Solving satisfiability problems by fluctuations: The dynamics of stochastic local search algorithms

Wolfgang Barthel, Alexander K. Hartmann, and Martin Weigt
Institut für Theoretische Physik, Universität Göttingen, Bunsenstrasse 9, D-37073 Göttingen, Germany

(Received 15 January 2003; published 12 June 2003)

Stochastic local search algorithms are frequently used to numerically solve hard combinatorial optimization or decision problems. We give numerical and approximate analytical descriptions of the dynamics of such algorithms applied to random satisfiability problems. We find two different dynamical regimes, depending on the number of constraints per variable: For low constraintness, the problems are solved efficiently, i.e., in linear time. For higher constraintness, the solution times become exponential. We observe that the dynamical behavior is characterized by a fast equilibration and fluctuations around this equilibrium. If the algorithm runs long enough, an exponentially rare fluctuation towards a solution appears.

DOI: 10.1103/PhysRevE.67.066104 PACS number(s): 02.50.Ga, 05.40.−a, 89.20.Ff

I. INTRODUCTION

The last years have seen a fruitful exchange between theoretical computer science and statistical mechanics [1,2]. Due to the formal analogy between various combinatorial optimization problems and certain spin-glass models, substantial progress in the understanding of hard combinatorial questions could be made by using tools that were originally developed in the statistical mechanics of disordered systems.

The most striking results so far were obtained in the description of the solution-space structure of the random satisfiability problem [3–7], of the number partitioning problem [8,9], of vertex covers [10–12] or colorings [13] of random graphs. In these cases, equilibrium methods from statistical mechanics can be applied directly, including, e.g., the replica and cavity approaches. The main result is that these models undergo phase transitions from an easily solvable, underconstrained phase to a hard, highly constrained one. The latter is characterized by the existence of glasslike states, i.e., the solution space is subdivided into a large number of disconnected clusters, and there are exponentially many excited states hindering even the best local algorithms from finding optimal solutions in subexponential time (where exponential means, here and in the following, exponential in the system size, as given, e.g., by the number of discrete degrees of freedom or, in a more computer-science oriented language, in the number of bits needed to encode an instance of the problem under consideration).

Up to now, much less is understood about the dynamical behavior of algorithms that are used to numerically solve the combinatorial problems. Also these are known to undergo algorithm-dependent phase transitions from phase space regions, where the problems are typically efficiently solvable, to regions where solutions are exponentially hard to construct. Some understanding was obtained for heuristics, i.e., approximate algorithms running in linear time, see, e.g., Refs. [14–16], for complete solvers [17–19] that are guaranteed to find an optimal solution, and finally for randomized versions of these complete algorithms [20,21]. The problem in analyzing algorithms is that they are intimately related to nonequilibrium statistical mechanics, which frequently is technically much harder to handle. In addition, algorithms are not forced to fulfill physical criteria such as detailed balance, which again complicates the analysis.

In this paper, we are going to analyze a different class of algorithms: *stochastic local search algorithms*, in particular, variants of the so-called *walk*-SAT algorithm [22] which is one of the most popular and successful solvers for satisfiability problems. Whereas the full problem is to hard to attack successfully by means of analytical tools, we will give some approximation methods that allow us to draw a qualitative picture on how these algorithms solve an optimization problem.

The paper is organized as follows: In Sec. II, the considered models are introduced. We first introduce the random K-satisfiability (K-SAT) problem and give an overview of the current state of knowledge. Then, we introduce a second model, the random K-XOR-satisfiability (K-XOR-SAT) problem. Being in many aspects similar to the K-SAT, it has recently attracted some interest due to its better analytical tractability. In the last part of Sec. II, we give a short introduction to some stochastic local search algorithms, in particular, to the famous walk-SAT algorithm which will be analyzed in the present paper. We then show some numerical observations in Sec. III. These are analytically explained in Secs. IV and V. The first of these two sections deals with the linear-time behavior, whereas the second one describes the exponential-time behavior. Our results are summarized in the last section.

