ones. The resulting configuration $\hat{x}$ has $\mathcal{G}(\hat{x}) = 1$. Clearly the constraints V_i (see Eq.(5.0.22)) are satisfied by $\hat{x}$ for all i fixed by UCP (because they are "frozen" by their neighbors). We easily see that this partial assignment is the unique one that can give $\mathcal{G} = 1$. Using the fact that the subgraph produced by UCP has no boundary condition and that the unique solution for $\mathcal{G} = 1$ on that subgraph is the all-$*$ one, we see that the proposed configuration is indeed the unique solution.

Note also that every solution of $\mathcal{F} = 1$ will coincide with $\hat{x}$ in the $-1, 1$-assigned variables of the latter, because these variables were fixed by UCP and thus are forced in every satisfying configuration. Moreover, if one takes an index i such that $\hat{x}_i$ is $*$, then there is at least one solution of $\mathcal{F}(s) = 1$ with $s_i = 1$ (resp. -1): by fixing s_i and applying again UCP one cannot get any contradiction (i.e. a 0-clause) because the subgraph has no loops nor 1-clauses. The remaining graph is still loop-free, and thus trivially satisfiable.

NOTE. Section 5.4 is part of a joint work with V. Napolano and R. Zecchina. After finishing this work, we became aware of two other independent researchs [**47, 7**] on related subjects.

CHAPTER 6

Discussion

In a first part of this work we have studied the cavity method from statistical physics for k-SAT and q-coloring, which is an heuristic probabilistic method to obtain several statistical properties of the random k-SAT and random $q-$coloring ensembles, including approximations to the critical thresholds and average ground-state energy. We have shown how to derive Survey Propagation (SP), a concrete new algorithm to solve typical large instances of these two random combinatorial problems, by applying the "average case" cavity equations to single samples. The version of SP for q-coloring and $k-$SAT have been analyzed numerically and shown unseen performances, in fact establishing solving records several orders of magnitude bigger than what was possible to solve with known solving algorithms (and to the authors knowledge, most still "unbroken").

In the second part we focused in SP for k-SAT. We have shown by elementary means that these SP equations can be interpreted and derived as belief propagation equations for the marginals over a modified combinatorial problem. An important consequence of this fact is a clarification of the hypothesis behind the SP algorithm. It is to be expected that the essential hypothesis making BP to work is the un-correlation of the marginals of distant (or cavity) variables. Under the shown mapping, this directly implies that the hypothesis behind SP (and in a way, of its definition of clusters) is the un-correlation of the cavity (or distant) variables over the solutions of $\mathcal{G}$. In physical terms, the frozen/unfrozen probabilities of distant variables, that is an un-correlation among different "clusters" (we are using loosely the term "clusters" for the solutions of $\mathcal{G}$). The "pure states" hypothesis, which

is claimed to be used by the cavity method **[13]**, states that the uncorrelation between cavity fields $h_{i\to a}$ holds "inside pure states". This hypothesis in fact must be viewed as the *definition* of "pure state" or cluster for a given (finite size) formula. More precisely, we define a "pure state" α as a solution to the local equilibrium equation $\mathcal{G} = 1$ of Section 5.1; this definition is even more attractive from the mathematical point of view, as variables are not "directed" like the cavity fields $h_{i\to a}$. Moreover, this definition gives a precise mathematical sense to the expression in Eq. 3.3.5, it is simply the distribution of the random variable h_i among the solutions h of $\mathcal{G} = 1$ (this space with uniform probability). It strikes under this view that the cavity "directed" variables $h_{i\to a}$ are more an artifice of the method to describe the statistics of h_i (the BP equations) than inherent of the definition of clusters.

Under this light one can think of the SP procedure of obtaining $\mathcal{G}$ from $\mathcal{F}$ as a way of collapsing the internal structure of pure states: the resulting solution space $\mathcal{G} = 1$ has many pure states but with zero internal entropy (i.e. just one solution per cluster). Note that this is a completely different limit case with respect to the "one pure state" hypothesis in which BP (more precisely DBP) is assumed to work correctly and to predict an accurate entropy (which we remind is by definition the complexity of the original $\mathcal{F}$). Both situations are classified under the replica scheme as "replica-symmetric". Intuitively, collapsing (ignoring) the internal structure of clusters would reduce or eliminate correlations among distant (or cavity) variables, making the SP equations more accurate than the BP ones for the original problem.

Finally, it may be useful to compare the results for XOR-SAT in Section 3.5 and $k-$SAT. We have seen that variables for the XOR-SAT problem can be separated into two groups: the ones eliminated during the leaf removal process (the non-core) and the ones remaining (the core). Solutions can be thus grouped in "clusters": each cluster is formed by just one solution in the core (the "seed") joined to a compatible set of solutions in the non-core. Variables belonging to the non-core are "free", i.e. can be flipped and a new solution in the same cluster can

be found by just adjusting $O(1)$ other non-core variables. In the XOR-SAT problem thus "free" variables play the role of $h_i = *$ variables in the SAT problem; the difference is that in the latter *which* variables are free do depend on the particular cluster, whereas in the former it doesn't (they are determined a priori by the topology of the graph). The situation for k-SAT seems more likely to admit a generalization to other problems.

