BELIEF PROPAGATION WITH INFORMED INITIALIZATION IN COMBINATORIAL OPTIMIZATION PROBLEMS

PROPAGACIÓN DE MENSAJES CON INICIALIZACIÓN INFORMADA EN PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA

D. Machado[a, b, c], R. Mulet[a], A. Pérez[a†]

a) Center for Complex Systems and Department of Theoretical Physics, Faculty of Physics, University of Havana, Cuba. CP10400, La Habana, Cuba. alexperez28403@gmail.com†
b) Dipartimento di Fisica, Sapienza Università di Roma, P.le Aldo Moro 5, 00185 Rome, Italy.
c) CNR - Nanotec, unità di Roma, P.le Aldo Moro 5, 00185 Rome, Italy.
† corresponding author

Recibido 5/2/2025; Aceptado 20/12/2025

In this work, we study the behavior of the Belief Propagation (BP) algorithm in solving two combinatorial optimization problems: 3-SAT and 3-XORSAT. We examine the performance of BP on planted instances, applying the algorithm with a fraction of the variables fixed from the beginning. The probability of reaching any solution using this informed initialization displays abrupt changes with the number of fixed variables. In both problems, BP requires fixing a smaller fraction of variables than the Monte Carlo algorithm to solve a given instance. We also extended a known analytical description of the Unit Clause Propagation Algorithm to accurately predict BP's behavior with an informed initialization for 3-XORSAT in random regular graphs.

En este trabajo estudiamos el comportamiento del algoritmo Belief Propagation (BP) en la resolución de dos problemas de optimización combinatoria: 3-SAT y 3-XORSAT. Aplicamos BP a la versión plantada de estos problemas cuando una fracción de las variables está fija desde el inicio. La probabilidad de resolver una instancia usando esta inicialización informada muestra cambios abruptos con el número de variables fijadas. En ambos problemas estudiados BP requiere menos variables fijadas que el algoritmo de Monte Carlo para converger a una solución. Finalmente, extendemos una conocida descripción analítica del algoritmo Unit Clause Propagation para predecir con precisión el comportamiento de BP aplicado al 3-XORSAT en grafos aleatorios regulares con inicialización informada.

Keywords: combinatorial optimization (optimización combinatoria), planted model (modelo plantado), belief propagation (propagación de creencias)

I. INTRODUCTION

Most optimization problems are formulated via the extremization of some function $f(\vec{x})$, of N variables ($\{x\}$) that must satisfy specific constraints. For continuous variables, if the function to extremize and the constraints are linear, the exact solution of the problem is obtained using Linear Programming, a well-known general algorithm [1]. On the other hand, when the function is non-linear and the variables are discrete, we step into a field where problems are usually more difficult [2]. In computer science, these problems are usually recognized as *combinatorial optimization problems* and have common grounds with the field of disordered systems in statistical physics [3].

This connection between combinatorial optimization and statistical physics has triggered the efforts to understand the structure of the solution space of some problems [4, 5] and the (sometimes surprising) capacity of algorithms to find solutions [6–8]. However, despite the advances in the field, we lack a general algorithm to efficiently solve every problem of this type and must approach them with tailored heuristics [9–11].

Therefore, understanding the performance of general and (in many cases) successful algorithms for combinatorial optimization problems, such as Simulated Annealing [12, 13] and Belief Propagation (BP) [14, 15], continues to be a very active research area.

Here, we move forward in this direction by studying BP's performance solving K-SAT and K-XORSAT problems when it is provided with an informed initialization. Both K-SAT and random K-XORSAT are very well-known combinatorial optimization problems and have been widely used as benchmarks to test algorithmic performance [16–20]. In our informed initialization, a fraction of the variables is fixed from the beginning and then BP is run to find the right assignment of the rest of the variables of the system. Moreover, we extend some of the results in Ref. [14] to understand the behavior that we observe when BP is applied to K-XORSAT instances on random regular graphs.

The rest of the work is organized as follows. In the next section, we present in detail the two problems, including some known results to contextualize our work. The algorithm, BP, is introduced in the following section titled *Belief Propagation*. Then, we compare the performances of BP and Monte Carlo (MC) when we use informed initializations. Finally, we present the conclusions of our work.

II. TWO PROBLEMS

The K-SAT problem deals with clauses formed by the logical OR of K variables. Its computational hardness depends on the density of clauses $\alpha = \frac{M}{N}$, where M is the number of clauses and N is the number of variables. The problem is typically defined on random factor graphs with Poisson (Erdös–Rényi) connectivity. As α increases, the solution space undergoes a sequence of sharp phase transitions. For small α the system is always solvable. At any given K, there is a threshold α_s such that, for $\alpha > \alpha_s$ there are no satisfying assignments with high probability when the size N of the formula is large. On the other hand, for $\alpha < \alpha_s$ a formula will have a satisfying assignment with high probability when the size N of the formula is large. For K=3, for example, we have $\alpha_s \approx 4.267$ [4, 21]. This sharp change around α_s is called the SAT-UNSAT transition.

Notably, there is another transition at some $\alpha_d < \alpha_s$ [4, 22] where the system enters a dynamical glassy phase, characterized by clustering of solutions and slow relaxation dynamics, often referred to as the dynamical transition. For K=3, we have $\alpha_d \approx 3.86$. It is often considered that hard-to-solve instances appear in the region $\alpha_d < \alpha < \alpha_s$, identifying the location of α_d as the onset of algorithmic hardness. However, the actual relevance of this threshold for real algorithms is not completely clear, since one can find several heuristics capable of reaching solutions well beyond α_d [6, 15, 23–26].

On the other hand, the K-XORSAT problem involves logical clauses that are satisfied when the sum modulo 2 (i.e., XOR) of K Boolean variables equals a fixed value. When this problem is translated into statistical physics via a mapping to spin variables—replacing Boolean values $\{0, 1\}$ with Ising spins (± 1), it becomes equivalent to the ferromagnetic p-spin model ($p = K$), a paradigmatic system in spin-glass theory [16, 17]. In this formulation, each clause corresponds to a K-body interaction that favors the alignment of the spins, leading to a ferromagnetic Hamiltonian. This connection has made K-XORSAT a valuable benchmark for studying algorithmic thresholds and phase transitions in disordered systems [16–18].

