Entropy landscape and non-Gibbs solutions in constraint satisfaction problems

L. Dall'Asta* and A. Ramezanpour†
The Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34014 Trieste, Italy

R. Zecchina‡
Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy

(Received 26 October 2007; published 17 March 2008)

We study the entropy landscape of solutions for the bicoloring problem in random graphs, a representative difficult constraint satisfaction problem. Our goal is to classify which types of clusters of solutions are addressed by different algorithms. In the first part of the study we use the cavity method to obtain the number of clusters with a given internal entropy and determine the phase diagram of the problem—e.g., dynamical, rigidity, and satisfiability-unsatisfiability (SAT-UNSAT) transitions. In the second part of the paper we analyze different algorithms and locate their behavior in the entropy landscape of the problem. For instance, we show that a smoothed version of a decimation strategy based on belief propagation is able to find solutions belonging to subdominant clusters even beyond the so-called rigidity transition where the thermodynamically relevant clusters become frozen. These nonequilibrium solutions belong to the most probable unfrozen clusters.

DOI: 10.1103/PhysRevE.77.031118 PACS number(s): 05.70.Fh, 89.20.Ff, 75.10.Nr

I. INTRODUCTION AND MOTIVATIONS

Many disciplines have at their root random constraint satisfaction problems (CSPs). Examples are information theory where they are used to design error-correcting codes [1,2] or computer science where they constitute elementary models for studying the onset of exponential regimes in algorithms [3]. More generally, random CSPs capture some of the optimization aspects of complex systems found in physics (e.g., spin-glasses and packing problems), in economics (e.g., financial markets) [4,5], and in biology (e.g., gene networks reconstruction and learning in neuroscience [6,7]). A random CSP is characterized by an extensive list of constraints, each one forbidding some of the joint assignments of the (discrete) variables it involves. In packing problems, for instance, overlapping positions of the elementary tiles on a given lattice are forbidden. Given an instance of a CSP, one wants to know whether there exists a solution—that is, an assignment of the variables—which satisfies all the constraints (e.g., a proper tiling or a proper coloring of a graph). When such an assignment exists, the instance is called SAT and one wants to find it. Most of the interesting CSPs are *NP*-complete [8,9]: in the worst case the number of operations needed to decide whether an instance is SAT or not is expected to grow exponentially with the number of variables. The interesting limit for random CSPs is the thermodynamic one where both the number N of independent variables and the number M of constraints go to infinity at fixed constraint density $\alpha \equiv M/N$. The most intriguing phenomenon is certainly the appearance of sharp thresholds [10,11]. At some critical ratio α_s the probability of the existence of solutions jumps from 1 to 0. Just below such a threshold, most of the known heuristic algorithms are observed to undergo a dramatic slowing down. Such a phenomenon has been put in connection with the onset of a clustering phase, where the space of solution becomes divided in a large (exponential) number of different clusters (or states) and variables develop nontrivial long-range correlations.

The scope of this study is to go a step further in the statistical physics analysis of the connection between clustering and the behavior of algorithms. By an analytic estimate of the internal entropy of the clusters found by different algorithms on large problem instances and by a large deviation analysis of clusters distribution with respect to their internal entropy, we are able to display which types of clusters are addressed by different algorithms. For random CSPs in the clustering phase, we observe that local search algorithms may be attracted by a large spectrum of clusters (not surprisingly as happens in out-of-equilibrium physical systems). Quite surprisingly, we also show that there exist simple message-passing (MP) processes that are capable of finding efficiently solutions even in the harder region where the thermodynamically dominant clusters become frozen. In such a region local search algorithms are observed numerically to undergo an exponential slowing down due to the global rearrangements needed to correct the errors made along the search process. On the contrary, the MP processes that we study continue to find solution efficiently by addressing clusters which are more rare than the dominating ones (i.e., those which would be seen by sampling solutions with uniform measure over the solution space). These results, together with the evidence coming from fully connected CSPs that even frozen solutions may be found by MP [6], shed light on how MP algorithms can be utilized and suggest that some further algorithmic progress may be at hand.

In what follows we first provide a very brief review of the known results and next apply our arguments to the so-called bicoloring problem, which has some analytical advantages compared to other *NP*-complete problems like *K*-satisfiability (*K*-SAT) or coloring while retaining all the conceptual features.

*dallasta@ictp.it
†aramezan@ictp.it
‡riccardo.zecchina@polito.it

The paper is organized as follows. In Sec. II an intuitive introduction to clustering is given. The definition of the specific problem under study is provided in Sec. III together with a summary of previously known results. The cavity method for large deviations is presented in Sec. IV. The numerical methods used to solve the cavity equations and extract the complexity curves are described in Sec. V. In Sec. VI the equations are solved in some special cases in order to obtain the main properties of the phase diagram of the problem. The algorithms used to find solutions and locate them in the entropy landscape are described in Sec. VII. Section VIII is devoted to a summarization of the main results of the paper and to giving some concluding remarks and discussing perspectives. In the Appendixes we report the details of deriving the cavity equations and some quantities that are essential in computing the free energy.

II. GEOMETRY OF SOLUTIONS AND FREEZING

The set of solutions of a random CSP should be thought of as a portion of the phase space which may undergo a fragmentation into clusters for values of the density of constraints right below the SAT threshold. This scenario can be made rigorous in few cases, the simplest one being the random XOR-satisfiability (XOR-SAT) problem [12,13]: the density of constraints where clustering appears corresponds to the percolation of particular structures in the underlying graph of constraints. This fact can be used to define clusters, which, by linearity of XOR-SAT, turn out to be all identical. One may prove that a finite fraction of the variables have to be frozen—that is, must take the same value in all solutions belonging to a cluster. This picture is, however, far from general: both the definition of clusters and the analysis of their fluctuations in size are difficult tasks, which require application of the cavity method in a rather advanced setting [14–18]. One important feature of clusters in CSPs concerns the presence or absence of frozen variables. It may happen that clusters with frozen variables coexist with totally unfrozen clusters of larger internal entropy, with large effects on the hardness of the associated combinatorial optimization problem. The intuitive reason why the presence of frozen variables is believed to be relevant is well summarized by the idea of rearrangements [18]: "given an initial solution of a CSP and a variable i that one would like to modify, a rearrangement is a path in configuration space that starts from the initial solution and leads to another solution where the value of the ith variable is changed with respect to the initial one. The minimal length of such a path is a measure of how constrained the variable i was in the initial configuration. In intuitive terms this length diverges with the system size when the variable was frozen in the initial cluster." The idea is that when freezing takes place in Gibbs states, then the rearrangements are responsible for a critical slowing down of local search algorithms. On the contrary, when dominant clusters are not frozen, even relatively simple local algorithms may find solutions by incrementally adding constraints until the full problem is satisfied. Recent arguments and numerical studies have shown that one can still obtain a solution beyond the dynamical transition as long as the so-called *jamming transition* has not occurred [19]. Following Ref. [19], one can imagine adding the constraints one by one, in each step recording the number of flips that are required in order to find the new solution. Close to the jamming transition the number of flips diverges and makes it difficult to find a solution to the CSP [18].

A constrained satisfaction problem is defined by a set of constrains $\mathcal{C}=\{I_a[\underline{\sigma}]\in\{0,1\}|a=1,\ldots,M\}$ on a number of variables. The constraints depend on the configuration of variables $\underline{\sigma}\equiv\{\sigma_i|i=1,\ldots,N\}$ and the problem is called *satisfiable* if all constraints are satisfied—i.e. $I_a=1, \forall\ a$. A *solution* of the problem is a configuration of the variables that satisfies all constraints. In analogy with statistical physics models, we define the *energy* $E[\underline{\sigma}]$ of configuration $\underline{\sigma}$ as the number of unsatisfied constraints in $\underline{\sigma}$. Given an instance of a CSP, one is interested in deciding whether it is satisfiable (i.e., $E[\underline{\sigma}]=0$) and, in such a case, in explicitly finding solutions to the problem.

More generally, one can define an ensemble of instances of the problem considering all possible random assignments of the M constraints among N variables, with fixed density of constraints $\alpha=M/N$. Varying α it was shown that the system passes from a phase in which it is always possible to find a solution, the *SAT phase*, to the *UNSAT phase* where a fraction of the constraints are not satisfied. Examples of studies for root problems such as the random satisfiability problem, XOR-SAT, and graph coloring can be found in [20–25]. The main tool for analyzing the satisfiability of typical problem instances is the *cavity method*, originally developed to study the thermodynamic properties of diluted spin-glass systems [26] and recently reconsidered in the context of CSPs [22,27,28]. Actually, the cavity method at the ensemble level allows one to study typical properties as well as large deviations from typical behaviors [13,14,16]. The key feature of the cavity method which is of interest for computer science stems from the discovery that it can be converted to an algorithm for analyzing single problem instances [22,33], becoming an efficient solver for the random combinatorial problems.

III. DEFINITION OF THE PROBLEM AND KNOWN RESULTS

Consider a hypergraph of N nodes $i=1,\ldots,N$ and M hyperedges $a=1,\ldots,M$. For simplicity we consider regular random hypergraphs, or (K,L)-hypergraph, where each hyperedge connects K nodes and each node contributes in $L=KM/N=K\alpha$ hyperedges. A node in this hypergraph has state $\sigma_i\in\{0,1\}$, and each hyperedge imposes a constraint on the values of the associated variables. In the bicoloring problem this constraint just forbids the configurations in which all the nodes belonging to an hyperedge have the same value. In the context of circuit logic the bicoloring problem is known as not-all-equal-satisfiability (NAE-SAT) problem, in physics it is a spin model with antiferromagnetic interactions.

We may represent bicoloring as a factor graph [30]. This is a bipartite graph where the variables and constraints are represented with two different types of nodes: variable and function nodes, respectively. Each function node is con-

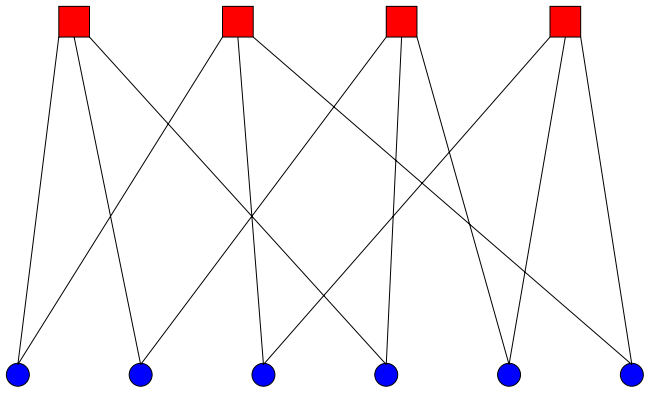

FIG. 1. (Color online) A regular random factor graph with function nodes (squares) of degree $K=3$ and variable nodes (circles) of degree $L=2$.

nected to all the variable nodes that should satisfy the associated constraint. Figure 1 shows an example of regular random factor graph.

The hypergraph bicoloring problem is *NP*-complete for $K \geq 3$ [31]. The case $K=3$ with a Poisson degree distribution for the variable nodes has already been studied in [32]. The authors found dynamical and SAT and UNSAT transitions within the single- and multiple-cluster approximations. In spin-glass language these approximations are called the replica symmetric (RS) and one-step replica symmetry breaking (1RSB) approximations, respectively. In the latter case the authors only consider the most numerous clusters.

