

Monte Carlo algorithms are very effective in finding the largest independent set in sparse random graphs

Maria Chiara Angelini¹ and Federico Ricci-Tersenghi^{1,2}

¹*Dipartimento di Fisica, Università “La Sapienza,” Piazzale Aldo Moro 5, 00185 Rome, Italy*

²*INFN, Sezione di Roma 1, and Rome Unit, CNR Nanotec, Piazzale Aldo Moro 5, 00185 Rome, Italy*



(Received 30 April 2019; published 2 July 2019)

The effectiveness of stochastic algorithms based on Monte Carlo dynamics in solving hard optimization problems is mostly unknown. Beyond the basic statement that at a dynamical phase transition the ergodicity breaks and a Monte Carlo dynamics cannot sample correctly the probability distribution in times linear in the system size, there are almost no predictions or intuitions on the behavior of this class of stochastic dynamics. The situation is particularly intricate because, when using a Monte Carlo-based algorithm as an optimization algorithm, one is usually interested in the out-of-equilibrium behavior, which is very hard to analyze. Here we focus on the use of parallel tempering in the search for the largest independent set in a sparse random graph, showing that it can find solutions well beyond the dynamical threshold. Comparison with state-of-the-art message passing algorithms reveals that parallel tempering is definitely the algorithm performing best, although a theory explaining its behavior is still lacking.

DOI: [10.1103/PhysRevE.100.013302](https://doi.org/10.1103/PhysRevE.100.013302)

I. INTRODUCTION

Discrete optimization problems defined on graphs are widespread among many scientific disciplines and commonly found in real-world applications. Depending on the properties of the underlying graph, these optimization problems may become so hard to solve that all known algorithms find only very suboptimal solutions, while the optimal ones remain unreachable to algorithms running in polynomial time.

A common benchmark to test the effectiveness of search algorithms is represented by optimization problems defined on random graphs (typical case analysis). In this case, the hardness of the optimization problem can usually be controlled by varying continuously a model parameter (e.g., the random graph mean degree or the solution size), and different algorithms can be quantitatively compared on the basis of how close to optimality they can go.

Unfortunately, in optimization problems that, in the worst case analysis, are NP-hard and also hard to approximate, a large algorithmic gap is often present in the typical case analysis, i.e., all known algorithms stop working at an algorithmic threshold which is bounded far away from the optimal (information-theoretic) threshold. Computing the ultimate algorithmic threshold in these hard problems and understanding whether and why such an algorithmic threshold remains below the optimal one are fundamental open questions. The present work takes a step towards answering these questions, by studying the problem of finding a large independent set (IS) in a random regular graph (RRG).

Given a graph $G = (V, E)$, an IS is a subset of vertices $S \subset V$ such that no vertices in S are adjacent, that is, $(ij) \notin E \forall i, j \in S$. Finding the largest IS in a graph is a fundamental problem that is hard in the worst case (NP-hard) and tightly related to minimum vertex cover and maximum clique [1]. In physics, the problem is known under the name of hard-core

model [2], because vertices in S can be seen as particles that have a hard-core interaction and cannot be adjacent. The largest IS thus corresponds to the densest packing configuration in the hard-core model.

We call ρ the relative size of the IS, that is, $|S| = \rho|V| = \rho N$. On RRGs of constant degree d it has been proved that, in the large- N limit, ISs with $\rho < \rho_{\max} \sim 2 \log d/d$ do exist with high probability for d large enough [3,4]. However, algorithms running in polynomial time cannot find ISs with $\rho > \rho_{\text{alg}} \sim \log d/d$ for d large enough [5]. Actually, this algorithmic threshold ρ_{alg} can be achieved with very simple algorithms [6]. The algorithmic gap, which is the strict inequality $\rho_{\text{alg}} < \rho_{\max}$, has been proven for a class of local algorithms in the large- d limit [7]. In this case, the origin of the algorithmic failure is due to the ergodicity breaking taking place at ρ_{alg} ; this is a common phenomenon in optimization problems [8,9], also called clustering or shattering of the solution space.

One expects the ergodicity breaking taking place at ρ_{alg} to affect also other types of algorithms. In particular, the sampling of the optimal solutions through numerical methods based on the Monte Carlo Markov chain should become much slower when ergodicity is broken, due to the need to overcome large barriers. However, if one is just interested in finding a single optimal or very close to optimal solution, Monte Carlo methods may work better than expected. This is an issue that is one of the main motivations for the present work.

We are going to analyze the performances of different algorithms, dedicating particular attention to those based on Monte Carlo Markov chains, and we will try to relate such performances to the relevant phase transitions taking place in the space of ISs in the limit of large RRGs. Indeed, studying the thermodynamics of the problem via the cavity method, the authors of Ref. [10] showed how the space of ISs changes while increasing ρ : For $d < 16$ it undergoes a continuous

phase transition from a replica symmetric (RS) phase to a phase described by a full replica symmetry breaking (FRSB) solution, while for $d \geq 16$ the space of ISs undergoes a random first-order transition (RFOT) and can be described by a solution with one step of replica symmetry breaking (1RSB).

Let us briefly review the important phase transitions in the RFOT case, each one corresponding to a drastic change in the structure of the set of ISs. At small densities ρ , the ISs form a single large cluster [two ISs are considered adjacent if they differ in $o(N)$ vertices] and can be well described by an RS solution that assumes the existence of a single state. Increasing the density, one first finds a dynamical threshold ρ_d above which the space of ISs is divided into an exponential number N of distinct clusters. This is the ergodicity breaking phase transition that affects local search algorithms and Monte Carlo methods for sampling. At the condensation threshold $\rho_c > \rho_d$ the number of clusters becomes subexponential, and beyond the maximum density ρ_{\max} there are no more ISs. This last threshold is equivalent to the satisfiability/unsatisfiability threshold in constraint satisfaction problems (CSPs).

Besides the above thermodynamic transitions, another property has been conjectured to be important for understanding the origin of the algorithmic complexity in CSPs: the concept of frozen clusters [9,11,12]. A cluster of solutions is said to be frozen if it contains frozen variables that take the same value in all the solutions of that cluster. The rigidity threshold ρ_r is defined such that for $\rho > \rho_r$ typical clusters are frozen, while above the freezing transition ρ_f all clusters are frozen. In CSPs many smart algorithms can find solutions in the clustered phase, but even the best performing ones do not find frozen solutions [13]. For this reason, the freezing threshold is conjectured to be the ultimate algorithmic threshold. Unfortunately, its analytic computation is a very difficult task, which has been achieved only in random hypergraph bicoloring at present [14].

In Ref. [10] ρ_d , ρ_c , and ρ_{\max} were computed for $d \leq 100$. We just want to emphasize that even for $d = 100$ the value of ρ_{\max} is still far from its large- d limit $\rho_{\max} \sim 2 \log d/d$. In this work we will show that for $d \leq 100$, $\rho_{\text{alg}} > \rho_d$ and far from the large- d limit $\rho_{\text{alg}} \sim \rho_d \sim \log d/d$. This clearly indicates that the finite- d behavior is extremely different from that of the large- d limit, to which rigorous results are often relegated.

We will analyze different kinds of algorithms running in polynomial times. We avoid using algorithms that are known to find the largest IS in time typically growing exponentially in the graph size since these are impractical. Three main classes of polynomial algorithms will be considered: greedy algorithms, Monte Carlo methods, and message passing algorithms. Greedy algorithms are very popular [15–17] because they are extremely fast and often provide a reasonably large IS.

We will mainly focus on Monte Carlo-based algorithms that have been much less studied. Indeed, the common belief is that a slow enough simulated annealing (SA) is able to reach densities not larger than the bottom of the equilibrium states at ρ_d [18]. Above ρ_d ergodicity is broken and Monte Carlo methods should not be able to sample correctly the equilibrium properties of the model. However, it is always possible that there are states accessible to the out-of-equilibrium

dynamics that terminate at densities $\rho > \rho_d$ and thus an out-of-equilibrium process can find very large ISs with $\rho > \rho_d$.

Recently, it has been proposed to enhance the weight of deep large states in an efficient way by coupling some replicas of the system, for example, in an SA algorithm [19]. The replicated SA (RSA) has been seen to enhance the performances of learning in some models of neural networks. Here we apply RSA to the problem of finding the largest IS problem, discovering indeed that this algorithm is able to find solutions when the standard SA is not able to, well beyond ρ_d . However, this seems to be true only if the transition is strongly discontinuous (RFOT). In the case where the transition is weakly discontinuous (or continuous), RSA and SA show similar performances.