The SAT-UNSAT and dynamic transitions also occur in random K-XORSAT at some α_d and α_s, respectively. However, there is a key difference with K-SAT: solutions of K-XORSAT can be found in polynomial time for any α up to α_s, using Gaussian elimination. In other words, we already know an efficient algorithm that solves the problem whenever this is possible. This does not mean that K-XORSAT is an easy problem. It is widely accepted that, when one restricts oneself to local algorithms such as BP or MC, the onset of hardness occurs exactly at α_d. For large K, it has been rigorously proven that finding a solution of K-XORSAT is indeed hard in the region $\alpha_d < \alpha < \alpha_s$ if one uses a local algorithm [27, 28].

In the physics literature, these two problems are usually defined using N binary variables $s_i = \pm 1$, with $i = 1, \ldots, N$. Inspired by condensed matter physics, these variables are usually called spins. The combinatorial optimization task is to find a configuration $\vec{s}^*$ that minimizes the following Hamiltonians:

1. **K-XORSAT**:

$$H(\vec{s}) = -\sum_{a=1}^{M} J_a \prod_{i \in a} s_i \tag{1}$$

where $J_a = 1$ in the original formulation of the problem, corresponding to the ferromagnetic p-pspin model. Here, we will also allow $J_a = \pm 1$ with equal probability, which in turn corresponds to the disordered p-spin. To simplify the setting, for K-XORSAT we construct $H(\vec{s})$ such that each variable i belongs to the same number of clauses, denoted by c. This builds a c-regular K-uniform random hypergraph.

2. **K-SAT**:

$$H(\vec{s}) = \sum_{a=1}^{M} \prod_{i \in a} \frac{1 - J_i^a s_i}{2} \tag{2}$$

where J_i^a indicates whether variable s_i is affirmed ($J_i^a = 1$) or negated ($J_i^a = -1$) in clause a. As usual, here we choose the clauses in the formula uniformly at random among all possible clauses. Then, each variable i belongs to a number of clauses that is Poisson distributed with mean αK.

In the physics nomenclature, the optimal configuration $\vec{s}^*$ is the ground state of the corresponding Hamiltonian at temperature $T = 0$.

To study the performance of algorithms, it is convenient to use *planted instances*, where the existence of at least one solution is guaranteed [29, 30]. The procedure used to build the so-called planted solution depends on the problem of interest, but the general idea is usually the same. One starts by pre-assigning values to N variables (initially disconnected), and then defines a graph of interactions such that the pre-assigned configuration is a solution to the problem.

In our case, these instances are constructed as follows:

1. **random K-XORSAT**: Choose N variables $s_i = \pm 1 \ \forall i$ with probability 1/2 to form a configuration $\vec{s}^*$. Then group these variables into M clauses (with indexes $a = 1, \ldots, M$) that are also selected uniformly at random. Regardless of the outcome, $\vec{s}^*$ will be the solution of an instance where the J_a are defined as $J_a = \prod_{i \in a} s_i \ \forall a$. The Hamiltonian (1) takes its minimum value ($H(\vec{s}^*) = -M$) when we build $\vec{s}^*$ and all the J_a following these rules [30].

2. **random K-SAT**: Choose N variables $s_i = 1 \ \forall i$ to form $\vec{s}^*$. Then group these variables into M clauses (with indexes $a = 1, \ldots, M$) that are also selected uniformly at random. All the clauses, of course, contain K variables and one needs to define the corresponding K parameters J_i^a. For every clause a, exclude only the case where $\forall i \in a$, and randomly extract one among all other possible choices for J_i^a, $J_i^a = -1$ and randomly extract one among all other possible choices for J_i^a. This automatically constructs a

Hamiltonian (Eq. (2)) such that $H(\vec{s}^*) = 0$, thus $\vec{s}^*$ is a solution. In practice, we set the clauses with k negated variables to appear with probability p_k, $p_0 = 1/4$, $p_1 = 0$, and $p_2 = 1/4$and $p_3 = 0$ following reference [29].

This work addresses a key question on algorithmic performance in planted instances: *What is the minimum fraction of variables that must be fixed to find a solution using Belief Propagation?*. This is answered in two contexts: i) when the variables are fixed randomly, and ii) when the variables are fixed to their correct values. In each scenario, we compare the behavior of the BP algorithm with the results of the MC algorithm.

III. BELIEF PROPAGATION

Combinatorial optimization problems involve minimizing a certain cost function, called *Hamiltonian* or energy in physics. Written generically, this function is:

$$H(\vec{s}) = \sum_{a=1}^{M} \Psi_a(s_i, i \in a) - \sum_{i=1}^{N} h_i s_i \tag{3}$$

where the term Ψ_a is a function of the spins contained in the clause with index a. Essentially, all Ψ_a functions must be minimized simultaneously. Another important definition is the average magnetization:

$$m(\vec{s}) = \frac{1}{N} \sum_{i=1}^{N} s_i. \tag{4}$$

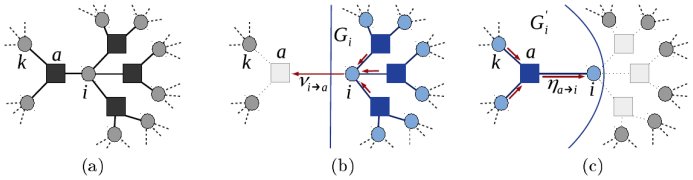

Figure 1. Visualization of the BP algorithm in the factor graph section of panel (a). In panel (b), the connection between i and the clause a is removed. The message $\nu_{i\to a}(s_i)$ from i to a is the local probability distribution associated with the spin s_i in the remaining subgraph G_i (colored in blue). In panel c), almost all the connections involving s_i have been removed, leaving only its connection with a. The message $\eta_{a\to i}(s_i)$ from a to i is the local probability distribution of s_i in the subgraph G_i' (colored in blue).

