

Algorithmic Thresholds in Combinatorial Optimization Depend on the Time Scaling

M. C. Angelini^{1,2}, M. Avila-González³, F. D'Amico^{1,4}, D. Machado^{1,3,4}, R. Mulet⁵, and F. Ricci-Tersenghi^{1,2,4}

¹*Dipartimento di Fisica, Sapienza Università di Roma, Piazzale le Aldo Moro 5, 00185 Rome, Italy*

²*Istituto Nazionale di Fisica Nucleare, Sezione di Roma I, Piazzale le Aldo Moro 5, 00185 Rome, Italy*

³*Department of Theoretical Physics, Physics Faculty, University of Havana, 10400, La Habana, Cuba*

⁴*CNR-Nanotec, unità di Roma, Piazzale le Aldo Moro 5, 00185 Rome, Italy*

⁵*Group of Complex Systems and Statistical Physics and Department of Theoretical Physics, Physics Faculty, University of Havana, 10400, La Habana, Cuba*



(Received 26 May 2025; accepted 16 December 2025; published 3 March 2026)

In the past decades, many efforts have focused on analyzing typical-case hardness in optimization and inference problems. Some recent work has pointed out that polynomial algorithms exist, running with a time that grows more than linearly with the system size, which can do better than linear algorithms, finding solutions to random problems in a wider range of parameters. However, a theory for polynomial and superlinear algorithms is in general lacking. Here, we examine the performance of the simulated annealing algorithm, a standard, versatile, and robust choice for solving optimization and inference problems, in the prototypical random K -SAT problem. For the first time, we show that the algorithmic thresholds depend on the time scaling of the algorithm with the size of the system. Indeed, one can identify not just one but different thresholds for linear, quadratic, and cubic regimes (and so on). This observation opens new directions in studying the typical case hardness in optimization problems.

DOI: [10.1103/dw9m-95vv](https://doi.org/10.1103/dw9m-95vv)

Subject Areas: Complex Systems,
Statistical Physics

I. INTRODUCTION

Computing the performance of algorithms solving optimization and inference problems is crucial for both fundamental research and real-world applications. In particular, it is primarily important to identify the so-called *algorithmic threshold*, that is, the limit of the region where a specific algorithm can work efficiently to find a solution to the problem.

In many relevant problems, the best known algorithmic threshold is far from the *statistical threshold*, which can be obtained by computing the limit of the existence of solutions. In other words, there exists a region (often very broad) where solutions are likely to be present, but no algorithm can find them efficiently. This region is called the *hard phase* and it is at the origin of the computational-to-statistical gap that plagues many optimization and inference problems. To close this gap, or at least reduce it, one should move forward the algorithmic threshold.

The vast majority of algorithms whose performances can be analyzed are linear algorithms, that is, algorithms whose running time scales linearly with the problem size. This is a

major limitation in several aspects. First of all, superlinear but still polynomial algorithms may show better performances and may have improved thresholds. Not every algorithm can be scaled up to a superlinear regime, but even the scalable ones have not been studied so far. The present work represents the first systematic study of the superlinear regime in a general class of algorithms. We expect that it may open a new approach to the determination of algorithmic thresholds.

Second, the algorithmic threshold for linear algorithms is often computed via extrapolations from the linear regime as the point where the time to reach a solution diverges (see, e.g., insets in Figs. 1 and 5 of Ref. [1]). These estimates not only require taking the large size and large time limits, but are unavoidably quite noisy due to the extrapolation procedure. Here we show how to evaluate algorithmic thresholds in a much better way by going to the superlinear regime (in particular, the quadratic regime) and exploiting well-known techniques from the study of critical phenomena. We consider that this new approach can become the standard for the determination of algorithmic thresholds.

To achieve the goals stated above, we focus our study on an algorithm and a problem that are general, representative, and with nontrivial properties. We analyze the performances of algorithms based on Monte Carlo Markov chains (MC in short), in particular, simulated annealing (SA) [2] and the zero-temperature (greedy) Monte Carlo algorithm, in the well-known K -satisfiability (K -SAT)

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

problem. Both have widespread applications in many diverse fields of science and technology.

Despite the robustness and flexibility of MC, properties that make them reference algorithms in many situations (especially when real-life instead of random problems are considered), there are no known works that precisely identify the thresholds reached by these algorithms on the prototypical random K -SAT optimization problem.

In the present work, for the first time, we identify different algorithmic thresholds for MC algorithms, depending on the scaling of the number of Monte Carlo sweeps (MCSs) with the problem size N . Since each MCS is defined as the attempt to change every variable in the configuration once, it takes N iterations of the algorithm to complete one MCS. Therefore, a fixed number of MCSs corresponds to an algorithm linear in N . Leaving the number of MCSs to grow with N , we get quadratic, cubic, and other superlinear algorithms. Apart from estimating a different algorithmic threshold for every scaling regime, our analysis combining finite-size scaling and simulations in the large size limit allows us to identify the “ultimate” algorithmic threshold and the optimal scaling between times and size to reach it.

Interestingly enough, our results provide an explanation for earlier empirical observation that the running time must be scaled cubically with the problem size to achieve the best performance of specific heuristics in the three-coloring problem [3]. In recent literature, there have been few works showing that polynomial algorithms, with running time that scales at least quadratically with the system size, could beat very efficient linear-time algorithms in different optimization and inference problems [4–7].

MC algorithms are ideal for running such an experiment, given that one can decide the total number of MCSs before running the algorithm, and such a number can be varied when the system size increases.

We expect the behavior we uncover in this work to be possibly shared by other classes of algorithms. For example, neural networks can be trained and tested for a number of epochs that could be changed accordingly with the size of the problem. Our work thus opens new perspectives on the study of typical case hardness: How could we explain theoretically the presence of different thresholds related to different polynomial time scaling?

II. K -SAT AND TYPICAL CASE HARDNESS

The K -SAT is a pillar in the theory of algorithmic complexity. It was the first to be classified as NP complete, actually, in the same article where Cook introduced the very concept of NP completeness [8]. The task in K -SAT is to find an assignment to N boolean variables $\vec{x} = \{x_1, \dots, x_N\}$ that satisfies a set of M clauses that involve K variables each. In the random K -SAT problem, the K variables in each clause are chosen at random, and thus the hardness of the formula is controlled mainly by the parameter

$\alpha = M/N$. This problem undergoes a SAT-UNSAT transition at a satisfiability threshold α_s [9], such that for $\alpha < \alpha_s$ solutions exist almost surely in the infinite-size limit ($N \rightarrow \infty$) while for $\alpha > \alpha_s$ in the same limit there are no solutions with high probability: This transition has been proved to be a sharp one when $N \rightarrow \infty$ [10,11].

One could associate with each configuration of the N variables a cost function that counts the number of violated clauses. In this way, the problem of finding a satisfying assignment is recast as an optimization problem, for which one wants to identify a configuration of zero cost. The K -SAT problem is thus seen as the paradigm of combinatorial optimization problems. For this reason, we will concentrate our attention on this problem, keeping in mind that what is discussed in the following is shared by many other random optimization problems, as we discuss at the end of the paper, presenting results for a different random optimization problem.

While the “worst-case” NP completeness of K -SAT problems was stated many years ago [8], in recent years many efforts have been focused on studying typical-case hardness [12–17]. One would like to identify the best algorithm that can find optimal configurations (i.e., solutions) of the problem within the shortest time in typical instances of the problem, for which we know that solutions exist up to α_s in the infinite-size limit. However, in practice, search algorithms are not able to find solutions, although they do exist, in a region $\alpha_{\text{alg}} < \alpha < \alpha_s$ [18]. The precise determination of the threshold α_{alg} in typical instances, which could be specific to every algorithm, has been the object of an intense effort in the past decades [1,19–23], because for practical applications, the algorithmic threshold α_{alg} matters much more than the SAT-UNSAT transition at α_s .

Exploiting the so-called one-step replica symmetry breaking cavity method formalism, theoretical physicists have characterized different phase transitions in the geometrical structure of the space of solutions [14,16,24]. This structure is certainly related to the algorithmic complexity of the problem, but a clear connection is still missing. In the following, we summarize the phase transitions taking place in random K -SAT and several conjectures relating phase transitions to algorithm thresholds.

Increasing α , the first phase transition the model undergoes is the *dynamical transition* at α_d . Above α_d , the uniform equilibrium measure over the solutions breaks into an exponential number of disconnected clusters [13,15,16]. The *condensation transition* takes place at $\alpha_c \geq \alpha_d$ and defines the threshold above which the equilibrium measure is dominated by a subexponential number of clusters [16]. Given that above α_d the ergodicity is spontaneously broken (because of the existence of many clusters separated by barriers), it was initially conjectured α_d to be an upper bound for the local search algorithms [9]. Later, it was found that the belief propagation algorithm returns exact

marginals until α_c , and consequently, α_c was conjectured to be the algorithmic threshold [16]. This conjecture was further improved in Ref. [25].

However, there are several observations of local search algorithms overcoming both bounds α_d and α_c [1,19,22] solving almost all typical instances in a time that scales polynomially with the problem size for α very close to the satisfiability threshold α_s in some optimization problems (included the K -SAT problem for small enough K). An intuitive solution to this contradiction [26] rests on the idea that some algorithms violate the detailed balance condition and are completely out of equilibrium. Therefore, they do not necessarily fit within the hypothesis of the equilibrium computations that lead to the theoretical thresholds α_d and α_c . Even speaking about algorithms satisfying the detailed balance condition, like MC algorithms, they are often not used in the regime of very large times, but on a finite timescale: In such a regime, the algorithm could be visiting configurations with a measure different from the equilibrium one [27]. This includes both algorithms that we will study here: SA and the zero-temperature Monte Carlo algorithm. Their performances are thus not well described by equilibrium computations.

Forgetting the equilibrium measure, one could “follow” the states out of equilibrium [28]. By doing so, even above the clustering threshold, there could exist a phase displaying the so-called “canyon-dominated landscape” where out-of-equilibrium states go down to zero energy with a large basin of attraction. The region of the parameters where these canyon states do not exist anymore could correspond to the hard phase for algorithms. This hypothesis was checked in p -spin glass models for the simulated annealing algorithm [29]. The canyon-dominated threshold is upper bounded by the so-called *freezing transition*, above which any cluster of solutions contains an extensive number of variables that are frozen, that is, taking the same value in all the solutions belonging to the cluster [15,30]. Finding solutions in frozen clusters is very hard, as no error can be made for any frozen variable. Since the number of frozen variables is extensive, the probability of making no error fixing them goes to zero exponentially fast in the problem size. For this reason, the freezing transition has been conjectured to be an upper bound to the algorithmic threshold (again, under the hypothesis that solutions are sampled according to the equilibrium measure).

A new approach to algorithmic hardness in random problems was recently introduced: the so-called overlap gap property (OGP), based on the topology of the solution space [31]. Such a property, already discovered in Refs. [13,15] but named OGP only in Ref. [17], allows one to rigorously demonstrate the ineffectiveness of a large class of algorithms, named *stable algorithms* [32], in solving the corresponding optimization problem. Stable algorithms include low-degree polynomial algorithms [33–35], approximate message passing algorithms [36], and Monte Carlo

algorithms run for a fixed (i.e., not growing with N) number of steps. One could conjecture the OGP to be plausible evidence for computational intractability, not just for stable algorithms, but even for more general algorithms. There are, however, counterexamples: In a recent work [37], it has been proved that the shortest-path optimization problem has the OGP in sparse graphs, but it can be solved by $O(\log N)$ -degree (nonstable) polynomial algorithms. Thus, in this case, the OGP is not predictive of algorithmic intractability for an average-case optimization problem. Something similar also happens in the case of the XOR-SAT problem [38], which displays the OGP [39] while it is easy to solve through Gaussian elimination algorithms [40]. Both the exact nature of the stability of the algorithms and the existence of the OGP are problem dependent and sometimes extremely technical to prove.