The intensive entropy s is defined by the number of solutions, $\mathcal{N}=e^{Ns}$. Using the Bethe approximation in the replica-symmetric phase we find the entropy as

$$s^{RS} = \ln\left[2\left(1-\frac{1}{2^{K-1}}\right)^{L}\right] - (K-1)\alpha \ln\left[1-\frac{2}{2^{K}}\right]. \quad (1)$$

This quantity vanishes at

$$L_s^{RS} = -K\frac{\ln 2}{\ln\left(1-\frac{1}{2^{K-1}}\right)}, \quad (2)$$

where for $K \gg 1$ gives

$$L_s^{RS} \simeq K2^{K-1} \ln 2\left[1+O\left(\frac{1}{2^K}\right)\right]. \quad (3)$$

If there exists more than one cluster of solutions, we define the complexity Σ by $\mathcal{N}_c=e^{N\Sigma}$ where $\mathcal{N}_c$ is the number of clusters. Notice that for very large N the above complexity is dominated by typical clusters. In the 1RSB approximation and considering only typical clusters, the complexity reads [32]

$$\Sigma_{typ} = \ln[A_L] - (K-1)\alpha \ln\left[1-\frac{1}{2}(1-\eta)\left(1-\frac{\eta^{L-1}}{A_{L-1}}\right)\right], \quad (4)$$

where

TABLE I. Numerical values of L_d^0 and L_s (in the RS and 1RSB approximations). In each case we have given the smallest integer degree larger than or equal to the precise value.

K	L_d^0	L_s^{RS}	L_s^{1RSB}
3	–	8	7
4	17	21	20
5	40	54	53
6	91	131	130
7	200	309	307
8	428	708	705
9	905	1594	1592
10	1894	3546	3543

$$A_L = 2\left(1-\frac{1-\eta}{2}\right)^{L} - \eta^{L} \quad (5)$$

and η is determined by the equation

$$\eta = 1-2\left[\frac{1}{2}\left(1-\frac{\eta^{L-1}}{A_{L-1}}\right)\right]^{K-1}. \quad (6)$$

A nontrivial solution ($\eta \neq 1$) for the above equation results in a nonzero complexity. Let us assume $K \gg 1$ and find the point where for the first time a nontrivial solution appears. We try $\eta=1-\frac{c}{2^{K-1}}$ and find c in a self-consistent way. From the above equation we obtain

$$\frac{c}{2} \simeq \exp\left[-\frac{K-1}{2}e^{-c(L-1)/2^K}\right]. \quad (7)$$

It means that to have a finite solution for c we need L diverges as

$$L_d^0 \simeq 2^K\frac{1}{c}\left[\ln K - \ln 2 - \ln\ln\left(\frac{2}{c}\right) + o(1)\right]. \quad (8)$$

At the SAT-UNSAT transition Σ_{typ} vanishes and we can use this fact to determine α_s. This value behaves, asymptotically, as α_s^{RS} in Eq. (3). In Table I we compare the numerical values of L_d^0 and L_s obtained with the above methods.

IV. CAVITY METHOD: A LARGE DEVIATION STUDY

A more complete picture of the distribution of clusters is given by a large deviation study [14,16,29]. We define the partition function at zero temperature as

$$Z = \sum_{\underline{\sigma}} \prod_a I_a(\sigma_{\partial a}) \exp\left(x\sum_i (\sigma_i - \sigma_i^*)^2\right). \quad (9)$$

Here x is a Lagrange multiplier that controls the distance between the solutions and a reference point $\underline{\sigma}^*$ and $\sigma_{\partial a} = \{\sigma_i | i \in V(a)\}$ where $V(a)$ is the set of neighbors of function node a. For $x=0$ we recover the total number of solutions. If there is only one cluster of solutions, we can safely use the standard *belief propagation* (BP) equations [30] to obtain an estimate of the cavity marginals (see Appendix A):

$$\mu_{i\to a}(\sigma_i) = \frac{1}{Z_{i\to a}} \sum_{\sigma_{\partial i\to a}} \left(\prod_{b\in V(i)\backslash a} I_b(\sigma_{\partial b}) \times \left[\prod_{j\in V(b)\backslash i} \mu_{j\to b}(\sigma_j) \right] \right) e^{x(\sigma_i - \sigma_i^*)^2}. \quad (10)$$

Here $Z_{i\to a}$ is a normalization constant, $V(i)$ is the set of neighbors of variable node i, and $\sigma_{\partial i\to a} = \{\sigma_j | j\in V(b), b\in V(i)\backslash a\}$. We will write the above equation in short as

$$\mu_{i\to a}(\sigma_i) = \mathcal{S}[\mu_{j\to b}]. \quad (11)$$

We also define the free energy $f(x)$ as

$$Z = e^{Nf(x)} = \sum_d e^{N[s(d)+xd]}, \quad (12)$$

where $d=\frac{1}{N}\sum_i(\sigma_i-\sigma_i^*)^2$ and $e^{Ns(d)}$ is the number of solutions at distance d from the reference point. In the Bethe approximation,

$$f(x) = \sum_i \Delta f_i - \sum_a (K_a - 1)\Delta f_a, \quad (13)$$

where

$$e^{N\Delta f_i} = \sum_{\sigma_i} \left[\prod_{a\in V(i)} \mu_{a\to i}(\sigma_i) \right] e^{x(\sigma_i - \sigma_i^*)^2},$$

$$e^{N\Delta f_a} = \sum_{\sigma_{\partial a}} I_a(\sigma_{\partial a}) \prod_{i\in V(a)} \mu_{i\to a}(\sigma_i). \quad (14)$$

Using the above free energy we can determine the entropy $s(d)$ by a Legendre transform:

$$s(d) = \min_x [f(x) - xd(x)], \quad d(x) = \frac{\partial f(x)}{\partial x}. \quad (15)$$

If the replica symmetry is broken, we would have different clusters of solutions and the cavity marginals fluctuate from one cluster to another one. This is described by a generalized partition function defined by

$$\mathcal{Z} \equiv e^{N\mathcal{F}(m)} = \sum_c e^{mNs_c} = \int ds\, e^{N[\Sigma(s)+ms]}. \quad (16)$$

Here c labels the clusters and s_c is the internal entropy of cluster c. Again, in the Bethe approximation,

$$N\mathcal{F}(m) = \sum_i \ln \mathcal{Z}_i - \sum_a (K_a - 1)\ln \mathcal{Z}_a, \quad (17)$$

where

$$\mathcal{Z}_i \equiv \int \prod_{a\in V(i)} \prod_{j\in V(a)\backslash i} d\mathcal{P}_{j\to a}[\mu_{j\to a}] e^{mN\Delta s_i},$$

$$\mathcal{Z}_a \equiv \int \prod_{i\in V(a)} d\mathcal{P}_{i\to a}[\mu_{i\to a}] e^{mN\Delta s_a}. \quad (18)$$

Having the generalized free energy we can determine the complexity $\Sigma(s)$ by a Legendre transformation:

$$\Sigma(s) = \min_m [\mathcal{F}(m) - ms], \quad s(m) = \frac{\partial \mathcal{F}(m)}{\partial m}. \quad (19)$$

In Appendix B we explain the origin of the above relations. Notice that $\Delta s_i = \Delta f_i(x=0)$ and $\Delta s_a = \Delta f_a(x=0)$, where the free energy shifts are given by Eq. (14). Moreover, $\mathcal{P}_{i\to a}[\mu_{i\to a}]$ is the probability that, in a randomly selected cluster, we find the cavity marginal $\mu_{i\to a}$ on edge (i,a) of the factor graph. This probability distribution is determined by the following self-consistency equation:

$$\mathcal{P}_{i\to a}[\mu_{i\to a}] = \frac{1}{\mathcal{Z}_{i\to a}} \int \prod_{b\in V(i)\backslash a} \prod_{j\in V(b)\backslash i} d\mathcal{P}_{j\to b}[\mu] e^{mN\Delta s_i} \times \delta(\mu_{i\to a} - \mathcal{S}[\mu]), \quad (20)$$

where $\mathcal{Z}_{i\to a}$ is a normalization constant and $\mathcal{S}[\mu]$ is the same as in Eq. (11) with $x=0$. The factor $e^{mN\Delta s_i}$ is to sample correctly the clusters when we add the new variable i. Let us multiply the two sides of Eq. (20) by $2\mu_{i\to a}(\sigma)$ to find the new probability distribution $\mathcal{Q}^{\sigma}_{i\to a}[\mu] = 2\mu_{i\to a}(\sigma)\mathcal{P}_{i\to a}[\mu]$, which will be useful in the special case of $m=1$. On the right-hand side we can replace $\mu_{i\to a}(\sigma)$ with its definition in Eq. (10). Rearranging the terms we get

$$\mathcal{Q}^{\sigma_i}_{i\to a}[\mu] = \frac{2}{\mathcal{Z}_{i\to a}} \int \sum_{\sigma_{\partial i\to a}} \prod_{b\in V(i)\backslash a} I_b(\sigma_{\partial b}) \times \prod_{j\in V(b)\backslash i} \left(\frac{1}{2} d\mathcal{Q}^{\sigma_j}_{j\to b}[\mu] \right) e^{(m-1)N\Delta s_i} \delta(\mu_{i\to a} - \mathcal{S}[\mu]). \quad (21)$$

In the following we will split $\mathcal{P}_{i\to a}[\mu]$ into frozen and unfrozen parts as

$$\mathcal{P}_{i\to a}[\mu_{i\to a}] = \frac{1-\pi}{2}[\delta(r) + \delta(r-1)] + \pi\rho(r), \quad (22)$$

where $\mu_{i\to a}(0)=r$, $\mu_{i\to a}(1)=1-r$, and $\rho(r)$ is the probability distribution of unfrozen fields. The above arguments become much simpler in random (K,L)-hypergraph where all the links and nodes are statistically equivalent. For example, Eq. (17) is replaced with

$$\mathcal{F}(m) = \ln\left[\int D_i\mathcal{P}[\mu] e^{mN\Delta s_i} \right] - \alpha(K-1)\ln\left[\int D_a\mathcal{P}[\mu] e^{mN\Delta s_a} \right], \quad (23)$$

where $\int D_i\mathcal{P}[\mu]$ and $\int D_a\mathcal{P}[\mu]$ denote integrations over all cavity marginals that contribute in Δs_i and Δs_a, respectively.

V. ENTROPY LANDSCAPE: NUMERICAL METHOD

The main equation that we should solve is Eq. (20). One can use the population dynamics method [27,28] to get rid of summing over a large number of continuous variables.

A. In a single hypergraph

Given the factor graph we represent $\mathcal{P}_{i\to a}[\mu]$ by a population of $\mathcal{N}_p$ cavity probabilities (or fields), Fig. 2. At the

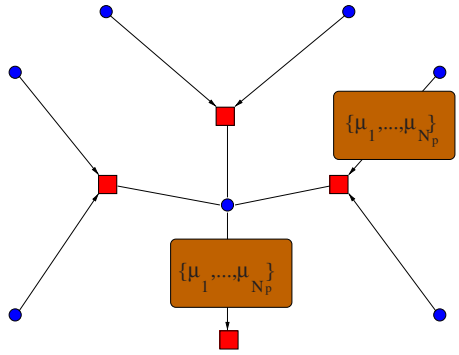

FIG. 2. (Color online) Population dynamics works with a population of fields on each link of the factor graph.

initial point all the cavity fields are of a frozen kind with equal probability for $r=0$ and $r=1$. With this initial condition we will not miss a nontrivial solution with frozen fields, if any. At each step of the population dynamics we select a link $(i\to a)$ randomly and do in the following way.

