Numerical solution-space analysis of satisfiability problems

Alexander Mann*
II. Institute of Physics, University of Göttingen, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany

A. K. Hartmann†
Institute of Physics, University of Oldenburg, Carl von Ossietzky Straße 9-11, 26129 Oldenburg, Germany
(Received 24 April 2010; revised manuscript received 6 September 2010; published 3 November 2010)

The solution-space structure of the three-satisfiability problem (3-SAT) is studied as a function of the control parameter α (ratio of the number of clauses to the number of variables) using numerical simulations. For this purpose one has to sample the solution space with uniform weight. It is shown here that standard stochastic local-search (SLS) algorithms like average satisfiability (ASAT) exhibit a sampling bias, as does "Metropolis-coupled Markov chain Monte Carlo" (MCMCMC) (also known as "parallel tempering") when run for feasible times. Nevertheless, unbiased samples of solutions can be obtained using the "ballistic-networking approach," which is introduced here. It is a generalization of "ballistic search" methods and yields also a cluster structure of the solution space. As application, solutions of 3-SAT instances are generated using ASAT plus ballistic networking. The numerical results are compatible with a previous analytical prediction of a simple solution-space structure for small values of α and a transition to a clustered phase at $\alpha_c \approx 3.86$, where the solution space breaks up into several non-negligible clusters. Furthermore, in the thermodynamic limit there are, even for $\alpha=4.25$ close to the SAT-UNSAT transition $\alpha_s \approx 4.267$, always clusters without any frozen variables. This may explain why some SLS algorithms are able to solve very large 3-SAT instances close to the SAT-UNSAT transition.

DOI: 10.1103/PhysRevE.82.056702 PACS number(s): 05.10.−a, 89.75.Fb, 89.20.Ff, 75.40.Mg

I. INTRODUCTION

The application of notions, analytical approaches, and numerical algorithms from statistical mechanics has lead to a better understanding [1–3] of NP-hard (NP: nondeterministic polynomial) optimization problems [4,5]. One main underlying question is why these optimization problems are computationally hard, which means no fast algorithms are available, where the running times increase only polynomially with the problem size. The progress of gaining insight into this phenomenon has been considerable in particular for the *typical-case complexity*, where ensembles of random instances are studied as a function of control parameters. These ensembles often exhibit phase transitions where changes in the effective "hardness" of the problem can be observed. Often, these transitions are connected to changes in the structure of the solution space, comparable to energy landscapes in physics. In particular, one is interested in the question of how the change in the solution-space structures influences the performance of exact and stochastic algorithms. For example, for the vertex-cover problem, which is defined on graphs, a clustering transition has been found analytically [6] and numerically [7,8] when increasing the edge density of Erdős-Rényi random graphs. This transition coincides with a change of the typical-case complexity from polynomial to exponential [9]. For other optimization problems, the situation is less clear, as for the *satisfiability (SAT) problem*, which we study in this work.

As we will explain, exact enumeration of solutions works well in one region of the phase diagram, close to the SAT-UNSAT phase transition (see below), whereas Monte Carlo (MC) approaches perform well in the opposite part of the phase diagram, away from the SAT-UNSAT transitions. Unfortunately, the clustering transition is located right between these extreme parts, hence numerically difficult to study. We use a stochastic algorithm in combination with a correction of the sampling bias introduced by the stochastic algorithm to study the clustering phenomena.

The outline of the paper is as follows. In the second section, we give the necessary background on SAT and on clustering of solution landscapes. In the third section, we briefly explain the algorithms we use to sample solutions and show that they exhibit a bias. Next, we introduce ballistic networking and related methods, which we use to correct for the bias. In Sec. V, we show the results we have obtained for random three-satisfiability problem (3-SAT). Finally, we provide a conclusion and an outlook.

II. BACKGROUND

A. Satisfiability

Satisfiability is one of the fundamental problems of computer science and has attracted a lot of attention over the past years, also by physicists, due to its similarity to spin-glass problems. It is the first problem proven to belong to the class of NP-complete problems [10], a class of problems for which no algorithm has been found yet that exhibits a polynomial worst-case running time as a function of the problem size. Therefore, it is still a challenge to find algorithms which perform well on typical instances and to understand the underlying structure of the solution space which may hinder the performance of algorithms.

*amann@uni-goettingen.de
†a.hartmann@uni-oldenburg.de

Satisfiability belongs to the class of constraint satisfaction problems [11]: given N Boolean variables $x_i \in \{0,1\}$ and a Boolean formula F describing a set of constraints, each of which forbids a certain assignment of values to some of the variables, one is to decide whether F can be satisfied, i.e., whether there is an assignment $\vec{x}=(x_1,\ldots,x_N)$ such that all constraints are fulfilled simultaneously. In the K-SAT formulation, F is given in conjunctive normal form,

$$F = \bigwedge_{m=1}^{M} (l_1^m \vee l_2^m \vee \cdots \vee l_K^m),$$

which describes a logical conjunction of M constraints (clauses) C_m each containing a disjunction of K literals l_k^m $(m=1,\ldots,M;\ k=1,\ldots,K)$, which are either a variable x_i or a logically negated variable $\overline{x_i}$.

A certain assignment of values to the variables is called a configuration in the following. If a configuration satisfies all clauses in F, it is called a solution. In the random K-SAT ensemble each clause is chosen randomly and uniformly among the $2^K\binom{N}{K}$ possible combinations in which no variable appears twice.

One defines a control parameter $\alpha=M/N$ which is the number of clauses M divided by the number of variables N. For low α the problem is typically satisfiable, whereas for high values of α there typically is no solution [12,13]. The existence of such a threshold α_s between a satisfiable and an unsatisfiable phase for $N\to\infty$ has been proven [14] for all K, but this still allows for α_s not converging to a certain value. While the position of the threshold for $K=2$ is known exactly [15], for larger K there are only numerical estimates. In this paper we will stick to the case of $K=3$, where every clause contains exactly three literals. The satisfiability transition is located in this case at $\alpha_s(K=3)=4.267$ [16].

B. Cluster phenomena

In addition to the SAT-UNSAT transition, analytical calculations [17,18] give rise to evidence that there are further ("structural") phase transitions which refer to the formation of disconnected clusters of solutions for high values of the control parameter α in the satisfiable phase. Formally, clusters in constraint satisfaction problems can be defined as extremal Gibbs measures which give the following picture for satisfiability: for small values of α all solutions are contained in one connected component (cluster). When α grows, more and more solutions disappear, so that at some point the cluster decomposes into smaller clusters which initially, up to a threshold α_d, make up only an exponentially small fraction of all solutions, whereas above α_d many clusters contribute to the statistical behavior. Above a higher critical value α_c we enter another type of clustered phase which is dominated by a small number of large clusters. The case of 3-SAT is special, as here $\alpha_d=\alpha_c$, i.e., we directly enter the phase dominated by a few clusters. The position of the dynamical threshold to the clustered phase is predicted to be at $\alpha_c(K=3)\approx 3.86$ [19].

This value is compatible with recent numerical results [20], where the cluster structure was investigated using the detection of community structures. Unfortunately, the sampling was performed using an algorithm, which does not exhibit uniform sampling of the solutions (see below). Anyway, there is no general rule on how to translate the formal definition of clusters, which holds in the thermodynamic limit, to finite system sizes; hence, other approaches besides community structures are possible.

For numerical studies often a very appealing approach is used, where a cluster is defined as the connected components in a graph where each solution is represented by a vertex and edges connect solutions differing in only one variable. This definition of a cluster will be used in this work as well. For every two solutions belonging to the same cluster there is therefore a "path" of configurations which all solve the SAT instance at hand. Unfortunately, this path can be long and peppered with many dead ends or loops, which makes it very difficult to decide whether two configurations belong to the same cluster. The main problem when discussing clusters in high-dimensional discrete solution spaces like that of satisfiability is that one is tempted to think of clusters as bloblike, well-separated, and homogeneous structures in configuration phase like, e.g., nanoclusters formed by agglomeration of atoms. The clusters which occur in high-dimensional discrete solution spaces are yet of a completely different nature in that they are more like fragmented and interweaved structures with lots of dead ends, loops, and holes, which makes it difficult to speak of spatially separated clusters.

The existence of a clustered phase has been proven for $K\geq 8$ [17]. In the language of statistical physics this clustering corresponds to one-step replica-symmetry breaking (1-RSB) [21,22]. A further substructure in terms of another clustering of solutions taken from one cluster, giving a hierarchical structure of clusters, is suspected where 1-RSB becomes instable and higher steps of replica-symmetry breaking occur [23].