Here, we introduce a new crucial ingredient in the analysis of these algorithms: the scaling of the running time with the size of the system. As mentioned before, when MC are run for a number of MCSs that is fixed with respect to the size of the problem, the running time scales linearly with the problem size N . Under these conditions, one can often demonstrate that MC algorithms are stable algorithms, and their performances are thus limited by the existence of OGP [41]. However, MC algorithms have not been proved to be stable when running for a number of MCSs that increases with the problem size, and nothing is known about their performance.

III. METHODS

The standard methodology to determine algorithmic thresholds measures the time an algorithm needs to find a solution [in short, time to solution (TTS)] and studies how it scales with the system size at different values of α . However, this analysis does not provide a clear estimate for the algorithmic threshold. Indeed, for the sizes one can study, $TTS(N)$ makes just a smooth crossover from a “slow” to a “fast” growth around the algorithmic threshold.

We follow, instead, two other routes for the analysis of the performance of MC-based algorithms in solving K -SAT problems and the determination of the algorithmic threshold for SA. Our two methods give consistent results for the K -SAT problem with $K = 3, 4$.

First, we use an approach that exploits finite-size effects to determine the algorithmic threshold α_{SA} , similar to the approach used to study the critical behavior in statistical physics models. We run the algorithm (e.g., SA in our case) for a polynomial time $t \sim N^z$, where the “dynamical” exponent z controls the time scaling. Then we examine the mean extensive energy U reached after a time t or the probability of solving the problem P_{SAT} in a time t , as a function of α . Both quantities, U and P_{SAT} , have a clear crossing at the same algorithmic threshold α_{SA} , supporting the scenario that below α_{SA} every problem can be solved in the large N limit if SA is run for a time $t = O(N^z)$, while

none is solvable by SA above α_{SA} . We will show how this algorithmic threshold depends on the value of z , that is, on the time scaling chosen for running the algorithm.

Our main results are obtained for $z = 2, 3$ (i.e., quadratic and cubic running times). These results provide clear evidence that the algorithmic threshold depends on the time scaling.

Determining the SA threshold for times growing less than quadratic (e.g., linear times) is a more difficult task. An MCS corresponds to the update of every variable once, and it takes a time $O(N)$. Thus, the linear-time regime actually corresponds to running an MC algorithm for a *fixed* (not growing with N) number of MCSs. Even below the algorithmic threshold, it is not surprising that running an MC algorithm for a fixed number of MCSs makes the success probability dependent on the system size (the number of MCSs that are enough for a given problem are probably not enough for a much larger problem). Indeed, in this linear-time regime, we find severe finite-size effects that make the techniques we have used for $z = 2, 3$ useless. To overcome this problem, we have studied MC at zero temperature, which corresponds to a sort of greedy algorithm without thermal noise, that we find to run in almost linear time. Results are reported in Appendix A.

As a complementary approach, we also work in the “large size limit,” where this limit must be understood as “the system size is large enough that finite-size effects are negligible.” In this limit, self-averaging quantities have vanishing fluctuations and concentrate on a unique curve (and from this unique curve, we need to extract the requested information). For example, considering the intensive energy u reached after a running time t , one expects a “relatively fast” decay of $u(t)$ below the algorithmic threshold and a “much slower” decay above the threshold. We will provide more quantitative statements below, but in practice, in this limit, the threshold value $\alpha_{\text{SA}}^{\infty}$ is defined as the critical value such that $u(t)$ decays as a power law of the running time t . This is very similar to what has been observed in Ref. [21] for the hypergraph bicoloring problem. It is worth stressing that in the large size limit, the time required to solve the problem is always formally infinite, and this is the reason why the algorithmic threshold must be estimated from the way the energy decays for large times, rather than from the time to reach a solution.

IV. SIMULATED ANNEALING ALGORITHM

The hardness of K -SAT is in the structure of the Boolean formula f , which is the conjunction (logic *and*) of M smaller expressions called *clauses*. A clause is a disjunction (logic *or*) of K variables or their negations. The following simple example has $K = 3$, $M = 2$ clauses and only $N = 3$ variables:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (-x_1 \vee -x_2 \vee x_3).$$

While in this example it is easy to find a satisfying assignment by simple inspection, this is not necessarily the case when N and M increase. In the random K -SAT, the variables entering each clause are randomly chosen among the N possible ones and negated with probability 0.5. The parameter $\alpha = M/N$ controls the hardness of the formulas. Empirical evidence and theoretical studies [13,14,16,17,24,42] support the existence of a hard phase before the SAT-UNSAT threshold: In the hard phase, solutions do exist, but searching algorithms require times scaling superpolynomially with the problem size to find a solution.

In combinatorial optimization problems, the most natural choice for the energy function is the number of unsatisfied clauses $U(\vec{x})$. This energy function is bounded between 0 and M , reaching the minimum value $U = 0$ for every satisfying assignment.

MC algorithms are the standard technique for sampling the configuration space of the system according to the Boltzmann-Gibbs distribution $P(\vec{x}) = e^{-\beta U(\vec{x})}/Z$. Here, Z is a normalization factor and $\beta = 1/T$ is the inverse temperature. While solutions of the optimization problem correspond to the equilibrium configurations at $T = 0$, a positive temperature allows *uphill* moves that facilitate overcoming the local barriers.

A single MC step implementing the Metropolis dynamics [43] works as follows. Given a configuration \vec{x} , a new configuration \vec{x}' is proposed where a single variable is flipped. Among the N variables, the one to be flipped is chosen uniformly at random. The proposed change in configuration is accepted with probability

$$r(\vec{x} \rightarrow \vec{x}') = \begin{cases} 1 & \text{if } \Delta U(\vec{x} \rightarrow \vec{x}') \leq 0 \\ e^{-\beta \Delta U(\vec{x} \rightarrow \vec{x}')} & \text{if } \Delta U(\vec{x} \rightarrow \vec{x}') > 0, \end{cases} \quad (1)$$

where $\Delta U(\vec{x} \rightarrow \vec{x}') = U(\vec{x}') - U(\vec{x})$.

The SA algorithm starts implementing MC at a high temperature (low β), where there are no barriers and the configuration can be easily updated. Then the temperature T is slowly decreased, performing some MCSs at each selected temperature. This can be implemented using different schemes [44,45]. For simplicity, we restricted ourselves to the traditional choice of a constant temperature change ΔT [2]. The aim is to have an SA algorithm that keeps sampling equilibrium configurations while decreasing the temperature. If this can be achieved until $T = 0$, then a solution with $U = 0$ can be obtained (if it exists). In practice, when dealing with finite-size systems and finite running times, we do not expect SA to be able to sample from the equilibrium distribution at each time. From this point of view, real-world implementations of SA are typically out-of-equilibrium processes.

The *cooling schedule* is the sequence of chosen temperatures (T_0, T_1, \dots, T_n) used by SA and the number of MCSs the algorithm will take at each temperature. In general, the

initial temperature T_0 is required to be high enough to be in the ergodic phase and the final temperature T_n should be (close to) $T = 0$. In our simulations, we consider a very simple schedule (from T_0 to $T_n = 0$ with fixed $\Delta T = T_0/n$), and one MCS is performed at each temperature, such that the SA algorithm is run for a total of n MCSs. Although the schedule is probably not the optimal one, we choose it to be quasiadiabatic (the ΔT value is the smallest possible), and we expect that changing T_0 and T_n would change only prefactors and not the scaling between size and time (which is what we are interested in). In general, it is advisable to use a simple schedule that can provide a clear indication of the time scaling, although the absolute running time could be further optimized.

In contrast to our choice, some other implementations of SA use a schedule that stays more than one MCS at each temperature [44,45], in an attempt to ensure equilibration (this is possible only in the ergodic phase $\alpha < \alpha_d$). In any case, when jumping from T_k to T_{k+1} , the algorithmic dynamics exhibits a transient regime and then reaches equilibrium at T_{k+1} after a certain number of steps. Thus, the number of steps τ spent in this jump must be adjusted so that τ is greater than the transient time $\tau_{\text{eq}}(T_k \rightarrow T_{k+1})$.

However, if the difference $\Delta T = T_{k+1} - T_k$ decreases, so does τ_{eq} . In general, when $\Delta T \rightarrow 0$, we also have $\tau_{\text{eq}} \rightarrow 0$. That is one of the reasons why, in our schedule, we follow a different strategy to ensure convergence to equilibrium at each temperature. By taking $\Delta T = T_0/n$ equal for all transitions $T_k \rightarrow T_{k+1}$ and approaching the large n limit, we get that ΔT is so small that a single MCS must always be enough to equilibrate, provided that one is in the ergodic phase $\alpha < \alpha_d$ (while for $\alpha > \alpha_d$ any schedule would need a number of steps growing exponentially with N to reach equilibrium). Even for finite n , we expect that taking the smallest steps in temperature produces a dynamics that remains closer to equilibrium at each step in the ergodic phase. Moreover, our choice avoids the introduction of the parameter τ that would make the analysis more cumbersome.

Summarizing, the reason why it is worth studying the SA algorithm is that it is known to be general (i.e., it can be used to optimize any energy function) and very robust. The use of the temperature cooling is a standard approach when exploring a rough energy landscape to allow for the bypassing of energy barriers. The slow convergence to the zero-temperature limit focuses the Gibbs measure on the solutions to the problem, thus formally ensuring the optimum could be achieved in the infinite time limit.

V. ALGORITHMIC THRESHOLDS FROM FINITE-SIZE ANALYSIS

In many cases, the running time of algorithms based on MC, such as SA, is limited to a fixed number of MCSs and, therefore, grows linearly with the system size.

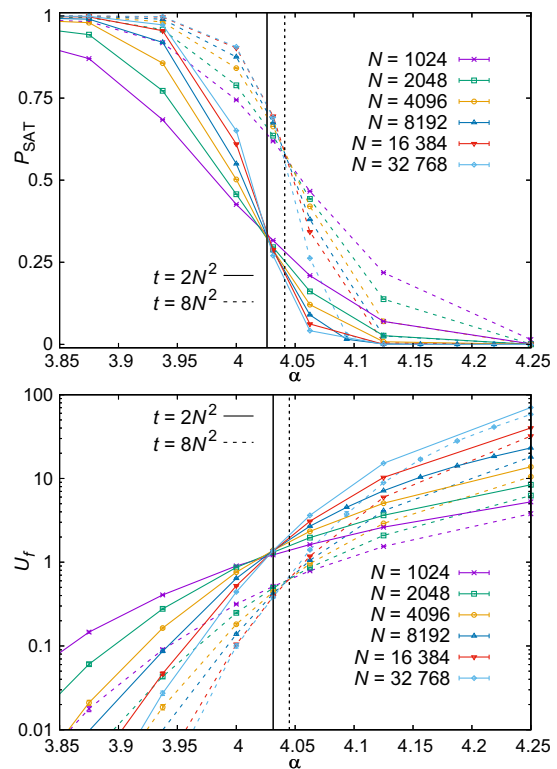


FIG. 1. Simulated annealing thresholds for quadratic time scaling in random 3-SAT instances. For several system sizes, we measure the probability P_{SAT} of finding a solution (upper panel) and the average extensive energy U_f (lower panel) after the SA algorithm has been run for a quadratic time, $t = AN^2$. The cooling schedule starts at $T_0 = 0.2$, and each point represents an average over 10^4 different runs. The error bars are reported, but are smaller than the points. Continuous lines are for $A = 2$ and dashed lines are for $A = 8$. The algorithmic thresholds are $\alpha_{\text{SA}}^{N^2} = 4.031(2)$ for $A = 2$ and $\alpha_{\text{SA}}^{N^2} = 4.045(5)$ for $A = 8$.