(i) For each $b\in V(i)\backslash a$ and $j\in V(b)\backslash i$, randomly select a member of the population on link $(j\to b)$.

(ii) Using these $(L-1)(K-1)$ fields, calculate the new $\mu_{i\to a}$ by Eq. (10).

(iii) Calculate the weight $w_{i\to a}=e^{mN\Delta s_i}$ from Eq. (14) at $x=0$. This weight is zero if there is any contradiction.

(iv) With probability $\frac{w_{i\to a}}{w_{i\to a}^{\max}}$ replace a randomly selected member of the population with the new one. Here $w_{i\to a}^{\max}$ is the maximum weight $w_{i\to a}$ observed in the evolution from the beginning.

In a sweep of the algorithm we choose all the links of the factor graph randomly. Having the populations we can obtain the free energy as

$$N\mathcal{F}(m)=\sum_i \ln \mathcal{Z}_i-(K-1)\sum_a \ln \mathcal{Z}_a,$$

$$\mathcal{Z}_i=\langle e^{mN\Delta s_i}\rangle_{pop},$$

$$\mathcal{Z}_a=\langle e^{mN\Delta s_a}\rangle_{pop}, \tag{24}$$

where $\langle\cdots\rangle_{pop}$ means averaging over the populations. We stop the updates as soon as the free energy, and so the weights $w_{i\to a}^{\max}$, reaches the steady state. Then the entropy reads

$$Ns(m)=\sum_i \overline{\Delta s_i}-(K-1)\sum_a \overline{\Delta s_a},$$

$$\overline{\Delta s_i}=\frac{\langle \Delta s_i e^{mN\Delta s_i}\rangle_{pop}}{\langle e^{mN\Delta s_i}\rangle_{pop}},$$

$$\overline{\Delta s_a}=\frac{\langle \Delta s_a e^{mN\Delta s_a}\rangle_{pop}}{\langle e^{mN\Delta s_a}\rangle_{pop}}. \tag{25}$$

Figure 3 shows the results for choices of the factor graph parameters that correspond to different phases.

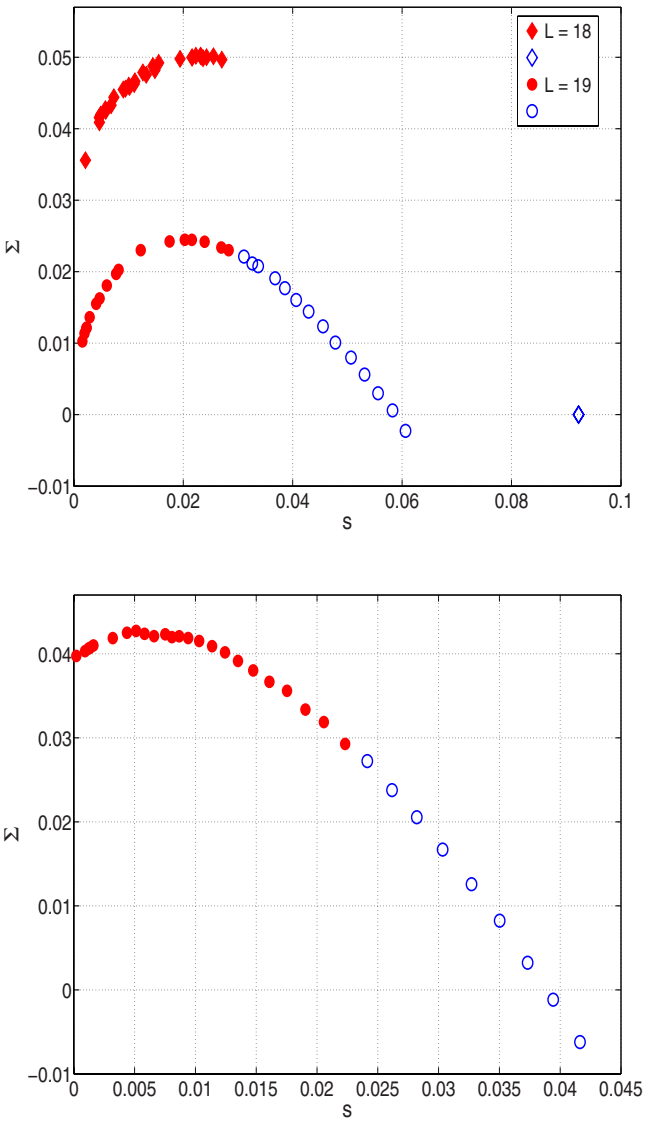

FIG. 3. (Color online) Complexity vs entropy for $K=4$, $L=18,19$ with $\mathcal{N}_p=10^5$ (top) and $K=6$, $L=121$ with $\mathcal{N}_p=2\times10^4$ (bottom) in the one-link approximation. Solid and open symbols represent frozen and unfrozen parts, respectively. The statistical errors are about 0.001.

B. In the one-link approximation

In a regular random hypergraph all links of the associated factor graph are equivalent. Therefore we can forget about different populations on different links and work with only one large population of fields. The way we obtain the stationary distribution $\mathcal{P}[\mu]$ is the same as above. The only difference is that we always select the fields from the single population. In Fig. 4 we compare the complexity computed on a single (4,19)-hypergraph of size $N=10^4$ with complexity obtained in the one-link approximation.

VI. ENTROPY LANDSCAPE: ANALYTICAL RESULTS

To locate different phase transitions in the solution space we need to calculate the generalized free energy $\mathcal{F}$ which is

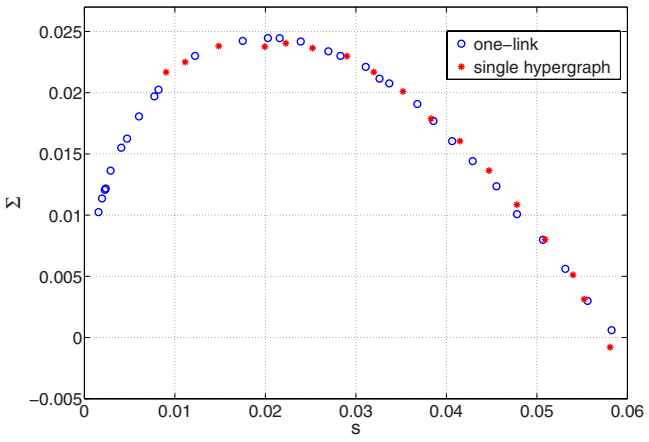

FIG. 4. (Color online) Comparing $\Sigma(s)$ in a single (4,19)-hypergraph ($N=10^4$, $\mathcal{N}_p=10^2$) and in the one-link approximation with $\mathcal{N}_p=10^5$. The statistical errors are about 0.001.

given in terms of $\mathcal{Z}_i$ and $\mathcal{Z}_a$ in Eq. (18). These quantities in turn depend on the fraction of frozen variables π and $\rho(r)$ which should be determined by Eq. (20) for $\mathcal{P}[\mu]$. In the following we study some special cases that allow us to calculate the above quantities and determine the phase diagram of the problem. For clarity here we only state the results of calculations that will be presented in more details in Appendix C.

A. Case $m=1$

The study of $m=1$ clusters is relevant in determining the dynamical, rigidity, and condensation transitions [16]. These are in fact the thermodynamically relevant clusters before the condensation transition.

For $m=1$ the generalized free energy reads

$$\mathcal{F}=\ln\left(2\left[1-\frac{1}{2^{K-1}}\right]^L\right)-(K-1)\alpha\ln\left(1-\frac{2}{2^K}\right). \quad (26)$$

Comparing with the RS entropy s^{RS} in Eq. (1) we see that $\mathcal{F}(m=1)=\Sigma(m=1)+s(m=1)=s^{RS}$. Therefore, as long as the $m=1$ clusters are the thermodynamically relevant ones the RS approximation gives the correct total entropy. From Eq. (21) we can also find the probability of having a frozen field with $r=1$:

$$\frac{1-\pi}{2}=\frac{1}{\mathcal{Z}_{i\to a}}\left\{\left[1-\frac{1}{2^{K-1}}\right]^{L-1}-\left[1-\frac{1}{2^{K-1}}-\left(\frac{1-\pi}{2}\right)^{K-1}\right]^{L-1}\right\}. \quad (27)$$

For small L the above equation has only one solution $\pi=1$, where the $m=1$ clusters are unfrozen. Increasing L, one reaches the rigidity point L_r where another solution $\pi\neq 1$ appears. It is where a finite fraction of the variables in these clusters become frozen. We find that for $K<6$ the rigidity transition always happens after the SAT-UNSAT transition. Simplifying the above equation we obtain

$$\pi=\left[1-\frac{\left(\frac{1-\pi}{2}\right)^{K-1}}{1-\frac{1}{2^{K-1}}}\right]^{L-1}. \quad (28)$$

Assuming $K\gg 1$ and $\pi=\frac{c}{K}$ one obtains

$$c\simeq K\exp\left[-\frac{(L-1)}{2^{K-1}}e^{-c}\right], \quad (29)$$

which suggests

$$L_r\simeq 2^{K-1}e^c[\ln K-\ln c+o(1)]. \quad (30)$$

We see that, like L_d^0, the leading term in L_r diverges as $2^K\ln K$. Compare it with the leading term of L_s^{RS} which scales as $2^K K$.

B. Case $m=0$

The typical or most numerous clusters are the $m=0$ ones. The study of these clusters provides us with an estimate of the SAT-UNSAT transition (in that they are the last clusters to disappear). Indeed the previous studies of the complexity in 1RSB phase focus on these type of clusters. For $m=0$ the generalized free energy reads

$$\mathcal{F}=\ln\left\{2\left[1-\left(\frac{1-\pi}{2}\right)^{K-1}\right]^L-\left[1-2\left(\frac{1-\pi}{2}\right)^{K-1}\right]^L\right\}-(K-1)\alpha\ln\left(1-2\left(\frac{1-\pi}{2}\right)^K\right). \quad (31)$$

Using Eq. (20) one can easily write an equation for the fraction of frozen marginals:

$$\frac{1-\pi}{2}=1-\frac{\left[1-\left(\frac{1-\pi}{2}\right)^{K-1}\right]^{L-1}}{2\left[1-\left(\frac{1-\pi}{2}\right)^{K-1}\right]^{L-1}-\left[1-2\left(\frac{1-\pi}{2}\right)^{K-1}\right]^{L-1}}. \quad (32)$$

The above equation is another way of writing Eq. (6), which has been obtained in the previous studies. Notice that

$$\pi=\frac{\eta^{L-1}}{A_{L-1}}, \quad (33)$$

with A_{L-1} and η given by Eqs. (5) and (6). A nontrivial solution for π appears at L_d^0, where for the first time a maximum appears in the curve $\Sigma(s)$. According to Eq. (19) the complexity of $m=0$ clusters is $\Sigma(m=0)=\mathcal{F}$. The point that this quantity vanishes defines the SAT-UNSAT transition L_s. One can show that, like L_s^{RS}, the leading term in L_s scales as $2^K K$.