As far as the connection between solutions of $\mathcal{G}$ and $\mathcal{F}$ is concerned, things are particularly simple over tree factor graphs (see also [**17**] for results concerning propagation of messages): Indeed, for any fixed boundary condition (i.e. an assignment for the leaf variables), there is at most one solution of $\mathcal{G} = 1$, and it is easy to prove that all solutions of $\mathcal{F} = 1$ correspond to the same connected component of the solution space (i.e. every two solutions can be joined by a path of solutions in which successive configurations in the path differ by exactly one spin flip). The situation on loopy graphs (corresponding for instance to random formulas) is obviously more complicated. Our interpretation is that not only the recursive DBP/SP equations themselves are accurate in a probabilistic sense (i.e. when the factorization of the corresponding input joint probability is sound) to compute the statistics of the ground states of $\mathcal{G}$, but also that the accuracy of the interpretation of the ground states of $\mathcal{G}$ in terms of clustering of the ground states of $\mathcal{F}$ relies on this hypothesis being true. We have shown some preliminar enumerative numerical results (unfortunately only for relatively small formulas) that seems to support this hypothesis.

A second consequence of the link between SP and BP is that the SP equations can benefit from known results about the BP ones. Most notably, this result gives automatically a new algorithm based on the variational result of Thm. 4.2.6 to obtain fixed points of the SP equations. Other extensions to BP, like the double-loop belief propagation[**38**] can be also applied to SP. Both these methods ensure convergence to a fixed point if $\mathcal{F} > 0$ and in consequence, a "smooth approximation" to the SP equations has automatically algorithms with guaranteed convergence.

Finally we would like to mention some of the (in our opinion) most important related "open problems" (which, fortunately or not, are quite a few). We enumerate them in suspected order if increasing difficulty.

- To formally generalize the connection between SP and DBP in the case of finite y. The LEC "energy function" would take the form $\hat{H} = \lambda \sum_{a \in A} H_a + \sum_{i \in I} A_i$, where λ plays the role of the so called Parisi re-weighting parameter y [**51**].
- To generalize this connection to other combinatorial problems.
- To complete the identification between clusters of solutions of $\mathcal{F} = 1$ and solutions of $\mathcal{G} = 1$ for random $k-$SAT. We suspect that (unlike the results already obtained) this will strongly depend on the statistics of the ensemble.
- To analyze rigorously the combinatorial problem of finding solutions of $\mathcal{G} = 1$ for random k-SAT. The cavity approach suggests that this should be easier to analyze analytically than the initial problem $\mathcal{F} = 1$. One possible approach is to analyze it with a (rigorously justified) RS method. Possibly the most difficult step in this proof would be (unsurprisingly) to prove the independence of the cavity variables.
- Despite the fact that is widely known and used, there are unfortunately too few rigorous known results about Belief Propagation when $\mathcal{F}$ is allowed to take the value 0: it would be interesting to have conditions of convergence and of existence of the fixed points and estimations of convergence times. All these results would be of course automatically inherited by SP.
- To use iteratively this method to interpret the k-RSB equations for $k \geq 2$ in appropriate problems.

We believe that the strength of the SP algorithm itself gives enough evidence that further study of the cavity method may be of great interest for mathematicians and computer scientists, and we hope that this work will help to motivate in this direction.

Bibliography

[1] D. Achlioptas and C. Moore, Proc. Symp. on Theory of Computing (STOC) 2002.

[2] D. Achlioptas, Phd. Thesis. `http://www.research.microsoft.com/~optas/papers/thesis.ps`

[3] Achlioptas, D. *A survey of lower bounds for random 3-sat via differential equations.* Theoretical Computer Science 265 159–185 (2001)

[4] J. Culberson and Ian P. Gent, Theor. Comp. Sci **265** 227 (1991).

[5] D. Achlioptas, C. Moore, Random k-SAT: Two Moments Suffice to Cross a Sharp Threshold, preprint (2002)

[6] D. Achlioptas and F. Friedgut. A sharp threshold for k-Colorability. Random structures and algorithms, Vol 14, 1, pp 63-70. (1999).

[7] D. Achlioptas. Private communication.

[8] D. Aldous, Random Structures and Algorithms **18** 381 (2001)

[9] K. Appel and W. Haken, Illinois J. Math. **21**, 421 (1977); Illinois J. Math. **21**, 491 (1977).

[10] W. Barthel, A.K. Hartmann,*Clustering analysis of the ground-state structure of the vertex cover problem*, cond-mat/0403193

[11] D. Battaglia, M. Kolář, R. Zecchina, Minimizing energy below the glass thresholds, preprint `cond-mat/0402529`, to appear on Phys. Rev. E

[12] D. Battaglia, A. Braunstein, J. Chavas, R. Zecchina, in preparation.