Unfortunately, in this linear time regime, it is not easy to estimate the algorithmic threshold. We discuss this regime for SA solving K -SAT in Appendix A. Here, we show how better algorithmic thresholds can be estimated by going to superlinear time scalings.

To explore this idea, we ran SA with the schedule mentioned above for different system sizes N and a number of MCSs growing linearly in N , implying $t = O(N^2)$. As expected, in the upper panel of Fig. 1, the probability of finding a solution P_{SAT} decreases as α increases. The drop in the probability becomes sharper as N increases. All the curves cross at the same value of α (within the statistical uncertainty of our data), which represents the algorithmic threshold α_{SA} . Indeed, we expect the curves to become a step function in the large N limit, such that $P_{\text{SAT}} = 1$ for $\alpha < \alpha_{\text{SA}}$ and $P_{\text{SAT}} = 0$ for $\alpha > \alpha_{\text{SA}}$.

The bottom panel of Fig. 1 shows an analogous behavior in the final mean extensive energy U_f achieved by SA with the same quadratic scaling. The final energy grows with α with a slope increasing with N . Data for different problem

sizes crosses at the same α_{SA} computed from the crossing of P_{SAT} curves, providing stronger and consistent evidence for a sharp algorithmic transition at α_{SA} . Indeed, in the large N limit, U_f tends to zero for $\alpha < \alpha_{\text{SA}}$, while it diverges for $\alpha > \alpha_{\text{SA}}$.

It is worth stressing that the nice crossings in Fig. 1 have never been observed before because MC algorithms (as many other algorithms) were studied only in the linear regime. A reliable algorithmic threshold for SA can be obtained only by scaling times at least quadratically with the problem size.

The specific value of α_{SA} not only depends on the quadratic scaling of the running time with the problem size, $t = AN^2$, but also on the prefactor A , as can be appreciated in Fig. 1, where data for $A = 2$ and $A = 8$ are reported. Let us call $\alpha_{\text{SA}}^{N^2}(A)$ the algorithmic threshold obtained in the large N limit, scaling running times as $t = AN^2$. We expect a range of thresholds for SA running in quadratic times, $\alpha_{\text{SA}}^{N^2} \in [\alpha_{\text{SA}}^{N^2}(0), \alpha_{\text{SA}}^{N^2}(\infty)]$. In particular, the lower value $\alpha_{\text{SA}}^{N^2}(0)$ is the algorithmic threshold for SA running in linear time, which is hard to compute studying the linear regime, as it needs extrapolations, but is computable coming from the quadratic regime. The interested reader can take a look at Appendix B, where we determine the algorithmic threshold for a small value of the coefficient ($A = 2^{-5}$). The upper value $\alpha_{\text{SA}}^{N^2}(\infty)$ corresponds to the algorithmic threshold for SA running in quadratic time. Although it is in practice unreachable, we can get a bound from the best results in Fig. 1. The scaling $t = 8N^2$ provides $\alpha_{\text{SA}}^{N^2}(\infty) \gtrsim 4.045$. This critical value has been computed from the crossings of U_f curves, which are, in general, more accurate and robust with respect to P_{SAT} data. This value is just a lower bound to the best algorithmic threshold for SA running in a quadratic time. However, we have noticed that $\alpha_{\text{SA}}^{N^2}$ does not change much by varying the prefactor A .

A way of getting an upper bound to $\alpha_{\text{SA}}^{N^2}$ is to study the performance of SA running in cubic times, $t = O(N^3)$. The results are reported in Fig. 2 for random 3-SAT and random 4-SAT. Data in the upper panel show a clear crossing, implying $\alpha_{\text{SA}}^{N^3} \simeq 4.11$ (the threshold depends again on the prefactor, but such a dependence is very mild, and we may ignore it for our purposes). Interestingly enough, this threshold is far beyond the dynamical transition $\alpha_d = 3.86$ [24], and very close to what can be achieved by the best local search algorithm, focused Metropolis search (FMS), discussed in Appendix C.

Data in the lower panel of Fig. 2 are for SA solving random 4-SAT in cubic time and present strong finite-size corrections that manifest themselves by the large shifting in the crossing points $\alpha_{\text{cross}}(N, 2N)$ between data of size N and $2N$. The reason for these large finite-size corrections can be understood by looking at the residual energy U_f at

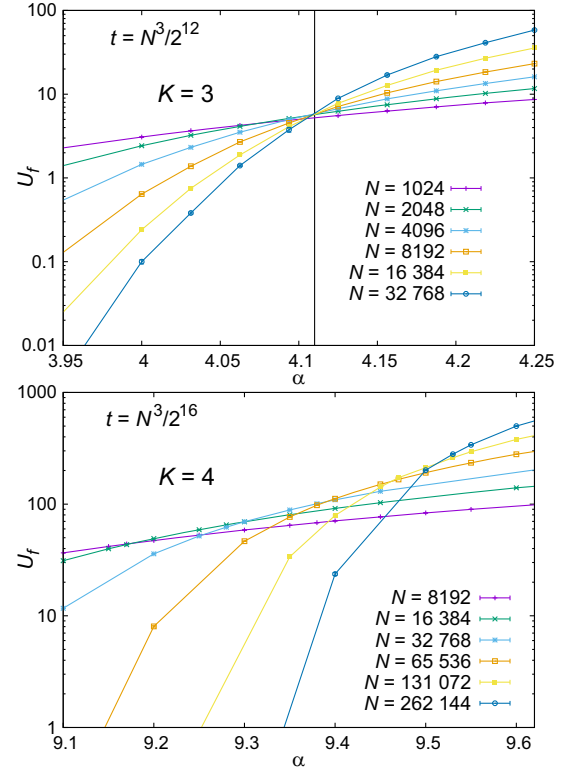


FIG. 2. Simulated annealing thresholds for cubic time scaling in random 3-SAT and random 4-SAT. For each system size, we measure the average extensive energy U_f after a running time t scaling cubically with the problem size. The cooling schedule starts at $T_0 = 0.2$, and each point represents an average over 10^4 different runs. The error bars are reported, but smaller than the points. For $K = 3$ (top panel) we used $t = N^3/2^{12}$, while for $K = 4$ (bottom panel) we have $t = N^3/2^{16}$. The predicted algorithmic threshold for $K = 3$ is located at the crossing points of the curves in the upper panel, $\alpha_{\text{SA}}^{N^3} = 4.110(4)$. In random 4-SAT, finite-size effects are strong and estimating the algorithmic threshold requires a more accurate analysis and extrapolation (see text and Fig. 3).

the crossing points. In the random 3-SAT case, the residual energy at the (unique) crossing point is small, $U_f = O(1)$ (remember we are considering the extensive energy). Instead, for random 4-SAT, U_f at the crossing points grows to much larger values. This, in turn, requires the study of larger systems to actually access the relevant values of U_f at criticality. Hence, the severe finite-size effects.

A quantitative estimate of the algorithmic thresholds for $K = 4$ requires the extrapolation of the crossing points $\alpha_{\text{cross}}(N, 2N)$ in the large N limit. In the upper panel of Fig. 3, the crossing points are plotted as a function of the correction to scaling term $N^{-\omega}$. The optimal value for ω has been chosen by minimizing the χ^2 value of the linear fit to $\alpha_{\text{cross}}(N, 2N)$ versus $N^{-\omega}$ data (as shown in the inset). There are two sources of error in this extrapolation. The crossing values have a statistical uncertainty associated

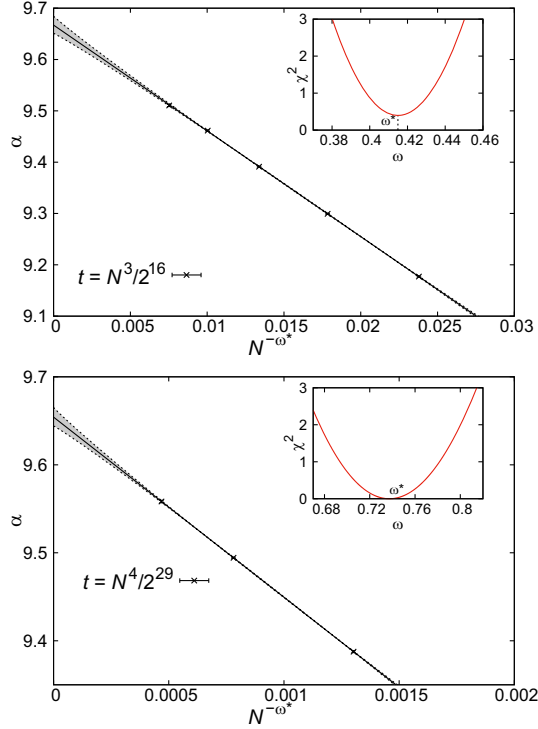


FIG. 3. Determination of the algorithmic thresholds for different time scalings in the random 4-SAT. We include cubic (top panel) and quartic (bottom panel) scalings. Each point in the panels corresponds to the crossing $\alpha_{\text{cross}}(N, 2N)$ between the curves $U_f(\alpha)$ for sizes N and $2N$. For cubic scaling, the data are shown in the bottom panel of Fig. 2. For quartic scaling, the data (not shown) includes the sizes $N = 8192, 16\,384, 32\,768, 65\,536$. The cooling schedule starts at $T_0 = 0.2$, and each point represents an average over 10^4 different runs. The error bars are reported, but smaller than the points. Continuous lines are linear fits of the type $\alpha_{\text{cross}}(N, 2N) = \alpha_{\text{SA}} - cN^{-\omega}$ made with the optimal exponent ω^* that minimizes the quality of the fit $\chi^2(\omega)$, shown in the insets. The shaded regions show the uncertainty on the fit considering statistically acceptable values of ω around ω^* (see the text for more details). Final results are $\omega^* = 0.4149$ and $\alpha_{\text{SA}}^{N^3} = 9.665(15)$ for cubic scaling and $\omega^* = 0.7373$ and $\alpha_{\text{SA}}^{N^4} = 9.650(15)$ for quartic scaling.

with the measurement of the energies U_f (the corresponding error bars are shown in Fig. 3), but these are very small. The largest uncertainty in the extrapolation comes from the unknown value of the correction-to-scaling exponent ω . For each value of ω we have performed a linear fit to the data (as shown in Fig. 3) and computed the quality of the fit through the χ^2 (shown in the inset). The optimal value ω^* is the minimizer of $\chi^2(\omega)$. However, all values of ω such that $\chi^2(\omega) - \chi^2(\omega^*) < 1$ are statistically acceptable. For a numerical confirmation of the robustness of this extrapolation method, we refer the reader to the Appendix B.