The magnetization $m(\vec{s})$ tells us about the collective orientation of the spins, which can also be guided by the local external fields h_i in Eq. (3).

The cavity method provides a way to determine local probability distributions in tree-like graphs [3,4,31]. The main idea is to make small modifications of the graph to define new *cavity* marginals or messages $\nu_{i\to a}(s_i)$ and $\eta_{a\to i}(s_i)$ (see Fig. 1). These are used afterwards to get back the marginals in the original graph $p(s_i)$.

One key point of the scheme is the BP self-consistency equations, which must be updated iteratively to obtain the fixed-point messages:

$$\nu_{i\to a}(s_i) = \frac{e^{\beta h_i s_i}}{Z_{i\to a}} \prod_{b\in \delta i\setminus a} \eta_{b\to i}(s_i) \tag{5}$$

$$\eta_{a\to i}(s_i) = \frac{1}{Z_{a\to i}} \sum_{s_{a\setminus i}} e^{-\beta\Psi_a(s_j, j\in a)} \prod_{k\in a/i} \nu_{k\to a}(s_k) \tag{6}$$

where $\beta = 1/T$ is the inverse temperature and $Z_{i\to a}$ and $Z_{a\to i}$ are normalization constants. The symbol $b \in \delta i \setminus a$ represents the set of all clauses that contain i, except a. On the other hand, $a \setminus i$ is the set of variables that belong to clause a, except for i. Therefore, the sum $\sum_{s_{a\setminus i}}$ goes over all the configurations of the spins associated with the spins in $a \setminus i$.

The BP equations provide an exact solution for problems defined on locally tree-like graphs, meaning the probability of encountering closed cycles is null or very low. Note that if $h_i \to \pm\infty$, the variable-to-factor node messages in Eq. (5) become $\nu_{i\to a}(s_i) = \Theta(h_i s_i)$, where $\Theta(x)$ is the Heaviside function. In other words, the spin is fixed towards the orientation of the local field. Once the cavity marginals $\nu_{i\to a}(s_i)$ and $\eta_{a\to i}(s_i)$ are obtained, the local probabilities can be determined as:

$$p(s_i) = \frac{e^{\beta h_i s_i}}{Z_i} \prod_{a\in\delta i} \eta_{a\to i}(s_i) \tag{7}$$

$$p(s_i, i \in a) = \frac{1}{Z_a} e^{-\beta\Psi(s_i, i\in a)} \prod_{i\in a} \nu_{i\to a}(s_i) \tag{8}$$

where Z_i and Z_a are normalization constants and $\Psi_a(s_a)$ is the weight function. Notice that, by construction, at zero temperature BP concentrates on the ground state of the corresponding Hamiltonian (with all the products equal to one). This occurs because in the limit $\beta \to \infty$, the terms related to the lowest energies dominate the exponential functions in Eq. (8). Thus, if the fixed-point iteration converges, BP gives positive weight only to configurations compatible with the ground state.

IV. RESULTS

This section is divided into two parts. First, we show and discuss the behavior of BP and MC on the random 3-XORSAT problem, and then we move towards the study of the random 3-SAT problem. We remind the reader that we define our graphs of interest using the planted model defined above, in the section "Two Problems". Then, the goal is to test whether BP is able to find the planted or another solution as a function of some initial information about it.

The results are compared with the output of the MC algorithm, also with informed initialization. For the latter, we chose the well-known Metropolis dynamic rule [32]. In each step, this algorithm chooses a variable uniformly at random and

proposes to invert its value. If the change diminishes the number of unsatisfied clauses, *i.e.*, if the energy's variation $\Delta H(s_i \to -s_i)$ is negative, the change is accepted. If $\Delta H(s_i \to -s_i) > 0$, the proposal is accepted with a probability that is proportional to $\exp(-\beta \Delta H(s_i \to -s_i))$.

IV.1. 3-XORSAT

We start our runs by fixing first a fraction (ϕ) of the spins. We consider two possibilities: i) these spins are fixed randomly, and ii) they are fixed as defined by the planted model. Once a fraction of the spins are fixed, we run BP and MC at very low temperatures (looking for the ground state of the system). In all the cases considered here, BP converged.

In Fig. 2 we show the mean absolute local magnetization $|m| = \sum_i |m_i|/N$ as a function of ϕ as obtained from BP, where $m_i = \sum_{s_i} s_i\, p(s_i)$ is the local magnetization. In the upper panel, a fraction ϕ of the spins is fixed as in the planted model. In the lower panel, a fraction ϕ of the spins is fixed in random positions.

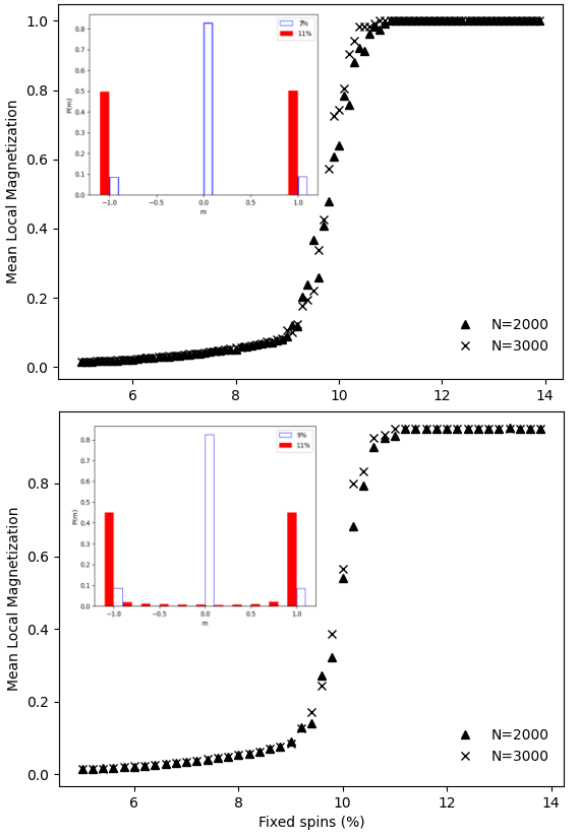

Figure 2. Behavior of the average absolute local magnetization returned by BP as a function of the percentage of fixed spins. Results are obtained for planted 3-XORSAT instances in random regular hypergraphs with connectivity $c = 4$. We imposed strong local fields in random positions and waited for BP's convergence at $T = 0.1$. Each point represents the average of 100 different runs. The inserted graphic displays a frequency histogram of local magnetizations for two distinct percentages of fixed spins. **Top panel:** Variables are fixed towards the planted solution. **Bottom panel:** Variables are fixed to random values.