Finally, we will analyze the behavior of parallel tempering (PT). Although PT has been invented to sample at equilibrium the very rough energy landscape of disordered systems and posterior distributions [20,21], it can be used in the out-of-equilibrium regime to try to reach some of the lowest-energy configurations [22]. Recently, PT has been applied to the planted IS problem, allowing one to find the planted configuration in the supposedly hard regime (i.e., when the planted IS is very small) in a time that seems to scale polynomially with the system size [23]. In the random case, we show here that PT is able to find solutions above the algorithmic threshold of the SA and of all the other analyzed algorithms, included belief propagation with reinforcement, which is usually the best-performing message passing algorithm in other optimization problems, able to go beyond the rigidity transition [24]. We will measure the scaling of the convergence time for PT, showing that indeed it stays polynomial for $\rho > \rho_d$.

II. DEFINITION OF THE PROBLEM AND DESCRIPTION OF ALGORITHMS ANALYZED

In this section we report the details of the problem and of the algorithms whose performances are analyzed in the rest of the paper.

The optimization problem we try to solve is to find the largest IS in a given RRG. Since K is the size of an IS, we call $\rho = K/N$ its density. Finding the largest IS problem is clearly a zero-temperature problem since it imposes strong constraints on any pair of nearest-neighbor vertices not to be in the IS. As usual in a statistical mechanics approach, we can add a temperature parameter $T = 1/\beta$ and relax the strong constraints into soft ones. The probability measure can be written as

$$P(\underline{n}) \propto \exp \left[\mu \sum_{i=1}^N n_i - \beta \sum_{(ij) \in E} n_i n_j \right], \quad (1)$$

where $n_i \in \{0, 1\}$. In the $T \rightarrow 0$ limit, vertices with $n_i = 1$ form the IS and the largest IS can in principle be achieved by sending $\mu \rightarrow \infty$ afterward.

In practice, we are going to approach such a limit ($T \rightarrow 0$ and $\mu \rightarrow \infty$) in two different ways. In the first way, we fix the IS size K such that the first term in the measure in Eq. (1) is constant and can be ignored and we study the problem in temperature. In the second way we fix $T = 0$, making

constraints hard, that is, we rewrite the measure as

$$P(\underline{n}) \propto \exp \left[\mu \sum_{i=1}^N n_i \right] \prod_{(ij) \in E} (1 - n_i n_j), \quad (2)$$

and we study the problem by increasing μ .

We will use many different algorithms, described in the following. Each algorithm will show its own algorithmic threshold ρ_{alg} above which that algorithm is not able to find the IS.

Greedy algorithm (GA). Greedy algorithms are linear time algorithms where variables are set just once during the process of finding an IS. They differ according to the rule which is used to select the next vertex to include in the growing IS. Schematically, they work as follows.

- (i) Start with all $n_i = 0$.
- (ii) At each step choose a vertex v from the graph and add it to the IS, i.e., set $n_v = 1$.
- (iii) The vertex is chosen uniformly at random in the random vertex version and so as to have the smallest degree in the minimum degree version.
- (iv) All the neighbors of the chosen vertex are removed from the graph.

The random vertex version has been designed by Karp and Sipser [25] and produces with high probability an IS of size $N \log(d + 1)/d$ at both finite and large d . The minimum degree version has been introduced in Ref. [26] and gives better results, at least for finite d , while it has the same scaling at large- d values. The computational time of the greedy algorithm scales as $O(dN)$, which is linear in the graph size.

Monte Carlo in temperature (β MC). We fix the size K of the IS we would like to find and the temperature $T = 1/\beta$ to be used in the Monte Carlo algorithm. The algorithm will sample configurations with exactly K variables set to $n = 1$, that is, $\sum_i n_i = K$; each of these configurations can be equivalently described in terms of the subset of vertices containing a particle $\mathcal{I} \equiv \{i \in V : n_i = 1\}$. With each configuration we associate the energy $E(\underline{n}) = \sum_{(ij) \in E} n_i n_j$ counting how many pairs of nearest neighbors are filled ($n = 1$). A configuration of zero energy is an IS of size K .

We start by choosing \mathcal{I} as a random subset of K vertices of V . At each step of the algorithm we propose to move a randomly chosen particle to a randomly chosen empty vertex; the particle and the empty vertex do not need to be nearest neighbors, so the algorithm is not standard diffusion. Calling \underline{n} the current configuration and \underline{n}' the proposed configuration, we follow the standard Metropolis rule for accepting the proposed configuration, that is, we accept the change with probability 1 if $E(\underline{n}') \leq E(\underline{n})$ and with probability $\exp[\beta(E(\underline{n}') - E(\underline{n}))]$ otherwise. As done conventionally, we define a Monte Carlo sweep (MCS) as the attempt to move a randomly chosen particle, repeated K times. We stop the algorithm when a configuration $\underline{n}_{\text{IS}}$ with $E(\underline{n}_{\text{IS}}) = 0$ is found, which corresponds to an IS.

Parallel tempering in temperature (β PT). We consider N_β replicas, each one with exactly K variables set to $n = 1$ as in the β MC method discussed above. Each replica undergoes a standard Metropolis evolution at inverse temperature $\beta_i = \beta_{\text{max}} - i\Delta\beta$, $i \in [0, N_\beta - 1]$. Every five steps of β MC a temperature swapping step is attempted for each pair of

configurations at nearby temperatures β_i and β_{i+1} ; the temperature swap is accepted with probability

$$p = \min(1, e^{(\beta_i - \beta_{i+1})(E_i - E_{i+1})}), \quad (3)$$

where E_i is the current value of the energy of the i th replica. The algorithm is stopped when a replica (usually the one with the lowest temperature) reaches a zero-energy configuration.

Simulated annealing in chemical potential (μ SA). Working directly at zero temperature, i.e., sampling the measure in Eq. (2), we run a simulated annealing scheme in the following way. We start from the empty configuration $n_i = 0 \forall i$ that certainly satisfies all the constraints and from a null chemical potential $\mu = 0$. At each step of the SA algorithm we increase the chemical potential by $\Delta\mu$ and we do a Monte Carlo sweep, which corresponds to the attempt to update each of the N variables n_i following the usual Metropolis rule: In practice, if $n_i = 0$, we set $n_i = 1$ only if all the nearest neighbors are empty, and if $n_i = 1$, we set $n_i = 0$ with probability $\exp(-\mu)$. We stop the SA algorithm at a value μ_{max} where we observe the IS density $\rho = \sum_i n_i/N$ no longer increasing on any reasonable timescale. The algorithm, at fixed parameter $\Delta\mu$, is linear in the size N .

Replicated simulated annealing in chemical potential (μ RSA). In Ref. [19] a replicated version of the SA was proposed to sample with higher probability states with larger entropy. To define the replicated SA, we introduce R replicas of the variables on the same RRG and a coupling between the different replicas according to the following measure:

$$P(\underline{n}^1, \dots, \underline{n}^R) \propto \exp \left[\mu \sum_{a=1}^R \sum_{i=1}^N n_i^a + \gamma \sum_{a < b} \sum_{i=1}^N n_i^a n_i^b \right] \times \prod_{a=1}^R \prod_{(ij) \in E} (1 - n_i^a n_j^a). \quad (4)$$

We then run the SA algorithm on this replicated system, fixing the value of γ and incrementing the value of μ as in the μ SA. At variance with numerical experiments in Ref. [19], where γ is incremented during the annealing, we prefer to keep γ fixed as we have seen that varying γ does not improve the final result.

Parallel tempering in chemical potential (μ PT). We consider N_μ replicas of the system, each replica being at a different chemical potential: $\mu_i = \mu_{\text{max}} - i\Delta\mu$, $i \in [0, N_\mu - 1]$. For each replica, we run five Metropolis Monte Carlo sweeps at the corresponding chemical potential and then we try to swap configurations between close by values of the chemical potential with probability

$$p = \min(1, e^{(\mu_i - \mu_{i+1})(-K_i + K_{i+1})}), \quad (5)$$

where K_i is the actual number of variables set to 1 in the i th replica. We stop the simulation if a replica (usually the one of index 0) reaches the IS size K we aim at.