The colored regions in Fig. 3 represent the fits obtained using these acceptable ω values and provide the correct uncertainty on the extrapolated value. This procedure

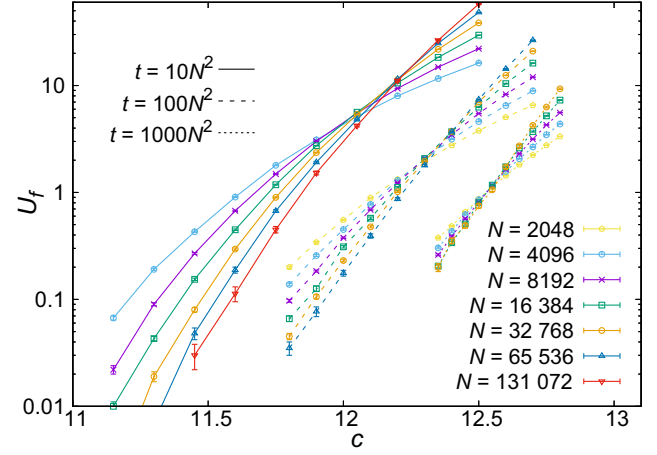


FIG. 4. Simulated annealing thresholds for quadratic time scaling in five-coloring random graphs. For each graph of size N and mean degree c , we measure the average extensive energy U_f after a fixed running time, $t = AN^2$, with $A = 10, 10^2, 10^3$ (from left to right). The cooling schedule starts at $T_0 = 1.0$, and each point represents an average over 10^4 different runs for $N \leq 32768$, and between 10^3 and 10^4 runs for $N \geq 65536$. The crossing point identifies the SA algorithmic threshold.

provides the algorithmic thresholds $\alpha_{\text{SA}}^{N^3} = 9.665(15)$ and $\alpha_{\text{SA}}^{N^4} = 9.650(15)$ for cubic and quartic scaling, respectively. The observation that $\alpha_{\text{SA}}^{N^3}$ and $\alpha_{\text{SA}}^{N^4}$ are compatible within errors implies that moving from the cubic to the quartic scaling there is no improvement. We will come back later to discuss this point.

To support the robustness of our findings, we consider a different combinatorial optimization problem, namely, q coloring of random graphs. Given an Erdős-Rényi random graph of N nodes and mean degree c , the problem is to assign to each node a color taking values in $\{1, 2, \dots, q\}$, such as to avoid monochromatic edges connecting two nodes with the same color. The SA can also be applied to this problem, where the energy function counts the number of monochromatic edges in a given configuration of the variables.

Results for $q = 5$ are reported in Fig. 4, where we plot the final energy U_f reached by SA in a quadratic time, $t = AN^2$, as a function of graph mean degree c . As in the K -SAT case, we observe a clear crossing at the algorithmic threshold α_{SA} (which again depends on the prefactor A). This confirms what we already obtained from K -SAT as a more general behavior.

VI. ALGORITHMIC THRESHOLDS IN THE LARGE SIZE LIMIT

Here, we discuss a complementary estimate of the algorithmic thresholds for SA in solving random K -SAT problems obtained in the so-called large size limit. In practice, we simulate very large instances of size $N = 10^5$

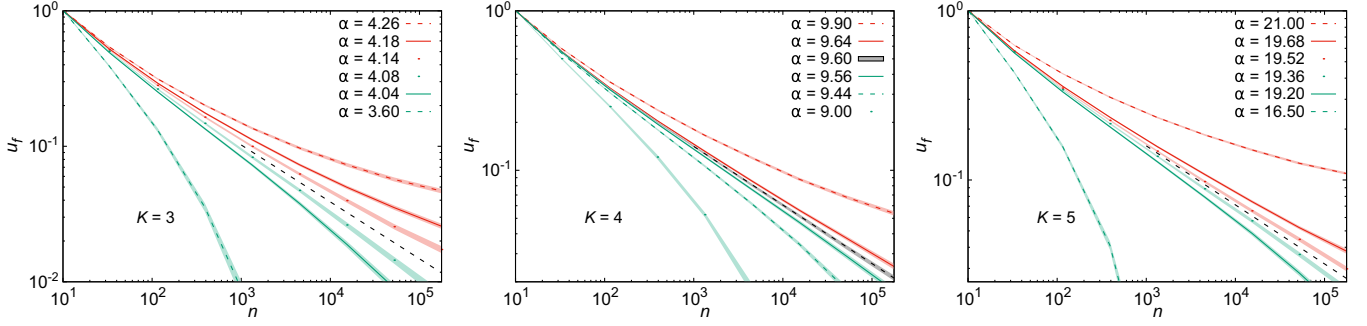


FIG. 5. Estimating the algorithmic threshold in the large size limit. Simulated annealing with a linear temperature schedule is run, starting at $T_0 = 1.0$, for n MCS steps to solve very large ($N = 10^5$) random K -SAT instances with $K = 3$ (left), $K = 4$ (center), and $K = 5$ (right). We compute the average of the final intensive energy u_f (fraction of clauses violated) as a function of the total number n of MCSs. Axes are on a logarithmic scale to highlight when the function $u_f(n)$ follows a power-law decay at the critical algorithmic threshold. Green curves decay faster than a power law. Red curves decay slower than a power law. Curves have been rescaled so that they all start at the same point (1,10) to make their curvature easier to identify. Shaded areas represent the measurement uncertainty (one standard deviation). The curve for $\alpha = 9.60$ in the center panel is colored in black because it is not possible to distinguish it from a straight line when n is large. Black dashed lines represent the critical power laws $u_f \propto n^{-b(K)}$ with $b(3) = 0.42$, $b(4) = 0.37$, and $b(5) = 0.35$. Estimated algorithmic thresholds are $\alpha_{SA}^\infty(K = 3) = 4.11(1)$, $\alpha_{SA}^\infty(K = 4) = 9.60(2)$, and $\alpha_{SA}^\infty(K = 5) = 19.44(3)$.

for $K = 3, 4, 5$, such that finite-size effects are negligible. We run SA as explained above: a linear schedule in temperature with $T_0 = 1.0$ and $T_n = 0$ with 1 MCS per temperature (the most adiabatic scheduling), for a total of n MCSs. We measure the final energy U_f , or equivalently its intensive value $u_f = U_f/N$, as a function of the running time n . The results are presented in a doubly logarithmic scale in Fig. 5.

For each value of $K = 3, 4, 5$, we have run simulations in a range of α values which is fully contained in the SAT phase (the reader can find the values of the SAT-UNSAT critical threshold α_s in Table I). This means that the equilibrium energy is exactly zero. However, when the SA is run for a finite number n of MCSs in a very large problem (remember we are working in the large size limit), it will soon or later fall out of equilibrium at low temperatures, and the final energy $U_f(n)$ will be strictly positive.

In the large size limit, the intensive energy $u_f(n)$ is a decreasing function of n with practically no fluctuations at

any finite value of n (this is a consequence of the energy being a self-averaging quantity). Indeed, curves in Fig. 5, which have been measured for problems of size $N = 10^5$, are smooth and very close to the large size limit curve $u_f(n)$ (finite-size effects are discussed below when commenting Fig. 6).

In Fig. 5, we plot the data in a doubly logarithmic scale because we want to stress which curves decay faster than a power law (green curves) and which ones decay slower than a power law (red curves). We observe that in a small range of α the curvature of the data shown in Fig. 5 changes from negative (green curves) to positive (red curves). This observation suggests that the energy decays as a power law $u_f(n) \propto n^{-b}$ at a single value of α (or in a very narrow

TABLE I. Summary of thresholds for random K -SAT. The values for the dynamical transition α_d and the condensation transition α_c have been taken from Table 1 in Ref. [24]. The SAT-UNSAT thresholds α_s are from Ref. [14]. We report the algorithmic thresholds for SA, obtained from finite-size scaling in the quadratic regime $\alpha_{SA}^{N^2}$, with a prefactor $A \in [0.25, 8]$, from finite-size scaling in the cubic regime $\alpha_{SA}^{N^3}$ and in the large size limit α_{SA}^∞ .

K	α_d	α_c	α_s	$\alpha_{SA}^{N^2}$	$\alpha_{SA}^{N^3}$	α_{SA}^∞
3	3.86	3.86	4.267	4.025–4.045	4.110(4)	4.11(1)
4	9.38	9.547	9.931	9.15–9.4	9.665(15)	9.60(2)
5	19.16	20.80	21.117	19.44(3)

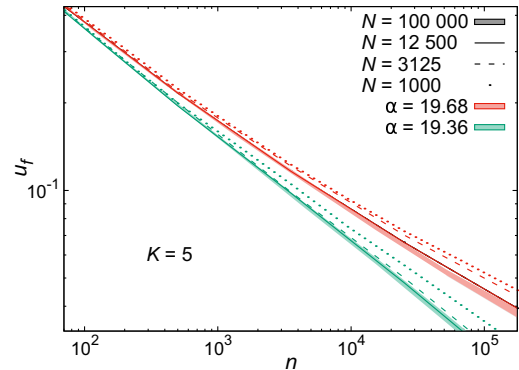


FIG. 6. Finite-size effects on simulated annealing behavior. The decay measured on problems of size $N = 10^5$ persists also for smaller sizes, $N \sim O(10^4)$. On even smaller sizes, the slope changes a little, but the curvature is preserved, thus making the proposed procedure to estimate α_{SA}^∞ robust. Error bars are reported only for the largest size with a shadowed area.

interval of α values). Let us call critical such α value (or interval of).

We argue that the algorithmic threshold $\alpha_{\text{SA}}^{\infty}$ corresponds to the critical α value (or to the upper limit of the critical interval). Our argument goes as follows: First, we prove that at criticality the time to find a solution by SA in a problem of size N scales like a power law, $t \sim N^{1+1/b}$; then, we prove that for $\alpha > \alpha_{\text{SA}}^{\infty}$ the time to find a solution scales superpolynomially.

At a critical α value, we have $u_f(n) \propto n^{-b}$ in the large size limit. Thus, the typical time to reach a solution in a problem of size N , which is finite, but large enough, is given by the n value such that $u_f(n) \sim 1/N$. Indeed, $u_f = a/N$ implies $U_f = a$, and if $a < 1$, then there is a nonzero probability that SA finds a solution (to have a mean extensive energy equal to $U_f = a < 1$, at least a fraction $1 - a$ of samples must have zero energy, given the energy takes only integer values). This argument states that SA finds solutions with a nonzero probability if run with $n \sim N^{1/b}$ MCSs, which requires a time $t \sim N^z$ with a dynamical exponent $z = 1 + 1/b$, considering that every MCS takes a time $O(N)$.

For $\alpha > \alpha_{\text{SA}}^{\infty}$ the curve $u_f(n)$ has a positive curvature (in a log-log scale), which means that the local slope decreases as n grows. We can use again the argument above, but now the effective b exponent becomes smaller for larger n , and eventually goes to zero for $n \rightarrow \infty$. This implies that the typical time to reach a solution grows faster than any power law, thus proving SA is a superpolynomial algorithm in this regime.

We estimate the algorithmic thresholds for SA in the large size limit $\alpha_{\text{SA}}^{\infty}(K)$ by bounding the critical α values. In Fig. 5 we report data for $u_f(n)$ measured with $K = 3, 4, 5$ (panels from left to right). In each case, the critical α value is between the higher green curve and the lower red one. The estimated algorithmic thresholds are $\alpha_{\text{SA}}^{\infty}(3) = 4.11(1)$, $\alpha_{\text{SA}}^{\infty}(4) = 9.60(2)$, and $\alpha_{\text{SA}}^{\infty}(5) = 19.44(3)$.

From the power-law decay at criticality, we can estimate the dynamical exponent $z(K)$ ruling the scaling of times with problem size, $t \propto N^z$. Our estimates are $z(3) = 3.4(3)$, $z(4) = 3.7(2)$, and $z(5) = 3.9(1)$. We note that all exponents are between 3 and 4 (and this is an argument supporting the lack of any gain in going from the cubic to the quartic scaling, observed in Fig. 3 and discussed in more detail below).