II. THE MODELS

A. Random K satisfiability

A random K-satisfiability (K-SAT) formula F consists of M logical clauses $\{C_\mu\}_{\mu=1,\dots,M}$ which are defined over a set of N Boolean variables $\{x_i=0,1\}_{i=1,\dots,N}$ which can take the values 0=FALSE and 1=TRUE. Every clause contains K randomly chosen Boolean variables that are connected by logical OR operations ($\vee$) and appear negated with probability 1/2, e.g., $C_\mu=(x_i\vee\overline{x_j}\vee x_k)$ for $K=3$. Because of the OR conjunction a K-SAT clause is satisfied if at least one of the K variables has the correct assignment. In the formula F, all clauses are connected by logical AND operations ($\wedge$),

$$F=\wedge_{\mu=1}^{M}C_\mu, \tag{1}$$

so all clauses have to be satisfied simultaneously in order to satisfy the formula. For $K=2$, i.e., if each clause connects only two variables, the problem is easy, and polynomial-time algorithms are known [24]. On the other hand, the problem becomes nondeterministic polynomial (NP) complete for all $K>3$ [24], so one expects that no efficient algorithm to solve generic K-SAT formulas in polynomial time can be found.

The considerable attention attracted by the random K-SAT problem was initiated about one decade ago, when the model was numerically observed [25] to undergo a characteristic phase transition which is parametrized by the clause-to-variable ratio $\alpha=M/N$. For $\alpha<4.26$ and sufficiently large system sizes N, almost all 3-SAT formulas were found to be satisfiable. For $\alpha>4.26$, this behavior changes drastically; the formulas are found to be unsatisfiable with a probability approaching 1 in the thermodynamic limit $N\rightarrow\infty$. Even more interestingly, this transition was observed to coincide with a strong exponential peak in the algorithmic solution time of complete algorithms. The hardest to solve formulas are thus located close to the phase boundary, and are said to be critically constrained.

The observation of this phase transition finally led to the application of analytical tools developed in the statistical mechanics of disordered systems, since random K-SAT can be mapped to a spin-glass model on a random hypergraph. After the pioneering work by Monasson and Zecchina [3] providing the first analytical approximation to the phase transition using the replica method, many efforts were done to improve the analytical understanding. In Ref. [5], on the basis of a variational approach, a second phase transition was suggested to appear inside the satisfiable phase: For very low α, the set of all solutions to a K-SAT formula was found to be unstructured, with the exponentially large number of solutions collected in one large connected cluster. For larger α, the solution space breaks into an exponential number of clusters. Using the cavity approach, the (probably) exact location of this transition was established recently for $K=3$. It is given by $\alpha_d=3.92$ [6,7].

B. A simpler but similar model: Random K-XOR-SAT

A model showing a very similar behavior, but being analytically much more tractable, is given by the random K-XOR-SAT problem [in the physical literature initially denoted as K-hSAT (hyper-SAT) [26]]. The difference to K-SAT is that the variables appearing in the clauses are connected by logical XOR operations ($\oplus$) instead of OR. A clause is thus satisfied if an odd number of variables is assigned correctly, i.e., to TRUE if the variable appears non-negated, and to FALSE if it appears negated.

The $\oplus$ operation is equivalent to an integer addition modulo 2. Using this equivalence, we can map each clause to a linear equation (modulo 2), and the formula, consequently, to a coupled set of M linear equations. The solution of this system can be easily found in $O(N^3)$ steps. Hence, XOR-SAT formulas can be solved efficiently by a *global* algorithm, i.e., by exploiting the global information about the instance and its structure in every step. If we use, however, *local* algorithms like the ones used also for K-SAT, we observe a very similar behavior of both models.

Again, the system can be conveniently parametrized by $\alpha=M/N$. The numbers given below are valid for $K=3$, but the qualitative picture is valid for any $K>3$. For $\alpha<0.818$, the formula is typically easy to solve, the solution space consisting of one large cluster. In the region $0.818<\alpha<0.918$, the formulas are still satisfiable with probability tending to 1 for $N\rightarrow\infty$, but the solution state decays into an exponential number of clusters. In addition, there are also exponentially many metastable states that prevent even the best local algorithms from fast convergence to a solution. For $\alpha>0.918$, the system is almost surely unsatisfiable.

These values were originally calculated using the replica method which are believed to be exact, but still lacks a rigorous foundation. A very beautiful result for K-XOR-SAT was recently obtained in two independent works [27,28]: The results given above, including the ones obtained by one-step replica symmetry broken calculations, were reproduced using mathematically rigorous methods.