Belief propagation with reinforcement (BPR). The belief propagation (BP) equations for the present problem were already derived in Ref. [10],

$$\pi_{i \rightarrow j} = \frac{e^\mu \prod_{k \in \partial i \setminus j} (1 - \pi_{k \rightarrow i})}{1 + e^\mu \prod_{k \in \partial i \setminus j} (1 - \pi_{k \rightarrow i})}, \quad (6)$$

TABLE I. Relevant physical thresholds ρ_d , ρ_c , and ρ_{\max} , computed in the thermodynamic limit via the 1RSB ansatz in Ref. [10] and the algorithmic thresholds found in this work for many different algorithms searching for the largest IS in a RRG of degree $d = 20$ and $d = 100$. The random vertex (RV) and minimum degree (MD) are two versions of a greedy algorithm (GA) whose computational time scales as $O(dN)$. In this case ρ_{alg} has been estimated for size $N = 10^6$, which is large enough that finite-size effects are not present. For the Monte Carlo–based algorithms (βMC , βPT , μSA , μRSA , and μPT), a linear dependence on N is hidden in the single MCS. In addition to this, for βMC , βPT , and μPT one should study the number of MCSs needed to reach the wanted solution to understand their computational complexity. For βMC we did not perform a detailed study because it is a suboptimal algorithm; we just extracted ρ_{alg} for sizes $N = 5 \times 10^4$ for which finite-size scaling is no longer present (see Fig. 1 and Table II). For βPT and μPT , the extrapolation of ρ_{alg} has been done by fitting data with Eq. (10), which includes a power-law dependence of the convergence time both on ρ and on N (see Figs. 2 and 6 and Table III). In our approach μSA and μRSA are linear algorithms because we use a rate $\Delta\mu = 10^{-7}$ independent of N . Here ρ_{alg} is extracted as the average maximum density reached for $\mu \rightarrow \infty$ for size $N = 5 \times 10^4$, which is large enough to not have finite-size effects.

d	ρ_d	ρ_c	ρ_{\max}	RV GA	MD GA	βMC	βPT	μSA	μRSA	μPT	BPR
20	0.1830	0.1833	0.1948	0.1512(1)	0.1737(1)	0.1906(4)	0.1943(2)	0.1937(1)	0.19370(5)	0.1945(1)	0.1933(1)
100	0.0638	0.0664	0.0674	0.0447(1)	0.0572(2)	0.0642(1)	0.0657(1)	0.06470(3)	0.06479(1)	0.0655(1)	0.0650(1)

where $\pi_{i \rightarrow j}$ is the probability to have $n_i = 1$ in a modified graph where edge (ij) has been removed. These equations for $\mu < \mu_c$ converge to a homogeneous paramagnetic fixed point (FP). To turn the BP equations into a solver, one can add a reinforcement term, initially introduced in Ref. [27], with two parameters γ and dt that tune, respectively, the strength and the speed of update of the reinforcement term. Practically, the equations for the update of the messages becomes

$$\pi_{i \rightarrow j}^{t+1} = \frac{e^{\mu[\theta_i(t)]^{1-\gamma_t}} \prod_{k \in \partial i \setminus j} (1 - \pi_{k \rightarrow i}^t)}{1 + e^{\mu[\theta_i(t)]^{1-\gamma_t}} \prod_{k \in \partial i \setminus j} (1 - \pi_{k \rightarrow i}^t)}, \quad (7)$$

with $\theta_i(t) = \prod_{k \in \partial i} (1 - \pi_{k \rightarrow i}^{t-1})$ and $\gamma_t = \gamma^{\lfloor t \, dt \rfloor}$.

The FP reached when reinforcement is present is a completely magnetized one, that is, the marginal probabilities for the values of n_i are such that $P[n_i = 1] \in \{0, 1\}$, and thus each variable is surely in the IS or surely outside of it. Thus the FP reached by BPR does correspond to an IS.

The attentive reader probably notices that the above-discussed algorithms do not include all possible Monte Carlo schemes: For example, (replicated) simulated annealing in temperature and simple Monte Carlo in chemical potentials are missing. For this reason, we briefly discuss our choice of the analyzed algorithms and explain how the present work is organized.

The first algorithms discussed are greedy algorithms. They are clearly suboptimal and have been run just to give an idea of the IS size, which is very easy to find in linear time. This information will also be useful to set the parameters of more refined algorithms as PT. The algorithmic thresholds for the GA, as for all the other analyzed algorithms, are reported in Table I.

We then analyze in Sec. III the algorithms at fixed density, βMC and βPT . In these algorithms, the size of the IS one is looking for is fixed to K and what is changed is the inverse temperature parameter β , which in turn varies the number of links within the set \mathcal{I} representing the putative IS. Naturally, in the limit $\beta \rightarrow \infty$, no more links inside \mathcal{I} are allowed and we obtain a true IS. We start the discussion about stochastic algorithms with the analysis of βMC because this is an adaptation of the IS problem of commonly used local search algorithms, e.g., WALKSAT or ASAT [28,29], which have been applied with success to problems like random K -SAT or

random graph coloring, the main difference being that βMC respects the detailed balance condition, while WALKSAT or ASAT do not. We then study the βPT algorithm because this is the most common way to improve Monte Carlo sampling methods in glassy systems. This algorithm seems to scale superlinearly, but still polynomially, with the problem size N as shown in detail in Sec. III.

Then in Sec. V we analyze stochastic algorithms that work directly at zero temperature, μSA , μRSA , and μPT , where links inside \mathcal{I} are not allowed and the tuning parameter is the chemical potential μ . We do not study the βSA because the extrapolation of the algorithmic threshold in that case is a long and difficult task [30]: One should find the threshold for any given μ and then extrapolate in the $\mu \rightarrow \infty$ limit. The extrapolation of the algorithmic threshold is instead direct for the μSA algorithm, and for this reason, we prefer to study this version of SA. We will see that the μPT has an algorithmic threshold similar to the βPT one, thus showing that the performances of PT are rather robust.

Finally, in Sec. VI we compare the results obtained via the stochastic algorithms with the outcome of BPR, which is a powerful message passing algorithm, widely used to solve problems defined on random graphs.

We will mainly analyze the problem at $d = 20$, where the transition is still near the continuous one, and $d = 100$, where the transition is distinctly 1RSB. In Table I, the values for ρ_d , ρ_c , and ρ_{\max} , together with the thresholds for the maximum density reached by the analyzed algorithms, are reported.

III. MAXIMUM DENSITY REACHED BY FIXED-DENSITY ALGORITHMS

In this section we look at the performances of the fixed-density algorithms, namely, βMC and βPT . For these algorithms, if we measure the running time in MCSs, a linear dependence on N is hidden in the single MCS (which takes a time proportional to N) and we can limit ourselves to measuring the number of MCSs needed to reach the wanted solution in order to understand the computational complexity of this class of algorithms. In Fig. 1 we show the number of MCSs needed by βMC and βPT to converge to an IS of a given density ρ . Also, the results for μPT are shown for comparison.