In both cases, when the fraction of fixed spins $\phi < 10\,\%$ is low, BP finds paramagnetic solutions, $|m| \sim 0$. The local magnetizations m_i at most sites i are close to zero (see the histograms of m_i in the insets). Around $\phi \sim 10\,\%$ the absolute magnetization $|m|$ grows abruptly, and as ϕ increases, the spins magnetize ($|m| \sim 1$). Notice, however, that when the fraction ϕ of the spins is initially fixed in random directions (bottom panel), the algorithm finds a solution in which some of the spins are not fully magnetized. In this case, the solution of BP can not be directly transformed into a ground state configuration.

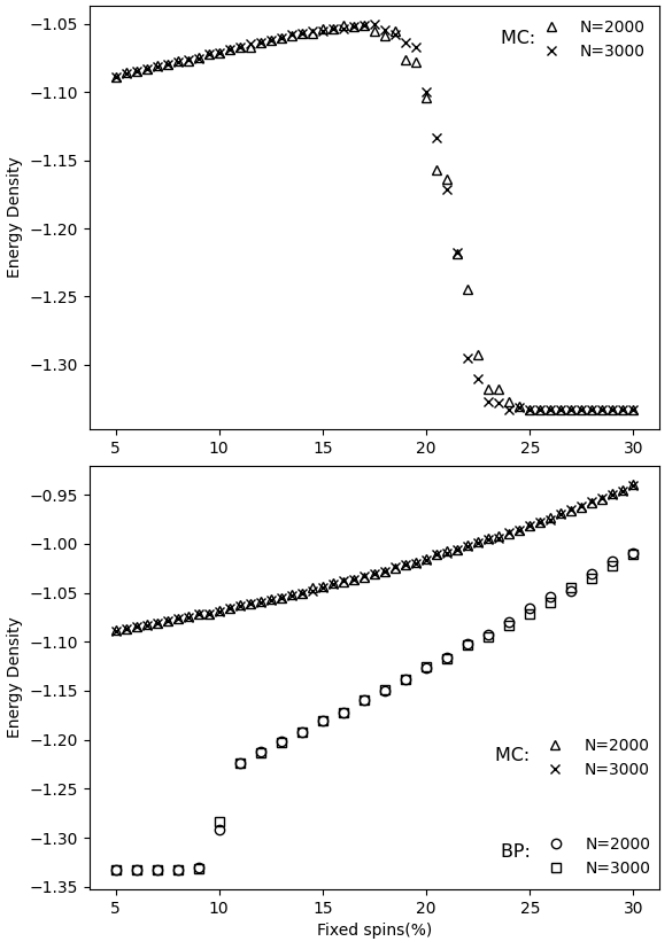

Figure 3. Final energy density returned by MC and BP as a function of the percentage of spins fixed. Results are obtained for the planted 3-XORSAT in random regular hypergraphs with connectivity $c = 4$. We imposed strong local fields in random positions and waited until convergence at $T = 0.1$. Each point represents the average of 100 different runs of the corresponding algorithm. **Top panel:** Spins are fixed towards the planted solution. Points are the result of MC runs. **Bottom panel:** Spins are fixed to random values. The panel shows results for MC and BP.

Indeed, to determine whether the distributions obtained from BP correspond to solutions, it is important to look at their local magnetization. When $m_i = 1$ ($m_i = -1$), one can safely set the corresponding spin to $s_i = 1$ ($s_i = -1$). In the cases where $|m_i| < 1$ for some i, one would have to design a subsequent way to assign s_i. Even when giving local distributions that are not fully magnetized, BP can output a small energy. In the particular case of K-XORSAT, when the local distributions $p(s_i, i \in a)$ give zero weight to the configurations that do not satisfy the clause a, we get the minimum energy. It is not necessary that $p(s_i, i \in a)$ *selects* one

specific configuration among the satisfying assignments. It is the informed initialization, fixing a subset of variables, which forces BP to converge to a specific solution.

Therefore, it does not make sense to plot the energy obtained from BP when the informed initialization coincides with the planted solution. This energy is always the minimum possible. At this point, we remind the reader that the minimum energy density (energy per variable) for the K-XORSAT is $e_{min} = -M/N$, which for regular hypergraphs with connectivity $c = 4$ and $K = 3$ coincides with $-c/K = -4/3 \approx -1.333$. However, this is not the case with the MC algorithm, which does not output local probability distributions but actual configurations.

In the top panel of Fig. 3, we show the results of MC when a fraction ϕ of the spins is oriented as in the planted model. When ϕ is small, the informed initialization does not seem to be helping MC. The energy indeed increases slowly when ϕ increases away from zero. Then, around $\phi \approx 18\,\%$, the informed initialization actually starts helping the algorithm to find the way towards the solution. Notice that the fraction of spins needed to actually recover the minimum energy of the system with MC ($\phi > 22\,\%$) is larger than the one obtained with BP ($\phi > 10\,\%$).