The procedure we have proposed and used to estimate $\alpha_{\text{SA}}^{\infty}$ assumes finite-size effects are somehow benign. We check our assumption in Fig. 6 through the study of the decay of $u_f(n)$ in the large n regime for very different problem sizes N . The largest size reported, $N = 10^5$, is the same one used in Fig. 5 to estimate $\alpha_{\text{SA}}^{\infty}$ and comes with a shaded area representing the statistical uncertainty (one standard deviation). The first important observation is that the data collected in solving problems of a size smaller by

an order of magnitude, $N \sim O(10^4)$ still follow closely the decay observed for larger sizes. The second observation is that, for even smaller problems, although finite-size effects are visible, the curvature (in a log-log scale) is preserved, i.e., it is negative for $\alpha < \alpha_{\text{SA}}^{\infty}$ and positive for $\alpha > \alpha_{\text{SA}}^{\infty}$. These observations imply our procedure to estimate $\alpha_{\text{SA}}^{\infty}$ is robust and can be applied even on problems of a modest size.

VII. SUMMARY AND DISCUSSION

We start by summarizing our results. The numerical values of the algorithmic thresholds we have estimated are reported in Table I, together with other relevant thresholds obtained from the thermodynamic solution to the random K -SAT problem [14,16,24] and described in the Introduction.

The algorithmic thresholds for SA in solving random K -SAT problems can be well defined only for superlinear regimes (we discuss in Appendix A the linear regime). The estimates for α_{SA} in the quadratic and the cubic regimes are clearly different, as can be seen from the thresholds summarized in Table I. This key result requires a revision of the concept of algorithmic threshold, which can no longer be considered unique, given the problem and the algorithm, but it can depend on the scaling between the problem size and the resolution time.

The algorithmic threshold grows when moving from a quadratic to a cubic scaling regime. However, as shown in Fig. 3, going to a quartic scaling regime is not helpful. A possible explanation for this observation comes from the estimates we got via the analysis of the large size limit. Let us consider that the algorithmic threshold $\alpha_{\text{SA}}^{\infty}$ obtained from the latter provides the largest threshold for any polynomial scaling regime; that is, for $\alpha > \alpha_{\text{SA}}^{\infty}$ the SA running times grow superpolynomially. Given that at $\alpha_{\text{SA}}^{\infty}$ the SA running times scale like $t \propto N^z$ with $z < 4$ (for the K values studied), then running SA for a time growing quartically with N cannot push the algorithmic threshold beyond $\alpha_{\text{SA}}^{\infty}$. Moreover, since $\alpha_{\text{SA}}^{\infty}$ is statistically compatible with $\alpha_{\text{SA}}^{N^3}$, we conclude that thresholds in the cubic and quartic regimes should be statistically compatible. Remarkably, this is not the only case in the literature where an exponent between $z = 3$ and $z = 4$ arises naturally from the running times. While analyzing the results of another efficient algorithm in the three-coloring problem, the authors of Ref. [3] noted that $O(N^{3.5})$ update steps were enough for them to obtain accurate statistics on the set formed by all the solutions to the problem, once one of them was found.

Among all possible polynomial time regimes, we observe the cubic one to be optimal. This is again related to the exponent z measured at criticality in the large size limit, which is slightly larger than 3. The optimal α_{SA} thresholds can be compared with the phase transitions in the structure of solutions. For every value of K studied, the

optimal α_{SA} is located well above the dynamical transition α_d . This is a nontrivial observation. Indeed, given that at α_d the ergodicity of MC-based algorithms is broken, one could imagine α_d to be the algorithmic threshold for SA and any MC-based algorithm [9,16]. However, the ergodicity breaking happening at α_d is related to the sampling of the equilibrium measure, which is uniformly distributed over all the solutions. By sampling solutions according to a nonuniform measure, it is well known that the dynamical threshold can change [21,46]. The observation that SA can efficiently solve random K -SAT instances well beyond α_d implies SA is not sampling solutions uniformly, but in a more efficient way that accesses a subset of solutions even beyond α_d .

Although we have chosen to study the SA algorithm for its generality, rather than for its performance, it is still worth comparing it to the best local search algorithm, FMS, discussed in Appendix C. FMS thresholds in the linear regime are $\alpha_{\text{FMS}} = 4.205(5)$ for $K = 3$ and $\alpha_{\text{FMS}} = 9.615(5)$ for $K = 4$, and grow to $\alpha_{\text{FMS}} \simeq 4.22$ for $K = 3$ and $\alpha_{\text{FMS}} \simeq 9.7$ for $K = 4$ in a quadratic regime. Compared to the FMS thresholds, the performance of SA is sub-optimal (as expected) but not too far from optimality. Thus, restricting our study to the SA algorithm has not been a too detrimental choice in terms of performance.

We discuss now two fundamental questions. Why is the optimal performance of SA (and probably of many other algorithms) achieved in a superlinear regime, with running times scaling with a nontrivial power of the problem size, $t \sim N^z$? How can several critical regimes with different time scaling (e.g., quadratic and cubic) coexist? These questions are nontrivial because, in the simplest scenario, one could imagine the existence of just two phases separated by an algorithmic threshold α_{alg} : an “easy” phase for $\alpha < \alpha_{\text{alg}}$ with no algorithmic barriers, where the algorithm converges to a solution quickly [47], and a “hard” phase for $\alpha > \alpha_{\text{alg}}$, where the presence of algorithmic barriers impedes fast convergence to a solution and requires superpolynomial times (e.g., when an extensive free-energy barrier needs to be surpassed by a stochastic algorithm satisfying detailed balance). In this simple scenario, the algorithmic threshold α_{alg} is unique for any polynomial scaling.

The results presented in this work reveal a novel scenario, where the easy phase is itself divided into different regions that are accessible only by scaling running times with the proper power law of the problem size. Roughly speaking, we can say that the easy phase gets broken into easier and less easy phases. Every transition between two of these polynomially accessible phases corresponds to a different algorithmic threshold between polynomial regimes, while the largest among these thresholds corresponds to the ultimate transition to a superpolynomial regime.

We conjecture that some (if not all) of these polynomially accessible phases are controlled by entropic barriers. Indeed, as shown in Appendix A, there are greedy algorithms that are not allowed to climb over energetic barriers and still find solutions in the region accessible by SA, scaling times quadratically with the problem size. This observation strongly suggests that, in this “quadratic” region, solutions are difficult to access by SA only because the stochastic dynamic has to find the right path in a complex energy landscape with many apparently equivalent directions (i.e., an entropic barrier).

Increasing α , the energy landscape may become more and more complicated to explore for purely entropic reasons, even without developing relevant energetic barriers. So, a plausible explanation for the many different algorithmic thresholds we have found is that, even when some solutions are polynomially accessible, these are few and connected by tiny “corridors” to the rest of the configurational space, so as to make it very hard to find them, especially by a purely stochastic algorithm, like SA (we discuss in Appendix C that a focused algorithm performs much better, probably because it can enter in these tiny and rare corridors more easily).

The picture we have in mind for the space of solutions in these polynomially accessible phases above α_d is similar to the one found in other models, where it has been proved that regions exist such that the majority of solutions are clustered and cannot be sampled efficiently, but a sub-dominant subset of solutions can be algorithmically achieved. This is the case, for example, of the binary perceptron, where the majority of solutions are isolated and unaccessible [48,49], but algorithms can still work [50] by reaching the few accessible states, which are “flat” minima of very large local entropy connected by paths of low probability [48–54].

A scenario in which the accessible states that can be algorithmically reached are flat, i.e., marginal, is very common in disordered systems [55]. Assuming such a scenario, the timescale needed to solve a problem by SA (or by any other algorithm) is likely to be related to the entropic difficulty of finding a descending path in a very flat landscape, rather than to the hardness of jumping over an energetic barrier. In such a scenario, the dynamical exponent z ruling the scaling between problem size and time to find a solution is probably related to the dimension of the marginal manifold where the searching dynamics is taking place and the dimension (i.e., the number) of descending directions that eventually lead to a solution. Future studies will better clarify this relation. However, it should be noted that SA is often out of equilibrium and does not sample solutions according to the uniform measure [21]. Thus, analytic computations are much harder than in equilibrium statistical physics.

We conclude by remarking that the methodology presented in this work can be easily applied to any other combinatorial optimization problem. For example, although the formulations of both K -SAT and q coloring include only local constraints, it is possible to go beyond and study problems with mixed local and global constraints, such as planted clique, where MC algorithms have also been successfully used [56]. We hope that it will become the standard procedure for computing the algorithmic thresholds in several problems solved by different algorithms.

Among the problems to which our procedure can be extended, it is worth pointing out artificial neural networks. Indeed, in this class of problems, both the training time and the inference time can be scaled with the problem size to achieve optimal performances [57,58], but to the best of our knowledge a systematic study of different time-scaling regimes is lacking.

Another interesting case is inference problems, which can be recast as optimization problems with a known optimum (corresponding to the signal to be inferred). Recent findings in the literature rigorously proved that, in some cases, superlinear algorithms can do better than linear ones, the latter also including message-passing algorithms whose running times typically scale like $O(N \log(N))$ [4–6,59]. The results in the present work strongly suggest that the optimal algorithmic performance requires higher-order time scaling (e.g., quadratic or cubic in N) in random optimization problems. It would be very interesting to extend the present study to the determination of thresholds for MC algorithms run at different timescales in solving inference problems.

We hope that our work can help in building a new general framework to explain these results. We point out that the recent OGP theory for algorithmic hardness [31] is not enough, because MC algorithms running in superlinear time have not been proved to be stable algorithms, and thus their performances are not ruled by the OGP.

ACKNOWLEDGMENTS

This research has received financial support from the “National Centre for HPC, Big Data and Quantum Computing–HPC,” Project No. CN_00000013, CUP B83C22002940006, NRP Mission 4 Component 2 Investment 1.4, Funded by the European Union–NextGenerationEU, and by PRIN 2022 PNRR, Project No. P20229PBZR, CUP B53D23028410001, Mission 4, Component C2, Investment 1.1, Funded by the European Union–NextGenerationEU. The numerical simulations were conducted using the DARIAH HPC-AI cluster at CNR-NANOTEC in Lecce, funded by the “MUR PON Ricerca e Innovazione 2014-2020” project, code PIR01_00022 and H2IOSC Project–Humanities and cultural Heritage Italian Open Science Cloud funded by the European Union–NextGenerationEU–NRRP M4C2, Project No. IR0000029.

DATA AVAILABILITY

The data that support the findings of this article are openly available [60].

APPENDIX A: ESTIMATING ALGORITHMIC THRESHOLDS IN THE (QUASI)LINEAR TIME REGIME

Here we first provide evidence that studying the performance of SA in the linear time regime, i.e., when SA is run for a *fixed* number of MCSs, is complicated and does not provide a clear estimate for the algorithmic threshold. Then we show that a *greedy* version of the MC sampling, where the temperature is set to zero and the algorithm can never increase the energy, can still find solutions in almost linear time.