The K-XOR-SAT problem is also interesting from a physical point of view, because it is equivalent to a diluted K-spin model. Such models are frequently discussed in connection to the glass transition, see, e.g., Ref. [29].

C. Stochastic local search algorithms

As already mentioned in the introduction, here we are not interested in the solution space structure of K-SAT and K-XOR-SAT, but in the *nonequilibrium dynamics* of so-called stochastic local search algorithms.

The idea behind these algorithms is that, if a formula is satisfiable, a solution can frequently be found more quickly if *randomized* algorithms are used. In general, these algorithms are *incomplete*, i.e., they stop once they have found a solution, but they are not guaranteed to really find one. Due to their random character, they are also not able to prove the unsatisfiability of a formula. In the case where there is no solution the algorithm just runs forever, or until some running-time cutoff is reached.

Here, we mainly concentrate on the *walk*-SAT *algorithm* introduced in Ref. [22]. Its most recent implementations are available in the SATLIB [30], and they are one of the best stochastic local search algorithms for random K-SAT. The algorithm starts with a random assignment to all N variables. Within this assignment, there is a number $\alpha_s N$ of satisfied clauses, whereas the other $\alpha_u N=(\alpha-\alpha_s)N$ are unsatisfied.

In every step, the algorithm selects an unsatisfied clause C randomly and then one of its K variables v^* (a) with probability q randomly (*walk* step), (b) with probability $1-q$ the variable in C occurring in the least number of satisfied clauses (*greedy* step).

The current assignment of v^* is inverted. All clauses containing v^* that were unsatisfied before become now satisfied. Clauses that were satisfied behave differently for the two models under consideration: For K-SAT, a previously satisfied clause becomes unsatisfied if and only if v^* was the only correctly assigned variable in this clause. For K-XOR-SAT, every previously satisfied clause containing v^* becomes unsatisfied.

These steps are repeated until no unsatisfied clause is left. Then, the algorithm has found a solution of formula F and stops. As noted earlier the algorithm will run forever if no solution exists.

There are variants for the greedy step: The algorithm could also select the variable in C leading to the minimal number of unsatisfied clauses (''maximal gain''), or the one minimizing the number of previously satisfied clauses that become unsatisfied (''minimal negative gain''). The second case is equivalent to our choice for K-XOR-SAT. For K-SAT, they are different due to the fact that not all satisfied clauses become unsatisfied.

A completely different heuristic is the GSAT heuristic [31] which, in the greedy step, globally selects the variable leading to the minimal number of unsatisfied clauses. In numerical studies, this selection is outperformed by walk-SAT [32]. There also other heuristic variations of walk-SAT and GSAT are discussed. For reasons of clarity, we concentrate completely on the algorithm given above. We expect, however, that the approximate approach developed in this paper can also be extended to more involved cases, as long as the dynamics can be considered as a Markov process.

A different iteration of variable flips was introduced by Schöning [33]. He suggested to stop the algorithm after $3N$ steps, and to restart it by selecting a new random initial assignment to all N Boolean variables. For $q=1$, i.e., for a pure random walk dynamic, he was able to prove that the worst case solution time goes down from 2^N iterations to only $(4/3)^N$ steps, i.e., the algorithm is exponentially accelerated. This simple algorithm shows, up to a refinement leading to 1.3303^N steps [34], the currently best known worst case behavior of all SAT algorithms.

In the following sections, we will analyze both models for exponential waiting times and for an exponential number of random restarts. We will concentrate on formulas that are satisfiable, i.e., on variables-to-clauses ratios inside the satisfiable phase of the model under consideration. In the unsatisfiable phase there are no solutions, thus the algorithm cannot terminate by construction.

III. NUMERICAL RESULTS ON THE BEHAVIOR OF WALK-SAT

Now we present some numerical observations on the behavior of walk-SAT applied to randomly generated satisfiability formulas. We look to K-SAT as well as to K-XOR-SAT, and we mainly concentrate on the solution times needed by walk-SAT, and the dynamical evolution of the number of unsatisfied clauses, while the algorithm is running. Explaining these observations will be the final aim in the following sections.