[13] K. Binder and A. P. Young, Rev. Mod. Phys. **58**, 801 (1986); M. Mézard, G. Parisi and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore 1987); K. H. Fisher and J. A. Hertz, *Spin Glasses* (Cambridge University Press, Cambridge U.K. 1991).

[14] G. Biroli and R. Monasson, and M. Weigt, Eur. Phys. J. **B 14**, 551 (2000)

[15] A. Braunstein, M. Leone, F. Ricci-Teresenghi, R. Zecchina. *Complexity transitions in global algorithms for sparse linear systems over finite fields.* J. Phys. A: Math. Gen. 35 7559-7574, 2002.

[16] A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, R. Zecchina, *Polynomial iterative algorithms for coloring and analyzing random graphs,* Phys. Rev. E 68, 036702 (2003)

[17] A. Braunstein, M. Mézard, R. Zecchina, Survey propagation: an algorithm for satisfiability, preprint 2002, to appear in *Random Structures and Algorithms*, cs.CC/0212002

[18] A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina, preprint cond-mat/0212451 (2002)

[19] A. Braunstein, V. Napolano, R. Zecchina *Clustering in random SAT*, in preparation

[20] A. Braunstein, R. Zecchina, *Survey propagation as local equilibrium equations.* J. Stat. Mech: Theor. Exp. P06007 (2004)

[21] G. J. Chaitin, M.A. Auslander, A.K. Chandra, J. Cocke, M.E. Hopkins, and P. Markstein, *Computer Languages* **6**, 47 (1981).

[22] S. Cocco, R. Monasson, A. Montanari, and G. Semerjian, Approximate analysis of search algorithms with "physical" methods, preprint cs.CC/0302003 (2003)

[23] S. Cocco, O. Dubois, J. Mandler, and R. Monasson, preprint `cond-mat/0206239`.

[24] J. Culberson. Graph Coloring Page, `http://www.cs.ualberta.ca/~joe`

[25] S. Y. Chung, G.D. Forney, T.J. Richardson, R. Urbanke, IEEE Communications Letters, vol. 5, no. 2, 58 - 60 (2001)

[26] O. Dubois, Y. Boufkhad, and J. Mandler, Typical random 3-SAT formulae and the satisfiability threshold, in *Proc. 11th ACM-SIAM Symp. on Discrete Algorithms*, 124 (San Francisco, CA, 2000); A. Kaporis, L. Kirousis, and E. Lalas, The probabilistic analysis of a greedy satisfiability algorithm, in *Proceedings of the 4th European Symposium on Algorithms* (ESA 2002), to appear in series: Lecture Notes in Computer Science, Springer

[27] O. Dubois, R. Monasson, B. Selman and R. Zecchina editors. Special Issue on *NP-hardness and Phase transitions*, Theor. Comp. Sci. **265**, Issue: 1-2 (2001)

[28] O. Dubois, J. Mandler: *The 3-XORSAT Threshold.* FOCS 2002: 769-778

[29] P. Erdös and A. Rènyi, Publ. Math. (Debrecen) **6**, 290 (1959).

[30] J. Franco, Theoretical Computer Science **265**, 147 (2001); D. Achlioptas, G. Sorkin, *41st Annu. Symp. of Foundations of Computer Science, IEEE Computer Soc. Press*, 590 (Los Alamitos, CA, 2000)

[31] S. Franz and M. Leone, preprint `cond-mat/0208280`.

[32] F. Friedgut. Sharp Thresholds of Graph Proprties, and the k-sat Problem. *J. Amer. Math. Soc.* 12 (1999), no. 4, 1017–1054

[33] M. R. Garey and D. S. Johnson, *Computers and intractability* (Freeman, New York, 1979).

[34] F. Guerra, Comm. Math. Phys. **233**, 1 (2003); S. Franz, M. Leone J. Stat. Phys. **111**, 535 (2003)

[35] F. Guerra and F.L. Toninelli, preprint `cond-mat/0208579`.

[36] B. Hayes, On the Threshold, AmSci online, 347, http://www.americanscientist.org/content/AMSCI/AMSCI/ArticleAltFormat/2003423105145_546.pdf

[37] T. Hogg, B.A. Huberman, C. Williams, C. (eds), Artificial Intelligence **81** I & II (1996)

[38] T. Heskes. Stable fixed points of loopy Belief Propagation are minima of the Bethe free energy. In S. Thrum S. Becker and K. Obermayer, editors. *Advances in Neural Information Processing Systems* 15, pp. 343-350, MIT Press, Cambridge, MA (2003)

[39] I. Kanter and H. Sompolinsky, J. Phys. A **20**, L673 (1987).

[40] A. C. Kaporis, L.M. Kirousis, and Y.C. Stamatiou, Electr. J. Comb. **7**, R29 (2000).

[41] S. Kirkpatrick, B. Selman, Science **264**, 1297 (1994)

[42] D. Knuth, *The art of computer programming*, vol. I: fundamental algorithms, Addison-Wesley, New-York (1968)

[43] F. R. Kschischang, B. J. Frey and H.-A. Loeliger, *IEEE Trans. Infor. Theory* **47**,498 (2002)

[44] T. Luczak, Combinatorica **11**, 45 (1991).