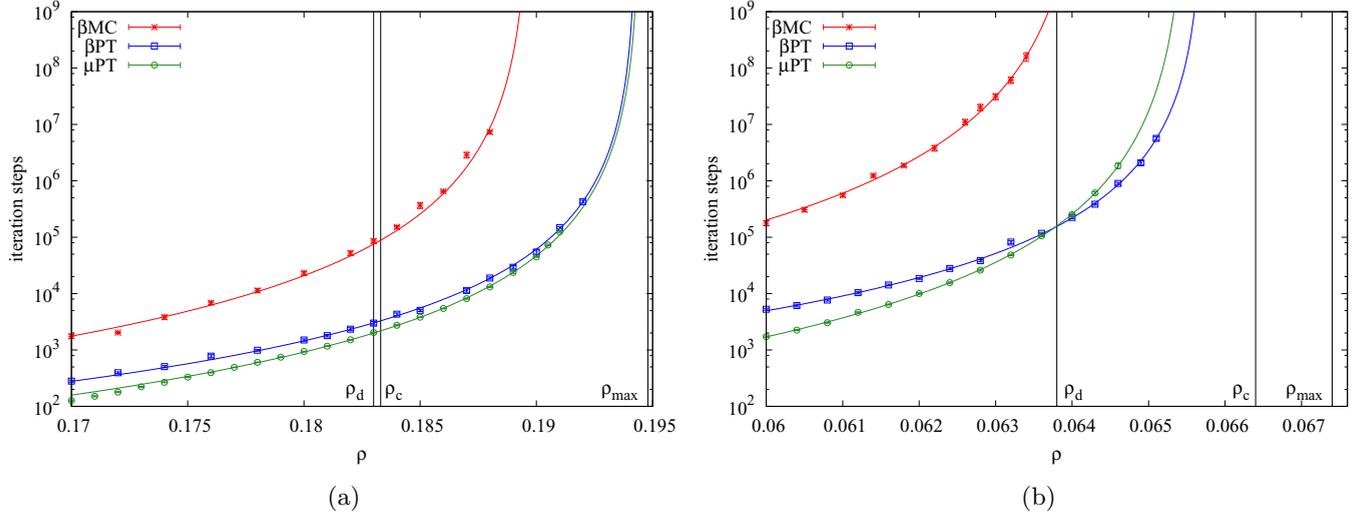


FIG. 1. Convergence time for β MC (with parameter $\beta = 11$), β PT (with parameters $\beta_{\max} = 11$, $\Delta\beta = 0.4$, and $N_\beta = 20$), and μ PT algorithm (with parameters $\mu_{\max} = 6$, $\Delta\mu = 0.2$, and $N_\mu = 20$) for $N = 5 \times 10^4$ and (a) $d = 20$ and (b) $d = 100$. The vertical lines show the theoretical thresholds for comparison.

For what concerns β MC, the optimal value of β maximizing the probability of reaching an IS, i.e., a zero-energy configuration, is likely to depend on N . Consequently, the convergence time will depend on N , since we expect the Monte Carlo dynamics to slow down when the temperature is decreased. Nevertheless, we are not going to make this detailed study because, as shown in Fig. 1, a standard Monte Carlo run at a single temperature is easily outperformed by parallel tempering.

The time to find an IS of a given density ρ is clearly diverging approaching the algorithmic threshold ρ_{alg} . In order to estimate the algorithmic threshold we need to perform an extrapolation. The best data interpolation is obtained via a power-law divergence

$$\tau = \frac{C}{(\rho_{\text{alg}} - \rho)^\nu}, \quad (8)$$

where C , ν , and ρ_{alg} are the fitting parameters (specific to each different algorithm). The best-fitting curves are shown with solid lines in Fig. 1. The extrapolated algorithmic thresholds are reported in Table I, while the best-fitting values for the ν exponent can be found in Table II. Data in Fig. 1 are for size $N = 5 \times 10^4$, which is large enough that finite-size effects are not present in the estimation of ρ_{alg} . The dependence of C and ν on the size will be discussed in Sec. III. We notice that both versions of PT (in temperature and chemical potential) have very similar algorithmic thresholds. This suggests that at that

TABLE II. Best-fitting values for ν . The divergence of the convergence time shown in Fig. 1 is fitted via the power law $\tau = C(\rho_{\text{alg}} - \rho)^{-\nu}$.

d	β MC	β PT	μ PT
20	4.2(2)	3.12(4)	3.34(7)
100	4.0(1)	4.2(2)	3.2(1)

density value there is some unavoidable hardness that affects both versions of PT. Our PT scheduling is not particularly optimized on purpose, because we believe that if an unavoidable algorithmic barrier arises at a certain density value, this should affect any version of Monte Carlo-based algorithms. The only parameter that we decide to fix in an (almost) optimal way is β_{\min} , i.e., the lowest value for the inverse temperature: Indeed, a too low β_{\min} requires a larger running time without any performances improvement (too many replicas at high temperature are useless), while a too large β_{\min} does not allow the configurations to decorrelate fast enough. We find that a very good choice for β_{\min} is the inverse temperature such that the actual density of the larger IS among the K variables with $n = 1$ is almost the maximum IS density reached by the best greedy algorithm. This means that the replica at β_{\min} can easily travel in the whole configurational space and this is enough for the PT algorithm to work properly.

Scaling with N for β PT

We have seen that the PT algorithm is able to find solutions in a region of ρ where other algorithms fail. The next important question to answer is how the number of PT iterations needs to be scaled with N in order to find an IS of density ρ . The issue is particularly relevant above ρ_d and approaching ρ_{alg} where the convergence time diverges. To analyze the scaling with N , we implement an optimized choice of the temperatures in the PT algorithm, whose derivation is in the Appendix. The optimized temperatures scheduling requires a number of replicas in a range $\beta \in [0, \beta_{\max}]$ that scales as \sqrt{N} . However, the replicas in the range $\beta \in [0, \beta_{\min}]$ are useless and can be safely ignored without altering PT performances. In practice, we end up with $N_\beta \sim 40$ in the worst case studied ($d = 100$, $N = 10^5$, and $\rho = 0.0646$).

To study the size dependence of the convergence time, we run all our β PT simulations with the temperature set defined in Eq. (A4) with $r = r_{\text{opt}}$, between β_{\min} and β_{\max} . In Fig. 2 we show for $d = 100$ the results in a wide range of densities

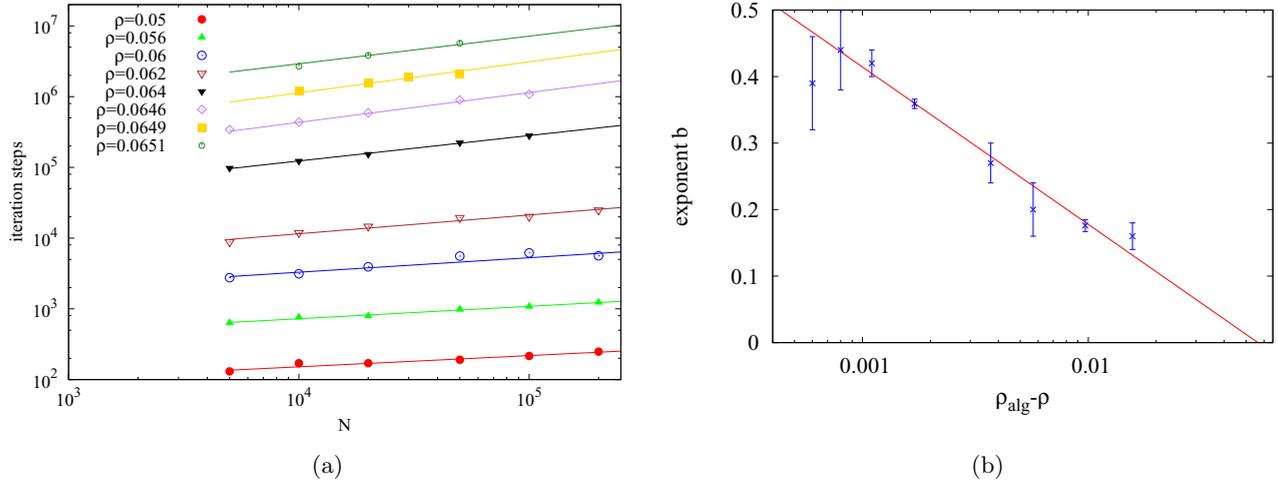


FIG. 2. (a) Number of MCSs to find a solution for $d = 100$ for different values of ρ as a function of the size N of the graph for the optimized β PT. Errors are smaller than points. The fits are of the kind $\tau(N) = aN^b$. (b) Dependence of the exponent b on the distance from the algorithmic threshold $\rho_{\text{alg}} - \rho$. The fit is of the type $b = c_1 + c_2 \log(\rho_{\text{alg}} - \rho)$. The right border of the plot corresponds to $\rho = 0$.

(similar behavior is observed for $d = 20$). The running times grow as a power law in N ,

$$\tau(N) = a(\rho)N^{b(\rho)}, \quad (9)$$

where the main ρ dependence is in the prefactor $a(\rho)$ that diverges at ρ_{alg} as in Eq. (8). However, there is also a slight dependence on ρ in the exponent b . We plot b as a function of ρ in Fig. 2(b), together with a fit of the type $b(\rho) = c_1 + c_2 \log(\rho_{\text{alg}} - \rho)$, which interpolates nicely the data. We notice that this behavior is the one to make Eqs. (8) and (9) compatible, since they are particular cases of the more general expression

$$\log[\tau(\rho, N)] = \log(c) - \nu' \log(\rho_{\text{alg}} - \rho) + c_1 \log(N) + c_2 \log(N) \log(\rho_{\text{alg}} - \rho). \quad (10)$$

For a fixed value of N we recover Eq. (8) with $C = cN^{c_1}$ and $\nu = \nu' - c_2 \log(N)$.