In Fig. 7 we plot the probability that SA finds a solution after being run for a fixed number of MCSs, that is, for a time growing linearly with the problem size N . We notice that both for $K = 3$ (upper panel) and $K = 4$ (lower panel) the curves move leftward when the problem size is increased. The explanation for this observation is straightforward: The same number of MCSs may be enough to

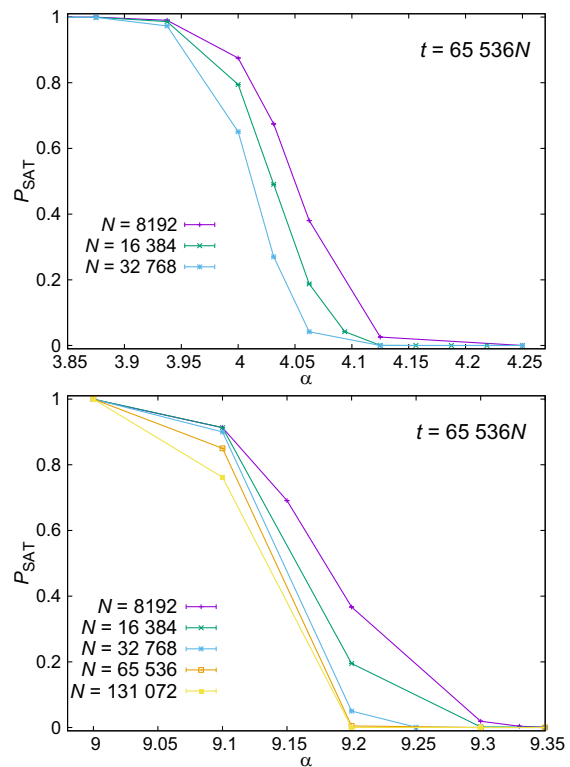


FIG. 7. Performances of simulated annealing in solving random K -SAT problems in linear time. The probability that SA finds a solution in a fixed (2^{16}) number of MCSs is plotted as a function of α for $K = 3$ (upper panel) and $K = 4$ (lower panel). The curves move to the left by increasing system size and have no crossing at all, thus making it hard to estimate the corresponding algorithmic threshold.

solve a smaller problem, but may not suffice to solve a larger problem. As a consequence, the data plotted in Fig. 7 show no crossing at all and just provide *upper bounds* to the actual algorithmic threshold in the linear regime. A more precise estimate for such a threshold would require uncontrolled extrapolations (there is no theory for these finite-size corrections) and would eventually provide an estimate with a large systematic error.

This is the reason why we started exploring superlinear regimes in SA solving random K -SAT. We discovered that clear algorithmic thresholds for SA can be obtained only in these regimes (and this is the main point of our work).

1. Performances of a quasilinear greedy algorithm

Let us consider now the limit $T \rightarrow 0$, or equivalently $\beta \rightarrow \infty$, in Eq. (1). The resulting Metropolis algorithm will evolve the configuration without ever increasing the system energy. This greedy algorithm is known as zero-temperature Monte Carlo (ZTMC) algorithm, and it also has a special role in the interplay between statistical physics and combinatorial optimization [9,61,62].

For several values of α and N with both $K = 3$ and $K = 4$, we have run the ZTMC algorithm until convergence to a solution (since we restrict ourselves to the SAT phase, this will happen sooner or later). The results have been averaged over 10^4 problems for each value of α and N , but for the largest size ($N = 819200$), in which case, only 500 problems were studied.

In Fig. 8, we report some results obtained by running ZTMC at a reasonably large value of α (a few percent below the estimated algorithmic threshold α_G) for both $K = 3$ (upper panel) and $K = 4$ (lower panel). In the main panels and for a fixed value of α , one can see that the cumulative distribution $F(\tau_s)$ of the times to reach a solution shifts to the right when the system size is increased. Both the median and the mean time to reach a solution scale with a small power of $\ln(N)$. Since the time unit we use is a Monte Carlo sweep, which already scales linearly with N , the average solution time $\bar{\tau}_s$ for the ZTMC grows slightly faster than linearly in N .

We observe that, also in this case, the estimation of the algorithmic threshold requires some extrapolation, given that the curves in Fig. 8 move right without any crossing. So, to estimate the algorithmic threshold, we assume that the mean time to solution follows the law $\bar{\tau}_s \sim (\ln N)^\delta (\alpha - \alpha_G)^{-\epsilon}$ with fixed exponents δ and ϵ for every K value. In the insets of Fig. 8, we plot $\bar{\tau}_s$ scaled by $(\ln N)^\delta$ such that, for every value of α , data at different values of N perfectly collapse. The collapsed data can be further scaled (by raising it to the power $1/\epsilon$) such that they align well and can be safely interpolated by a linear fit. We estimate the algorithmic threshold for the ZTMC algorithm by the point where the linear fit extrapolates to zero. The best estimates are $\alpha_G \simeq 4.04$ using $\delta = 2.5$ and $\epsilon = 2.9$ for $K = 3$, and $\alpha_G \simeq 8.83$ using $\delta = 1.5$ and $\epsilon = 2.7$ for $K = 4$.

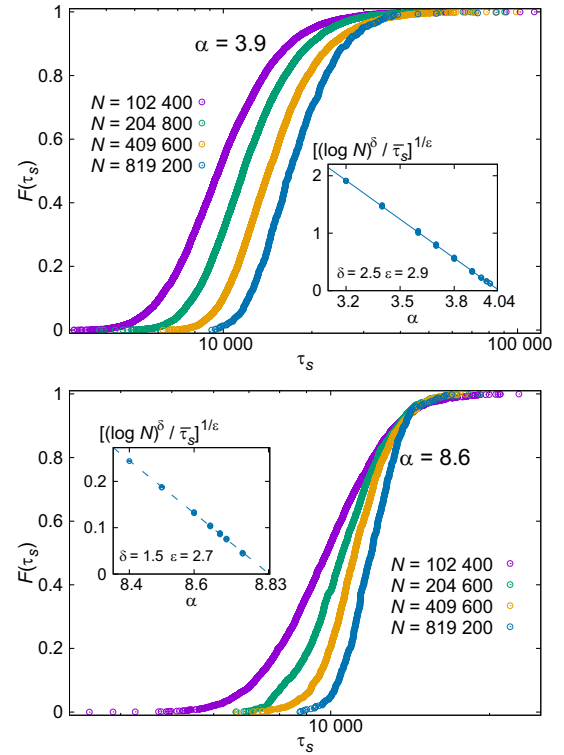


FIG. 8. Algorithmic threshold for the greedy zero-temperature Monte Carlo algorithm in solving random K -SAT problems with $K = 3$ (top panel) and $K = 4$ (bottom panel). The main panels show the cumulative distribution of the time τ_s needed to reach a solution. In both cases, the α value has been chosen to be a few percent below the algorithmic threshold. By increasing the problem size, at a fixed α value, the mean time $\bar{\tau}_s$ grows like $(\ln N)^\delta$. In the insets, we report the fits estimating the algorithmic threshold α_G , assuming the mean value follows the law $\bar{\tau}_s \sim (\ln N)^\delta (\alpha - \alpha_G)^{-\epsilon}$. Best-fit parameters are $\alpha_G \simeq 4.04$ with $\delta = 2.5$ and $\epsilon = 2.9$ for $K = 3$, and $\alpha_G \simeq 8.83$ with $\delta = 1.5$ and $\epsilon = 2.7$ for $K = 4$.

Now, where does the $(\ln N)^\delta$ factor come from in an algorithm without any thermal noise? If the ZTMC were always taking steps to decrease the energy, it would not take more than linear time to find a local energy minimum. Indeed, the initial energy of a random configuration is $2^{-K}\alpha N$. If the ZTMC algorithm would always decrease the energy at every step, it would reach the energy minimum in linear time. However, the ZTMC algorithm can also move between configurations with the same energy. When the algorithm reaches a region with many flat directions, it may take some time before finding a good direction to decrease further the energy. We numerically find that this time scales like $(\ln N)^\delta$. So, even with a greedy algorithm, it is not possible to find a solution in linear time because of the existence of entropic barriers (flat regions of the energy landscape).

Although the ZTMC algorithmic threshold for $K = 4$ is below α_d and well below the best results of SA, it is surprising that for $K = 3$ this greedy algorithm goes well

beyond α_d and its threshold is comparable, within the error, with that for the SA algorithm run with quadratic time scaling. The different behavior between $K = 3$ and $K = 4$ could be due to the different type of clustering transition in the two cases, continuous for $K = 3$ and discontinuous for $K = 4$ [24].

APPENDIX B: ROBUSTNESS OF ALGORITHMIC THRESHOLDS OBTAINED EXTRAPOLATING THE CROSSING POINTS

The ideal scenario to determine the algorithmic thresholds from a finite-size analysis is the one in Fig. 1, with one single crossing point $(\alpha_{\text{cross}}, U_f^*)$ for all system sizes. We can simply report $\alpha_{\text{alg}} = \alpha_{\text{cross}}$. On the other hand, the bottom panel of Fig. 2 shows a more delicate case where the crossings $[\alpha_{\text{cross}}(N, 2N), U_f^*(N, 2N)]$ depend on the size N . To estimate the algorithmic threshold in the large N from this data, we assume the usual correction to scaling $\alpha_{\text{cross}}(N, 2N) = \alpha_{\text{alg}} - cN^{-\omega}$, with $c > 0$ and $\omega > 0$, and extrapolate to get the limit $\lim_{N \rightarrow \infty} \alpha_{\text{cross}}(N, 2N) = \alpha_{\text{alg}}$.

Figure 3 illustrates two cases in which we determine the algorithmic threshold by extrapolation. This is done for the SA algorithm applied to the 4-SAT problem with cubic and quartic time scalings, where, for computational reasons, we are limited to using very small coefficients A for the time scalings $t(N) = AN^2$. However, the extrapolated thresholds are consistent with the independent estimate given in the central panel of Fig. 5.

To corroborate the validity of the method based on the extrapolation of the crossing points and help the reader understand how to use it in other problems, we apply this method to a more controlled scenario: SA solving the 3-SAT problem with quadratic scaling. Figure 1 gives us, without extrapolation, the thresholds $\alpha_{\text{SA}}^{N^2} = 4.031(2)$ for $A = 2$ and $\alpha_{\text{SA}}^{N^2} = 4.045(5)$ for $A = 8$. Now, we set up a much smaller coefficient $A = 2^{-5}$, which provokes stronger finite-size effects and makes extrapolation necessary. Here, we show that the new $\alpha_{\text{SA}}^{N^2}$ is consistent with the other thresholds and highlight some practical aspects of the method.

Figure 9 is analogous to Fig. 3 in the main text. We run SA for different system sizes $N = 2^k$, with $k = 10, \dots, 20$, in each case for a time $t(N)$ (measured in MCSs) assigned by the aforementioned quadratic scaling. We can obtain the algorithmic threshold α_{SA} by fitting the crossing points of the curves for the extensive energies U_f to the law $\alpha_{\text{cross}}(N, 2N) = \alpha_{\text{SA}} - cN^{-\omega}$. However, the selected coefficient $A = 2^{-5}$ causes SA to spend too few iterations in a small system [for example, $N = 1024$ will get as few as $t(N) = 32$ MCSs]. It is thus natural that we have to discard the crossings $\alpha_{\text{cross}}(N, 2N)$ that are produced running SA for too short times $t(N)$.