[45] D. J. C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, MA (2003)

[46] D. J. C. MacKay, R.M. Neal, IEEE Electronics Letters, 32, 18, 1645-1655 (1996)

[47] Maneva, E. Mossel, E. and M. J. Wainwright. *A new look at survey propagation and its generalizations.* Technical report 669. Department of statistics, U. of California, Berkeley. http://www.arxiv.org/PS_cache/cs/pdf/0409/0409012.pdf

[48] S. Mertens, M. Mézard and R. Zecchina, Threshold values of Random K-SAT from the cavity method, preprint, cs.CC/0309020 (2003)

[49] S. Mertens, M. Mezard, and R. Zecchina, "Dynamic and static thresholds in random K-satisfiability problems", in preparation.

[50] S. Mertens, Computing in Science and Engineering, **4**, 31 (2002).

[51] M. Mézard, G. Parisi, M.A. Virasoro, Europhys. Lett. **1**, 77 (1986); M. Mezard, G. Parisi, Eur. Phys. J. **B 20**, 217 (2001); M. Mezard, G. Parisi, J. Stat. Phys. **111**, 1 (2003)

[52] M. Mézard, G. Parisi, R. Zecchina, Science 297, 812 (2002) (Sciencexpress published on-line 27-June-2002; 10.1126/science.1073287)

[53] M. Mézard and G. Parisi, J. Stat. Phys. 111 (2003)

[54] M. Mézard and G. Parisi, *Eur. J. Phys. B* **20**, 217 (2001)

[55] M. Mézard, R. Zecchina, Phys. Rev. E **66**, 056126 (2002)

[56] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002), published online 27 june 2002, 10.1126/science.1073287

[57] M. Mézard and G. Parisi, J. Stat. Phys **111**, 1 (2003).

[58] M. Mézard and R. Zecchina, *Phys. Rev. E* **66**, 056126 (2002)

[59] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, preprint `cond-mat/0207140`.

[60] R. E. Miller and J. W. Thatcher (eds.), Complexity of Computer Computation, Plenum Press. New York, 85 (1972).

[61] M. Molloy, Rand. Struct. Alg. **7**, 159 (1996).

[62] R. Monasson and R. Zecchina, *Phys. Rev. Lett.* **75**, 2432 (1995).

[63] R. Monasson and R. Zecchina, *Phys. Rev. E* **56**, 1357 (1997).

[64] R. Monasson, *Phys. Rev. Lett.* **75**, 2847 (1995).

[65] A. Montanari, G. Parisi, and F. Ricci-Tersenghi, J. Phys. A 37, 2073 (2004)

[66] A. Montanari, F. Ricci-Tersenghi, On the cooling-schedule dependence of the dynamics of mean-field glasses preprint, cond-mat/0401649 (2004)

[67] A. Montanari and F. Ricci-Tersenghi, preprint `cond-mat/0301591`.

[68] A. Montanari and F. Ricci-Tersenghi, Phys. Rev. Lett. 90, 017203 (2003).

[69] J. van Mourik and D. Saad, *Phys. Rev. E* **66**, 056120 (2002).

[70] R. Mulet, A. Pagnani, M. Weigt and R. Zecchina, *Phys. Rev. Lett.* **89**, 268701 (2002).

[71] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* (Oxford University Press, 2001)

[72] G. Parisi, *On the survey-propagation equations for the random K-satisfiability problem*, ArXiv: xxx.lanl.gov/ps/cs.CC/0212009 (2002); G. Parisi, *On local equilibrium equations for clustering states* ArXiv: xxx.lanl.gov/ps/cs.CC/0212047 (2002)

[73] G. Parisi, Some remarks on the survey decimation algorithm for K-satisfiability, preprint cs.CC/0301015 (2003).

[74] J. Pearl, *Probabilistic reasoning in inteligent systems, network of plausible inference*, Morgan Kaufmann (1988).

[75] B. Pittel, J. Spencer, N. Wormald, J. Comb. Th. B **67**, 111 (1996).

[76] F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, *Phys. Rev. E* **63**, 026702 (2001).

[77] T. Richardson and R. Urbanke, An introduction to the analysis of iterative coding systems, in *Codes, Systems, and Graphical Models*, edited by B. Marcus and J. Rosenthal (Springer, New York (2001)

[78] The most recent implementations of walk-SAT can be downloaded from the SATLIB at `http://www.satlib.org`.

[79] C. E. Shannon, "A mathematical theory of communication", Bell Systems Technical Journal, vol. 27, pp.379-423 (1948).

[80] N. Sourlas, Nature **339**, 693 (1989).

[81] The code for SP for $k-$SAT and GBP is released under the GPL (general public licence) and is available at `http:/www.ictp.trieste.it/~zecchina/SP`.

[82] M. Talagrand *Spin Glasses: A challenge for mathematicians. Mean-field models and cavity method*, Springer-Verlag, to appear.