What makes cluster phenomena interesting from the algorithmic point of view is the question if (and if so in what way) clustering has an influence on the performance of local-search heuristics. Usually it is assumed that the existence of many clusters is an indication for a complicated "rugged" energy landscape, which then also gives rise to many local minima, hindering the performance of local-search heuristics [23]. Reference [24] argues that the assumption that clustering is directly responsible for the bad performance of local-search algorithms does not hold, but instead it is the size of the domains of attraction of different clusters. In the same way, but with a slightly different focus, Krząkała and coworkers [25,26] proposed that the appearance of locally frozen variables in clusters is responsible for the slowdown of heuristic algorithms close to the SAT-UNSAT threshold. A locally frozen variable is a variable which takes the same value over all solutions belonging to one cluster. A cluster containing at least one frozen variable is called frozen. One defines the freezing transition α_f as the smallest value of α above which all solutions belong to frozen clusters. Hard to solve problems can be designed by creating instances exhibiting a maximum number of frozen variables [27].

To clarify the influence of phase transitions on the average computational hardness, one can study the performance of stochastic algorithms as a function of the control parameter α. Of particular interest is the algorithm-dependent value of

α up to which an algorithm shows linear-time performance, and compare this to threshold values of α [28]. Studies of stochastic algorithms such as average satisfiability (ASAT) [28], WalkSat [29] and ChainSat [30] have shown however that those algorithms have linear behavior up to values considerably beyond the clustering transition. This suggests that the cluster transition has no impact on the performance of local-search algorithms, as long as the algorithms can access large regions of the configuration space. This is in particular the case for algorithms which also allow for local steps which increase the number of unsatisfied clauses with a suitably chosen probability. It is remarkable that ChainSat exhibits this behavior although it is greedy "in a weak sense" as it never allows steps which increase the number of unsatisfied clauses. Naively, one would therefore expect it to get trapped in local minima very easily. The authors of [30] interpreted this as evidence for the belief that true local minima are very rare in high-dimensional search spaces. These results could also indicate that indeed it is more the (non)existence of frozen clusters which is responsible for the performance of local-search algorithms.

For small instances there are always some frozen variables. Therefore in [31] a different notion of frozen clusters via the *whitening core* is used. There one looks, for each solution, iteratively for variables which can be flipped since they appear only in clauses satisfied by other variables or which contain variables already detected in the whitening core. The position of this freezing transition was then calculated by exact enumeration and clustering of all solutions for sufficiently small system sizes and is expected to lie at $\alpha_f(K=3)=4.254$ close to but below the satisfiability transition.

Furthermore, Ref. [32] finds a cluster condensation transition in the solutions generated by ASAT very close to α_s; again these results rely on a nonuniform sampling of the solutions. Anyway, these results are compatible with the observation of a good performance of local-search algorithms close to the threshold α_s.

C. Algorithmic treatment of SAT

Algorithms for SAT include a broad spectrum, both stochastic and exact, from simple and straightforward algorithms such as RandomWalksat [33] and WalkSat [34,35] to complex algorithms like Davis-Putnam-Logemann-Loveland (DPLL) [36] and message-passing algorithms such as belief propagation and survey propagation [22]. For small systems exact enumeration of all solutions is possible using one of the numerous standard algorithms [37,38] such as the aforementioned DPLL. It can be shown [39] that deterministic algorithms have longest average run times close to α_s, reflecting the difficulty of deciding whether a given SAT formula is satisfiable or not. The problem with exact enumeration is that it is limited to small systems due to hardware restrictions, especially because of the memory needed to store the huge number of configurations, as the number of solutions grows exponentially with the system size. Furthermore the number of solutions is not a continuous function when crossing the satisfiability threshold, but it drops from a finite value to zero. This corresponds to a nonzero entropy at the phase transition. The entropy per variable grows approximately linearly with decreasing α [40,41]. In turn this means that even very close to the satisfiability threshold the number of solution grows exponentially and quickly becomes so large that it is not feasible to enumerate all solutions even in this regime. From counting all solutions using DPLL for systems up to $N=144$ we can estimate the solution entropy per variable near the phase boundary at $\alpha=4.25$ to be roughly between 0.07 and 0.08 (averaging the logarithm only over satisfiable instances), which is in agreement with the 1-RSB prediction [23]. The average number of solutions turns out to be about 1.5×10^9 for $N=144$ and $\alpha=4.1$, i.e., already storing all solutions becomes very difficult due to memory constraints.

To overcome these limitations one can turn to stochastic algorithms which, starting at an arbitrary configuration, do successive changes either completely randomly or based on a heuristic evaluating information about the local configurational neighborhood. Stochastic algorithms are not guaranteed to find a solution, even if solutions do exist, but they may be able to find solutions significantly faster than deterministic algorithms. It is thus possible to obtain solutions for much larger systems, but on the other hand stochastic algorithms can never prove that there is no solution, i.e., tests for unsolvability can only be done by using deterministic algorithms.

In this paper we study the cluster structure numerically for $K=3$, which requires unbiased sampling of the solution space. Different types of sampling algorithms are studied and shown to be biased. We therefore present an algorithm that uses a different approach to create a survey of the cluster structure of satisfiability instances from which it is then possible to derive unbiased samples. It is an improvement on the ballistic search algorithm which has originally been applied to spin glasses [42–44]. The main advantage of the algorithm is that it is able to provide an overview of the cluster structure of the solution space without having to enumerate all solutions, which is no longer possible already for moderate numbers of variables.

III. SAMPLING ALGORITHMS

A. Bias in stochastic local-search algorithms

If stochastic local-search (SLS) algorithms found all solutions with the same probability, one could use them directly to probe the solution space. Unfortunately, this is not the case as has been shown earlier for WalkSAT [45]. Using a combination with simulated annealing, the sampling could be improved, but uniform sampling could not fully be restored. Later on in this paper we will use ASAT as solution generator, so we use it here for an exemplary presentation of the bias in SLS algorithms.

ASAT is a simplified variant of Focused Metropolis Search [29] and was first described in 2006 [28]. It starts at a random configuration and in each step picks a variable from an unsatisfied clause. This variable is flipped if either this decreases the number of unsatisfied clauses or otherwise with a constant probability w_{ASAT} which is a tuning parameter of

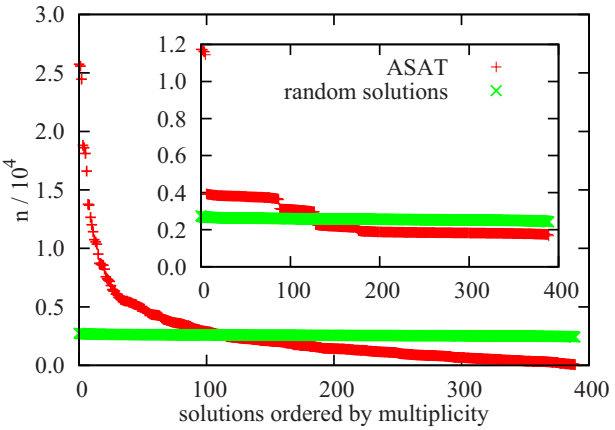

FIG. 1. (Color online) Multiplicities n of solutions found by ASAT in 10^6 runs for a randomly chosen instance with $N=50$ and $\alpha=4.0$, compared to an unbiased distribution. Inset: multiplicities of the ASAT solutions after an additional $T=0$ MC step with ten sweeps.

the algorithm. ASAT has run times linear in the system size at least up to $\alpha=4.21$ on 3-SAT. For instances of moderate size like those we study here, it can well be used beyond this point [32]. The tuning parameter of ASAT is chosen to be $w_{\mathrm{ASAT}}=0.21$, which is the optimal value as given in [28].

The test procedure is very simple: for a randomly chosen small instance we run ASAT again and again starting each time from a different randomly chosen configuration and count how often each solution returned by ASAT is found. If there were no bias we would expect the histogram of solution multiplicities to be flat except for statistical fluctuations around a plateau value.

Figure 1 shows the resulting histogram, in comparison to a histogram filled with the same number of random integers drawn from a truly flat distribution over the range corresponding to the number of solutions of the SAT instance, showing what the distribution should look like as if there were no bias. Clearly there is a strong bias favoring some solutions over others. To quantify the deviations we use a χ^2 test and calculate the p value giving the probability that an unbiased sampling process yielded a sample deviating at least as much as the one at hand. The p values numerically are smaller than 10^{-323} (i.e., the resolution of our double numbers).

To test whether this bias can be corrected in a simple way, we did a further check, where instead of using the solutions returned by ASAT directly, for each solution found by ASAT a solution from the same cluster was generated using a $T=0$ MC search starting at the ASAT result. The outcome of this modification is shown in the inset of Fig. 1 for the same SAT instance as before. The distribution now clearly has five plateaus corresponding to the five clusters of the solution space and looks much flatter but exhibits still some bias. One sees that most of the ASAT solutions stem from the smallest cluster; hence, the sampling does not respect the cluster size. Hence, the bias can be decreased by additional $T=0$ MC simulations, but not completely. Further checks showed that the bias persists independently of the system size.