From the data shown in Fig. 2 it is evident that the exponent b is positive even in the “easy” region and it seems to go to zero only for $\rho \simeq 0$. This means that using PT to find IS always requires a running time growing more than linearly in N . We think this is due to the fact that PT is a sophisticated algorithm developed to find solutions when the energy landscape is complex. For $\rho < \rho_d$, when there is just a single state, PT is thus suboptimal. (Maybe with a different choice of parameters it could become a linear algorithm in this region; this kind of optimization is however beyond our scope: We introduced PT to reach solutions in the hard region.)

The time divergence as a power law approaching a given density, as in Eq. (8), is reminiscent of what happens in a first-order phase transition, thus suggesting that at ρ_{alg} an extensive barrier develops that makes it impossible to reach states with $\rho > \rho_{\text{alg}}$ in polynomial time. The weak dependence on N instead suggests that some long-range correlations may develop in the states in which the dynamics fall for $\rho < \rho_{\text{alg}}$ (this is discussed in the next section).

IV. FREEZING

As already mentioned, it has been conjectured for other optimization problems that the threshold for the appearance of hardness in polynomial time algorithms corresponds to the freezing threshold, which is the lowest density such that all clusters are frozen. We want to check this conjecture in the present problem.

For practical purposes let us define a cluster as the set of solutions (i.e., valid ISs) that are “connected” via paths where each step is the flip of just two variables. A cluster of solutions is frozen if it contains frozen variables, that is, if there is at least one variable fixed to a given value in all the configurations of the cluster. Above the rigidity threshold almost all the dominant clusters are frozen (but clusters with larger internal entropy might be not frozen). The freezing threshold corresponds to the density at which each cluster of solutions is frozen.

In this section we study the escape time t_{esc} , which is the time needed by an algorithm that moves only between solutions to go away from the initial configuration. More precisely, we first find a solution with a given algorithm, then we apply the β MC algorithm at $\beta = \infty$ (that is a kind of diffusive dynamics at fixed zero energy and fixed size of the IS), and we measure the time needed to “free” each variable from its starting value, that is, to find that variable in a value different from the starting one. Looking at Fig. 3, the first important observation is that all the analyzed algorithms show the same t_{esc} at a fixed density of the IS. This means that they all find the same kind of solutions (when they can find one).

The escape time diverges as a power law at a threshold density ρ_r (see the fits in Fig. 3). From the data we estimate $\rho_r(d = 20) = 0.1890(6)$ and $\rho_r(d = 100) = 0.0639(2)$. The observation that the same threshold holds for different kind of algorithms suggests the conjecture that ρ_r does actually correspond to the rigidity threshold, which is the density where the typical clusters become frozen and the escape time from it thus diverges.

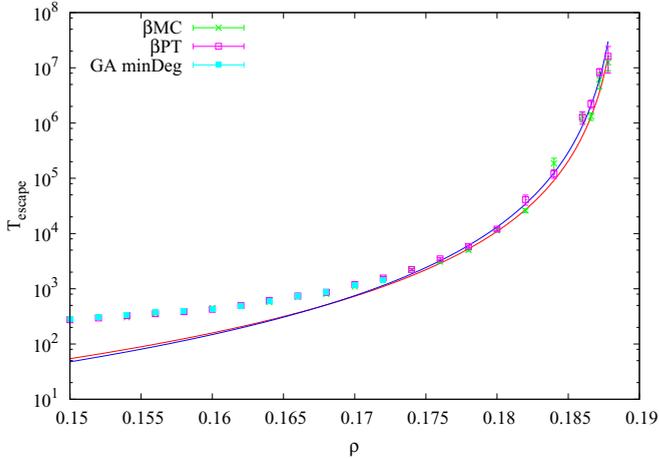
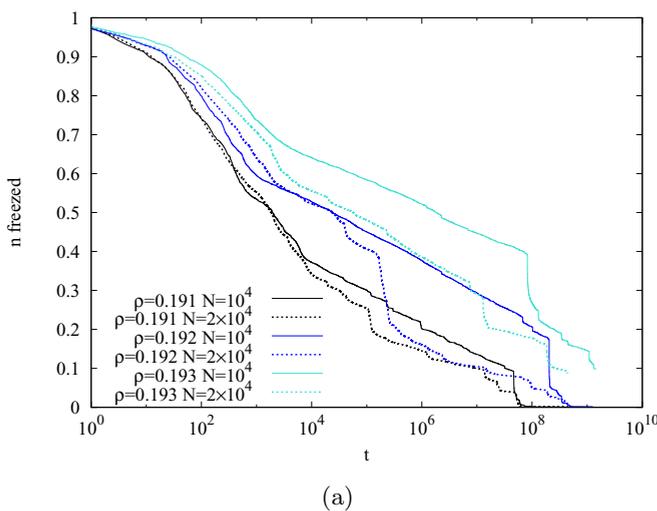


FIG. 3. Escape time from the solution reached by different algorithms ($d = 20$ and $N = 5 \times 10^4$).

The values of ρ_r are compatible with the thresholds for the β MC algorithm, while β PT and μ PT can find solutions of densities greater than ρ_r . At this point, it is natural to check whether the solutions found by the PT algorithms at densities larger than ρ_r are frozen or not. To answer this question we find a solution at density $\rho > \rho_r$ with the β PT algorithm, then we run the β MC algorithm at $\beta = \infty$ (the diffusive algorithm), and we look at the persistence, that is, the fraction of variables that have not changed during the diffusive dynamics.

The results are shown in Fig. 4 for a single sample: The fraction of frozen variables seems to decrease in an extremely slow way, mostly logarithmically in time with evident jumps (corresponding to avalanches of variables that are set free altogether). It is worth noticing that the slowness of the diffusive dynamics around the initial solution found by PT is only due to entropic effects, given that the diffusive dynamics keeps the energy constant.



In Fig. 4 we also notice some interesting finite-size effects. For the largest sizes, the diffusive dynamics eventually makes every variable unfrozen, although the escape time is several orders of magnitude larger than the time needed by PT to reach that particular solution (suggesting that PT follows a smart path that is not affected by entropic barriers). For smaller sizes, frozen variables persist longer and eventually we observe that the diffusive dynamics is not able to leave the cluster: The fraction of frozen variables becomes constant in time. This is strong evidence that the ISs found by PT for small enough N belong to frozen clusters (a similar phenomenon has been observed also in other models when solved, for example, via the reinforcement algorithm [31]).

The above observations support the following scenario: The PT algorithm is able to find ISs beyond the rigidity threshold ρ_r and in this rigid phase, for small enough sizes, there is a nonzero probability that PT finds a solution in a rare frozen cluster. However, for large N , the solutions found by PT seem to be all unfrozen and thus we deduce that the PT algorithmic threshold is bounded above by the freezing threshold. We are strongly tempted to conjecture that the two thresholds, ρ_{alg} for PT and ρ_f , do actually coincide, but we do not have firm arguments in support.

We also check that the solutions found by the PT algorithms above ρ_d are not equilibrium solutions. To do this, we find a solution at $\rho > \rho_d$ with the β PT or μ PT algorithms. We then initialize BP on that solution and we check whether BP converges to a fixed point close to the solution found by PT. If it is so, this means that the PT solution lies inside one of the states (and replica symmetry holds within a state) that form the 1RSB structure that characterizes the equilibrium measure for densities slightly above ρ_d . However, we find that BP does not converge (either to the paramagnetic fixed point or to a fixed point close to the PT solution). This lack of convergence suggests that the solution found by PT is probably inside a state that is not replica symmetric, but probably FRSB, as found in other models [18]. Indeed, it is well known that states

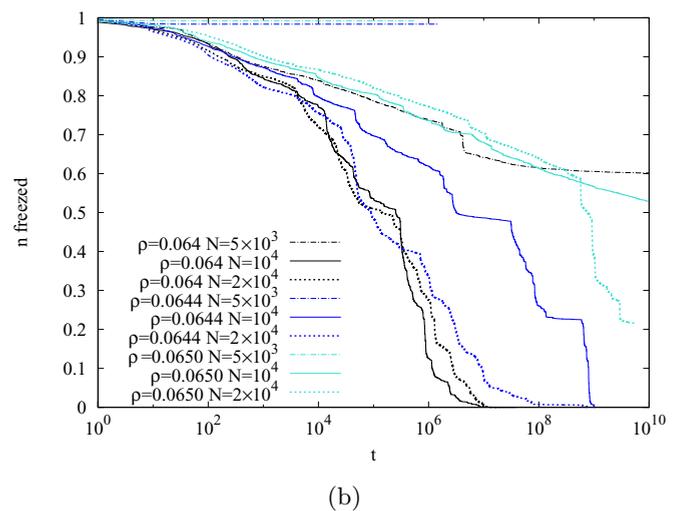


FIG. 4. Starting from an IS of density ρ found via the β PT algorithm, we measure the fraction of variables that have not changed their value during a pure diffusive dynamics (β MC algorithm with $\beta = \infty$). Results are for (a) $d = 20$ and (b) $d = 100$ and a single sample of the size indicated in the legend.

reached by the out-of-equilibrium dynamics may be FRSB even when equilibrium states are IRSB [32,33].