[83] M. Weigt and A.K. Hartmann, *Phys. Rev. Lett.*, **84**, 6188 (2000).

[84] M. Welling and Y.W. Teh. Belief Optimization: A stable alternative to belief propagation. *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence.* (2001)

[85] N.C Wormald, Ann. Appl. Probab. **5**, 1217 (1995)

[86] F. Y. Wu, Rev. Mod. Phys. **54**, 235 (1982).

[87] J. S. Yedidia, W.T. Freeman and Y. Weiss, Generalized Belief Propagation, in *Advances in Neural Information Processing Systems 13* eds. T.K. Leen, T.G. Dietterich, and V. Tresp, MIT Press 2001, pp. 689-695.

[88] J. S. Yedidia, W.F. Freeman, and Y. Weiss, technical reports TR-2002-35 and TR2004-040, Mitsubishi Electrical Research Laboratories, available online at `http://www.merl.com`

APPENDIX A

SP with external fields and compression

(Joint work with D. Battaglia, J. Chavas and R. Zecchina)

A.1. SP with external fields

The standard SP algorithm described in Section 4.3 has proved to be a powerful tool for the efficient determination of a truth value assignment. Even in the case of large formulas very close to the SAT/UNSAT transition point, the SP-inspired decimation is able to fix approximately 60% of the variables, producing as output a smaller and easily solvable subproblem. This is enough when one is just interested in finding *at least* one solution, but in many tasks the determination of a set of several satisfying assignments, eventually distant among them, can be required. The standard SP equations are often characterized by the existence of a single fixed-point during each convergence step (multiple runs with different random conditions seem to always fall in the same fixed point) and due to the deterministic nature of SID is able then to retrieve only one cluster of solutions for each given problem. The algorithm must then be generalized if one is interested in driving the decimation process toward a desired region of the space of the possible assignments.

It is easy to modify the SP iterations in order to accept a fixed external *probability preconditioning*. Given an arbitrary configuration $\mathbf{f}$ and a real number λ, the original factor graph is modified connecting to each variable node i an additional function node a_i whose G_{a_i}depends only on variable x_i and injects into the system a constant new u-survey $\eta^s_{a_i \to i} = \lambda\delta(s, x_i) + (1-\lambda)\,\delta(s)$ of a predefined value. When updating the ordinary u-surveys $\eta_{a\to i}$, these additional parameters $\eta_{x_i \to i}$ enter the equations (4.3.5,4.3.6) together with all the others η's internal to

the original factor graph, but they are never updated during the decimation process and their value is kept constant in time. They are simply "switched off" when the associated variable is fixed and the factor graph accordingly simplified. The presence of the external probability conditionings tries to drive the evolution of the cavity biases distributions toward the selection of clusters in which the variables are maximally aligned with the externally imposed direction $\mathbf{f}$.

The first remarkable effect of the introduction of an external forcing is the increased efficiency of the decimation itself. The paramagnetic collapse that takes place at a certain point when using the standard SP is avoided by the continuous "stimulation" carried on by the conditioning nodes, and the algorithm becomes able to determine completely a satisfying assignment without recurring to an auxiliary heuristics. The reaction of the internal cavity fields will be able to "repair" eventual contradictions present in the external forcing, if the intensity λ is not too big. Indeed, if the local directions of the external forcing are chosen randomly, a perfect alignment of the variables with the forcing field $\mathbf{f}$ will violate some clauses with probability one, and the iterative application of the zero temperature SP equations will never reach convergence. On the other hand, the imposition of an external field equal to a solution $\mathbf{s}$ with sufficiently large λ will cause an instantaneous polarization of the system along the direction of $\mathbf{s}$, and a perfect retrieval of the solution $\mathbf{s}$.

It is interesting to use the modified SP iterations (denoted by SP-ext in the following) for probing the geometrical structure of the space of ground state assignments of k-SAT. A first experiment can be performed in which an external forcing $\mathbf{f}$ of random direction and of small uniform intensity approximately equal to $\lambda = 0.1$ is imposed. A solution $\mathbf{s_f}$ is typically retrieved, at a normalized Hamming distance $d(\mathbf{s_f}, \mathbf{f}) \stackrel{\text{def}}{=} \frac{1}{n} H(\mathbf{s_f}, \mathbf{f})$ always significantly smaller than 0.5, signaling a non-trivial correlation between $\mathbf{s_f}$ and $\mathbf{f}$. The distance $d(\mathbf{s_f}, \mathbf{f})$ will be denoted in the following as d_p and referred as the a-priori distortion.

If the experiment is repeated with a number of different random fields $\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(\ell)}$ one obtains in general ℓ different solutions $\mathbf{s}_{\mathbf{f}^{(i)}}$. If the

selected k-SAT formula presents a clustered phase the a-priori distortion d_p between a solution and its corresponding forcing will be smaller than the typical distance d_1 between two different solutions $\mathbf{s}_{\mathbf{f}(i)}$ and $\mathbf{s}_{\mathbf{f}(i)}$.