Since we want to study clustering properties of the solution ensemble we need to remove the bias completely and sample solutions in proportion to the cluster sizes. To ensure this, we will perform reweighing using the ballistic search algorithm as described in Sec. IV. Before we go to the ballistic search, we will show in the next section that Metropolis-coupled Markov chain Monte Carlo (MCMCMC), another important sampling method, fails on sampling SAT solutions uniformly as well.

B. Bias in MCMCMC

The MCMCMC method, first proposed in 1991 by Geyer [46], also known as parallel tempering [47,48], is a powerful and versatile tool, commonly used in biophysics and statistical physics to perform equilibrium simulations and to generate unbiased samples in large configuration spaces. MCMCMC uses a set of replicas of single instances, simulated in parallel at different temperatures and linked by global updates in which replicas are swapped pairwise with an acceptance probability depending on their energy difference and temperature spacing (Metropolis-Hastings criterion), thus facilitating the tunneling through barriers separating local minima of the phase space [49].

To study the performance of MCMCMC on SAT we employ a histogram test similar to the one described in Sec. III A for the performance of ASAT. For several values of $\alpha=1.00,\ldots,4.25$ scattered over the satisfiable phase, the number of variables N is chosen such that the expected number of solutions is 1000. This, e.g., results for the smallest value of α considered here in a system size $N=14$, while for the highest value of α, $N=50$ is feasible.

We apply a straightforward implementation of MCMCMC to a set of 50 instances for each value of the control parameter α, where we use 15 temperatures, the lowest, at which the samples are taken, being initially $T_0=0.1$, the highest, such that the corresponding energy is found to be approximately $M2^{-K}$ which is the expected energy of a completely random configuration. At every 1000 steps the temperatures are adjusted to drive the replica exchange rate between neighboring temperatures toward 50%, while keeping the lowest temperature fixed at 0.1. The procedure chosen to adjust the temperatures leads to a distribution of temperatures where for the lowest temperatures the exchange rates indeed reach 50% on average, whereas the highest temperatures all gather in the random phase. This can be seen as an indication that the number of temperatures used is sufficient to allow the replicas to travel between the constraints, i.e., the highest temperatures are indeed located in the "paramagnetic phase." We take one sample every second sweep to generate a total of 10^6 samples. Only successful sampling steps are counted, i.e., those where the energy of the configuration at T_0 is zero.

The histograms with the resulting distribution look pretty close to those drawn from a flat distribution (not shown here). We again use the p values obtained from a χ^2 test to quantify the deviations. This criterion allows to easily detect even very slight deviations from the nonuniform sampling, due to the high number of samples generated. Hence, this test is very demanding. We find that in most cases MCMCMC gives reasonably flat distributions; hence, this method

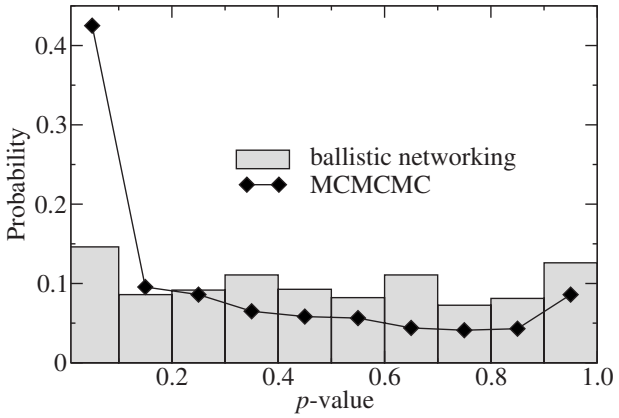

FIG. 2. Bias of MCMCMC: histogram of p values for 10^6 configurations sampled using MCMCMC (diamond symbols; lines are a guide to the eyes only) and for ballistic networking (bar graph; see Sec. IV B).

appears to exhibit on the first sight a much lower sampling bias. Nevertheless, there are also a number of histograms having a significant bias corresponding to very small p values. The higher is the value of the control parameter α chosen, the larger becomes the spread of the distribution of p values toward extremely small values. In Fig. 2 the distribution of p values is shown, integrated over all system sizes and values of the control parameter. Clearly, there is a bias toward nonuniform sampling, leading to a large fraction of small p values. Nevertheless, the situation is much better than for the application of the pure ASAT shown in Fig. 1, since a small majority of the instances are treated correctly by MCMCMC already. We have also investigated the subsets of instances exhibiting just one cluster. (The number of clusters can easily be calculated exactly for these small instances.) In that case the p-value distribution is flat (not shown), proving that MCMCMC works unbiased in this case. Thus, the presence of clustering leads to a bias or imbalance in the sampling process resulting in a peak at small p values.

Since we are interested in particular in those instances which exhibit many clusters, MCMCMC turns out to be not suitable as well since all instances have to be sampled correctly. Note that for larger system sizes, the number of instances having just one cluster, where MCMCMC seems to work well, will strongly decrease. Hence, for large system sizes, MCMCMC will exhibit a bias for basically all instances of interest. Note as well that MCMCMC by construction is able to sample all clusters with correct weights when run long enough, which can be observed for small instances. For larger instances, the bias we see comes from the fact that MCMCMC has not been run long enough. But in practice the time necessary to achieve a correct sampling is too long for MCMCMC to be useful in this case. To create an unbiased sample we need a different method which will be presented in Sec. IV.

IV. BALLISTIC SEARCH

Here, as mentioned in Sec. II B, we are using the neighbor-based definition of clusters: two solutions are considered to belong to the same cluster if there exists a path in solution space consisting of single-variable flips. We use ballistic search, which has been introduced in the year 2000 as a method for studying ground-state properties of spin glasses [42]. The approach is able to provide a survey of the cluster landscape using stochastic algorithms, in particular without the need to enumerate all ground states as it is usually necessary when one aims at clustering. The sheer number of ground states forbids exact enumeration when studying spin glasses, and—as mentioned above—the same holds for SAT. We therefore use this method which relies on generating a survey of the most important clusters.

The survey consists of a set $A=\{A_i\}$, where each element $A_i=(\{c_j^{(i)}\},\Sigma^{(i)})$ represents one cluster and consists of a (small) set of solutions $\{c_j^{(i)}\}$ from the cluster i and an estimate $\Sigma^{(i)}$ of the size of cluster i. The survey should cover all clusters or at least all but those which are negligibly small. One can then sample the whole solution space with correct weights by generating the desired number of solution samples from the representative sets of solutions for each cluster according to the respective cluster sizes.

Two main ingredients form the basis of the ballistic search algorithm. First is the above described data structure storing small sets of representative solutions for each cluster instead of all solutions. Second, a "ballistic path search" is used to analyze the cluster space and generate the survey from a given set of solutions. The basic operation of this procedure is that we have to determine for any given pair c_a,c_b of solutions, whether they belong to the same cluster. This has to work under the assumption that for the case of c_a,c_b belonging to the same cluster, a complete nearest-neighbor path of solutions between c_a and c_b is *not* contained in the set of already found solutions. Instead, one searches stochastically for paths between c_a and c_b by starting at one solution and subsequently changing a randomly chosen *free* variable. (A variable is called free if its value can be changed without violating any constraint, so that one never leaves the solution cluster.) This is repeated until either the target solution is reached or no free variable is left, because as an additional stop condition every variable shall be touched at most once. (Because of this additional constraint the path search is called "ballistic.") This implies that in a successful ballistic path search the number of steps taken is always equal to the Hamming distance of the solutions, i.e., the number of variables in which the two solutions differ.