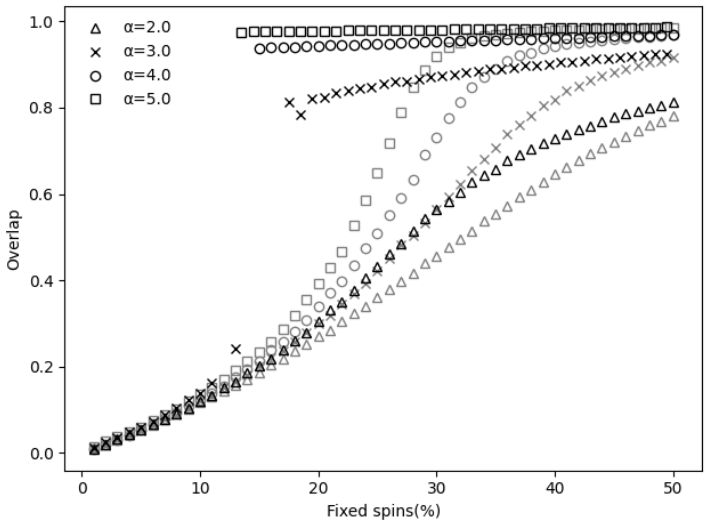

Figure 4. Comparison between the analytical description of UCP, the results of BP and MC runs for 3-XORSAT instances defined over random regular hypergraphs with connectivity $c = 4$. The graphic shows the overlap with the planted solution as a function of the percentage of spins fixed towards a planted solution. We imposed strong local fields in random positions and waited until convergence at $T = 0.1$. Note that BP and UCP reach the planted solution with a smaller fraction of fixed spins than MC. Each point represents the average of 100 different runs.

On the contrary, when the spins are fixed in random positions, it is useful to study the energy obtained after running BP, which can be seen in the bottom panel of Fig. 3. For $\phi < 10\,\%$, BP gives a small magnetization (see the bottom panel of Fig. 2) and the minimum possible energy $e \approx -1.333$. In this region, the output cannot be directly translated into actual solutions. For $\phi > 10\,\%$, as soon as the magnetization jumps to large values, the energy departs from the minimum, going up and away from the ground state. Although the fraction of fixed variables forces BP to converge to specific configurations, these are not solutions. The results indicate that even a small fraction of variables fixed in random positions can effectively obscure the planted solution. However, notice that BP consistently maintains lower energy values than MC in the same scenario. Therefore, from Figs. 2 and 3 we conclude that for $\phi > 10\,\%$ the output of BP is a set of marginals m_i, almost $90\,\%$ of which are $m_i = \pm 1$ (see the inserted graphic in the bottom panel of Fig. 2), and which correspond to relatively low energies. These can be easily translated into near-optimal configurations of the problem by taking $s_i = \text{sign}(m_i)$ for all sites.

We also measure the overlap between the solution found by the algorithms ($\vec{s}^{\,\text{BP}}$, or $\vec{s}^{\,\text{MC}}$) and the planted solution $\vec{s}^{\,*}$. One can define this overlap between any two configurations as a normalized dot product: $Q(\vec{s}, \vec{s}^{\,*}) = 1/N \sum_i s_i s_i^*$. In Fig. 4, we compare the overlap between the configurations found with BP and MC and the planted solution as a function of ϕ. The results are consistent with the analysis of the energy shown in Fig. 3. As can be easily seen, BP finds the planted solution already for $\phi > 10\,\%$, while MC converges to it only when $\phi > 22\,\%$ of the spins are fixed.

The results in Fig. 4 are analogous to the ones in Fig. 4 of Ref. [33]. However, there are some key differences. First of all, Ref. [33] deals with the ferromagnetic case ($J_a = 1$), while Fig. 4 is for the disordered case (3-XORSAT) with $J_a = \pm 1$. In the former, the magnetization is equivalent to the overlap with the ferromagnetic solution. In the latter, we plant $s_i = \pm 1$ at random and the magnetization is always zero, forcing us to measure the overlap. The second formal difference is that Ref. [33] compares MC with the results of the Cavity Master Equation (CME) [34], while we compare MC with BP. Nevertheless, Ref. [35] later demonstrated that, for dynamics that satisfy detailed balance, the fixed points of the CME are equivalent to BP. Thus, in this case, BP can also be seen as a faster way to compute the fixed points of the CME, improving the procedure used in Ref. [33].

To understand the behavior of BP, we can follow an approach developed in [14]. Let us imagine that what BP is doing is just following a simple rule: whenever $K-1$ variables in a clause are fixed, the remaining variable is also fixed by BP to the only possible value satisfying the clause. Indeed, remember that in the K-XORSAT the local constraints are satisfied only if the products $\prod_{i \in a} s_i$ are all equal to one. Therefore, knowing the values of the variables s_j, with $j \in a \setminus i$, directly gives only one possibility for the remaining s_i to satisfy the clause a.

The algorithm that proceeds like this, by propagating such logical implications throughout the graph, is called Unit Clause Propagation (UCP) and was already described using differential equations in Ref. [14] for random hypergraphs with Poisson connectivity. Here, we extend those results to the case of random regular hypergraphs. The initial condition for UCP is that a fraction ϕ of variables is already assigned (fixed). At each time step, UCP chooses a clause with $K-1$ fixed variables, and sets the remaining one to the only possible satisfying value. Therefore, if we measure the time t in Monte Carlo steps, each consisting of N simple UCP iterations, we

get a fraction of fixed variables that grows as $\theta(t) = \phi + t$. Thus, what we need to compute is the limit $\theta_\infty = \lim_{t\to\infty} \theta(t)$, to get the final fraction of fixed variables.

Let us define $\lambda(t) = 1 - \phi - t$ as the density of variables that are still unassigned at time t, and $\rho_\kappa(t)$ as the density of clauses containing κ unassigned variables at time t. For the latter, we have the differential equation:

$$\frac{d\rho_\kappa(t)}{dt} = -\delta_{\kappa,1} + \\ + \frac{(c-1)}{\sum_{\kappa'=0}^{K} \kappa' \, \rho_{\kappa'}(t)} \Big\{ (\kappa+1)\, \rho_{\kappa+1}(t) - \kappa\, \rho_\kappa(t) \Big\} \tag{9}$$

At each time step, UCP assigns a value to one variable, converting a clause with $\kappa = 1$ unassigned variables into a clause with $\kappa = 0$ unassigned variables. This explains the term $-\delta_{\kappa,1}$ in the derivative. By doing so, UCP also affects the other $c-1$ clauses that contain the newly assigned variable. At time t, each one of these other *neighboring* clauses will still have κ unassigned variables with probability $\kappa\, \rho_\kappa(t) / \sum_{\kappa'} \kappa' \, \rho_{\kappa'}(t)$. Therefore, ρ_κ must increase because some of the "neighboringçlauses with $\kappa+1$ unassigned variables will pass to have κ unassigned variables after each iteration. This explains the term $(c-1)\,(\kappa+1)\,\rho_{\kappa+1}(t) / \sum_{\kappa'} \kappa' \, \rho_{\kappa'}(t)$ in the derivative. In the same way, the term $(c-1)\,\kappa\,\rho_\kappa(t) / \sum_{\kappa'} \kappa' \, \rho_{\kappa'}(t)$ takes into account the neighboring clauses that go from κ to $\kappa - 1$ unassigned variables.