In the case of formulas extracted from the ensemble of random $k-$SAT formulas, distance scale d_1 is approximately 0.39 (average value of the inter-solution distance), smaller than 0.5, indicating that the retrieved solutions are concentrated along a preferential direction in the configuration space. This is an effect due to the fact that for finite size n the number of clauses $b_i^\sigma = |\{a \in i : J_{a,i} = \sigma\}|$ in which a variable i appears with sign σ is not always exactly identical to the number of times $b_i^{-\sigma}$ in which it appears with the opposite sign. The cluster distribution is then concentrated preferentially in a hypercone centered around a direction $\mathbf{b}$ individuated by the signs of the differences $b_i^+ - b_i^-$. An experimental value of q_1 extremely close to 0.5 can indeed be obtained by choosing a random graph ensemble in which b_i^+ is kept strictly equal to b_i^- for every i (see Figure A.1.1). The resulting ensemble of *bar-balanced formulas* has a phase diagram qualitatively similar to plain random k-SAT, but different values of the critical α's (for instance, for $k = 3$, $\alpha_d \simeq 3.0$ and $\alpha_c = 3.4$ only). The larger distance scale q_1 accounts for a more uniform distribution of the (addressable) clusters of solutions.

Another positive effect of the larger homogeneity, is the reduction of the a-priori distortion d_p: the more scattered the clusters, the more likely will be to find a cluster close to any given random forcing vector. The exact value of d_p will depend anyway on the value of α; the larger the complexity and the smaller d_p, in general. We have been able until now to obtain values of d_p down to 0.2 (using fixed-variable-connectivity formulas with $k = 7$), but we are confident that they can still be considerably reduced by choosing appropriate graphs.

The distance scale d_0 typical of the solutions lying inside a same cluster can be measured with a different experimental setup. One starts determining a random solution $\mathbf{s}$. At this point, a forcing $\mathbf{f}^{(d)}$ is generated by taking the solution $\mathbf{s}$ and flipping randomly nd of its spins. A

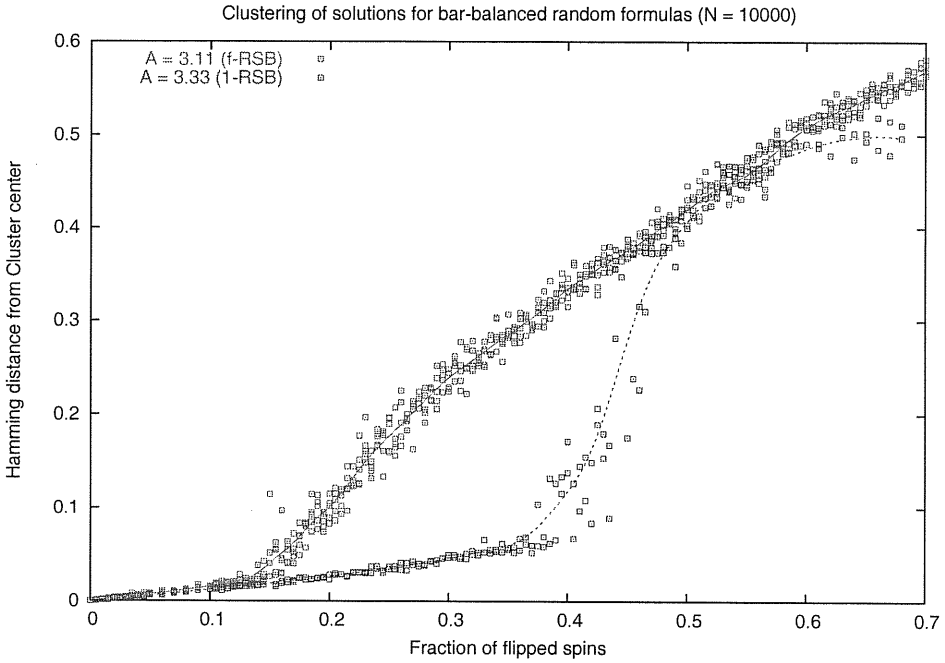

FIGURE A.1.1. Clustering of the solutions. Forcing the decimation along a vector increasingly distant from a specific solution $\mathbf{s}$ produces new satisfying assignments that belongs initially to the same cluster than $\mathbf{s}$. When the Hamming distance between the forcing and $\mathbf{s}$ becomes too large, the obtained solutions "escape" from the cluster. The difference between the f-RSB and the 1-RSB distribution of distances appears evident.

decimation is then performed imposing the resulting forcing with a not too large intensity ($\eta = 0.1$ in the present experiments), and retrieving a new solution $\mathbf{s}^{(d)}$.

The results of the experiments conducted over two bar-balanced formulas with $n = 10000$ variables and $\alpha = 3.11$ and 3.33 are shown in Figure A.1.1.