A. Random K-SAT

Let us start with random K-SAT. At first, we realize that the running time heavily depends on the ratio $\alpha=M/N$ of clauses to variables. Let us concentrate on the case $K=3$ and $q=1$ first, i.e., only walk steps are performed. For small α negating one variable in an unsatisfied clause rarely causes other clauses to become unsatisfied. Up to a critical threshold $\alpha_d \simeq 2.7$ a solution is found in a median time growing linearly with N, above this point running times grow exponentially, see Figs. 1 and 2. This observation does not depend on the fact whether we use the algorithm with or without restarts. In the following, we measure all running times in the number of Monte Carlo sweeps (MC sweeps), i.e., a single step of the algorithm leading to the negation of one variable is counted as $\Delta t = 1/N$. During a time interval of length one, every variable becomes thus negated on an average once. Note that in this representation linear solution times lead to a constant number of MC sweeps, whereas exponential iterations of walk-SAT correspond to exponentially many MC sweeps. In Fig. 3, we show a histogram of the resolution times inside the exponential regime. Obviously, this distribution can be well described by the mean of the logarithm of the running time. For such an exponentially dominated distribution this is equivalent to characterizing it by the median, whereas the average running time would be dominated by exponentially rare events with exponentially longer resolution times.

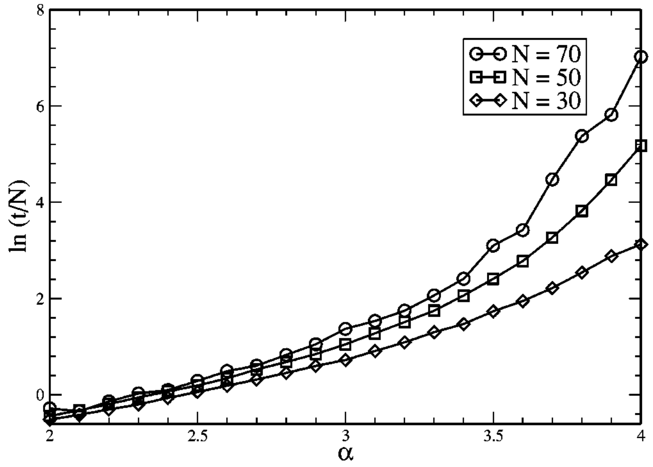

FIG. 1. 3-SAT: dependency of the running time of walk-SAT without restarts on the ratio α of clauses to variables.

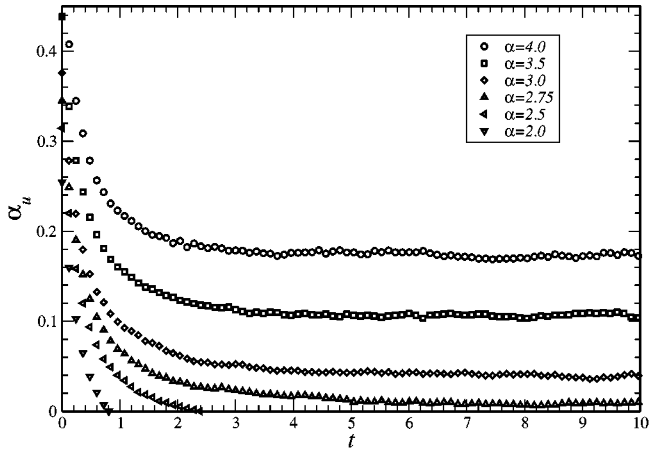

FIG. 2. 3-SAT: average number α_u of unsatisfied clauses per variable with sample size $N=50\,000$. Initially, this energy density quickly decreases. For $\alpha<\alpha_d\simeq 2.7$, it becomes zero after a finite time, for larger α a nonzero plateau is reached.

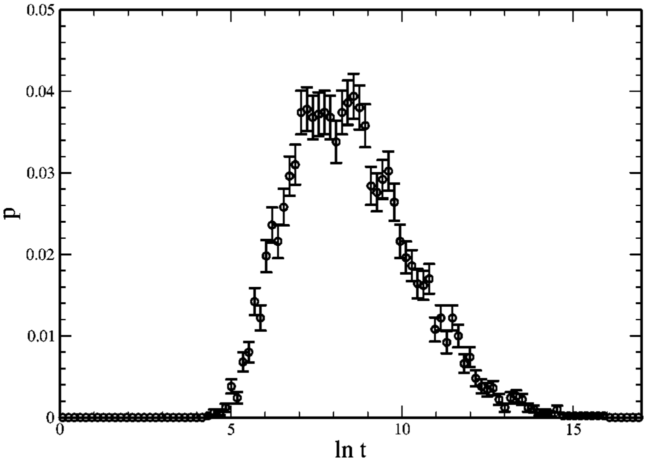

FIG. 3. 3-SAT: Histogram of the logarithm of the running times of walk-SAT without restarts for $\alpha = 3.5$ and $N = 100$.