C. Case $\pi=0$

This case is relevant to study very small clusters or close to the SAT-UNSAT transition where almost all variables are frozen and $\pi\simeq 0$. Notice that solving numerically for $\Sigma(s)$ is

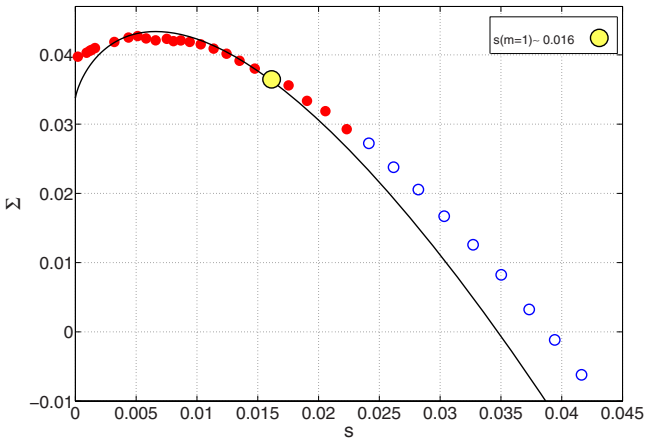

FIG. 5. (Color online) Comparing $\Sigma(s)$ for $K=6$ and $L=121$ (in the one-link approximation with $\mathcal{N}_p=2\times10^4$) with $\Sigma(s,\pi=0)$.

a heavy computational job and it would be useful to have other approximation methods to get a good estimate of the complexity. When $\pi=0$ the generalized free energy is given by

$$\mathcal{F}=\ln\left(2(2^{m-1}-1)\left[1-\frac{2}{2^{K-1}}\right]^L+2\left[1-\frac{1}{2^{K-1}}\right]^L\right)-(K-1)\alpha\ln\left(1-\frac{2}{2^K}\right). \tag{34}$$

Taking derivatives we obtain the entropy as

$$s=\frac{2^m\left[1-\frac{2}{2^{K-1}}\right]^L\ln 2}{2(2^{m-1}-1)\left[1-\frac{2}{2^{K-1}}\right]^L+2\left[1-\frac{1}{2^{K-1}}\right]^L}. \tag{35}$$

With the above quantities we can obtain the complexity of different clusters. In Fig. 5 we have compared this complexity with the one obtained numerically in the one-link approximation. As the figure shows the agreement is good especially for the smaller and frozen clusters.

Close to the condensation transition the $m=1$ clusters are nearly completely frozen and we expect the $\pi=0$ complexity to give a good estimate of $\Sigma(m=1)$. From the above equations we obtain

$$\Sigma(m=1)\simeq\ln\left(2\left[1-\frac{1}{2^{K-1}}\right]^L\right)-(K-1)\alpha\ln\left(1-\frac{2}{2^K}\right)-\left(\frac{1-\frac{2}{2^{K-1}}}{1-\frac{1}{2^{K-1}}}\right)^L\ln 2. \tag{36}$$

We use this approximated complexity to determine the condensation transition L_c where $\Sigma(m=1)$ vanishes. After some algebra we find that for $K\gg1$

$$L_c\simeq K2^{K-1}\left(\ln 2-\frac{K}{2^K}+o(1)\right). \tag{37}$$

Notice that the leading term in L_c is exactly the same as in L_s^{RS}.

D. Case $\pi=1$

The complexity can be nonzero even when the frozen fields are absent. In this case we can exactly compute the free energies of $m=0,1,2$ clusters. We use this fact to find an approximated free energy for $m\in[0,2]$. Having $\mathcal{F}(m=0)$, $\mathcal{F}(m=1)$, and $\mathcal{F}(m=2)$ we use the Lagrange interpolating polynomial function to write $\mathcal{F}(m)$ around $m=1$:

$$\mathcal{F}(m)=-\mathcal{F}(m=1)m(m-2)+\mathcal{F}(m=2)\frac{m(m-1)}{2}, \tag{38}$$

where we have used the fact that for $\pi=1$, $\mathcal{F}(m=0)=0$. The resulted entropy and complexity are

$$s(m)=-2(m-1)\mathcal{F}+\left(m-\frac{1}{2}\right)\mathcal{F}(m=2),$$

$$\Sigma(m)=m^2\left[\mathcal{F}(m=1)-\frac{1}{2}\mathcal{F}(m=2)\right]. \tag{39}$$

As we show in Appendix C, the free energy $\mathcal{F}(m=2)$ depends on the second moment of the $\rho(r)$:

$$\langle r^2\rangle=\frac{1}{\mathcal{Z}_{i\to a}(m=2)}\left[1-\frac{2}{2^{K-1}}+\langle r^2\rangle^{K-1}\right]^{L-1}. \tag{40}$$

It turns out that $\Sigma(m=1)$ is zero as long as $\langle r^2\rangle=1/4$—i.e., $\rho(r)=\delta(r-\frac{1}{2})$. The complexity becomes nonzero only when Eq. (40) suggests a nontrivial solution. We can rewrite Eq. (40) as

$$x=\left[1-\frac{1-x^{K-1}}{1+(1+x)^{K-1}(2^{K-1}-2)}\right]^{L-1},\quad x\equiv\frac{1}{2\langle r^2\rangle}-1. \tag{41}$$

Taking $K\gg1$ and $x=\frac{c}{K}$ we find

$$c\simeq K\exp\left[-\frac{L}{2^{K-1}}e^{-c}\right]. \tag{42}$$

The equation suggests that

$$L_d\simeq 2^{K-1}e^c[\ln K-\ln c+o(1)], \tag{43}$$

which behaves very similar to L_d^0 in Eq. (8).

E. Case of integer m

Suppose that we have computed $\mathcal{Z}_i(m_n)$ and $\mathcal{Z}_a(m_n)$ for $m_n=0,1,\ldots,N_m-1$. In Appendix C we write explicit relations for these quantities when $\rho(r)=\delta(r-\frac{1}{2})$. We can find an approximated free energy that interpolates between the free energy values at integer m's, $\mathcal{F}(m_n)$. To this end we use the Lagrange interpolating polynomial

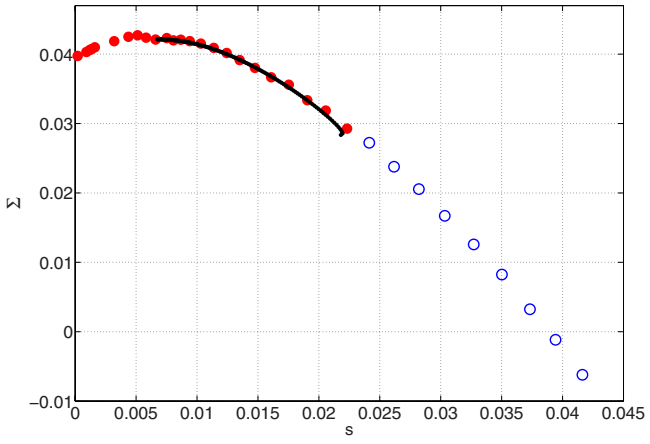

FIG. 6. (Color online) Comparing $\Sigma(s)$ for $K=6$ and $L=121$ (in the one-link approximation with $\mathcal{N}_p=2\times10^4$) with the complexity that has been obtained by the interpolation approximation ($N_m=10$).

$$\mathcal{F}(m)=\sum_{n=0}^{N_m-1}\mathcal{F}(m_n)\prod_{l\neq n}\frac{(m-m_l)}{(m_n-m_l)}. \tag{44}$$

To obtain the free energy we also need to determine π from Eq. (21). This equation depends on $\mathcal{Z}_{i\to a}(m)$, which again can be obtained in the above interpolation approximation. Using the above approximation we can obtain π and the free energy as long as $0\leq m\leq m_f$. Here m_f is the maximum value of m such that frozen variables do exist. Indeed for $m>m_f$ the fraction of frozen variables is zero and with a trivial $\rho(r)$ we find a zero complexity which is not always correct. The number of interpolation points is chosen such that the resulted complexity has a reasonable behavior. In Fig. 6 we compare the complexity obtained in this way with the one we obtained by the population dynamics. As the figure shows with the interpolation approximation we are able to reproduce the population dynamics results in the interval $0\leq m\leq m_f$. With the above approximation we can find an estimate of the freezing point L_f where all clusters become frozen. In Table II we compare L_f with degree values at the other phase transition points.

TABLE II. Numerical values of degree L at different transition points obtained in the 1RSB approximation with the methods described in the text.

K	L_d	L_r	L_f	L_c	L_s
3	6	–	–	7	7
4	18	–	–	20	20
5	49	53	–	53	53
6	114	119	126	130	130
7	250	257	297	306	307
8	534	543	663	705	706
9	1122	1136	1473	1591	1592
10	2333	2356	3202	3543	3543

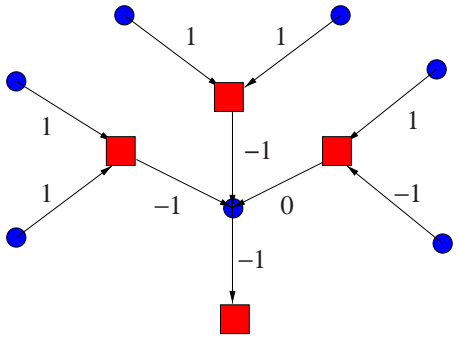

FIG. 7. (Color online) Warning propagation on a factor graph.

VII. ALGORITHMS AND THE ENTROPY LANDSCAPE

In this section we will use different algorithms to find some solutions of the bicoloring problem close to the SAT-UNSAT transition. We show that a smoothed BP decimation algorithm is able to find solutions even beyond the rigidity transition $L>L_r$. We will also see that, within our level of approximation and for fixed parameters, the algorithm always finds solutions that belong to the same kind of clusters. Interestingly enough, beyond the rigidity transition, we find solutions to clusters that are exponentially smaller in number compared to the thermodynamically relevant ones.

A. Cavity method as an algorithm to find solutions

Warning propagation (WP) is an elementary message-passing algorithm that uses cavity messages to find a solution of a constraint satisfaction problem. On each edges of the factor graph we define cavity messages $W_{a\to i}\in\{-1,0,1\}$, $W_{i\to a}\in\{-1,1\}$; the warning $W_{a\to i}=0$ means that variable i is free to take any value without worrying about constraint a. On the other hand, if $W_{a\to i}=-1,1$, variable i should take a value that satisfies constraint a. The message $W_{i\to a}=-1,1$ represents the color that variable i has to take to satisfy the other constraints. Given a factor graph we start with the initially random values of W's and in each sweep of the algorithm we update all the messages; see Fig. 7. For example, the messages on edge (i,a) are updated in the following way:

$$W_{a\to i}=\begin{cases}-1 & \text{if all } W_{j\to a}\text{'s are } 1,\\ 1 & \text{if all } W_{j\to a}\text{'s are } -1,\\ 0 & \text{otherwise,}\end{cases}$$

$$W_{i\to a}=\operatorname{sgn}\left(\sum_{b\in V(i)\backslash a}W_{b\to i}\right). \tag{45}$$

If the algorithm converges and no variable receives contradictory warnings, we can determine the solution according to the warnings. It has been shown that on tree factor graphs the above algorithm always converges and gives the solutions.