The normalized Hamming distance between $\mathbf{s}^{(d)}$ and $\mathbf{s}$ is plotted against d. When d is not too large it appears evident that $\mathbf{s}^{(d)}$ continues to belong to the same cluster of $\mathbf{s}$. After a certain critical d_c the retrieved solution escapes from the cluster and the distance from

$\mathbf{s}$ increases much faster (discontinuously when n is large enough). The difference between a typical f-RSB geometry and a 1-RSB landscape can be observed with astonishing clarity in Fig. A.1.1. For a formula taken in the frozen f-RSB phase (bar-balanced ensemble, $\alpha = 3.11$), after a critical distance $d_c \simeq 0.13$, there is a continuum spectrum of inter-solution distances between an intra-cluster distance scale $d_0 \leq 0.05$ and the inter-cluster distance scale $d_1 \geq 0.5$. In the case of a formula in the 1-RSB stable region (bar-balanced ensemble, $\alpha = 3.33$), the transition between $d_0 \leq 0.05$ and $d_1 \simeq 0.5$ is much sharper and takes place around $d_c \simeq 0.4$ (although some noise due to the tails of the distance probability distribution tail is present and causes a slight deviation from a perfect step shape, in the case of finite samples).

An ulterior confirmation of the clustering hypothesis comes from the analysis of the reciprocal distances between different solutions. Two solutions $\mathbf{s}^{(d)}$ taken from the "intra-cluster plateau" ($d < d_c$) in Fig. A.1.1 are at a distance of the same scale of the measured d_0; the distance between a solution $\mathbf{s}^{(d)}$ with $d < d_c$ and another one with $d > d_c$ is on the other hand significantly larger than the scale d_0 (it is of scale d_1 in the 1-RSB case); the same happens when considering two solutions with $d > d_c$, indicating that the "out-of-cluster" plateau is composed by solutions belonging to many different clusters approximately equidistant among them.

The capacity of the SP-ext iterations of probing, detecting and addressing the cluster structure of the ground state space can be exploited for practical purposes. For formulas belonging to the 1-RSB stable region, $d_c >> d_0$. This means that a solution vector corrupted in no more than d_c variable positions can still be used for addressing the same cluster of the original solution $\mathbf{s}$. An SP-ext decimation conducted using the corrupted vector as forcing can then be performed in order to produce a new vector $\mathbf{s}'$ which will be only at distance d_0 from the correct solution $\mathcal{S}$. A careful choice of the factor graph will allow to obtain quite remarkable solution-reconstruction capabilities.

The native cluster structure can also be exploited for realizing a lossy data compressor.

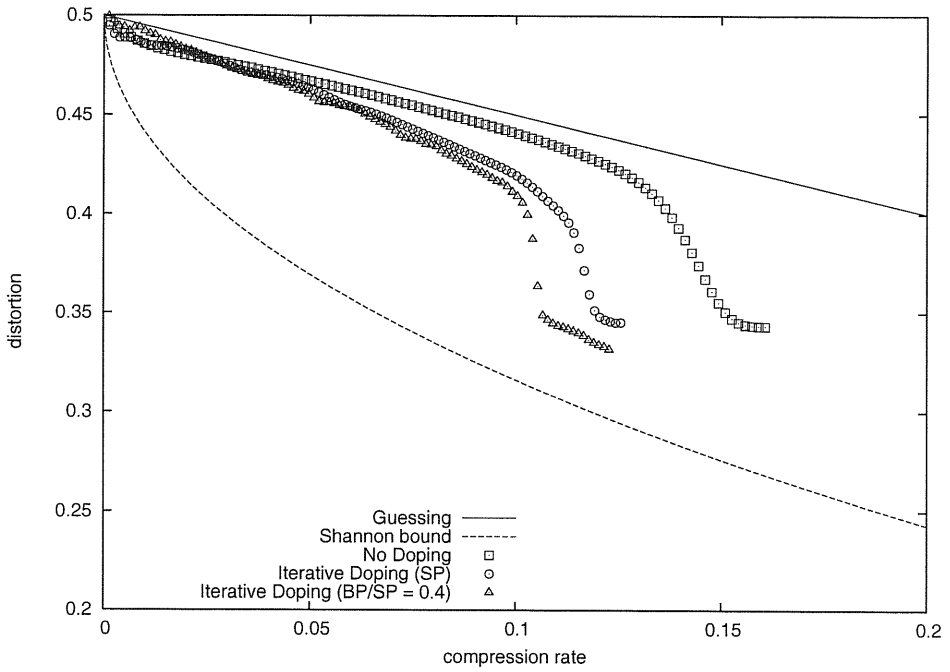

FIGURE A.1.2. Rate-distortion profile of a non-optimized SP-ext lossy compressor. The cluster-reconstruction capabilities of SP-ext are clearly shown by the deviation form the random guessing rate-distortion profile. Improvement in the codec performance can be obtained by the use of the iterative doping stage.