V. ZERO-TEMPERATURE ALGORITHMS

In this section we analyze a different kind of algorithms, the ones running directly at zero temperature. This means that links inside \mathcal{I} are not allowed, i.e., the algorithm always works with a valid IS. For this class of algorithms, the varying parameter is the chemical potential μ that changes the average density of the IS. The limit $\mu \rightarrow \infty$ should correspond to the largest possible IS.

First of all, we run μ SA. It is a common belief that a slow enough SA should reach the bottom of the equilibrium states at ρ_d . The algorithmic thresholds, computed as the average over 100 samples of the maximum density ρ reached when $\mu \rightarrow \infty$ in a SA with $\Delta\mu = 10^{-7}$ and $N = 5 \times 10^4$, are reported in Table I. As one can notice, for $d = 20$ the inequalities $\rho_{\text{alg}} > \rho_c > \rho_d$ hold, implying that the states that dominate the measure at ρ_d can be followed way beyond ρ_c . This is compatible with the fact that at $d = 20$ the transition is still nearly a continuous FRSB one and thus the ergodicity breaking is less pronounced. For $d = 100$ instead, $\rho_{\text{alg}} < \rho_c$, consistently with the fact that the transition is distinctly IRSB and ergodicity breaking takes place in a much more marked way.

We now analyze the μ RSA algorithm. We take inspiration from Ref. [19], where a replicated version of the SA was proposed to sample with higher probability states with larger entropy. In the context of ISs, one can identify a state in the following way: Starting from a maximal IS, that is, an IS that cannot be increased any further by just adding vertices to the IS itself, and considering this maximal IS as the “bottom of a valley” in a usual energy landscape, one can build a state by the set of ISs which are a subset of the maximal one (the construction has to be refined when one finds ISs which are a subset of more than one maximal IS, but we do not need such a detailed description for the present argument). According to this construction, it is likely that states corresponding to the largest IS are also those of largest entropy. So the use of an algorithm that favors states according to their entropy is likely to be beneficial also in the search for the largest IS.

We run μ RSA with parameters $R = 3$ and $\gamma = 1$. In Fig. 5 its performances are compared with those of μ SA in the case $d = 100$ (their algorithmic thresholds can be found in Table I). It is remarkable that the improvement of RSA with respect to SA is practically null for $d = 20$ and very tiny for $d = 100$. While for $d = 20$ one may claim that the improvement is absent because the model has a very weakly discontinuous phase transition (the range where the phase transition is continuous is very close by), for $d = 100$ the IRSB scenario clearly holds, but we do not see any improvement by reweighting states according to their internal entropy. This observation

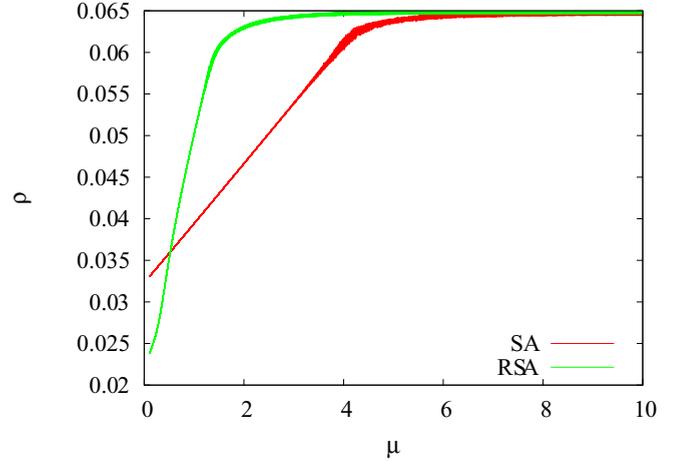


FIG. 5. Comparison between μ SA and μ RSA for 50 samples of size $N = 5 \times 10^4$ and $d = 100$ (the parameters are $\Delta\mu = 10^{-7}$, $R = 3$, and $\gamma = 1$).

raises some doubts about what RSA is actually doing and why it is not working as expected. Moreover, given that the performances of RSA are clearly worse than those of PT (see their algorithmic thresholds in Table I), we arrive at the conclusion that there are both more efficient and less efficient ways to couple replicas.

We move now to the analysis of the μ PT algorithm. We use $N_\mu = 21$ replicas evenly spaced by $\Delta\mu = 0.2$ in the range $\mu \in [2, 6]$ for $d = 20$ and $N_\mu = 31$ replicas evenly spaced by $\Delta\mu = 0.15$ in the range $\mu \in [2, 6.5]$ for $d = 100$.

We have already anticipated in Sec. III that the behavior of μ PT is equivalent to that of β PT and in particular the algorithmic thresholds of the two algorithms are compatible. In Fig. 6 we show that also the scaling of their running times with N is similar, as the time needed to reach a solution of a given density scales as $\tau = aN^b$. In Table III we make the comparison between the exponent b of the two algorithms at the same values of d and ρ . These data confirm that the PT algorithm is a very robust one.

VI. COMPARISON WITH ADVANCED MESSAGE PASSING ALGORITHMS

We have seen that Monte Carlo-based algorithms easily outperform greedy algorithms and can reach densities well above the dynamical threshold ρ_d , passing also the rigidity threshold ρ_r and for $d = 20$ even beyond the condensation threshold ρ_c , thus approaching closely the maximum density ρ_{max} . This looks like a great result, but in order to put it under the right light, we need a comparison with some other algorithm that is expected to work efficiently on this kind

TABLE III. Comparison between values of b for β PT and μ PT. The convergence time in Fig. 6 diverges as $\tau(N) = aN^b$.

Algorithm	$d = 20, \rho = 0.190$	$d = 20, \rho = 0.192$	$d = 100, \rho = 0.064$	$d = 100, \rho = 0.0646$
β PT	0.339(8)	0.69(7)	0.357(7)	0.42(2)
μ PT	0.40(1)	0.68(1)	0.336(8)	0.44(3)

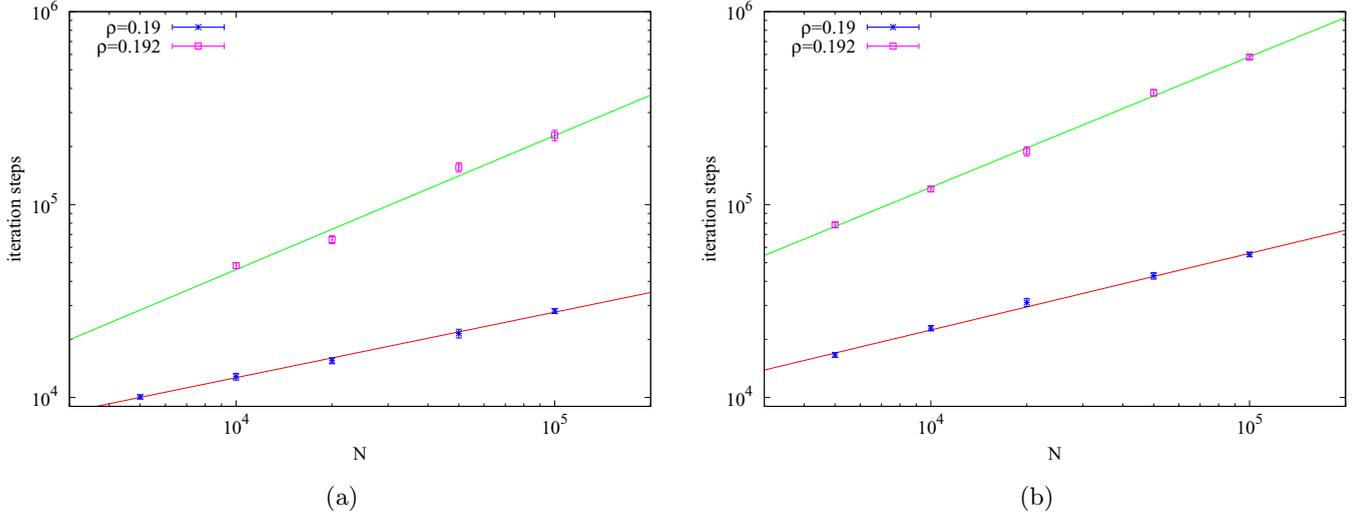


FIG. 6. Number of iterations to find a solution for $d = 20$ and $\rho = 0.19, 0.192$ as a function of the problem size N for the optimized (a) β PT and (b) μ PT algorithms. The behavior of the two algorithms is very similar.