More sophisticated message passing algorithms that work much better than WP are *belief propagation decimation*

(BPD) and *survey propagation decimation* (SPD) [33]. In these algorithms one replaces the messages $W_{a\to i}, W_{i\to a}$ with probabilities that come from a single-cluster (RS) or multiple-cluster (1RSB) approximation. For example, in BPD we have the beliefs $\mu_{a\to i}, \mu_{i\to a}$, which are updated according to the BP equations

$$\mu_{i\to a}(\sigma_i) = \frac{1}{Z_{i\to a}} \prod_{b\in V(i)\backslash a} \mu_{b\to i}(\sigma_i),$$

$$\mu_{b\to i}(\sigma_i) = \sum_{\sigma_{\partial b\backslash i}} I_b(\sigma_{\partial b}) \prod_{j\in V(b)\backslash i} \mu_{j\to b}(\sigma_j). \tag{46}$$

Starting from random initial values for the μ's we update them to reach a fixed point of the dynamics. After convergence we define the local fields

$$H_i \equiv \ln \frac{\mu_i(1)}{\mu_i(0)}, \tag{47}$$

where

$$\mu_i(\sigma_i) = \frac{1}{Z_i} \prod_{b\in V(i)} \mu_{b\to i}(\sigma_i). \tag{48}$$

Then the most biased variable is fixed according to the sign of its local field. Then we simplify the factor graph and repeat the above procedure until we obtain a paramagnet where $H_i=0$ for all the remaining variables. At this stage we can run a local search algorithm to complete the solution of our problem. In this paper we are going to use a smoothed version of BPD first introduced in [6]. The main idea is to introduce external fields h_i that at each step of the algorithm are updated according to the local fields. At the end, the external fields determine the preferred values of the variables. We call this algorithm *belief propagation reinforcement* [34].

More precisely, the BP reinforcement algorithm works as follows.

(i) Start with random initial values for $0\leq \mu_{i\to a}, \mu_{a\to i} \leq 1$ and $-\delta \leq h_i \leq \delta$ ($\delta \ll 1$).

(ii) For $t=1,\ldots,t_{\max}$,

(a) update all the μ's according to the BP equations in the presence of the external fields,

$$\mu_{i\to a}(\sigma_i) = \frac{e^{h_i\sigma_i}}{Z_{i\to a}} \prod_{b\in V(i)\backslash a} \mu_{b\to i}(\sigma_i),$$

$$\mu_{b\to i}(\sigma_i) = \sum_{\sigma_{\partial b\backslash i}} I_b(\sigma_{\partial b}) \prod_{j\in V(b)\backslash i} \mu_{j\to b}(\sigma_j); \tag{49}$$

(b) obtain local fields $H_i=\ln\frac{\mu_i(1)}{\mu_i(0)}$ and with probability $1-t^{-\gamma}$ update the external fields as $h_i \to h_i + sign(H_i)\delta$.

(c) If $\underline{\sigma}=\{\sigma_i=\mathrm{sgn}(H_i) | i=1,\ldots,N\}$ is a solution, return $\underline{\sigma}$.

Notice that instead of fixing variables one by one, here all the external fields are updated (with a rate that increases with t) during the run time. Moreover, in this algorithm we do not need to simplify the factor graph after each decimation.

For comparison, we also use other algorithms like *simulated annealing* (SA) and *focused simulated annealing* (FSA) to find solutions [35]. In the SA algorithm we start from a random configuration at small inverse temperature $\beta_i \equiv 1/T_i$ and decrease the temperature slowly. At each temperature we select all the variables in a random sequential way and flip a variable with probability $\min\{1, \exp(-\beta\,\Delta E_i)\}$. Here ΔE_i is the change in the number of unsatisfied constraints if we accept to flip variable i. After a sweep the inverse temperature increases by $\Delta\beta$. In the FSA algorithm we do the same as SA except that to flip a variable we only select those that belong to unsatisfied constraints.

To check the algorithms and their solutions we consider two different cases: (a) (4,19)-hypergraph ($L_d<L<L_s$), just after the dynamical transition and before the SAT-UNSAT transition where the thermodynamically relevant clusters are unfrozen, and (b) (6,121)-hypergraph ($L_r<L<L_f$), where the thermodynamically relevant clusters are frozen, but still there are some unfrozen clusters.

In case (a) we are able to find some solutions with all BPR, SA, and FSA algorithms in a reasonable time. On a (4,19)-hypergraph of $N=10^4$ variables it takes about 20 h for SA and FSA algorithms to find a solution, whereas the BPR algorithm does the job in about 10 min. In the SA and FSA algorithms the parameters are $\beta_i=0.1$ and $\Delta\beta=10^{-5}$. In the BPR algorithm we used $\gamma=0.05$ and $\delta=0.01$. With these parameters we could obtain a solution at the end of almost all runs.

In case (b) we could only find solutions with the BPR algorithm. Both SA and FSA algorithms were not able to give a solution in a couple of days even for $N=10^3$. However, on a hypergraph of size $N=10\,002$, the BPR algorithm still returns a solution for every instance after a few hours in about 20% of the runs starting with different initial conditions.

We expect the performance of the algorithm could be improved by further optimization. We did not pursue this line as we are interested in a proof of concept rather than in optimizing algorithms over academic benchmarks.

B. Entropy versus distance from a solution

Suppose that we have the number of solutions at distance d from a given solution, $e^{Ns(d)}$. If the solution belongs to a spherelike cluster of solutions, then $s(d)$ increases for $d \leq d^*$ and becomes zero at d^*. Clearly, for large N, the entropy s^* is a good (under)estimate of the total entropy of the cluster. If the solution space is more complex, we still expect $s(d)$ to be, up to a distance d^*, an increasing function of d. It may exhibit a maximum at d^* and decrease for larger distances. In any case we can take s^* as an approximation to the entropy of the cluster.

To obtain $s(d)$ for a given solution $\underline{\sigma}^*$ and distance d, we use a local BP as follows.

(i) Start with random initial values for $0\leq \mu_{i\to a}, \mu_{a\to i} \leq 1$ and a reasonable value of x.

(ii) For $t=1,\ldots,t_{\max}$,

(a) update all the μ's according to the BP equations around the given solution,

$$\mu_{i\to a}(\sigma_i) = \frac{e^{x(\sigma_i-\sigma_i^*)^2}}{Z_{i\to a}} \prod_{b\in V(i)\backslash a} \mu_{b\to i}(\sigma_i),$$

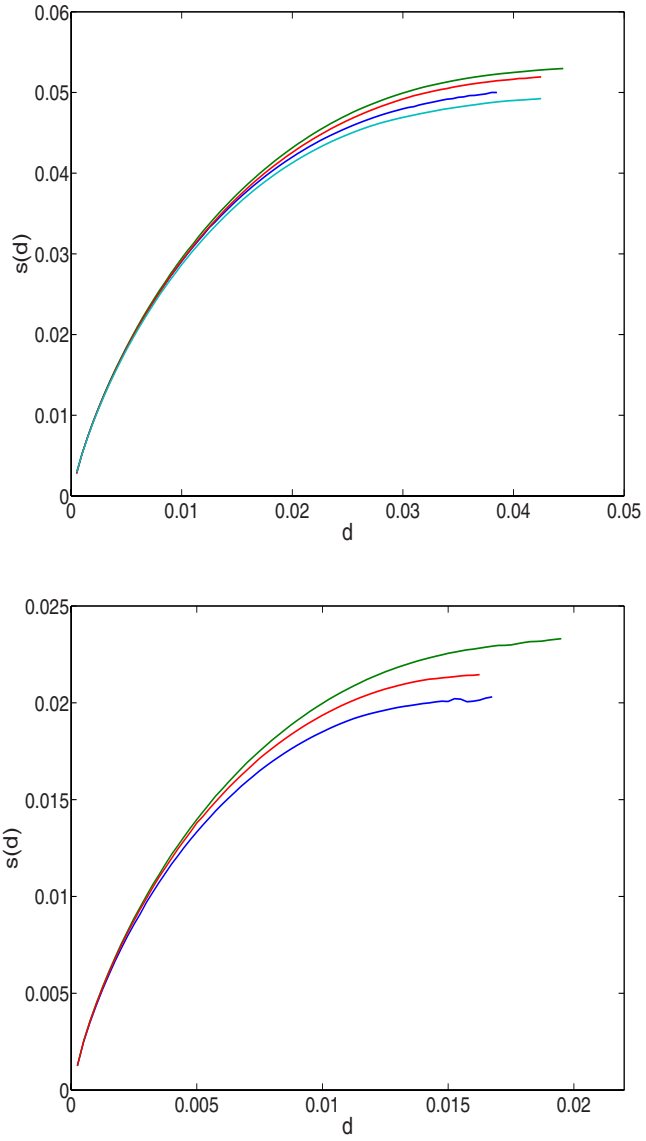

FIG. 8. (Color online) $s(d)$ for a few solutions in a (4,19)-hypergraph with $N=10\,000$ (top) and a (6,121)-hypergraph with $N=100\,02$ (bottom). Note that some of the curves have been selected to show the extremal behavior of $s(d)$.

$$\mu_{b\to i}(\sigma_i) = \sum_{\sigma_{\partial b\setminus i}} I_b(\sigma_{\partial b}) \prod_{j\in V(b)\setminus i} \mu_{j\to b}(\sigma_j); \qquad (50)$$

(b) obtain $f(x)$ [Eq. (13)] and find a new x such that $d = \frac{\partial f(x)}{\partial x}$.

(c) If converged, calculate the entropy [Eq. (15)] and return $s(d)$.

The SP version of this algorithm has been used in [36] to obtain the complexity as a function of distance from a solution in k-XOR-SAT problem.

Figure 8 shows $s(d)$ for a number of solutions obtained with BPR algorithm. As the figures show, we do not always reach the extremum point of the curve $s(d)$. It is even more difficult to observe the decreasing part of the entropy. Indeed, as we approach the maximum, the convergence time of the algorithm increases rapidly and exceeds the upper bound $t_{\max}=1000$. This happens, probably, when we encounter the other clusters where replica symmetry approximation is not valid any more. However, we could observe the decreasing part of $s(d)$ for small values of N, where computational time is not too large.