Figures 3 and 4 give a graphical description of how the ballistic search algorithm works. We start with some randomly generated solutions depicted as black circles in Fig. 3, all of which belong to the same cluster which is drawn in gray in a two-dimensional cartoon of the N-dimensional configuration space. For the sake of simplicity we assume here that all solutions belong to the same cluster; the generalization to more than one cluster is obvious. Running the ballistic path search we find that some of the solutions can be connected by paths drawn as lines in the picture, i.e., for these solutions the algorithms correctly find that they belong to the same cluster. The problem is that for low solution densities the average distance between solutions is large, and the efficiency of the ballistic path search strongly decreases with larger distances [42]. Therefore, we only find a few

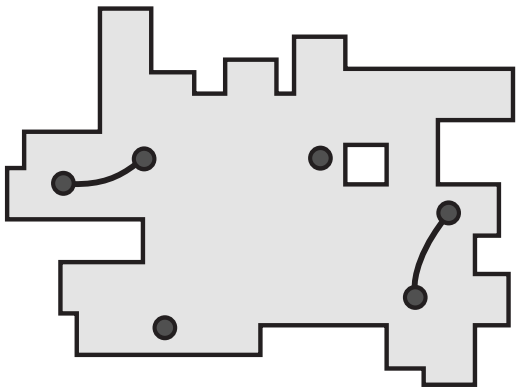

FIG. 3. Between those six solutions (black circles) from the same cluster (light gray) ballistic search has found only two paths. The apparent number of clusters is 4. We need to increase the density of solutions to make ballistic search more efficient.

paths and the apparent number of clusters in this example is larger than its true value (cf. Fig. 3). What we need to do is to increase the number of solutions by rerunning ASAT. For a few added solutions, the measured number of clusters will increase since only a few additional paths within clusters are detected, less than the number of added solutions. When generating even more solutions, we will find that the apparent number of clusters at some point no longer increases, but instead it decreases as more and more paths between solutions are found, until finally all solutions are correctly assigned to the same cluster as shown in Fig. 4.

A. Ballistic networking

Our studies have shown that the simple ballistic path search algorithm has very low efficiency when applied to satisfiability, which can be attributed to a high complexity of the solutions space or large sizes of the clusters. We therefore developed a refinement of the algorithm named "ballistic networking," which is a very general extension of the ballistic search, so that it can readily be applied to other problems.

The idea of the algorithmic refinement is to increase the probability of identifying two solutions, origin c_a and target

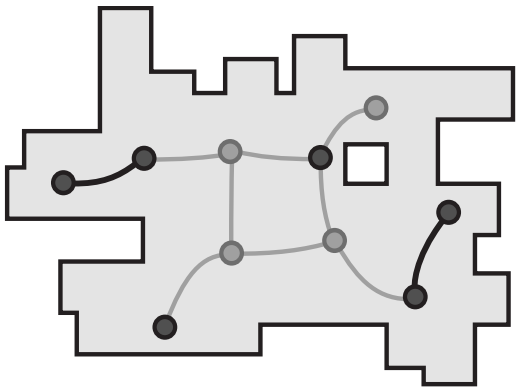

FIG. 4. Adding solutions (gray) has yielded a correct identification of the cluster, because more paths (gray) have been found, now connecting all solutions.

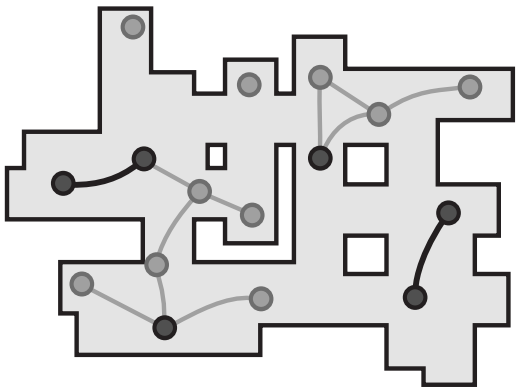

FIG. 5. This cluster has a more complex structure than the one in Fig. 4, illustrated by the additional holes. Here, adding even more new solutions (gray) than before does not work. Still not all solutions are recognized as belonging to the same cluster.

c_b, as belonging to the same cluster using ballistic path search by again increasing the number of solutions. But instead of using ASAT to generate more solutions we generate $2n_{\mathrm{add}}$ additional solutions by performing independent $T=0$ MC simulations starting at c_a and c_b, respectively. Hence, we are sure that the additional solutions belong to the same cluster as their respective "parent" solution. We then try to find connections using the ordinary ballistic path search between all $(n_{\mathrm{add}}+1)^2$ pairs of solutions, where one solution belongs to the origin and the other one belongs to the target. If at least one path is found, it is clear that c_a and c_b belong to the same cluster. We apply this test to all pairs of solutions which have not yet been found to belong to the same cluster. An artist's view of this improvement is given in Figs. 5 and 6. Figure 5 shows how the standard ballistic search fails due to a more complex structure of the cluster, although even more solutions than before have been used. In Fig. 6 the solutions found by the $T=0$ MC search are drawn as circles connected to their parent solution by arrows. We can see that the number of successful ballistic path searches (gray lines) does not have to be very high, but still is enough to correctly identify the cluster.

Indeed this procedure improves the performance of the search so much that it outweighs the additional effort of hav-

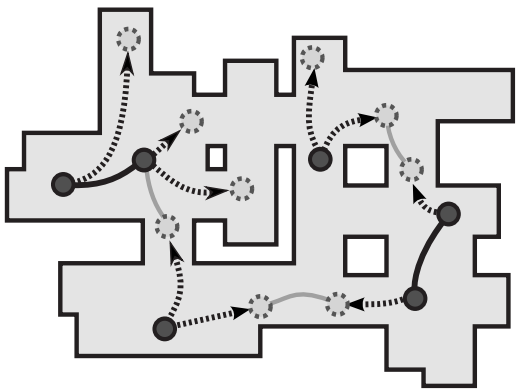

FIG. 6. Ballistic networking improves the result of ballistic search by not adding solutions randomly, but adding solutions from the same cluster using a $T=0$ MC search (arrows). Now all solutions are found to belong to one cluster.

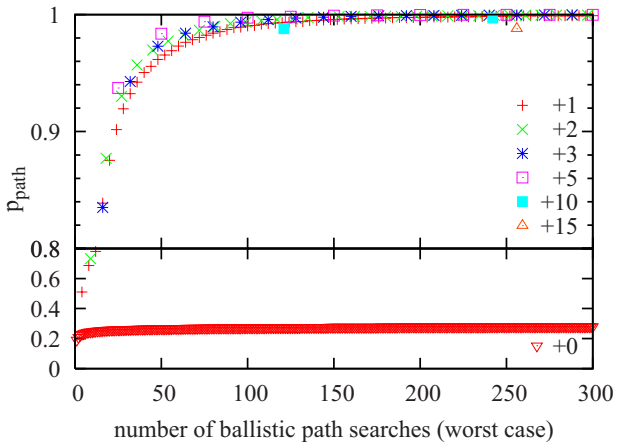

FIG. 7. (Color online) Comparison of ballistic path search without and with additional solutions (ballistic networking) for $N=128$ and $\alpha=3.0$. We show the probability for finding a path between two solutions generated from the same cluster, as a function of the total number of ballistic path searches between all pairs of parent and children configurations averaged over 1000 runs. The case +0 corresponds to the original ballistic path search.

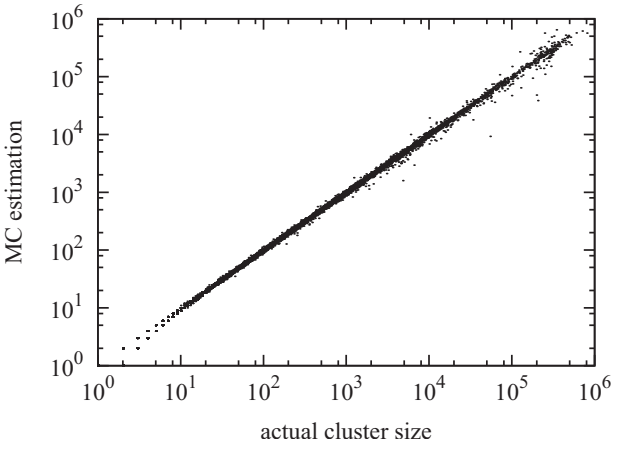

FIG. 8. Comparison of the actual number of configurations per cluster to the number estimated by Monte Carlo integration. Each point corresponds to one cluster.

ing to carry out $(n_{\text{add}}+1)^2$ ballistic path searches instead of one. Figure 7 shows a comparison of the performance of ballistic path search without and with additional solutions. The case "+0" corresponds to the bare ballistic path search. The horizontal axis shows the number of ballistic path searches which have to be carried out in the worst case, i.e., when no connecting path is found. We found $5<n_{\text{add}}<10$ to be a suitable range for the system sizes under study.

When creating the additional solutions to test whether a pair of solutions c_a, c_b belongs to the same cluster, to improve the success probability, one can think of introducing a bias into the $T=0$ MC search which pushes the additional solutions derived from the first solution c_a closer to the second solution c_b, and vice versa. Indeed we found that such a bias has a positive influence on the success probability of the ballistic path search. Yet we did not use this bias in the implementation, because the positive influence comes at a high cost. For each pair of solutions to be tested a dedicated biased set of additional solutions has to be generated which cannot be reused when comparing either c_a or c_b to a third solution. The necessary computational effort for generating each time new biased configurations by far outweighs the positive effect of the bias.