Now, we must solve Eq. (9) with the initial condition:

$$\rho_\kappa(0) = \frac{c}{K} \binom{K}{\kappa} (1-\phi)^\kappa \phi^{K-\kappa} \tag{10}$$

To do it, it is enough to follow the same procedure as in Appendix A.1 of Ref. [14], and we obtain:

$$\rho_1(t) = c\,(1-\phi)\,[\,b(t)\,]^{c-1} \Big\{ b(t) - 1 + \\ \Big[\, \theta + (1-\theta)\,(1 - [\,b(t)\,]^{c-1}) \,\Big]^{K-1} \Big\} \tag{11}$$

where $b(t) = [(1-\phi-t)/(1-\phi)]^{1/c}$.

Eq. (11) is valid only for the time interval when UCP is still running. As discussed above, the algorithm will stop at the time t^* when it exhausts the clauses with only one unassigned variable, that is, when $\rho_1(t^*) = 0$. To calculate the final fraction of variables assigned by UCP ($\theta(t^*) = \phi + t^*$), given the initial fraction of fixed variables ϕ, and the values of K and c, we need to numerically obtain the time t^* that solves the equation:

$$b(t^*) - 1 + \Big[\, \theta + (1-\theta)\,(1 - [\,b(t^*)\,]^{c-1}) \,\Big]^{K-1} = 0 \tag{12}$$

In Fig. 4 we also show this analytical description (continuous lines). We compare it with the overlap between the marginals of BP and the planted solution. The agreement is very good, and we believe the differences are due to finite-size effects in the numerical analysis with BP. Using UCP, we estimate with more precision that the abrupt change in overlap occurs at $\phi_c = 0.09575(5)$.

These results indicate that the fundamental mechanism used by BP to find a correct final assignment is analogous to the way UCP works. It propagates logical implications throughout the graph by assigning the variables that belong to clauses with other $K-1$ already assigned variables. Below ϕ_c, it does not go very far and assigns a few variables in total. On the other hand, a small fraction of initially fixed variables $\phi \geq \phi_c$ provokes an avalanche of assignments that ultimately covers the whole graph.

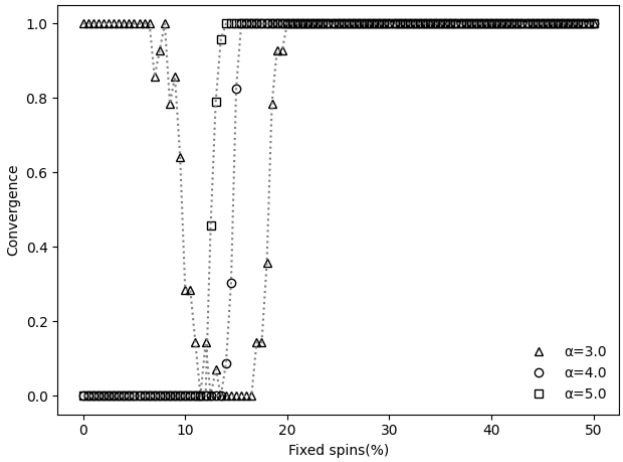

Figure 5. BP convergence probability as a function of the percentage of spins fixed towards a planted solution. Each point corresponds to the average of 30 BP runs on formulas with size $N = 10^4$ at temperature $T = 0.1$.

IV.2. 3-SAT

The K-SAT problem belongs to the NP-complete class. In particular, when defined over random graphs with $K = 3$, it is well known that BP with random initialization fails to converge above $\alpha_d = 3.86$ [3]. However, this is not necessarily the case for the planted ensemble of graphs, where the situation is more complex. In Fig. 5, we show how an informed initialization in planted instances changes the algorithm convergence. For $\alpha = 3$, below the convergence threshold $\alpha_d = 3.86$ of the ensemble of Poisson graphs, we can see that after fixing a few spins ($\phi < 10\,\%$) towards the planted solution, the convergence is unaffected. However, when the fraction of fixed variables passes $\phi \sim 10\,\%$ BP stops converging. We find a whole interval where the algorithm does not converge, that for $\alpha = 3$ extends from $\phi \approx 10\,\%$ to $\phi \approx 20\,\%$.

For $\alpha > \alpha_d$, instead, BP loses convergence also for small values of ϕ. Only when enough variables are fixed in the direction of the planted solution, we see that BP starts converging. This happens abruptly at a given value of ϕ ($\phi \approx 15\,\%$ for $\alpha = 4$ and $\phi \approx 12\,\%$ for $\alpha = 5$), in a behavior reminiscent of what we observed for the 3-XORSAT problem in Figs. 2, 3, and 4.

As said above, in the absence of fixed variables BP fails to converge for $\alpha > \alpha_d$ [3]. Thus, it is not surprising that BP converges for $\alpha = 3$ when $\phi = 0$, while it does not converge for $\alpha = 4$ and $\alpha = 5$ (see Fig. 5). However, the abrupt ascent observed when ϕ is large enough for all three values of α, and the non-monotonicity displayed for $\alpha = 3$, are non-trivial behaviors that we should discuss in more detail.