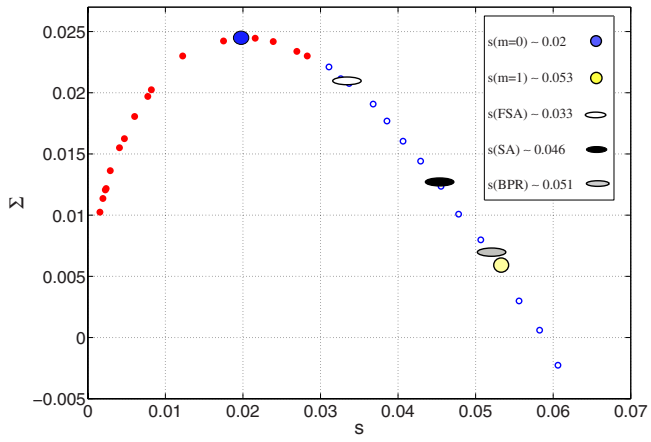

FIG. 9. (Color online) Comparing the attractive clusters of different algorithms in the entropy landscape. The large circles show the typical and thermodynamically relevant clusters. All FSA, SA, and BPR algorithms find solutions in the interval between frozen and thermodynamically relevant clusters with entropies $s(\mathrm{FSA}) < s(\mathrm{SA}) < s(\mathrm{BPR})$. In each case we found 20 solutions on a (4,19)-hypergraph of size $N=10\,000$. The standard deviations in the entropies is about 0.002

Finally, notice that one could obtain the cluster entropy by summing over all solutions: $\exp(Ns)=\Sigma_d \exp[Ns(d)]$. However, for large N the maximum entropy has the dominant contribution to the cluster entropy. To show this we calculated the cluster entropy s, using the above definition, and compared it with our estimation s^*, which is the maximum entropy. For instance, for three solutions of a (4,19)-hypergraph of size $N=10\,000$ we obtain $\delta s = s - s^* = 0.000\,115, 0.000\,118, 0.000\,121$ whereas $s^*=0.052, 0.050, 0.0497$, respectively. We see that the differences are very small compared to the cluster entropy.

Another source of systematic error is that the curves do not always reach the real extremum. However, as Fig. 8 shows, we observed that the maximum entropy is very close to the real one. Indeed, an extrapolation of the curves to higher distances gives a correction which is about 0.001.

C. $m=1$ vs $m\neq 1$ solutions

Using the method described in the previous subsection we can now locate our solutions in the entropy landscape to see to which clusters they belong. In Fig. 9 we show the attractive clusters of different algorithms after the dynamical transition and before the rigidity transition. In this case BPR finds solutions in clusters that are very close to the thermodynamically relevant ones. We think that the difference is due to the systematic errors in underestimating the cluster entropy. In addition, there is also some statistical error in the entropy value of the curve points. The figure also shows that SA ends up in smaller clusters compared to the thermody-

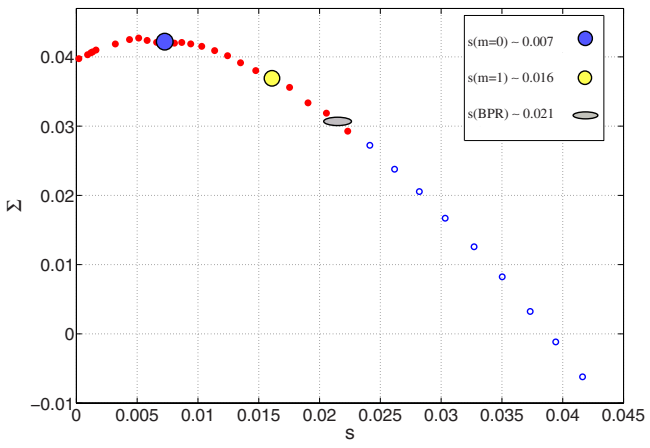

FIG. 10. (Color online) Comparing the attractive clusters of the BPR algorithm with the typical and thermodynamically relevant clusters (large circles). In this case the BPR algorithm finds solutions in the most numerous unfrozen clusters. The result obtained from 20 solutions on a (6,121)-hypergraph of size $N=100\,02$. The standard deviation in the entropy is about 0.002.

namically relevant ones. Moreover, FSA finds solutions in much smaller clusters close to the frozen ones.

The above results have been obtained with parameters given in Sec. VII A. We found that by decreasing γ (in BPR) or $\Delta\beta$ (in SA and FSA) the algorithms find solutions in larger clusters. In fact in a very slow annealing scheme, where one equilibrates the system at each step of the algorithm, we will finally find a solution in the thermodynamically relevant clusters.

Notice that all the algorithms end up in the region between the frozen and thermodynamically relevant clusters. Indeed, when N is large it is very difficult to find a solution in the frozen clusters; each time we flip a frozen variable, we go to another cluster of solutions and so an extensive number of flips is needed to accordingly rearrange the variables. On the other hand, it is not also easy to find a solution in very large clusters that are exponentially less numerous than the thermodynamically relevant ones.

As already mentioned, beyond the rigidity transition we could only find a solution by BPR algorithm. Figure 10 shows that in this case the solutions are very close to the boundary between frozen and unfrozen clusters. The difference is about the statistical errors and the error that we make by underestimating the entropy. We see that when the thermodynamically relevant clusters are frozen the algorithm ends up in the smallest unfrozen clusters. These are exponentially more numerous than the other unfrozen clusters.

We have checked that our solutions do, indeed, belong to the unfrozen clusters. This can be done with the so-called *whitening* process [37,38]. Given a solution one checks if a variable can be flipped without violating any constraint. If so, that variable is unfrozen and is denoted with a "$*$." The process goes on by checking one by one the other variables with the additional rule that a constraint with at least one star variable be always satisfied. This process is repeated up to the fixed point where the number of star variables is fixed. If a solution belongs to an unfrozen cluster, then at the end all the variables would be $*$.

VIII. CONCLUSION

In summary, we applied the large deviations cavity method to study the phase diagram of the bicoloring problem on regular random hypergraphs. Working in the one-step replica-symmetry-breaking framework we located the various phase transitions characterizing the structure of the solutions landscape at both the ensemble and single-instance levels. Notice that we did not check the stability of 1RSB solutions toward higher-order replica symmetry breaking. But as other studies show [39,40], we expect 1RSB solutions to give the correct qualitative picture and even exact results close to the SAT-UNSAT transition.

We also used different algorithms to find solutions and to locate them in the entropy landscape of the problem. This provided a rough characterization of the relations existing between the entropic properties of clusters of solutions and the different algorithms used to find them.

From an algorithmic point of view, the algorithms based on simulated annealing could not efficiently find solutions after the rigidity point. However, using BPR we showed that it is actually possible to go beyond the rigidity transition. In this case we obtained solutions that belong to the smallest and most numerous unfrozen clusters [18].

ACKNOWLEDGMENTS

We thank T. Mora and L. Zdeborova who let us know about their numerical methods. The work was supported by an EVERGROW EC grant and by the Technical Computing Initiative of Microsoft Research.

APPENDIX A: CAVITY EQUATIONS IN THE RS APPROXIMATIONS

We start from the partition function definition in Eq. (9) and derive the main equations in the first part of Sec. IV. Let $Z_{i\to a}(\sigma_i)$ denote the partition function in the absence of constraint a and when variable i has state σ_i. Then, in the absence of constraint a, the probability of finding variable i in state σ_i is

$$\mu_{i\to a}(\sigma_i)=\frac{Z_{i\to a}(\sigma_i)}{\sum_\sigma Z_{i\to a}(\sigma)}. \tag{A1}$$

On the other hand, assuming a tree structure for the factor graph we can write

$$Z_{i\to a}(\sigma_i)=\sum_{\sigma_{\partial i\to a}}\left(\prod_{b\in V(i)\backslash a} I_b(\sigma_{\partial b})\left[\prod_{j\in V(b)\backslash i} Z_{j\to b}(\sigma_j)\right]\right)e^{x(\sigma_i-\sigma_i^*)^2}. \tag{A2}$$

From the above equation we can derive a relation for the cavity marginals:

$$\mu_{i\to a}(\sigma_i)=\frac{1}{Z_{i\to a}}\sum_{\sigma_{\partial i\to a}}\left(\prod_{b\in V(i)\backslash a} I_b(\sigma_{\partial b})\times\left[\prod_{j\in V(b)\backslash i}\mu_{j\to b}(\sigma_j)\right]\right)e^{x(\sigma_i-\sigma_i^*)^2}, \tag{A3}$$

where $Z_{i\to a}$ is a normalization constant. It is more convenient if we write the above relation as

$$\mu_{i\to a}(\sigma_i) = \frac{e^{x(\sigma_i - \sigma_i^*)^2}}{Z_{i\to a}} \prod_{b\in V(i)\backslash a} \mu_{b\to i}(\sigma_i), \tag{A4}$$

where

$$\mu_{b\to i}(\sigma_i) = \sum_{\sigma_{\partial b\backslash i}} I_b(\sigma_{\partial b}) \prod_{j\in V(b)\backslash i} \mu_{j\to b}(\sigma_j). \tag{A5}$$

The free energy $f(x)$ is given by $\frac{1}{N}\ln Z$. In the Bethe approximation

$$f(x) = \sum_i \Delta f_i - \sum_a (K_a - 1)\Delta f_a, \tag{A6}$$

where Δf_i and Δf_a are the free energy shifts by adding variable node i and function node a, respectively.

Suppose that we have removed node i and all its function nodes from the factor graph. In this case the partition function reads

$$Z_{-\{i,V(i)\}} = \prod_{a\in V(i)} \left[\prod_{j\in V(a)\backslash i} Z_{j\to a} \right], \tag{A7}$$

whereas in the complete factor graph

$$Z = \sum_{\sigma_i,\sigma_{\partial i}} \prod_{a\in V(i)} \left[I_a(\sigma_{\partial a}) \prod_{j\in V(a)\backslash i} Z_{j\to a}(\sigma_j) \right] e^{x(\sigma_i - \sigma_i^*)^2}. \tag{A8}$$

Dividing the two quantities we get

$$\frac{Z}{Z_{-\{i,V(i)\}}} = e^{N\Delta f_i} = \sum_{\sigma_i,\sigma_{\partial i}} \prod_{a\in V(i)} \left[I_a(\sigma_{\partial a}) \prod_{j\in V(a)\backslash i} \frac{Z_{j\to a}(\sigma_j)}{Z_{j\to a}} \right] e^{x(\sigma_i - \sigma_i^*)^2}, \tag{A9}$$

and this gives the shift in the free energy by adding variable node i:

$$e^{N\Delta f_i} = \sum_{\sigma_i} \prod_{a\in V(i)} \mu_{a\to i}(\sigma_i) e^{x(\sigma_i - \sigma_i^*)^2}. \tag{A10}$$

We can do the same procedure for a function node. If we remove function node a from the factor graph, we have

$$Z_{-a} = \prod_{i\in V(a)} Z_{i\to a}, \tag{A11}$$

whereas the complete partition function can be written as

$$Z = \sum_{\sigma_{\partial a}} I_a(\sigma_{\partial a}) \prod_{i\in V(a)} Z_{i\to a}(\sigma_i). \tag{A12}$$

So

$$\frac{Z}{Z_{-a}} = e^{N\Delta f_a} = \sum_{\sigma_{\partial a}} I_a(\sigma_{\partial a}) \prod_{i\in V(a)} \frac{Z_{i\to a}(\sigma_i)}{Z_{i\to a}}, \tag{A13}$$

and the shift in the free energy is given by

$$e^{N\Delta f_a} = \sum_{\sigma_{\partial a}} I_a(\sigma_{\partial a}) \prod_{i\in V(a)} \mu_{i\to a}(\sigma_i). \tag{A14}$$

Finally using Eqs. (A6), (A10), and (A14) we obtain the free energy $f(x)$ in the Bethe approximation:

$$Nf(x) = \sum_i \ln Z_i - \sum_a (K_a - 1)\ln Z_a,$$

$$Z_i \equiv \sum_{\sigma_i} \left[\prod_{a\in V(i)} \mu_{a\to i}(\sigma_i) \right] e^{x(\sigma_i - \sigma_i^*)^2},$$

$$Z_a \equiv \sum_{\sigma_{\partial a}} I_a(\sigma_{\partial a}) \prod_{i\in V(a)} \mu_{i\to a}(\sigma_i). \tag{A15}$$

APPENDIX B: CAVITY EQUATIONS IN THE 1RSB APPROXIMATION

We start from the generalized partition function in Eq. (16) and explain the main equations in the second part of Sec. IV.