Let us consider a random binary vector $\mathbf{f}$ and a given fixed factor graph. Let impose a forcing field of moderate intensity ($\lambda_{com} = 0.3$ in the present experiments) along the direction $\mathbf{f}$. A solution $\mathbf{s_f}$ will be obtained, at the typical a-priori distance d_p from the forcing $\mathbf{f}$. Suppose now to take as forcing a subvector $\mathbf{c}$, composed of just the first nR components of the solution $\mathbf{s_f}$. If nR is larger than a certain critical nR_c and if the forcing λ_{dec} is sufficiently intense, a new vector $\mathbf{s}'$ still belonging to the same cluster of $\mathbf{s_f}$ will be retrieved, lying consequently at a distance of order d_0 from $\mathbf{s_f}$. Instead of performing a complete decimation in the decoding stage, one might just impose a really intense forcing along $\mathbf{c}$ ($\lambda_{dec} = 0.95$ in the present experiments) and fix all the variables according to the ranking obtained after the first convergence.

Because $d_0 << d_p$, the reconstructed string $\mathbf{f}'$ will be still at a distance from the original $\mathbf{f}$ comparable with the scale d_p.

The whole procedure can be considered as a lossy compression-decompression (codec) cycle [**45**]. The initial forcing vector $\mathbf{f}$ can be considered as a signal block emitted from a random uncorrelated binary source, and the solution chunk $\mathbf{c}$ plays the role of compressed block. After the decoding stage, one retrieves finally $\mathbf{f}'$ as decompressed string. If the critical compression rate R_c and the a-priori distortion d_p are small enough, and if the algorithm parameters λ_{com} and λ_{dec} are carefully tuned, one can obtain a large compression factor without having a too large distortion after the codec-cycle. It should be noticed that this algorithm, exploiting the built-in quantization of the solution-space, makes use of message-passing techniques both in the compression and in the decompression stage, differently from the case of LDPC and Turbo codes in which algorithms analogue to BP are used only for the decoding part [**80, 77**].

In Fig. A.1.2 it is possible to observe the rate-distortion curve obtained for a random uncorrelated source and for a factor graph with $n = 33600$, $k = 5$ and a constant variable node connectivity $\gamma = 84$. It is quite unlikely that our choice represent an optimal one: in the literature is known [**46, 25**] that the best code graphs are often highly inhomogeneous in the connectivity of both the function and the variable nodes, and that dramatic variation in the performance can be obtained realizing a suitable code-graph optimization.

The theoretical Shannon Bound is reported in Fig. A.1.2 toghether with the straight line referring to the rate-distortion profile of the trivial random guessing; this strategy consists simply in taking the input block $\mathbf{f}$, "compressing" it by cutting out and disregarding its last $(1 - R_c)n$ bits and reconstructing finally in a random way the missing information bits. One can see that the rate-distortion curve of the SP-ext codec deviates in a characteristic way from the random guessing line, exhibiting in a clear way its peculiar cluster-reconstruction properties.

The performance of the SP-ext codec can be considerably improved constructing the subvector $\mathbf{c}$ in a non random way, but selecting appropriately the most balanced variables. After the determination of $\mathbf{s_f}$, SP is run in absence of any external field and the most balanced variable (that is the one which maximizes $\min\left(H_i^+, H_i^-\right)$) is individuated. The spin value of this most balanced variable is then read from the solution $\mathbf{s_f}$ and used both for fixing the variable and as *first entry* of the compressed vector $\mathbf{c}$. These steps are iterated until when the desired length nR of $\mathbf{c}$ has been reached.

There is no need to store information about the location of the variables whose values in $\mathbf{s_f}$ are written in the entries of $\mathbf{c}$. In the decoding stage indeed, the determination of the most balanced variables is repeated, obtaining exactly the same result than in the coding stage. But during the decoding the way of fixing the variables is read sequentially in the entries of the compressed string $\mathbf{c}$. When the string $\mathbf{c}$ has been completely read, and the first nR most balanced variables have then been fixed according to $\mathbf{s_f}$, all variables left are fixed after just one single convergence to the direction of $\max\left(H_i^+, H_i^-\right)$ according to the standard SP decimation criterion.

This modification of the codec cycle is known as *iterative doping*. The distortion of the reconstructed vector $\mathbf{f}'$ is not particularly improved but it is achieved at a considerable better rate (see the second curve in Fig. A.1.2. We have empirically obtained a further improvement, performing the iterative doping stage with the BP/SP interpolating parameter ρ set at 0.4 (third curve in Fig. A.1.2), but we do not have any rigorous argument for explaining this interesting behavior.

A.2. Conclusion

The SP algorithm has proven to be an extremely powerful tool for the resolution of various random combinatorial optimization problems presenting a clustered phase. While the proliferation of metastable states is harmful for any local search heuristics, and the clustered distribution of the ground state seems to prevents ordinary BP from convergence, the SP procedure, based from the very beginning on an ansatz

equivalent to a 1-RSB description, can successfully determine the most biased variables and construct a satisfying assignment by decimation.

SP-ext, an extension of SP making use of external probability conditionings, allows to probe the geometrical structure of the solution space at an exciting level of detail, confirming the theoretical scenarios predicted in the past years. The capability of addressing and retrieving specific cluster of states open the road toward the realization of a new family of lossy compressors and associative memories, even if much work has yet to be done in order to achieve the optimal performance and to generalize the algorithms to make them compliant with more realistic and useful applications.