of optimization problem. Since the problem is defined on a random graph, we expect message passing algorithms to be particularly well suited. For this reason, we have run also BPR on this problem.

In Fig. 7 we show the average density of IS found by the BPR algorithm, as a function of the chemical potential μ , for different values of the BPR parameters. Let us just mention that below a certain chemical potential μ_L , the solutions found by the BPR algorithm are always $n_i = 0 \forall i$. The value of μ_L is the one that generates, using the RS solution of the model from [10], a density ρ_L that roughly corresponds to the threshold density for the random vertex GA [for both $d = 20$ and $d = 100$ we have $\mu_L = 2.15(5)$].

We have run the BPR algorithm in a broad range of chemical potentials and for different choices of the BPR parameters. The best results have been obtained with the choice $\gamma = 0.999$ and $dt = 10$. The maximum density reached can be deduced from the data shown in Fig. 7 and it is clearly lower than

the thresholds for the PT algorithms. Our best estimates are reported in Table I. We notice that the threshold density for the BPR algorithm is very similar to the one of the RSA algorithm and this is expected from Ref. [19].

VII. CONCLUSION

We have done a comparative study of algorithms to find the largest IS in a RRG of degree $d = 20$ and $d = 100$. Our aim was to understand the actual performances of different kinds of algorithms (greedy, message passing, and especially Monte Carlo based) and to connect their algorithmic thresholds with thermodynamical phase transitions. For both values of d the set of ISs undergoes a RFOT varying the IS density ρ ; however, for $d = 20$ the transition is weakly discontinuous because of the proximity to the range where the transition is continuous ($d < 16$); for $d = 100$ the transition is markedly discontinuous as in the large-degree limit.

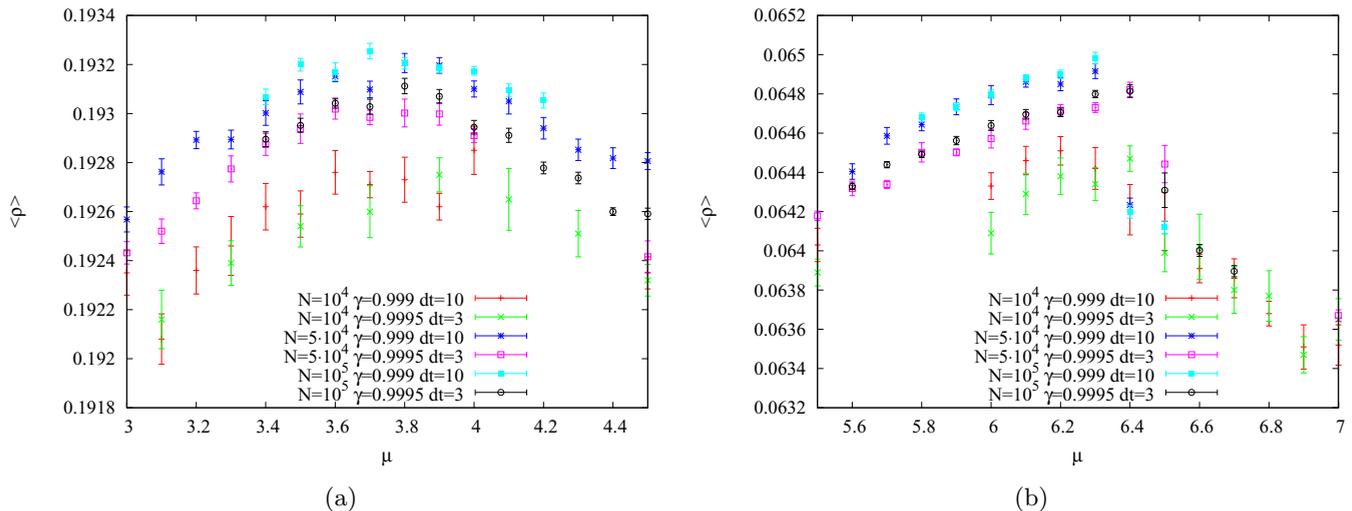


FIG. 7. Average density of the ISs found by BP plus reinforcement as a function of the chemical potential μ at different values of N and of the BPR parameters for (a) $d = 20$ and (b) $d = 100$.

While Table I summarizes thermodynamical and algorithmic thresholds, we list below the most relevant conclusions that we achieved.

(i) Only greedy algorithms get stuck below the dynamical threshold, while all the other algorithms easily pass beyond ρ_d ; the relevance of the dynamical threshold for smart optimization algorithms seems very limited.

(ii) The condensation threshold at ρ_c seems to play no role at all in describing the performances of the best optimization algorithms.

(iii) The simplest version of Monte Carlo algorithms seems to work roughly until the rigidity threshold at ρ_r , defined as the density where the time to diffuse away from a typical IS diverges.

(iv) More sophisticated Monte Carlo schemes (SA and PT) find ISs beyond ρ_r , but without frozen variables, thus showing the ability of finding ISs in atypical unfrozen states.

(v) Replicated SA does not show any sensible improvement over standard SA for this problem, especially for $d = 20$.

(vi) Belief propagation with reinforcement has an algorithmic threshold similar to replicated SA.

(vii) Parallel tempering is by far the best algorithm for solving this problem and can find ISs of a very large density that no other algorithm can find.

(viii) Different versions of PT (in temperature and chemical potential) show almost the same algorithmic threshold, and this strongly suggests a universal behavior linked to an underlying phase transition. We conjecture the PT algorithmic threshold to coincide with the freezing threshold, i.e., PT, is able to find an unfrozen IS as long as there is one.

(ix) Running times of PT are superlinear, but still polynomial in N . Algorithmic thresholds for superlinear algorithms are likely to be larger than those for linear algorithms, but a theory for the former is completely lacking.

Our results clearly show the need for a theory for advanced Monte Carlo algorithms, like parallel tempering, which is at present lacking. Only by understanding analytically this class of algorithms can we hope to approach the ultimate algorithmic threshold for a broad class of hard optimization problems.

ACKNOWLEDGMENTS

We thank Scott Kirkpatrick, Raffaele Marino, and Andrea Montanari for very interesting discussions. This research was supported by the European Research Council within the European Unions Horizon 2020 Research and Innovation Programme (Grant No. 694925, Project LoTGLasSy).