In the Bethe approximation,

$$N\mathcal{F}(m) = \sum_i \Delta\mathcal{F}_i - \sum_a (K_a - 1)\Delta\mathcal{F}_a. \tag{B1}$$

The generalized partition function can be written as

$$\mathcal{Z} = \sum_c e^{mN[s_{c,-\{i,V(i)\}} + \Delta s_{c,i}]}, \tag{B2}$$

where $\Delta s_{c,i}$ is the shift in the entropy of cluster c by adding node i and all its function nodes. At a fixed value of m the typical clusters would have nearly the same entropy and it seems safe to approximate $e^{mN\Delta s_i}$ with its average value among the clusters—i.e.,

$$\mathcal{Z} = \left[\sum_c e^{mNs_{c,-\{i,V(i)\}}} \right] \int d\mathcal{P}[\mu] e^{mN\Delta s_i}, \tag{B3}$$

where $\mathcal{P}[\mu]$ is the probability distribution of $\mu_{i\to a}$'s among the clusters. So the shift in the generalized free energy reads

$$e^{N\Delta\mathcal{F}_i} = \int \prod_{a\in V(i)} \prod_{j\in V(a)\backslash i} d\mathcal{P}_{j\to a}[\mu_{j\to a}] e^{mN\Delta s_i}. \tag{B4}$$

In the case of adding a function node, similarly we find

$$e^{N\Delta\mathcal{F}_a} = \int \prod_{i\in V(a)} d\mathcal{P}_{i\to a}[\mu_{i\to a}] e^{mN\Delta s_a}. \tag{B5}$$

Notice that Δs_i and Δs_a correspond to the free energy shifts, Eqs. (A10) and (A14), with $x=0$. Using Eqs. (B4) and (B5) along with the Bethe form of the generalized free energy we obtain

$$N\mathcal{F}(m) = \sum_i \ln \mathcal{Z}_i - \sum_a (K_a - 1)\ln \mathcal{Z}_a,$$

$$\mathcal{Z}_i \equiv \int D_i\mathcal{P}[\mu] e^{mN\Delta s_i},$$

$$\mathcal{Z}_a \equiv \int D_a \mathcal{P}[\mu] e^{mN\Delta s_a}, \tag{B6}$$

where $d\mathcal{P}[\mu]$ is determined by Eq. (20). The normalization constant in this equation is

$$\mathcal{Z}_{i\to a} = \int \prod_{b\in V(i)\backslash a} \prod_{j\in V(b)\backslash i} d\mathcal{P}_{j\to b}[\mu] e^{mN\Delta s_i}. \tag{B7}$$

We represented $\mathcal{P}[\mu]$ as

$$\mathcal{P}[\mu] = \frac{1-\pi}{2}[\delta(r)+\delta(r-1)] + \pi\rho(r), \tag{B8}$$

where $\rho(r)$ is the probability distribution of unfrozen marginals. By the normalization and symmetry of the problem

$$\int dr \rho(r) = 1,$$

$$\int dr\, r\rho(r) = \int dr(1-r)\rho(r) = \frac{1}{2}. \tag{B9}$$

Then for the $\mathcal{Q}^{\sigma}_{i\to a}[\mu] = 2\mu_{i\to a}(\sigma)\mathcal{P}_{i\to a}[\mu]$ we have

$$\mathcal{Q}^0_{i\to a}[\mu] = (1-\pi)\delta(1-r) + 2\pi r\rho(r),$$

$$\mathcal{Q}^1_{i\to a}[\mu] = (1-\pi)\delta(r) + 2\pi(1-r)\rho(r). \tag{B10}$$

Notice that for $\sigma\in\{0,1\}$,

$$\int d\mu \mathcal{Q}^{\sigma}_{i\to a}[\mu] = 1. \tag{B11}$$

APPENDIX C: CALCULATING THE GENERALIZED FREE ENERGY

To calculate the free energy $\mathcal{F}$ we need to obtain $\mathcal{Z}_i$, $\mathcal{Z}_a$, the fraction of unfrozen variables π, and $\rho(r)$. Let us start from Eq. (20) and multiply both sides of the equation with e^{-rt}. Integrating over r allows us to get rid of the δ function and we find

$$\begin{aligned}
&\frac{1-\pi}{2}(1+e^{-t}) + \pi\int \rho(r)e^{-rt}dr \\
&= \frac{1}{\mathcal{Z}_{i\to a}} \prod_{b\in V(i)\backslash a} \left[\sum_{n_b^0,n_b^1,n_b^*} \binom{K-1}{n_b^0,n_b^1,n_b^*} \right. \\
&\times \left(\frac{1-\pi}{2} \right)^{n_b^0+n_b^1} \pi^{n_b^*} \prod_{j\in V^*(b)} \int dr_b^j \rho(r_b^j) \Bigg] \Bigg[\prod_b \lambda_0(n_b,r_b) \\
&\left. + \prod_b \lambda_1(n_b,r_b) \right]^m e^{-t/\mathcal{Z}_{i\to a}\Pi_b \lambda_0(n_b,r_b)},
\end{aligned} \tag{C1}$$

where n_b^0 and n_b^1 are the number of frozen variables in $V(b)\backslash i$ that take values 0 and 1, respectively. Accordingly $V^*(b)\backslash i$ is the set of unfrozen variables in $V(b)\backslash i$ and n_b^* is the number of its elements. Notice that n_b's should satisfy $n^0+n_b^1+n_b^* = K-1$.

Using Eqs. (A3) and (A10) we write

$$r = \frac{1}{Z_{i\to a}} \prod_{b\in V(i)\backslash a} \lambda_0(n_b,r_b),$$

$$e^{N\Delta s_i} = \prod_{b\in V(i)} \lambda_0(n_b,r_b) + \prod_{b\in V(i)} \lambda_1(n_b,r_b), \tag{C2}$$

where

$$\lambda_0(n_b,r_b) = (1-\delta_{n_b^0,K-1}) \left[1 - \delta_{n_b^1,0}(1-\delta_{n_b^*,0}) \prod_{j\in V^*(b)\backslash i} r_b^j \right],$$

$$\lambda_1(n_b,r_b) = (1-\delta_{n_b^1,K-1}) \left[1 - \delta_{n_b^0,0}(1-\delta_{n_b^*,0}) \prod_{j\in V^*(b)\backslash i} (1-r_b^j) \right]. \tag{C3}$$

One can use Eq. (C1) to write some equations for different moments of $\rho(r)$. For example, for the second moment we obtain

$$\begin{aligned}
\frac{1-\pi}{2} + \pi\langle r^2\rangle &= \frac{1}{\mathcal{Z}_{i\to a}} \prod_{b\in V(i)\backslash a} \left[\sum_{n_b^0,n_b^1,n_b^*} \binom{K-1}{n_b^0,n_b^1,n_b^*} \right. \\
&\times \left(\frac{1-\pi}{2} \right)^{n_b^0+n_b^1} \pi^{n_b^*} \prod_{j\in V^*(b)\backslash i} \int dr_b^j \rho(r_b^j) \Bigg] \\
&\times \left[\prod_{b\in V(i)\backslash a} \lambda_0(n_b,r_b) + \prod_{b\in V(i)\backslash a} \lambda_1(n_b,r_b) \right]^m \\
&\times \left(\frac{\prod_b \lambda_0(n_b,r_b)}{Z_{i\to a}} \right)^2.
\end{aligned} \tag{C4}$$

To compute $\mathcal{F}$ we also need to find $e^{N\Delta s_a}$ in Eq. (A14):

$$\begin{aligned}
e^{N\Delta s_a} &= (1-\delta_{n_a^0,K})(1-\delta_{n_a^1,K}) \left[1 - \delta_{n_a^1,0}(1-\delta_{n_a^*,0}) \prod_{i\in V^*(a)} r_a^j \right. \\
&\left. - \delta_{n_a^0,0}(1-\delta_{n_a^*,0}) \prod_{i\in V^*(a)} (1-r_a^i) \right].
\end{aligned} \tag{C5}$$

The normalization constants in Eqs. (A2) and (B7) are

$$Z_{i\to a} = \prod_{b\in V(i)\backslash a} \lambda_0(n_b,r_b) + \prod_{b\in V(i)\backslash a} \lambda_1(n_b,r_b) \tag{C6}$$

and

$$\begin{aligned}
\mathcal{Z}_{i\to a} &= \prod_{b\in V(i)\backslash a} \left[\sum_{n_b^0,n_b^1,n_b^*} \binom{K-1}{n_b^0,n_b^1,n_b^*} \right. \\
&\times \left(\frac{1-\pi}{2} \right)^{n_b^0+n_b^1} \pi^{n_b^*} \prod_{j\in V^*(b)\backslash i} \int dr_b^j \rho(r_b^j) \Bigg] \\
&\times \left[\prod_{b\in V(i)\backslash a} \lambda_0(n_b,r_b) + \prod_{b\in V(i)\backslash a} \lambda_1(n_b,r_b) \right]^m.
\end{aligned} \tag{C7}$$

Finally for the main elements of the generalized free energy we have

$$\mathcal{Z}_i = \prod_{b\in V(i)} \left[\sum_{n_b^0,n_b^1,n_b^*} \binom{K-1}{n_b^0,n_b^1,n_b^*} \times \left(\frac{1-\pi}{2}\right)^{n_b^0+n_b^1} \pi^{n_b^*} \prod_{j\in V^*(b)\setminus i} \int dr_b^j \rho(r_b^j) \right] \times \left[\prod_{b\in V(i)\setminus a} \lambda_0(n_b,r_b) + \prod_{b\in V(i)\setminus a} \lambda_1(n_b,r_b) \right]^m \quad \text{(C8)}$$

and

$$\mathcal{Z}_a = \sum_{n_a^0,n_a^1,n_a^*} \binom{K}{n_a^0,n_a^1,n_a^*} \left(\frac{1-\pi}{2}\right)^{n_a^0+n_a^1} \pi^{n_a^*} \prod_{i\in V^*(a)} \int dr_a^i \rho(r_a^i) \times \left[(1-\delta_{n_a^0,K})(1-\delta_{n_a^1,K}) \left(1-\delta_{n_a^1,0}(1-\delta_{n_a^*,0}) \prod_{i\in V^*(a)} r_a^j - \delta_{n_a^0,0}(1-\delta_{n_a^*,0}) \prod_{i\in V^*(a)} (1-r_a^i) \right) \right]^m. \quad \text{(C9)}$$

In the following we will give the details of calculations in two special cases that need more explanation.