The second part of the cluster survey A consists of the sizes $\Sigma^{(i)}$ of the clusters. An exact calculation of the cluster size is possible, but takes too long since it typically grows exponentially with N. Hence, we restrict the exact calculation to small clusters exhibiting at most 1000 solutions, where the cluster enumeration can be done very quickly. For larger clusters, we estimate the cluster size. We have therefore examined several different estimation methods with respect to their reliability in giving a correct estimate for the cluster size by comparing the estimated cluster size to the exact cluster size on a random ensemble of clusters for different values of α and small system sizes N. The best method known to us has been found to be the estimation of the cluster size using a Monte Carlo integration as it has been used in [50] in an application to spin glasses.

The basic idea of this method is to define a test Hamiltonian $H_t=-\Sigma_i \delta_{c_i,s_i}$, where δ is the Kronecker delta, $\{c_i\}$ is a reference configuration from the cluster $\mathcal{C}$ to be estimated in size, and we restrict $\{s_i\} \in \mathcal{C}$. The size of the cluster then is given by $\exp[S_t(\beta)]|_{\beta=0}$, where S_t is the extensive entropy and β is the inverse temperature. We obtain from the microcanonical definition of the temperature $T=dE_t/dS_t$

$$\begin{aligned} S_t(0) &= S_t(0) - S_t(\infty) \\ &= \Delta S_t \\ &= \int_{E_t(\infty)}^{E_t(0)} \beta dE_t \\ &= \int_0^{\infty} [E_t - E_t(\infty)] d\beta \\ &= \int_0^{\infty} (E_t + N) d\beta, \end{aligned}$$

where the last but one equality comes from an integration by parts and the last equality comes from the uniqueness of the ground state of H_t with $E_t(\infty)=-N$.

The extensive energy $E_t(\beta)$ is obtained from a MC simulation, and a numerical integration using $w=\exp(-2\beta) \in [0,1]$ and $\Delta S_t=\int_0^1 [(N+E_t)/2w] dw$ yields an estimate of the cluster size. For the integration over w, we have chosen $\Delta w=0.05$. The number of MC sweeps was chosen automatically such that by doubling the number of sweeps the relative entropy change is less than a given threshold.

Figure 8 shows a comparison of the actual number of configurations in one cluster to the number as estimated by Monte Carlo integration. We used several combinations of N and α where the total number of solutions (over all clusters) did not exceed 5×10^6, such that all clusters could be calculated exactly, and afterward for each cluster the MC estimation was run.

B. Implementation

Combining ballistic networking and the cluster-size estimation the full algorithm is comprised of two alternating steps. The first step is to generate a given number (on the order of 1000) of solutions of the satisfiability instance at hand using the ASAT algorithm. In the second step ballistic networking of the solutions found by ASAT is done as described above to create the cluster survey, and then the sizes of the clusters are estimated. Afterward ASAT is run again and another set of new solutions is created. The cluster survey is then updated using ballistic networking on the new solutions and the solutions representing the so far found clusters in the existing survey. Here new clusters may be found; and, if so, their sizes are estimated. This is repeated until the cluster survey is considered complete, i.e., no more relevant clusters are found.

From the cluster survey for each instance a set $\mathfrak{U}$ of unbiased solutions can be generated using the cluster-size estimates. For each solution to be generated for a given instance, first a cluster from the survey is selected with a probability proportional to the cluster size. One solution is selected from the set of representative solutions, and starting from this solution a $T=0$ MC search is performed finally giving the solution to be used in the analysis.

Defining a good stopping criterion is a crucial point of the algorithm. As the cluster number in satisfiability can be rather large, we decided not to generate all clusters, but all except for those which contain only a negligible number of solutions. For this purpose we monitor the total cluster weight $\Sigma_i \Sigma^{(i)}$. We run the algorithm until the total cluster weight has not increased by more than 0.5% over the last half of solutions included in the clustering process. We store the order in which the solutions have been generated by ASAT and label each cluster with a number telling the position of the earliest solution which has been found to belong to this cluster.

When trying to optimize the number of new solutions added in each round one has to consider two competing effects: on one hand adding solutions—as in ordinary ballistic search—may reveal that two clusters actually are parts of the same cluster, connected maybe by only a narrow path in solution space which has been too hard to find with a fewer solutions. On the other hand increasing the number of solutions makes ballistic networking slower and, even worse, increases the probability of fake new clusters which in turn can lead to an ongoing increase in the cluster number and total cluster weight, and thus to a failure of the stopping criterion.

The system sizes which can be reached using the method described above depend, of course, on the control parameter α. For small α all solutions are contained in only one large cluster where there are many possible paths between configurations, so that ballistic search is very efficient and system sizes of a few hundred variables are possible. For high α values in the solvable phase, the number of solutions is small, so that in this regime ballistic search still is rather efficient due to the small extent of the clusters and relatively large system can be done.

Figure 9 shows the dependence of the success probability of ballistic path search on the Hamming distance d_{ham} between the configurations for $N=128$, for different values of the control parameter α. Up to $\alpha \approx 3.5$ the probability decreases strongly with increasing α, as the clusters develop more holes. Above this point the curves are approximately independent of α. We also find that the probability decreases weakly with increasing system size (not shown). The fact that the average distance between solutions decreases with α makes ballistic networking most difficult in the intermediate regime around $\alpha \approx 3.3$. Here, the number of ballistic path searches needed to find a connection between two solutions from the same cluster is highest. The cluster structure seems to be such that there are many "dead ends" in which the search may get stuck. Together with the high number of clusters, which enters quadratically in the running time, this limits the reachable system sizes. All in all, satisfiability instances of up to $N=144$ variables were doable in reasonable time over the whole range of interest $3.0 < \alpha < 4.2$, while for smaller intervals of the control parameter, we also studied $N=256$.

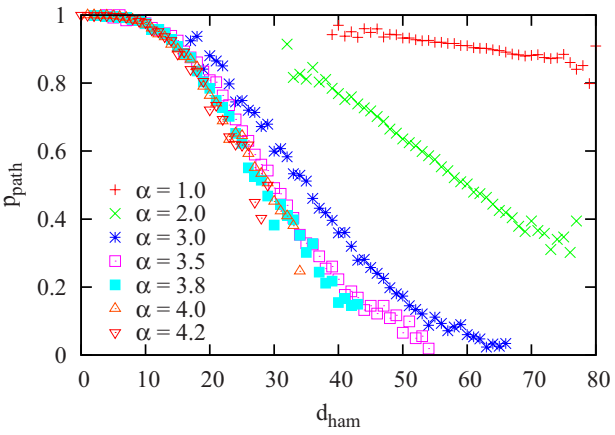

FIG. 9. (Color online) Dependence of the success probability p_{path} of the ballistic path search on α and d_{ham} for $N=128$. As the typical distance of two solutions depends on α, so do the ranges shown in the plot. For small distances $d_{\mathrm{ham}} \leq 15$, a path is usually found, while for large distances, the probability depends on the value of the control parameter α, which can be understood by a more and more complex structure of the solution clusters.

The behavior of our approach with respect to uniform sampling is shown in Fig. 2, which indicates that even this high-demanding criterion is fulfilled. Note that it is much easier to get the physical quantities right within error bars, since very often the average physical quantities will not much change between different clusters. Hence, e.g., using MCMCMC one could probably achieve in many cases the correct results by doing longer and longer simulations, even if the sampling has not converged at that point.

V. RESULTS

We study the behavior of random 3-SAT instances as a function of the parameter α. This is meant in the sense that we generate an instance using a given number N of variables and a set of (arbitrarily ordered) clauses C_m ($m=1,\ldots,M_{\max}$). We chose $M_{\max}=\alpha_{\max}N$, where $\alpha_{\max}$ is the

largest value of the control parameter we want to consider. We can study the behavior of each instance as a function of $M \leq M_{\max}$ by considering the clauses C_m for $m=1,\ldots,M$. Also, we can average over these distances for each value of the control parameter.

A. Hierarchical clustering

For the analysis [51] of the behavior of 3-SAT as a function of the control parameter α, we start by looking at the hierarchical structure of a set $\mathfrak{U}$ of solutions sampled for a typical 3-SAT instance. We have used "Ward's algorithm" [52,53], an agglomerative hierarchical matrix updating algorithm, on the set $\mathfrak{U}$ to extract a hierarchical clustering from which we can then draw a visual representation of the solution space.

Ward's algorithm has been applied in many different fields ranging from RNA secondary structures over optimization problems to spin glasses [8,20,44,54]. It is an iterative procedure where initially each configuration comprises a single item cluster. In each step those two clusters are merged which have minimal distance with respect to an effective distance measure chosen such that the sum of the variances in each cluster is minimized. After each merger, the distances of the remaining clusters to the new cluster have to be calculated; for details see, e.g., Ref. [53]. Finally, one reorders the configurations according to the hierarchy obtained in the iterative merging process and draws a color-coded visualization of the distance matrix.