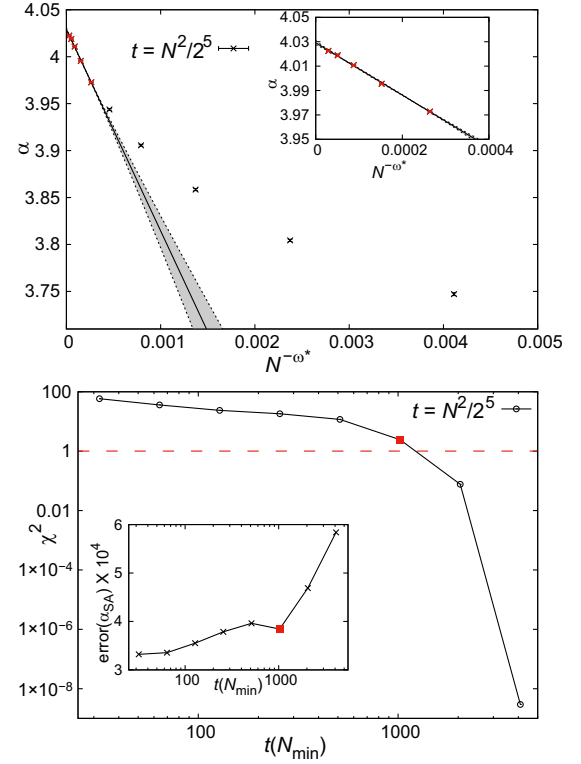


FIG. 9. Extrapolation of the crossing points to get the algorithmic threshold for quadratic scaling with small coefficient $A = 2^{-5}$ and $K = 3$. In the top panel, each point corresponds to the crossing $\alpha_{\text{cross}}(N, 2N)$ between the curves $U_f(\alpha)$ for sizes N and $2N$. The data include the sizes $N = 2^k$, with $k = 10, \dots, 20$. Continuous lines are linear fits of the type $\alpha_{\text{cross}}(N, 2N) = \alpha_{\text{SA}} - cN^{-\omega}$ made with the optimal exponent ω^* that minimizes the quality of the fit $\chi^2(\omega)$. The fit is done with the points marked in red. We set a minimum number of iterations $t(N) = 1024$ and discard the data obtained by running SA for fewer iterations. The shaded regions show the uncertainty on the fit considering statistically acceptable values of ω around ω^* . The inset is an enlargement of the red points. Final results are $\omega^* = 0.7638$ and $\alpha_{\text{SA}}^{N^2} = 4.0292(18)$ (for $A = 2^{-5}$). In the bottom panel, each point corresponds to a different fit, done by setting a different minimum number of iterations $t(N)$. The main graphic shows the χ^2 of the fits, which cross the horizontal line at 1 when we pass $t(N) = 1024$ (marked with a red square). In the inset, the corresponding errors on the estimated algorithmic threshold grow rapidly for times longer than $t(N) = 1024$ (marked with a red square).

The best fit is obtained considering only the data from $t(N) \geq 1024$. In the upper panel of Fig. 9, the points used in the fit are marked in red, while the discarded points are marked in black. The result is $\alpha_{\text{SA}} = 4.0292(18)$, a threshold that is smaller and compatible with those for larger coefficients $A = 2$ and $A = 8$, presented in Fig. 1.

To select the minimum time $t(N_{\min})$, we performed the fits with all possible N_{\min} values. The main graphic in the bottom panel of Fig. 9 shows that the χ^2 of the fit $\alpha_{\text{cross}}(N, 2N) = \alpha_{\text{SA}} - cN^{-\omega}$ is large when $t(N_{\min})$ is small.

When we remove points, the value of χ^2 decreases because the points are closer to each other, and it is easier to find a value of the exponent ω such that they lie in a straight line. But having fewer points for the fit means, in general, more uncertainty in the determination of the algorithmic threshold, as can be checked in the inset. We need to balance this trade-off by selecting an acceptable value of χ^2 to be reached. After setting $\chi^2 = 1$ as our goal, we determine that the best fit is the one marked with a red square both in the main panel and the inset in the bottom panel of Fig. 9. The best minimum time is $t(N_{\min}) = 1024$. In effect, taking $t(N_{\min})$ shorter than that, χ^2 becomes too large. Taking $t(N_{\min})$ longer than that, the error in the algorithmic threshold considerably increases.

The criterion of choosing a minimum time for the data to be included in the extrapolation should not be restricted to this specific case. It is particularly important for our SA, for which a short number of iterations means taking large steps in temperature, driving the algorithm strongly out of equilibrium even for high temperatures. We should remark that both extrapolations in Fig. 3 are done considering also $t(N_{\min}) = 1024$. The reader should note that a different time scaling might reach the same $t(N_{\min})$ for a different value of N_{\min} . For example, for the cubic and quartic

scalings in Fig. 1 we have $N_{\min} = 8192$, while for Fig. 9 we used $N_{\min} = 32768$. In all cases, however, the fits achieve reasonable values of the χ^2 (not much bigger than one) and the corresponding errors in the estimation of the algorithmic threshold are all in the third or fourth significant digit. This reinforces the idea that $t(N_{\min}) = 1024$ is a proper choice across different time scalings.

APPENDIX C: ALGORITHMIC THRESHOLDS OF FOCUSED METROPOLIS SEARCH

Focused Metropolis search is a well-known efficient algorithm introduced in Ref. [19] to solve K -SAT instances. It has been shown [19] to perform on par with the best versions of the famous WalkSAT [63–65].

Each iteration of FMS starts by selecting an unsatisfied clause in the Boolean formula, uniformly at random among all unsatisfied clauses. Then, the algorithm chooses a variable x_i inside that clause also uniformly at random and computes the change ΔU in the energy that would be provoked by flipping x_i . If $\Delta U \leq 0$, the flip is always accepted; if $\Delta U > 0$, the flip is accepted with probability $\eta^{\Delta U}$. Here, $\eta \in [0, 1]$ is an algorithmic parameter interpreted as the level of noise in the dynamics. The reader should

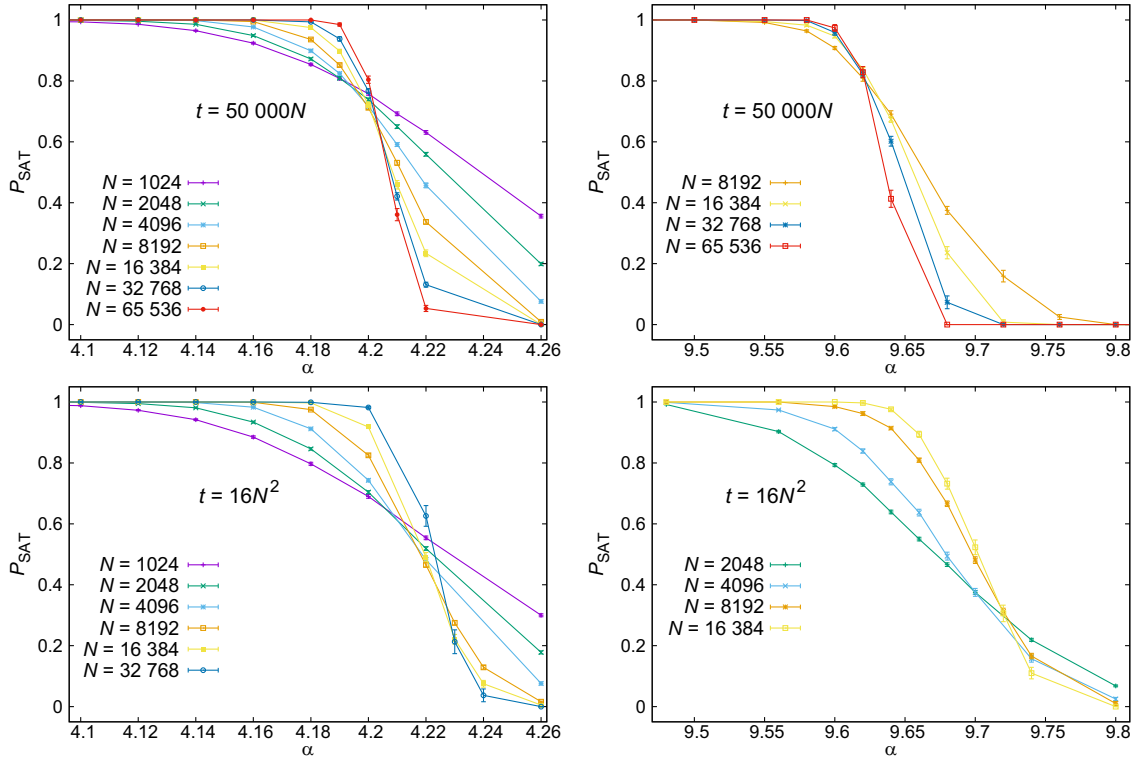


FIG. 10. Focused Metropolis search with linear and quadratic time scalings in random K -SAT. For several system sizes N , we plot the probability P_{SAT} to find a solution to random K -SAT instances ($K = 3$ in left-hand panels and $K = 4$ in right-hand panels) by running FMS for a time scaling linearly (upper panel) and quadratically (lower panel) with N . We used the algorithmic parameter $\eta = 0.37$ for $K = 3$ and $\eta = 0.293$ for $K = 4$. For the linear scaling, the algorithmic thresholds are $\alpha_{\text{FMS}}^N = 4.205(5)$ for $K = 3$ and $\alpha_{\text{FMS}}^N = 9.615(5)$ for $K = 4$.

note that, once the variable x_i is selected, FMS follows the Metropolis rule with $\eta = e^{-\beta}$ [see Eq. (1)].

Remarkably, the numerical results presented in Refs. [1,19] indicate that FMS is able to solve random K -SAT instances in linear time even beyond the condensation transition α_c . This is achieved once the algorithmic parameter η is properly tuned to its optimal values. We chose $\eta = 0.37$ and $\eta = 0.293$, for $K = 3$ and $K = 4$, respectively, which are close or exactly equal to the optimal parameters reported in Refs. [1,19].

These two works also study the divergence of the time to solution, which could allow the authors to locate the algorithmic threshold. The authors of Ref. [19] estimate the value $\alpha_{\max} = 4.24$ as the limit of the linear regime for $K = 3$. On the other hand, the data presented in Ref. [1] would suggest a divergence between $\alpha = 9.6$ and $\alpha = 9.65$ for $K = 4$. However, methods based on extrapolations may be quite noisy. In these contributions, the larger values of α where FMS is reliably shown to find the solution in linear time are $\alpha = 4.2$ for $K = 3$ and $\alpha = 9.6$ for $K = 4$.

Here, we estimate FMS thresholds using the more accurate method introduced in the main text. The upper panels of Fig. 10 show the results with a linear time scaling. For different system sizes, N , we measured the probability that FMS finds a solution in no more than 5×10^4 MCSs. The curves have nice crossings at $\alpha_{\text{FMS}}^N = 4.205(5)$ for $K = 3$ and $\alpha_{\text{FMS}}^N = 9.615(5)$ for $K = 4$. Note that FMS can solve random 3-SAT in linear time in almost all the SAT phase ($\alpha < \alpha_s = 4.267$), outperforming SA with super-linear time scalings. For $K = 4$, on the other hand, running FMS in linear time achieves a performance similar to that of running SA in superlinear time. Both estimates are compatible with the mentioned data in Refs. [1,19].

The quadratic scaling, $t = 16N^2$, shown in the bottom panels of Fig. 10, appreciably improves the performance obtained with a linear scaling. For $K = 4$ and $N = 16384$, FMS solves almost half of the instances with $\alpha = 9.7$, an outstanding result for a local search algorithm in random K -SAT. Determining the corresponding algorithmic thresholds is more difficult in this case because of strong finite-size effects.

The results in this appendix illustrate the generality of our method to estimate the algorithmic thresholds in hard combinatorial optimization problems. The more efficient FMS algorithm displays the same phenomenology discussed in the main text for the SA algorithm, with algorithmic thresholds depending on the time scaling.

-
- [1] M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, and S. Seitz, *Circumspect descent prevails in solving random constraint satisfaction problems*, *Proc. Natl. Acad. Sci. U.S.A.* **105**, 15253 (2008).
 [2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*, *Science* **220**, 671 (1983).