As identified in Ref. [14] and verified by us in the case of 3-XORSAT with homogeneous connectivity, fixing variables makes BP follow logical implications and effectively assign other variables, in analogy to the Unit Clause Propagation (UCP) algorithm. When the initial fraction of fixed variables ϕ exceeds a critical value ϕ_c, the final fraction of assigned variables $\theta(t^*)$ increases abruptly from a finite value to $\theta(t^*) \sim 1$. Once the assigned variables are removed, we are left with a trivialized constraint satisfaction problem for the remaining part of the system, and BP is able to converge. The latter is possible because the variables are set exactly to their value in the planted solution, avoiding the creation of local contradictions and causing the final ascent that we see for all values of α in Fig. 5.

While for $\alpha = 4$ and $\alpha = 5$ the ascent is the only relevant feature, for $\alpha = 3$ we first have a descent in the probability of convergence. The authors of Ref. [14] demonstrate, for the 3-XORSAT problem, that a large enough fraction ϕ of fixed variables can lead to a remaining constraint satisfaction problem that is inside the clustered phase, even though the solutions of the original problem were not clustered. In other words, it is possible to depart from some $\alpha < \alpha_d$ where the problem is easy to solve (and BP converges), and start fixing variables to a point where the remaining formula is effectively much harder to solve (and BP stops converging). Contradictions will arise because of the presence of cycles in the subjacent hypergraph, and BP's messages will oscillate indefinitely. Nevertheless, if we keep fixing new variables, at some ϕ_c the final fraction of assigned variables jumps to $\theta(t^*) \sim 1$ and the problem becomes easy once more. Thus, BP converges again. This is a plausible explanation for the non-monotonicity of the probability of convergence that we observe for $\alpha = 3$ in Fig. 5.

In Fig. 6, we present the energy (after convergence) for MC as a function of ϕ for different values of α. Notice that this algorithm recovers the ground state of the system only when ϕ is very large. Moreover, the curve is not monotonic. As with the 3-XORSAT in the top panel of Fig. 3, the energy first grows when ϕ is increased. However, once a certain number is fixed, the system begins recognizing the planted solution of the instance.

BP, on the other hand, needs a smaller fraction of fixed variables to find the planted solution. To show it, we plot the overlap defined as before (Fig. 7) as a function of ϕ, both for BP and for MC final configurations. Noticing that BP does not always converge, we include in Fig. 7 only the statistics obtained over the instances where BP converged. We observe again two different behaviors depending on the value of α. For $\alpha < \alpha_d$ ($\alpha = 2$ and $\alpha = 3$ in the figure), BP converges for small values of ϕ, but the final overlap with the planted solution is small. Once BP regains convergence for larger values of ϕ, it finds probability distributions that have a high overlap with the planted solution.

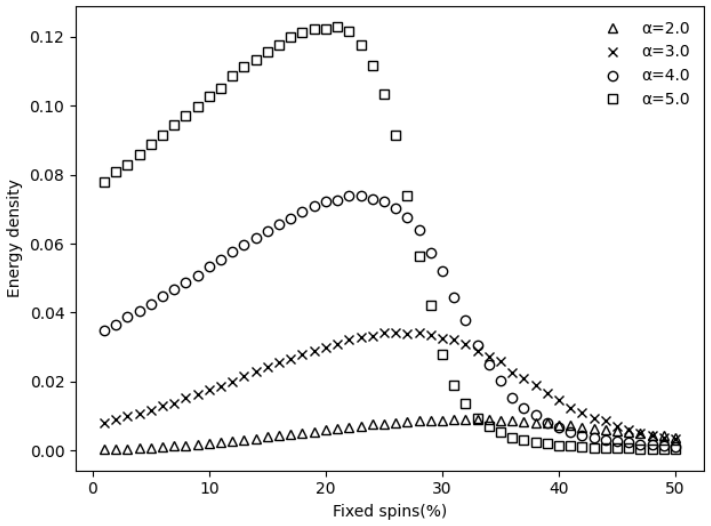

Figure 6. Intensive energy of the final configurations obtained by running MC on random 3-SAT instances as a function of the percentage of fixed spins. The variables are fixed towards a planted solution. The points represent the average of 100 MC runs over formulas of size $N = 1024$ at temperature $T = 0.1$.

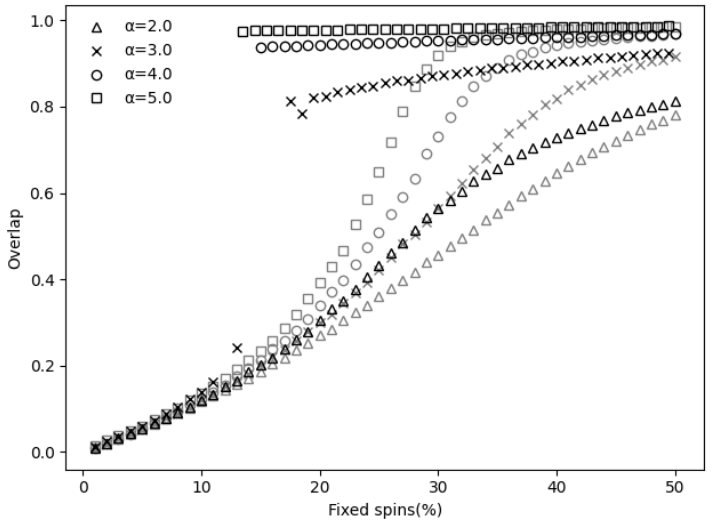

Figure 7. Overlap with the planted solution as a function of the percentage of fixed spins in the 3-SAT problem with $N = 10^4$ at temperature $T = 0.1$. Gray points represent the results of MC, and black points those of BP. The variables are fixed towards the planted solution. Each point represents the average of 20 BP algorithm runs (100 MC runs).

For $\alpha > \alpha_d$, as said before, BP does not converge for small ϕ. However, as soon as it starts converging, the overlap with the planted configuration is close to one. The overlap increases when α increases, as expected in the inference problem. Thus, our results show that running BP with an informed initialization serves two purposes simultaneously: it helps convergence and it drives BP towards the planted solution.