Next, we present some results for a typical instance. We chose one which exhibits its SAT-UNSAT transition close to the numerical estimate of the ensemble average $\alpha_s=4.267$ given in [16]. Figure 10 shows the color-coded distance matrices and the dendrogram which were generated for three different values of α. The difference in the solution landscape and cluster structure between the phases is clearly visible. For low α the Ward matrix is featureless and homogeneously gray. All solutions belong to one single cluster and the phase space shows no specific features. In the intermediate range one sees boxlike structures along the diagonal in a darker gray. These correspond to clusters, because darker means smaller Hamming distance and the solutions inside a cluster are closer to each other than to other solutions. Some of these boxes show a substructure which can be interpreted as the solutions from this cluster themselves forming subclusters. This is consistent with the theoretical prediction of replica-symmetry breaking beyond 1-RSB in the intermediate α range for the most numerous clusters [23], which we expect to be visible for finite sizes, even if indeed the 1-RSB behavior is dominating. Nevertheless, as mentioned in the introduction, it is to be expected that most of the clusters are not relevant in the thermodynamic limit and a small number of clusters contain almost all solutions. For higher values of α the substructures inside the clusters become washed out, whereas the first-level cluster structure becomes more pronounced as the cluster becomes smaller. In the replica-symmetry breaking framework this would be interpreted as a vanishing of higher level RSB above a certain threshold, but this cannot be deduced from looking at single instances, of course.

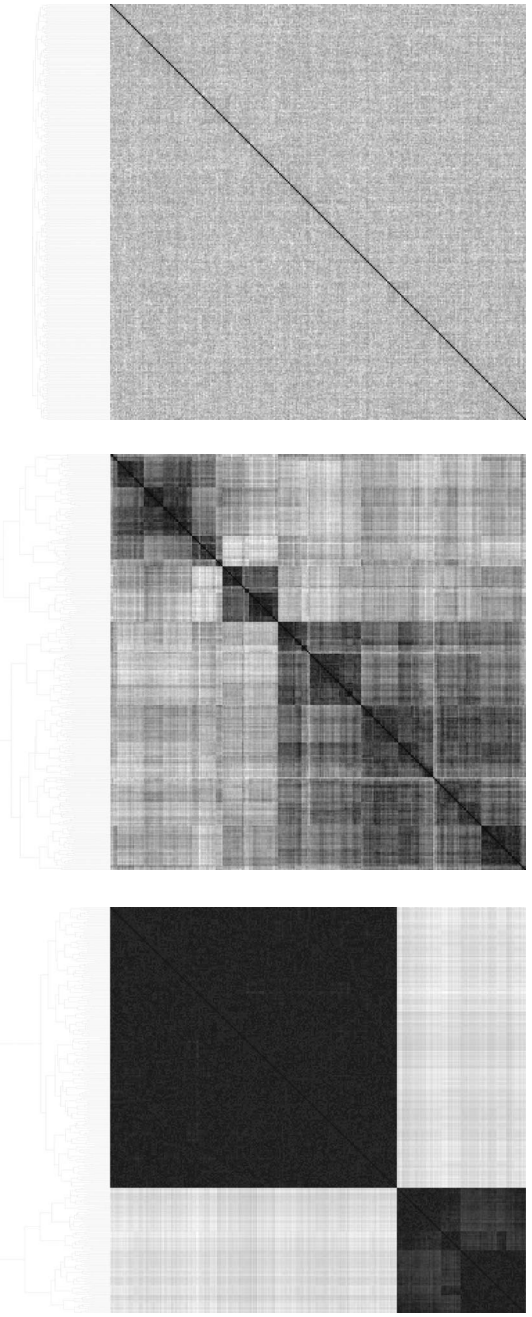

FIG. 10. The hierarchical structure resulting from Ward's algorithm visualized both as tree structure (dendrogram) and distance matrix, for $N=256$ and $\alpha=1.00$ (top), 4.00, and 4.25 (bottom). Darker gray scales correspond to smaller distances.

B. Averaged quantities

The complexity $c=\frac{1}{N}\ln N_c$ is defined as the logarithm of the number of clusters normalized to the system size. Figure 11 shows the average complexity as a function of α and for system sizes up to $N=256$ variables averaged over 200–500 satisfiable instances for each value of α and each value of N. The number of clusters was taken directly from the cluster surveys created using the ballistic-networking method described above.

For the "easy" part of the satisfiable phase, where the value of the control parameter α is small, there is only one

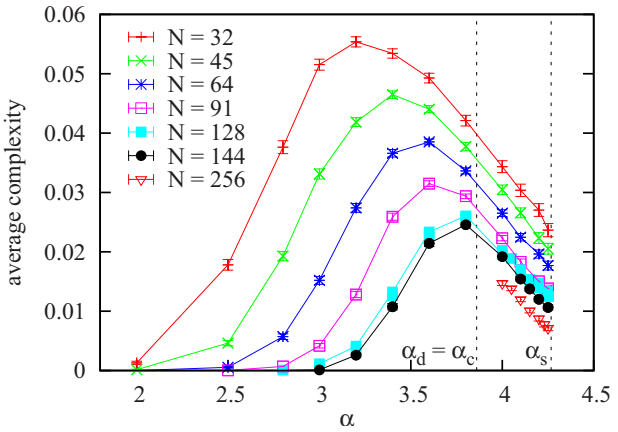

FIG. 11. (Color online) Average complexity c as a function of α and for several system sizes between 32 and 144. Lines have been drawn to guide the eye. The error bars give statistical errors.

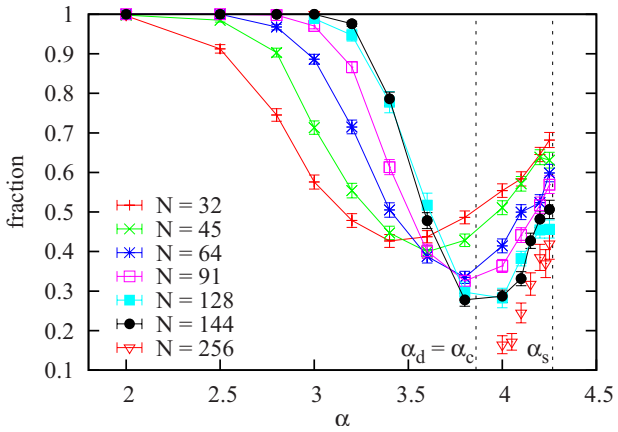

FIG. 12. (Color online) Fraction of the weight of the largest cluster with respect to the total weight of all clusters.

cluster; thus, the complexity is zero. In an intermediate range the number of clusters grows peaking at a value which is strongly affected by finite-size effects and then becomes smaller again. This behavior reflects the theoretical prediction of one single cluster in the low-α regime "crumbling" into smaller pieces when α is increased and the clustered phase is reached. For even higher α the vanishing of solutions leads to the disappearance of clusters and the cluster number decreases again. The peak of the complexity curves seems to converge with increasing system size toward a value near α_c.

Looking at this plot one has to keep in mind that the stopping criterion used in the algorithm is based on the number of solutions covered by the clusters that were found so far. In a phase with a large number of small clusters we might miss small clusters if they only comprise a negligible part of the solutions (in the sense of the stopping criterion) and therefore might underestimate the number of clusters. It is therefore natural that the complexity found here is lower than the one given in [31]. After all the complexity shown in the graph is only a lower bound for the true complexity respecting all clusters. Nevertheless, we expect that the true complexity behaves qualitatively the same as what we have observed. In the thermodynamic limit, since all clusters contribute to this figure, also the statistically unimportant ones, the complexity c will also exhibit a peak near α_c and may stay finite for $\alpha > \alpha_c$. From our results, it is not possible for us to decide whether c will converge to zero for $\alpha \to \alpha_s$ or not.

Figure 12 displays the fraction of the solutions contained in the largest cluster. For $\alpha < \alpha_c$ this value seems to increase with growing system size. At $\alpha = \alpha_c$ it exhibits a minimum, while for $\alpha > \alpha_c$ it decreases slightly with growing system size, but it is larger than the value found right at α_c. These results are also compatible with the analytical prediction [18], which states that only for a range $\alpha \geq \alpha_c$ more than one cluster is relevant in the thermodynamic limit. Nevertheless, we cannot deduce from the data, since system sizes we can reach are rather limited, whether for all values of $\alpha < \alpha_c$ this growing fraction converges to one or to a smaller value.