- [3] S. Boettcher and A. G. Percus, *Extremal optimization at the phase transition of the three-coloring problem*, *Phys. Rev. E* **69**, 066703 (2004).
 [4] M. C. Angelini, *Parallel tempering for the planted clique problem*, *J. Stat. Mech.* (2018) 073404.
 [5] M. J. Song, I. Zadik, and J. Bruna, *On the cryptographic hardness of learning single periodic neurons*, *Adv. Neural Inf. Process. Syst.* **34**, 29602 (2021).
 [6] I. Zadik, M. J. Song, A. S. Wein, and J. Bruna, *Lattice-based methods surpass sum-of-squares in clustering*, in *Conference on Learning Theory* (PMLR, London, 2022), pp. 1247–1248.
 [7] V. Erba, F. Behrens, F. Krzakala, and L. Zdeborová, *Quenches in the Sherrington–Kirkpatrick model*, *J. Stat. Mech.* (2024) 083302.
 [8] S. Cook, *The complexity of theorem-proving procedures*, in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing* (ACM, Shaker Heights, OH, 1971), p. 151.
 [9] M. Mezard, G. Parisi, and R. Zecchina, *Analytic and algorithmic solution of random satisfiability problems*, *Science* **297**, 812 (2002).
 [10] E. Friedgut and J. Bourgain, *Sharp thresholds of graph properties, and the k -sat problem*, *J. Am. Math. Soc.* **12**, 1017 (1999).
 [11] J. Ding, A. Sly, and N. Sun, *Proof of the satisfiability conjecture for large k* , in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, Portland, Oregon, USA* (Association for Computing Machinery (ACM), New York, NY, 2015), pp. 59–68.
 [12] M. Mézard and R. Zecchina, *Random k -satisfiability problem: From an analytic solution to an efficient algorithm*, *Phys. Rev. E* **66**, 056126 (2002).
 [13] M. Mézard, T. Mora, and R. Zecchina, *Clustering of solutions in the random satisfiability problem*, *Phys. Rev. Lett.* **94**, 197205 (2005).
 [14] S. Mertens, M. Mézard, and R. Zecchina, *Threshold values of random k -sat from the cavity method*, *Random Struct. Algorithms* **28**, 340 (2006).
 [15] D. Achlioptas and F. Ricci-Tersenghi, *On the solution-space geometry of random constraint satisfaction problems*, in *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, Seattle, WA, USA* (Association for Computing Machinery (ACM), New York, NY, 2006), pp. 130–139.
 [16] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, *Gibbs states and the set of solutions of random constraint satisfaction problems*, *Proc. Natl. Acad. Sci. U.S.A.* **104**, 10318 (2007).
 [17] D. Gamarnik and Q. Li, *Finding a large submatrix of a Gaussian random matrix*, *Ann. Stat.* **46**, 2511 (2018), <http://hdl.handle.net/1721.1/120593>.
 [18] A. Coja-Oghlan, *A better algorithm for random k -sat*, *SIAM J. Comput.* **39**, 2823 (2010).
 [19] S. Seitz, M. Alava, and P. Orponen, *Focused local search for random 3-satisfiability*, *J. Stat. Mech.* (2005) P06006.
 [20] R. Marino, G. Parisi, and F. Ricci-Tersenghi, *The backtracking survey propagation algorithm for solving random K -SAT problems*, *Nat. Commun.* **7**, 12996 (2016).

- [21] L. Budzynski, F. Ricci-Tersenghi, and G. Semerjian, *Biased landscapes for random constraint satisfaction problems*, *J. Stat. Mech.* (2019) 023302.
- [22] M. C. Angelini and F. Ricci-Tersenghi, *Monte Carlo algorithms are very effective in finding the largest independent set in sparse random graphs*, *Phys. Rev. E* **100**, 013302 (2019).
- [23] M. C. Angelini and F. Ricci-Tersenghi, *Limits and performances of algorithms based on simulated annealing in solving sparse hard inference problems*, *Phys. Rev. X* **13**, 021011 (2023).
- [24] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian, *Clusters of solutions and replica symmetry breaking in random k -satisfiability*, *J. Stat. Mech.* (2008) P04004.
- [25] F. Ricci-Tersenghi and G. Semerjian, *On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms*, *J. Stat. Mech.* (2009) P09001.
- [26] R. Lemoy, M. Alava, and E. Aurell, *Local search methods based on variable focusing for random k -satisfiability*, *Phys. Rev. E* **91**, 013305 (2015).
- [27] A. Giorgio Cavaliere and F. Ricci-Tersenghi, *Biased thermodynamics can explain the behaviour of smart optimization algorithms that work above the dynamical threshold*, *J. Stat. Mech.* (2025) 053405.
- [28] L. Zdeborová and F. Krzakala, *Generalization of the cavity method for adiabatic evolution of Gibbs states*, *Phys. Rev. B* **81**, 224205 (2010).
- [29] F. Krzakala and L. Zdeborová, *Performance of simulated annealing in p -spin glasses*, *J. Phys. Conf. Ser.* **473**, 012022 (2013).
- [30] J. Ardelius and L. Zdeborová, *Exhaustive enumeration unveils clustering and freezing in the random 3-satisfiability problem*, *Phys. Rev. E* **78**, 040101 (2008).
- [31] D. Gamarnik, *The overlap gap property: A topological barrier to optimizing over random structures*, *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2108492118 (2021).
- [32] D. Gamarnik and M. Sudan, *Limits of local algorithms over sparse random graphs*, *Ann. Prob.* **45**, 2353 (2017).
- [33] S. B. Hopkins and D. Steurer, *Efficient Bayesian estimation from few samples: Community detection and related problems*, in *Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Berkeley, CA, 2017), pp. 379–390.
- [34] S. B. Hopkins, P. K. Kothari, A. Potechin, P. Raghavendra, T. Schramm, and D. Steurer, *The power of sum-of-squares for detecting hidden structures*, in *Proceedings of the 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Berkeley, CA, 2017), pp. 720–731.
- [35] S. Hopkins, *Statistical inference and the sum of squares method*, Ph. D. thesis, Cornell University, 2018.
- [36] L. Zdeborová and F. Krzakala, *Statistical physics of inference: Thresholds and algorithms*, *Adv. Phys.* **65**, 453 (2016).
- [37] S. Li and T. Schramm, *Some easy optimization problems have the overlap-gap property*, [arXiv:2411.01836](https://arxiv.org/abs/2411.01836).
- [38] F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, *Simplest random k -satisfiability problem*, *Phys. Rev. E* **63**, 026702 (2001).
- [39] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, *Two solutions to diluted p -spin models and XORSAT problems*, *J. Stat. Phys.* **111**, 505 (2003).
- [40] F. Ricci-Tersenghi, *Being glassy without being hard to solve*, *Science* **330**, 1639 (2010).
- [41] D. Gamarnik, A. Jagannath, and A. S. Wein, *Low-degree hardness of random optimization problems*, in *Proceedings of the 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Durham, NC, 2020) pp. 131–140.
- [42] A. Braunstein, L. Dall’Asta, G. Semerjian, and L. Zdeborová, *The large deviations of the whitening process in random constraint satisfaction problems*, *J. Stat. Mech.* (2016) 053401.
- [43] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, *J. Chem. Phys.* **21**, 1087 (1953).
- [44] R. Eglese, *Simulated annealing: A tool for operational research*, *Eur. J. Oper. Res.* **46**, 271 (1990).
- [45] T. Guilmeau, E. Chouzenoux, and V. Elvira, *Simulated annealing: A review and a new scheme*, in *Proceedings of the 2021 IEEE Statistical Signal Processing Workshop (SSP), Rio de Janeiro, Brazil* (Institute of Electrical and Electronics Engineers (IEEE), 2021), pp. 101–105.
- [46] A. G. Cavaliere, T. Lesieur, and F. Ricci-Tersenghi, *Optimization of the dynamic transition in the continuous coloring problem*, *J. Stat. Mech.* (2021) 113302.
- [47] How quickly may depend on the nature of the algorithm, but most often in a time $O(N)$ or $O(N \log(N))$.
- [48] W. Krauth and M. Mézard, *Storage capacity of memory networks with binary couplings*, *J. Phys.* **50**, 3057 (1989).
- [49] H. Huang and Y. Kabashima, *Origin of the computational hardness for learning with binary synapses*, *Phys. Rev. E* **90**, 052813 (2014).
- [50] A. Braunstein and R. Zecchina, *Learning by message passing in networks of discrete synapses*, *Phys. Rev. Lett.* **96**, 030201 (2006).
- [51] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, *Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses*, *Phys. Rev. Lett.* **115**, 128101 (2015).
- [52] C. Baldassi, C. Borgs, J.T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, *Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes*, *Proc. Natl. Acad. Sci. U.S.A.* **113**, E7655 (2016).
- [53] C. Baldassi, C. Lauditi, E. M. Malatesta, G. Perugini, and R. Zecchina, *Unveiling the structure of wide flat minima in neural networks*, *Phys. Rev. Lett.* **127**, 278301 (2021).
- [54] C. Baldassi, E. M. Malatesta, G. Perugini, and R. Zecchina, *Typical and atypical solutions in nonconvex neural networks with discrete and continuous weights*, *Phys. Rev. E* **108**, 024310 (2023).
- [55] G. Folena, S. Franz, and F. Ricci-Tersenghi, *Rethinking mean-field glassy dynamics and its relation with the energy landscape: The surprising case of the spherical mixed p -spin model*, *Phys. Rev. X* **10**, 031045 (2020).
- [56] M. C. Angelini, P. Fachin, and S. de Feo, *Mismatching as a tool to enhance algorithmic performances of Monte Carlo*

- methods for the planted clique model*, *J. Stat. Mech.* (2021) 113406.
- [57] C. Snell, J. Lee, K. Xu, and A. Kumar, *Scaling LLM test-time compute optimally can be more effective than scaling model parameters*, [arXiv:2408.03314](https://arxiv.org/abs/2408.03314).
- [58] B. Brown, J. Juravsky, R. Ehrlich, R. Clark, Q. V. Le, C. Ré, and A. Mirhoseini, *Large language monkeys: Scaling inference compute with repeated sampling*, [arXiv:2407.21787](https://arxiv.org/abs/2407.21787).
- [59] A. S. Wein, A. El Alaoui, and C. Moore, *The Kikuchi hierarchy and tensor PCA*, in *Proceedings of the 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Baltimore, MD, 2019), pp. 1446–1468.
- [60] M. C. Angelini, M. Avila-González, F. D’Amico, D. Machado, R. Mulet, and F. Ricci-Tersenghi, *tScalingAlg-Thresholds*, Version number: 2.0.0 (2025), <https://github.com/d4v1d-cub/tScalingAlgThresholds.git>.
- [61] M. Mezard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).
- [62] M. Mezard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, New York, 2009).
- [63] C. Papadimitriou, *On selecting a satisfying truth assignment*, in *Proceedings of the 32nd IEEE Symposium on the Foundations of Computer Science* (IEEE Computer Society, New York, 1991), Vol. FOCS-91, pp. 163–169.
- [64] B. Selman and H. Kautz, *Domain-independent extensions to GSAT: Solving large structured satisfiability problems.*, in *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambéry, France* (International Joint Conferences on Artificial Intelligence (IJCAI), 1993), Vol. 1, pp. 290–295.
- [65] B. Selman, H. Kautz, and B. Cohen, *Local search strategies for satisfiability testing*, in *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, edited by D. S. Johnson and M. A. Trick (American Mathematical Society (AMS), Piscataway, NJ, 1996), Vol. 26, pp. 521–532.