APPENDIX: OPTIMIZING THE CHOICE OF TEMPERATURES IN THE PT ALGORITHMS

Here we explain how we have implemented an optimized choice for the temperatures in the parallel tempering algorithm. We assume that the energy is close to its equilibrium value that can be computed via the RS solution. For the β MC algorithm at a fixed density ρ , the RS mean energy can be computed by noting that the RS marginals are equal for each site and assuming the value $p_{RS}(\sigma) = \rho \delta_{\sigma,1} + (1 - \rho) \delta_{\sigma,0}$,

thus getting

$$e(\beta) = \frac{d}{2} \frac{\rho^2 e^{-\beta}}{\rho^2 e^{-\beta} + (1 - \rho^2)}. \quad (\text{A1})$$

In the large- N limit we can assume that the extensive energy at inverse temperature β is a Gaussian variables with mean $E(\beta) = Ne(\beta)$ and variance $\sigma^2(\beta) = -Ne'(\beta)$. This Gaussianity assumption (which is rather well satisfied, but in the vicinity of the ground state) allows us to compute the probability of swapping two replicas at inverse temperatures β_1 and β_2 ,

$$p_{\text{swap}}(\beta_1, \beta_2) = \int dz_1 dz_2 \frac{e^{-z_1^2/2 - z_2^2/2}}{2\pi} \times \min(1, e^{(\beta_2 - \beta_1)[E(\beta_2) + \sigma(\beta_2)z_2 - E(\beta_1) - \sigma(\beta_1)z_1]}). \quad (\text{A2})$$

In the limit $\Delta\beta = \beta_2 - \beta_1 \ll 1$ we can approximate $E(\beta_2) - E(\beta_1) \simeq Ne'(\beta)\Delta\beta$ with $\beta = (\beta_1 + \beta_2)/2$ and $\sigma(\beta_1) \simeq \sigma(\beta_2) \simeq \sigma(\beta) = \sqrt{-Ne'(\beta)}$, thus getting

$$p_{\text{swap}}(\beta, \Delta\beta) = \int dz_1 dz_2 \frac{e^{-z_1^2/2 - z_2^2/2}}{2\pi} \times \min(1, e^{\Delta\beta[Ne'(\beta)\Delta\beta + \sqrt{-Ne'(\beta)}(z_2 - z_1)])} = \text{erfc}\left(\frac{\Delta\beta\sqrt{-Ne'(\beta)}}{2}\right). \quad (\text{A3})$$

The best way to allow replicas to wander fast between temperatures is to fix a constant p_{swap} between any pair of successive temperatures and this can be achieved with the choice

$$\beta_{n+1} = \beta_n + \frac{r}{\sqrt{N|e'(\beta_n)|}}, \quad (\text{A4})$$

implying $p_{\text{swap}} = \text{erfc}(r/2)$. The optimal value for r can be obtained by maximizing the mean squared distance traveled by a random walker performing jumps of size r with probability $\text{erfc}(r/2)$, that is,

$$r_{\text{opt}} = \text{argmax}\left[\text{erfc}\left(\frac{r}{2}\right)r^2\right] \simeq 1.68376, \quad (\text{A5})$$

leading to an optimal swapping rate equal to $\text{erfc}(r_{\text{opt}}/2) \simeq 0.23381$ (this is the well-known 0.23 rule [34]).

With the set of temperatures defined in Eq. (A4), the optimized PT would require $O(\sqrt{N})$ replicas. However, we empirically observe that the time of convergence of the algorithm does not change if replicas in the range $\beta \in [0, \beta_{\min}]$ are removed. We find empirically that the largest possible value for β_{\min} roughly corresponds to the inverse temperature at which the equilibrium magnetization coincides with the maximum IS density reached by the greedy algorithm ρ_{GA} . This is very reasonable; indeed, for $\rho < \rho_{\text{GA}}$ we do not expect any relevant barrier to be present and so the PT replicas at β_{\min} can easily travel the entire configurational space.

For β_{\max} we choose the lowest inverse temperature at which the condition $E(\beta) - \sqrt{N|e'(\beta)|} < 0$ is satisfied, implying that a typical spontaneous fluctuation can lead the algorithm to find a configuration of zero energy.

We observe that the optimized version of β PT finds solutions up to $\rho_{\max}(d = 20) = 0.1941(5)$, compatible with the nonoptimized version, but with a smaller exponent $b = 2.4(3)$. For $d = 100$ the optimized β PT algorithm reaches $\rho_{\max}(d = 100) = 0.06572(9)$ with $b = 3.2(1)$.

Things are different for the μ PT algorithm. For this algorithm, the RS magnetization is the one written in Eq. (6) of Ref. [10]. However, in the hard region, the real magnetization is quite different and so we cannot use the RS result to optimize the μ PT algorithm.

-
- [1] B. Bollobás, in *Modern Graph Theory* (Springer, Berlin, 1998), pp. 215–252.
- [2] A. K. Hartmann and M. Weigt, *Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics* (Wiley, New York, 2006).
- [3] B. Bollobás and P. Erdős, *Math. Proc. Cambridge* **80**, 419 (1976).
- [4] A. M. Frieze, *Discrete Math.* **81**, 171 (1990).
- [5] A. Coja-Oghlan and C. Efthymiou, *Random Struct. Algor.* **47**, 436 (2015).
- [6] G. R. Grimmett and C. J. McDiarmid, *Math. Proc. Cambridge* **77**, 313 (1975).
- [7] D. Gamarnik and M. Sudan, *Proceedings of the Fifth Conference on Innovations in Theoretical Computer Science, Princeton, 2014* (ACM, New York, 2014), pp. 369–376.
- [8] M. Mézard, T. Mora, and R. Zecchina, *Phys. Rev. Lett.* **94**, 197205 (2005).
- [9] D. Achlioptas, A. Coja-Oghlan, and F. Ricci-Tersenghi, *Random Struct. Algor.* **38**, 251 (2011).
- [10] J. Barbier, F. Krzakala, L. Zdeborová, and P. Zhang, *J. Phys.: Conf. Ser.* **473**, 012021 (2013).
- [11] D. Achlioptas and F. Ricci-Tersenghi, in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, 2006* (ACM, New York, 2006), pp. 130–139.
- [12] L. Zdeborová and F. Krzakala, *Phys. Rev. E* **76**, 031131 (2007).
- [13] R. Marino, G. Parisi, and F. Ricci-Tersenghi, *Nat. Commun.* **7**, 12996 (2016).
- [14] A. Braunstein, L. Dall’Asta, G. Semerjian, and L. Zdeborová, *J. Stat. Mech.* (2016) 053401.
- [15] T. A. Feo, M. G. Resende, and S. H. Smith, *Oper. Res.* **42**, 860 (1994).
- [16] T. A. Feo and M. G. Resende, *J. Global Optim.* **6**, 109 (1995).
- [17] M. M. Halldórsson and J. Radhakrishnan, *Algorithmica* **18**, 145 (1997).
- [18] L. Zdeborová and F. Krzakala, *Phys. Rev. B* **81**, 224205 (2010).
- [19] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, *Proc. Natl. Acad. Sci. USA* **113**, E7655 (2016).
- [20] K. Hukushima and K. Nemoto, *J. Phys. Soc. Jpn.* **65**, 1604 (1996).
- [21] D. J. Earl and M. W. Deem, *Phys. Chem. Chem. Phys.* **7**, 3910 (2005).
- [22] J. Moreno, H. G. Katzgraber, and A. K. Hartmann, *Int. J. Mod. Phys. C* **14**, 285 (2003).
- [23] M. C. Angelini, *J. Stat. Mech.* (2018) 073404.
- [24] L. Dall’Asta, A. Ramezanpour, and R. Zecchina, *Phys. Rev. E* **77**, 031118 (2008).
- [25] R. Karp and M. Sipser, *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science, Nashville, 1981* (IEEE, New York, 1981), pp. 364–375.
- [26] N. C. Wormald, *Ann. Appl. Probab.* **5**, 1217 (1995).
- [27] A. Braunstein and R. Zecchina, *Phys. Rev. Lett.* **96**, 030201 (2006).
- [28] E. Aurell, U. Gordon, and S. Kirkpatrick, in *Proceedings of the 17th Conference on Advances in Neural Information Processing Systems, Vancouver, 2017*, edited by L. K. Saul, Y. Weiss, and L. Bottou (MIT Press, Cambridge, 2004), p. 804.
- [29] J. Ardelius and E. Aurell, *Phys. Rev. E* **74**, 037702 (2006).
- [30] L. Budzynski, F. Ricci-Tersenghi, and G. Semerjian, *J. Stat. Mech.* (2019) 023302.
- [31] L. Zdeborová, *Acta Phys. Slovaca* **59**, 169 (2009).
- [32] A. Montanari and F. Ricci-Tersenghi, *Eur. Phys. J. B* **33**, 339 (2003).
- [33] A. Montanari and F. Ricci-Tersenghi, *Phys. Rev. B* **70**, 134406 (2004).
- [34] A. Pelissetto and F. Ricci-Tersenghi, in *Large Deviations in Physics: The Legacy of the Law of Large Numbers*, edited by A. Vulpiani, F. Cecconi, M. Cencini, A. Puglisi, and D. Vergni, Lecture Notes in Physics Vol. 885 (Springer, Berlin, 2014), pp. 161–191.