Next, we have a closer look at the average structure of the solution space. As mentioned in the discussion of the Ward matrices, solutions belonging to the same cluster are more similar to each other, i.e., closer in terms of the Hamming distance, than pairs of solutions which belong to different clusters. The cluster structure is thus reflected in the set of all pairwise overlaps, where the overlap r_{ij} of two solutions i and j, for which the Boolean variables take the values given by $x_n^{(i)}$ and $x_n^{(j)}$, is defined as $r_{ij} := \Sigma_{n=1}^{N} \delta(x_n^{(i)}, x_n^{(j)})/N$. Figure 13 shows the overlap distribution for $\alpha = 4.0$ and several values of N. For each system size 1000 instances have been processed with the algorithm described in Sec. IV B and 500 solutions have been generated from each cluster survey.

Two peaks are visible. One peak is lying close to $\langle r \rangle = 1$ and due to the overlap of solutions belonging to the same cluster. With larger system size it moves slightly to lower $\langle r \rangle$ values and becomes sharper. The second peak at about $\langle r \rangle = 0.7$ is not discernible for the smallest system size, but only evolving with larger system sizes and only visible weakly against an also growing background. Note that a pure two-peak structure would correspond to the picture of one-step replica-symmetry breaking (1-RSB) [55]. Nevertheless, the result is not fully clear here since in addition to the peaks, there is also a continuous part between the two peaks. The overlap converges to zero for values of the overlap smaller

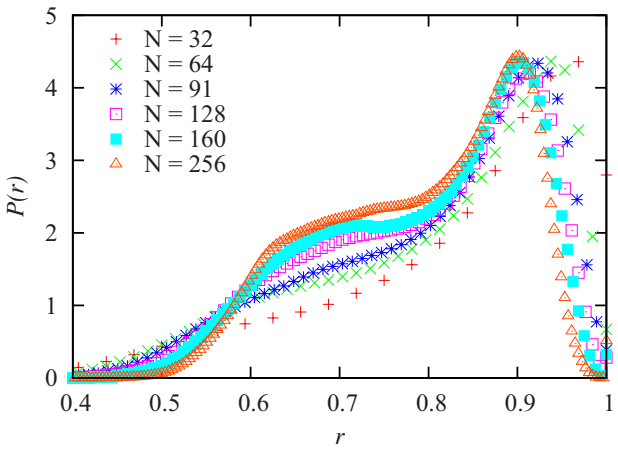

FIG. 13. (Color online) Average distribution of overlaps r for $\alpha = 4.0$. For $r < 0.4$ the curves are essentially zero.

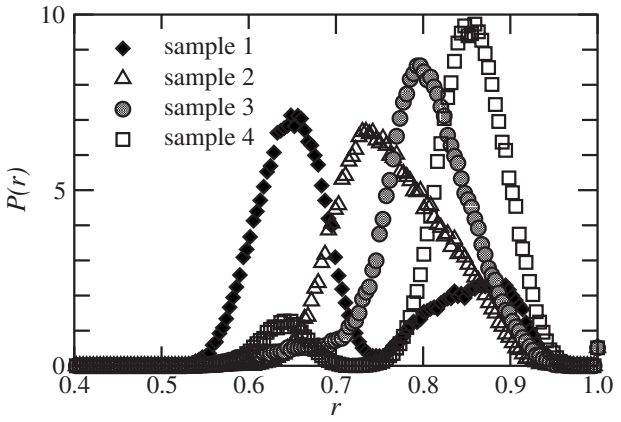

FIG. 14. Distribution of overlaps r of typical samples for $\alpha=4.0$ and $N=256$. For $r<0.4$ the curves are essentially zero.

than 0.5, which means that the number of solutions which differ in more than half the variables is very small. For $q>0.5$ the distribution may remain broad in the thermodynamic limit, which could be compatible with full RSB. Nevertheless, when we look at typical distribution of overlaps of single instances, we see mainly distributions with one or two peaks (see Fig. 14); only a few realizations exhibit three peaks. This is indeed in favor of a 1-RSB scenario. Thus, the broad behavior found in Fig. 13 is due to an average over many peaks at different locations. In any case, one maybe could somehow label each instance automatically with "RS," "1-RSB," or "f-RSB" depending on the number of peaks and study these fractions as a function of the system size. Nevertheless, larger systems would be needed for a thorough quantitative analysis of this kind. Note that we have found similar results for other values of the control parameter $\alpha_c<\alpha<\alpha_s$ (not shown).

C. Freezing transition

To complete the picture we also studied the freezing transition, which as mentioned in Sec. II B is defined as the smallest α above which all solutions belong to frozen clusters and has been found to lie at $\alpha_f=4.254$. To check directly whether a cluster contains frozen variables, we need to generate and compare all solutions from this cluster; therefore, cluster surveys do not help here. Using exact algorithms we find that for the system sizes we can reach, for all α near the SAT-UNSAT transition, there are always frozen variables in all clusters. This is probably due to too small system sizes.

Thus, we followed a different approach. For each instance, taken at $\alpha=4.20$ and 4.25 and for system sizes up to $N=2048$, we generated a solution using ASAT, which belongs with high probability to the largest cluster. Since ASAT is biased, as shown in Sec. III A, we cannot be sure that this is always the case, but according to our experience for small systems, larger clusters are found more often than smaller ones, but less often than proportional to the cluster size. Nevertheless, since our results presented next indicate the presence of clusters without frozen variables, for the range of the control parameter α we study, it is not relevant whether we indeed find the largest cluster. Finding large clusters just

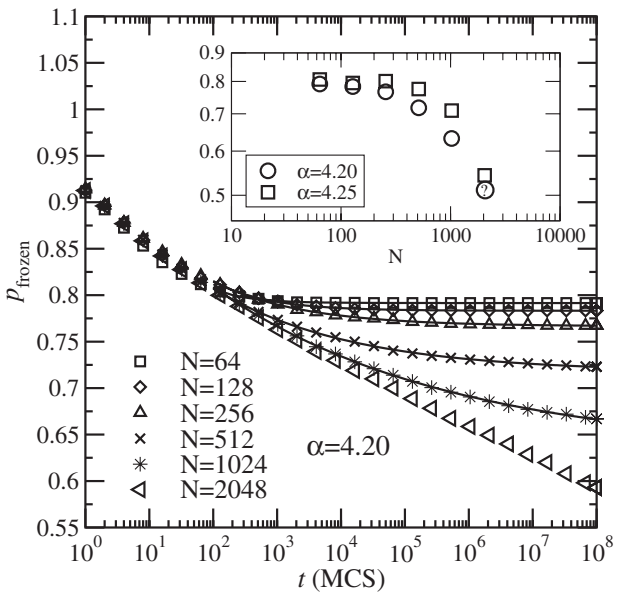

FIG. 15. Fraction $p_{\text{frozen}}(t)$ of never-flipped variables for the "largest" (i.e., first detected) cluster during $T=0$ MC simulations for $\alpha=4.20$ and different system sizes N. The solid lines indicate fits to functions of the form $p_{\text{frozen}}(t)=p_{\text{frozen}}^{\infty}+at^{-\beta}$ Inset: extrapolated values $p_{\text{frozen}}^{\infty}$ as a function of system size N in a double-logarithmic scale, for $\alpha=4.20$ and $\alpha=4.25$. The "?" symbol marks a point where the extrapolation failed and an upper limit was estimated by the eye.

helps. For each solution we found, we performed a very long $T=0$ simulation starting from this solution and measured the fraction p_{frozen} of variables which have never flipped while performing this random walk inside the solution cluster. We extrapolated this fraction to a large number of MC steps, yielding $p_{\text{frozen}}^{\infty}$ (see Fig. 15). With increasing system size, $p_{\text{frozen}}^{\infty}$ seems to converge to zero (see the inset of Fig. 15). Hence, for $\alpha=4.20$ and $\alpha=4.25$ the largest clusters seem to contain no frozen variables in the thermodynamic limit. This is compatible with $\alpha_f=4.254$, meaning that in the thermodynamic limit no frozen clusters occur below this value of α.

VI. CONCLUSION

In this work we have shown that stochastic local-search algorithms cannot be expected to produce correctly weighted samples of the solution space of satisfiability. The same holds true for MCMCMC which is widely used to sample configuration spaces in many fields of application, when the SAT solutions are spread over several clusters.