1. Case $\pi=1$

When $\pi=1$ the equation for $\mathcal{Z}_i$, Eq. (C8), is

$$\mathcal{Z}_i = \prod_{a=1,L} \left[\prod_{j=1,K-1} \int dr_a^j \rho(r_a^j) \right] \left[\prod_a (1-\prod_j r_a^j) + \prod_a \left(1 - \prod_j (1-r_a^j) \right) \right]^m. \quad \text{(C10)}$$

For $m=0,1,2$ we obtain

$$\mathcal{Z}_i(m=0) = 1,$$

$$\mathcal{Z}_i(m=1) = 2\left[1-\frac{1}{2^{K-1}}\right]^L,$$

$$\mathcal{Z}_i(m=2) = 2\left[1-\frac{2}{2^{K-1}}+\langle r^2\rangle^{K-1}\right]^L + 2\left[1-\frac{2}{2^{K-1}}+\left(\frac{1}{2}-\langle r^2\rangle\right)^{K-1}\right]^L. \quad \text{(C11)}$$

For $\mathcal{Z}_a$ from Eq. (C9) we find

$$\mathcal{Z}_a = \left[\prod_{i=1,K} \int dr_a^i \rho(r_a^i) \right] \left[1-\prod_i r_a^i - \prod_i (1-r_a^i) \right]^m. \quad \text{(C12)}$$

Again, for $m=0,1,2$,

$$\mathcal{Z}_a(m=0) = 1,$$

$$\mathcal{Z}_a(m=1) = \left[1-\frac{2}{2^K}\right],$$

$$\mathcal{Z}_a(m=2) = \left[1-\frac{4}{2^K}+2\langle r^2\rangle^K + 2\left(\frac{1}{2}-\langle r^2\rangle\right)^K\right]. \quad \text{(C13)}$$

To complete the free energy calculations we need to find $\langle r^2\rangle$ in $m=2$ clusters. The second moment of $\rho(r)$ can be obtained from Eq. (C4):

$$\langle r^2\rangle = \frac{1}{\mathcal{Z}_{i\to a}} \prod_{b=1,L-1} \left[\prod_{j=1,K-1} \int dr_b^j \rho(r_b^j) \right] \left[\prod_b \left(1-\prod_j r_b^j\right) + \prod_b \left(1-\prod_j (1-r_b^j)\right) \right]^{m-2} \left[\prod_b (1-\prod_j r_b^j) \right]^2. \quad \text{(C14)}$$

If $m=2$, the exact equation is

$$\langle r^2\rangle = \frac{1}{\mathcal{Z}_{i\to a}(m=2)} \left[1-\frac{2}{2^{K-1}}+\langle r^2\rangle^{K-1} \right]^{L-1}. \quad \text{(C15)}$$

Now we can use the Lagrange interpolating polynomial to approximate the free energy by

$$\mathcal{F}(m) = \mathcal{F}(m=0)\frac{(m-1)(m-2)}{2} - \mathcal{F}(m=1)m(m-2) + \mathcal{F}(m=2)\frac{m(m-1)}{2}. \quad \text{(C16)}$$

Since $\mathcal{F}(m=0)=0$, we get

$$\mathcal{F}(m) = -\mathcal{F}(m=1)m(m-2) + \mathcal{F}(m=2)\frac{m(m-1)}{2},$$

$$s(m) = -2(m-1)\mathcal{F}(m=1) + \left(m-\frac{1}{2}\right)\mathcal{F}(m=2),$$

$$\Sigma(m) = m^2\left[\mathcal{F}(m=1) - \frac{1}{2}\mathcal{F}(m=2)\right]. \quad \text{(C17)}$$

2. Case of integer m

Starting from Eq. (B7) we expand $\mathcal{Z}_{i\to a}$ for integer $m>0$ to get

$$\mathcal{Z}_{i\to a} = \sum_l \binom{m}{l} \prod_{b\in V(i)\setminus a} \left[\sum_{n_b^0,n_b^1,n_b^*} \binom{K-1}{n_b^0,n_b^1,n_b^*} \left(\frac{1-\pi}{2}\right)^{n_b^0+n_b^1} \times \pi^{n_b^*} \prod_{j\in V^*(b)} \int dr_b^j \rho(r_b^j) \lambda_0^l(n_b,r_b) \lambda_1^{m-l}(n_b,r_b) \right]. \quad \text{(C18)}$$

To simplify the results we approximate $\rho(r)$ by $\delta(r-\frac{1}{2})$. After some simplifications we obtain

$$\mathcal{Z}_{i\to a}=\sum_l \binom{m}{l}\Bigg\{1-2\left(\frac{1+\pi}{2}\right)^{K-1}+\pi^{K-1}+\pi^{K-1}\Bigg(1-\frac{1}{2^{K-1}}\Bigg)^m-\pi^{K-1}\left(1-\frac{1}{2^{K-1}}\right)^l-\pi^{K-1}\left(1-\frac{1}{2^{K-1}}\right)^{m-l}+\sum_n \binom{K-1}{n}\left(\frac{1-\pi}{2}\right)^{K-1-n}\pi^n\Bigg[\left(1-\frac{1}{2^n}\right)^l+\left(1-\frac{1}{2^n}\right)^{m-l}\Bigg]\Bigg\}^{L-1}. \tag{C19}$$

For $\mathcal{Z}_a$ we use Eq. (C9) and again $\rho(r)=\delta(r-\frac{1}{2})$ to get

$$\mathcal{Z}_a=\pi^K\left[1-\frac{2}{2^K}\right]^m-2\pi^K\left[1-\frac{1}{2^K}\right]^m+2\sum_n \binom{K}{n}\times\left(\frac{1-\pi}{2}\right)^{K-n}\pi^n\left[1-\frac{1}{2^n}\right]^m+\left[1+\pi^K-2\left(\frac{1+\pi}{2}\right)^K\right]. \tag{C20}$$

To obtain the free energy we still need to determine π. From Eq. (21) we have

$$1-\pi=\frac{2^{-(K-1)(L-1)+1}}{\mathcal{Z}_{i\to a}}\sum_{l=1,L-1}\binom{L-1}{l}[(1-\pi)^{K-1}]^l\times\Bigg[\sum_{\sigma_{\partial b\backslash i}} I_b(\sigma_{\partial b\backslash i})\int\prod_{j\in V(b)\backslash i} d\mathcal{Q}^{\sigma_j}[\mu]\Bigg(1-\delta_{n_b^1,0}(1-\delta_{n_b^*,0})\prod_{j\in V(b)\backslash i} r_b^j\Bigg)^{m-1}\Bigg]^{L-1-l}. \tag{C21}$$

Taking $\rho(r)=\delta(r-\frac{1}{2})$ we obtain an equation for π and for general m:

$$\frac{1-\pi}{2}=\frac{1}{\mathcal{Z}_{i\to a}}\Bigg\{\Bigg[1-\left(\frac{1+\pi}{2}\right)^{K-1}+\sum_n \binom{K-1}{n}\times\left(\frac{1-\pi}{2}\right)^{K-1-n}\pi^n\left(1-\frac{1}{2^n}\right)^m\Bigg]^{L-1}-\Bigg[1-\left(\frac{1+\pi}{2}\right)^{K-1}-\left(\frac{1-\pi}{2}\right)^{K-1}+\sum_n \binom{K-1}{n}\times\left(\frac{1-\pi}{2}\right)^{K-1-n}\pi^n\left(1-\frac{1}{2^n}\right)^m\Bigg]^{L-1}\Bigg\}. \tag{C22}$$

[1] R. G. Gallager, *Information Theory and Reliable Communication* (Wiley, New York, 1968).
[2] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, England, 2002).
[3] *NP-hardness and Phase Transitions*, edited by O. Dubois, R. Monasson, B. Selman, and R. Zecchina; [Theor. Comput. Sci. 265 (1-2) (2001)].
[4] D. Challet, M. Marsili, and R. Zecchina, Phys. Rev. Lett. **84**, 1824 (2000).
[5] T. Galla, M. Leone, M. Marsili, M. Sellitto, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **97**, 128701 (2006).
[6] A. Braunstein and R. Zecchina, Phys. Rev. Lett. **96**, 030201 (2006).
[7] C. Baldassi, A. Braunstein, N. Brunel, and R. Zecchina, Proc. Natl. Acad. Sci. U.S.A. **104**, 11079 (2007).
[8] S. Cook, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (ACM, Ohio, 1971), pp. 151–158.
[9] M. Garey and D. Johnson, *Computers and Intractability A Guide to the Theory of NP-completeness* (Freeman, San Francisco, 1979).
[10] S. Kirkpatrick and B. Selman, Science **264**, 1297 (1994).
[11] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyanski, Nature (London) **400**, 133 (1999).
[12] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, J. Stat. Phys. **111**, 505 (2003).
[13] S. Cocco, O. Dubois, J. Mandler, and R. Monasson, Phys. Rev. Lett. **90**, 047205 (2003).
[14] M. Mézard, M. Palassini, and O. Rivoire, Phys. Rev. Lett. **95**, 200202 (2005).
[15] T. Mora, Ph.D. thesis, LPTMS Université Paris-Sud, 2007 (unpublished).
[16] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborova, Proc. Natl. Acad. Sci. U.S.A. **104**, 10318 (2007).
[17] L. Zdeborova and F. Krzakala, Phys. Rev. E **76**, 031131 (2007).
[18] G. Semerjian, J. Stat. Phys. **130**, 251 (2008).
[19] F. Krzakala and J. Kurchan, Phys. Rev. E **76**, 021122 (2007).
[20] R. Monasson and R. Zecchina, Phys. Rev. E **56**, 1357 (1997).
[21] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).
[22] M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002).
[23] R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **89**, 268701 (2002).
[24] D. Achlioptas, A. Naor, and Y. Peres, Nature (London) **435**, 759 (2005).
[25] S. Mertens, M. Mézard, and R. Zecchina, Random Struct. Algorithms **28**, 340 (2006).
[26] M. Mezard, G. Parisi, and M. A. Virasoro, *Spin-Glass Theory and Beyond*, Vol. 9 of Lecture Notes in Physics (World Scientific, Singapore, 1987).
[27] M. Mézard and G. Parisi, Eur. Phys. J. B **20**, 217 (2001).
[28] M. Mézard and G. Parisi, J. Stat. Phys. **111**, 1 (2003).
[29] O. Rivoire, J. Stat. Mech.: Theory Exp. (2005), P07004.
[30] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, IEEE Trans. Inf. Theory **47**, 498 (2001).
[31] E. Friedgut, J. Am. Math. Soc. **12**, 1017 (1999).
[32] T. Castellani, V. Napolano, F. Ricci-Tersenghi, and R. Zecchina, J. Phys. A **36**, 11037 (2003).
[33] A. Braunstein, M. Mezard, and R. Zecchina, Random Struct. Algorithms **27**, 201 (2005).
[34] A. Braunstein, F. Kahyan, G. Montorsi, and R. Zecchina, in Encoding for the Blackwell Channel with Reinforced Belief

Propagation, *Proceedings of the IEEE International Symposium on Information Theory* (IEEE, Nice, France, 2007).
[35] S. Seitz, M. Alava, and P. Orponen, J. Stat. Mech.: Theory Exp. (2005), P06006.
[36] T. Mora and M. Mézard, J. Stat. Mech.: Theory Exp. (2005), P10007.
[37] G. Parisi (unpublished).
[38] D. Achlioptas and F. Ricci-Tersenghi, in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing* (ACM, Seattle., 2006), Session 4A, p. 130.
[39] A. Montanari and F. Ricci-Tersenghi, Eur. Phys. J. B **33**, 339 (2003).
[40] A. Montanari, G. Parisi, and F. Ricci-Tersenghi, J. Phys. A **37**, 2073 (2004).