Moreover, the planted solution always has a larger overlap

with the probability densities of BP than with the MC final configurations. The number of fixed spins needed by MC to find the solution is, in any case, much larger than the number of spins needed by BP. For example, when $\alpha = 4.0$ we see that MC starts having overlap close to one around $\phi \approx 40\,\%$, while BP has comparable overlaps already at $\phi \approx 15\,\%$.

V. CONCLUSIONS

Our results show that BP is more efficient than MC in finding solutions with an informed initialization. Both for 3-XORSAT and 3-SAT, BP requires a smaller proportion of fixed spins to find the planted solution. Even when the spins are fixed towards random directions, the final configurations obtained from BP for 3-XORSAT instances have low energies. This method could then be used as a starting point to find near-optimal configurations. Furthermore, using an informed initialization helps BP to converge in the planted 3-SAT, in regions where otherwise it would not achieve convergence.

Moreover, we have shown that the behavior of BP when run on 3-XORSAT instances can be accurately reproduced using UCP, a simple algorithm whose analytic description was already included in Ref. [14]. Here, we extended the equations to consider random regular hypergraphs, and show that the fundamental mechanism followed by BP with an informed initialization is the same included in the definition of UCP. Taking advantage that a warm start is given, with a fraction of variables fixed, the algorithm propagates logical implications throughout the graph by assigning the variables that belong to clauses with other $K-1$ already assigned variables.

Regarding 3-SAT, one should recall the close relation between the 1/4 model that we used to plant solutions, and the 3-XORSAT problem [29]. Indeed, in this specific case, the planted instances can be transformed into equivalent instances of 3-XORSAT. Thus, it is not surprising that we observe qualitative similarities between both scenarios. However, it must be said that the local algorithms, BP and MC, are not informed that such a transformation exists. If they exploit it in any way, it would be an interesting phenomenon to be investigated in future work.

REFERENCES

[1] R. Dorfman, P. A. Samuelson, and R. M. Solow, Linear programming and economic analysis, international student ed. (McGraw-Hill Kogakusha, New York, 1958).

[2] S. Cook, in Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (ACM, New York, 1971), p. 151.

[3] M. Mezard and A. Montanari, Information, physics, and computation (Oxford University Press, Oxford, 2009).

[4] M. Mezard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).

[5] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, and L. Zdeborova, Proc. Natl. Acad. Sci. **104**, 10319 (2007).

[6] M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, and S. Seitz, Proc. Natl. Acad. Sci. **105**, 15253 (2008).

[7] D. Achlioptas and A. Coja-Oghlan, in 2008 49th Annual IEEE Symposium on Foundations of Computer Science (IEEE, New York, 2008), pp. 793-802.

[8] E. Aurell, E. Domínguez, D. Machado, and R. Mulet, Phys. Rev. Lett. **123**, 230602 (2019).

[9] C. H. Papadimitriou, in Proceedings of the 32nd IEEE Symposium on the Foundations of Computer Science (IEEE Computer Society, New York, 1991), pp. 163-169.

[10] E. Aurell, U. Gordon, and S. Kirkpatrick, in Advances in Neural Information Processing Systems, Vol. 17, edited by L. Saul, Y. Weiss, and L. Bottou (MIT Press, Cambridge, 2004).

[11] S. Seitz, M. Alava, and P. Orponen, J. Stat. Mech. Theory Exp. **2005**, P06006 (2005).

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[13] M. C. Angelini and F. Ricci-Tersenghi, Phys. Rev. X **13**, 021011 (2023).

[14] F. Ricci-Tersenghi and G. Semerjian, J. Stat. Mech. Theory Exp. **2009**, P09001 (2009).

[15] L. Budzynski, F. Ricci-Tersenghi, and G. Semerjian, J. Stat. Mech. Theory Exp. **2019**, 023302 (2019).

[16] F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, Phys. Rev. E **63**, 026702 (2001).

[17] S. Franz, M. Leone, F. Ricci-Tersenghi, and R. Zecchina, Phys. Rev. Lett. **87**, 127209 (2001).

[18] A. Montanari and F. Ricci-Tersenghi, Phys. Rev. B **70**, 134406 (2004).

[19] M. Bellitti, F. Ricci-Tersenghi, and A. Scardicchio, Phys. Rev. Res. **3**, 043015 (2021).

[20] M. Bernaschi, M. Bisson, M. Fatica, E. Marinari, V. Martin-Mayor, G. Parisi, and F. Ricci-Tersenghi, Europhys. Lett. **133**, 60005 (2021).

[21] S. Mertens, M. Mezard, and R. Zecchina, Random Struct. Algorithms **28**, 340 (2006).

[22] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian, J. Stat. Mech. Theory Exp. **2008**, P04004 (2008).

[23] H. Zhou, Eur. Phys. J. B **73**, 617 (2010).

[24] F. Krzakala and J. Kurchan, Phys. Rev. E **76**, 021122 (2007).

[25] M. C. Angelini and F. Ricci-Tersenghi, Phys. Rev. E **100**, 013302 (2019).

[26] A. G. Cavaliere and F. Ricci-Tersenghi, J. Stat. Mech. Theory Exp. **2025**, 053405 (2025).

[27] D. Gamarnik and M. Sudan, in Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (Association for Computing Machinery, New York, 2014), pp. 369-376.

[28] D. Gamarnik and A. Jagannath, Ann. Probab. (2019).

[29] W. Barthel, A. K. Hartmann, M. Leone, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **88**, 188701 (2002).

[30] F. Krzakala and L. Zdeborova, Phys. Rev. Lett. **102**, 238701 (2009).

[31] F. Zamponi, Mean field theory of spin glasses, arXiv:1008.4844 (2014).

[32] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[33] E. Aurell, E. Domínguez, D. Machado, and R. Mulet, Phys. Rev. E **97**, 05103(R) (2018).
[34] E. Aurell, G. Del Ferraro, E. Domínguez, and R. Mulet, Phys. Rev. E **95**, 052119 (2017).
[35] E. Domínguez, D. Machado, and R. Mulet, J. Stat. Mech. Theory Exp. **2020**, 073304 (2020).