Another type of algorithm has been presented and applied for studying clustering phenomena in the solution landscape of the satisfiability problem. It is an improved version of the ballistic search algorithm which has been successfully used for studying spin glasses. Its guiding principle is to generate a survey of clusters of solutions represented by small sets of solutions rather than enumerating and clustering all solutions, which is unfeasible already for moderate system sizes. By using a different approach, ballistic networking, in the reconstructing process of the cluster structure the efficiency

of the ballistic search could be improved, so that its performance becomes reasonably high when used on satisfiability. The method presented here is general enough to be suitable for many other problems. Of course, it would be natural to study satisfiability for $K>3$ using ballistic networking, but the efficiency for ballistic path search seems to be still much lower than for the case of $K=3$, which sets very restricting limits on the system sizes which can be reached. Nevertheless, the approach presented here should be useful for many disordered systems like other types of combinatorial optimization problems.

In the case of satisfiability, the range of low values of α (where many solutions exist but belong to only one cluster) can be studied by MCMCMC. Furthermore, the case of high values of α close to the SAT-UNSAT transition (fewer solutions contained in several clusters) can be studied using exact enumeration of all solutions. In contrast, the algorithm presented here allows us to study the full satisfiable phase, but it is limited to moderate system sizes in the intermediate α range. Nevertheless, it is the only reliable method to generate unbiased samples in this regime.

Using the method described here the ensemble properties of satisfiability with moderate system size could be studied and analytical predictions about the cluster structure could be tested. To this aim we first did a visual inspection of the cluster landscape using a graphical representation in terms of Ward distance matrices. These show the expected structural differences of the different phases of satisfiability. Furthermore, we had a look at the complexity measure over the whole α spectrum in the easy phase, the fraction of the solutions contained in the dominating cluster, and the overlap distribution of the solutions for particular values of α. Our findings are in good agreement with the theoretical predictions and previous numerical studies using other methods.

ACKNOWLEDGMENTS

We have profited a lot from discussions with M. Alava, J. Ardelius, E. Aurell, P. Kaski, F. Krzakala, M. Mézard, A. Montanari, P. Orponen, S. Seitz, and L. Zdeborová. This work was supported financially by the VolkswagenStiftung within the program "Nachwuchsgruppen an Universitäten." We furthermore wish to acknowledge the allocation of computer time by the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, by the Institute of Theoretical Physics (University of Göttingen), and by the Cluster for Scientific Computing "GOLEM" (University of Oldenburg).

[1] S. Mertens, Comput. Sci. Eng. **4**, 31 (2002).
[2] A. K. Hartmann and M. Weigt, *Phase Transitions in Combinatorial Optimization Problems* (Wiley-VCH, Berlin, 2005).
[3] M. Mézard and A. Montanari, *Constraint Satisfaction Networks in Physics and Computation* (Oxford University Press, New York, 2007), http://www.lptms.u-psud.fr/membres/mezard/main.pdf
[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, New York, 1979).
[5] C. H. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1994).
[6] M. Weigt and A. K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000).
[7] W. Barthel and A. K. Hartmann, Phys. Rev. E **70**, 066120 (2004).
[8] A. K. Hartmann, A. Mann, and W. Radenbach, J. Phys.: Conf. Ser. **95**, 012011 (2008).
[9] M. Bauer and O. Golinelli, Eur. Phys. J. B **24**, 339 (2001).
[10] S. A. Cook, *Conference Record of the 3rd Annual ACM Symposium on Theory of Computing* (ACM, Pittsburgh, PA, 1971), pp. 151–158.
[11] *Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Vol. 35, edited by D. Du, J. Gu, and P. M. Pardalos (American Mathematical Society, Providence, RI, 1997); DIMACS Workshop, 1996 (unpublished).
[12] D. Mitchell, B. Selman, and H. Levesque, *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)* (AAAI Press/MIT Press, Cambridge, MA, 1992), p. 440.
[13] S. Kirkpatrick and B. Selman, Science **264**, 1297 (1994).
[14] E. Friedgut, J. Am. Math. Soc. **12**, 1017 (1999).
[15] A. Goerdt, in *Proceedings of the 17th International Symposium on Mathematical Foundations of Computer Science (MFCS'92), Prague, Czechoslovakia, 1992*, LNCS Vol. 629, edited by I. M. Havel and V. Koubek (Springer-Verlag, Berlin, 1992), pp. 264–274.
[16] S. Mertens, M. Mézard, and R. Zecchina, Random Struct. Algorithms **28**, 340 (2006).
[17] M. Mézard, T. Mora, and R. Zecchina, Phys. Rev. Lett. **94**, 197205 (2005).
[18] F. Krząkała, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, Proc. Natl. Acad. Sci. U.S.A. **104**, 10318 (2007).
[19] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian, J. Stat. Mech.: Theory Exp. (2008) P04004.
[20] H. Zhou and H. Ma, Phys. Rev. E **80**, 066108 (2009).
[21] M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002).
[22] M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002).
[23] A. Montanari, G. Parisi, and F. Ricci-Tersenghi, J. Phys. A **37**, 2073 (2004).
[24] L. Zdeborová and F. Krzakala, Phys. Rev. B **81**, 224205 (2010).
[25] F. Krzakala and J. Kurchan, Phys. Rev. E **76**, 021122 (2007).
[26] L. Zdeborová and F. Krząkała, Phys. Rev. E **76**, 031131 (2007).
[27] L. Zdeborová and M. Mézard, Phys. Rev. Lett. **101**, 078702 (2008).
[28] J. Ardelius and E. Aurell, Phys. Rev. E **74**, 037702 (2006).
[29] S. Seitz, M. Alava, and P. Orponen, J. Stat. Mech.: Theory Exp. (2005) P06006.

[30] M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, and S. Seitz, Proc. Natl. Acad. Sci. U.S.A. **105**, 15253 (2008).
[31] J. Ardelius and L. Zdeborová, Phys. Rev. E **78**, 040101(R) (2008).
[32] J. Ardelius, E. Aurell, and S. Krishnamurthy, J. Stat. Mech.: Theory Exp. (2007) P10012.
[33] C. H. Papadimitriou, *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'91), San Juan, Puerto Rico, 1991* (IEEE Computer Society Press, Los Alamitos, 1991), pp. 163–169.
[34] B. Selman, H. Kautz, and B. Cohen, in *Cliques, Coloring, and Satisfiability*, DIMAC Series in Discrete Mathematics and Theoretical Computer Science, edited by D. S. Johnson and M. A. Trick (American Mathematical Society, Providence, RI, 1996), pp. 521–532.
[35] H. Kautz, B. Selman, and Y. Jiang, in *Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Vol. 35, edited by D. Du, J. Gu, and P. M. Pardalos (American Mathematical Society, Providence, RI, 1997), pp. 573–586.
[36] M. Davis, G. Logemann, and D. Loveland, Commun. ACM **5**, 394 (1962).
[37] C. P. Gomes, H. Kautz, A. Sabharwal, and B. Selman, *Handbook of Knowledge Representation*, Foundations of Artificial Intelligence Vol. 3 (Elsevier, New York, 2008), pp. 89–134.
[38] R. J. Bayardo and J. D. Pehoushek, *Proceedings of the 17th AAAI* (AAAI Press, Menlo Park, 2000), pp. 157–162.
[39] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001).
[40] R. Monasson and R. Zecchina, Phys. Rev. Lett. **76**, 3881 (1996).
[41] G. Biroli, R. Monasson, and M. Weigt, Eur. Phys. J. B **14**, 551 (2000).
[42] A. K. Hartmann, J. Phys. A **33**, 657 (2000).
[43] A. K. Hartmann, Eur. Phys. J. B **13**, 539 (2000).
[44] G. Hed, A. K. Hartmann, D. Stauffer, and E. Domany, Phys. Rev. Lett. **86**, 3148 (2001).
[45] W. Wei, J. Erenrich, and B. Selman, *AAAI Conference on Artificial Intelligence: Proceedings of the 19th National Conference on Artificial Intelligence* (AAAI Press, Menlo Park, 2004), pp. 670–676.
[46] C. Geyer, *23rd Symposium on the Interface between Computing Science and Statistics* (Interface Foundation North America, Fairfax, 1991), p. 156.
[47] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[48] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, New York, 1999).
[49] J. J. Moreno, H. G. Katzgraber, and A. K. Hartmann, Int. J. Mod. Phys. C **14**, 285 (2003).
[50] A. K. Hartmann and F. Ricci-Tersenghi, Phys. Rev. B **66**, 224419 (2002).
[51] A. K. Hartmann, *Practical Guide to Computer Simulations* (World Scientific, Singapore, 2009).
[52] J. H. Ward, Jr., J. Am. Stat. Assoc. **58**, 236 (1963).
[53] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Advanced Reference Series (Prentice-Hall, Englewood Cliffs, NJ, 1988).
[54] P. G. Higgs, Phys. Rev. Lett. **76**, 704 (1996).
[55] M. Mézard, G. Parisi, N. Sourlas, G. Toulouse, and M. Virasoro, Phys. Rev. Lett. **52**, 1156 (1984).