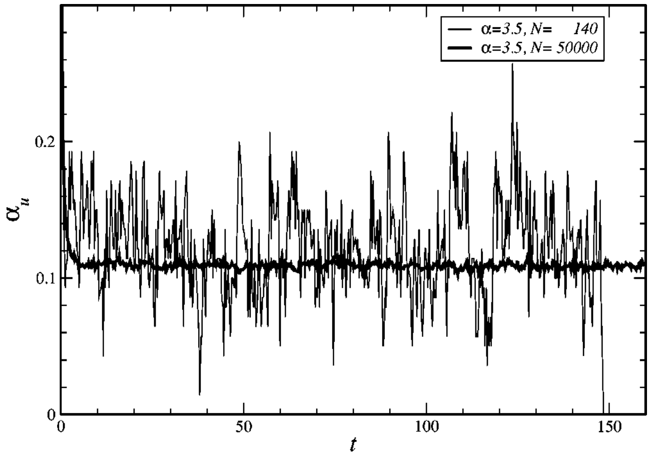

FIG. 4. After the initial decrease α_u fluctuates around its plateau value. Two different system sizes are shown. For the smaller one with $N = 150$ a fluctuation after about 145 MC steps was large enough to reach a solution of the formula.

The algorithm starts with an extensive number of unsatisfied clauses, and stops when their number reaches zero. To characterize the search process, we therefore look at the behavior of $\alpha_u(t)$, which is given as the number of unsatisfied clauses per variable. We can think of it as an energy density of the system of the N variables. In a randomly drawn starting configuration of the Boolean variables x_i, $i = 1, \ldots, N$, there are on an average 1/8 of all clauses unsatisfied, we thus have almost surely $\alpha_u(t=0) = (M/8)/N = \alpha/8$. Concentrating first on the linear-time behavior, i.e., to finite MC times, it is convenient to work with large systems, $N \gg 1$. These show a good separation of linear and exponential time scales but also minimize the influence of fluctuations. Numerically we find, in dependence on α, the following behavior.

(a) For $\alpha < \alpha_d$, a solution is found after a finite number of MC sweeps, i.e., $\alpha_u(t)$ becomes zero at finite MC times. This solution time grows with α, and diverges once we approach the dynamical threshold α_d.

(b) For $\alpha > \alpha_d$, the energy density $\alpha_u(t)$ initially decreases and quickly equilibrates to a nonzero plateau (Fig. 2). For larger times $\alpha_u(t)$ fluctuates around its plateau value, as can be seen for smaller system sizes, cf. Fig. 4. Eventually, and only if the formula is satisfiable, one of these fluctuations is large enough to reach $\alpha_u(t) = 0$.

This behavior explains the origin of the title of the paper: For $\alpha > \alpha_d$, the system equilibrates to a nonzero number of unsatisfied clauses, and only fluctuations around this equilibrium lead the dynamics to satisfying assignments, and the algorithm stops. Such macroscopic fluctuations appear, of course, only with exponentially small probability, giving rise to exponential solution times.

This observation leads to an obvious way of improving the algorithmic performance: We may choose a better heuristic having a lower equilibrium number of unsatisfied clauses. Exactly this is achieved by introducing a fraction $q > 0$ of greedy steps, see Fig. 5 where the plateau energy is determined as a function of q for two different values of $\alpha > \alpha_d$. We can see a minimum in the plateau energy for high values of q. The dynamical threshold itself also changes slightly and has a maximum at $q \simeq 0.85$. There formulas up to $\alpha \simeq 2.8$ can be solved in linear time.

B. Random *K*-XOR-SAT

A qualitatively similar behavior can be observed for random K-XOR-SAT, for $K = 3$ and $q = 1$ (pure walk dynamics). The main difference is of a quantitative nature; the dynamical threshold marking the onset of exponential solution times is located at $\alpha_d \simeq 0.33$. We therefore do not repeat the figures given for random 3-SAT, but the corresponding numerical data can be found in the following sections in comparison to analytical results.

IV. A RATE-EQUATION APPROACH TO THE LINEAR-TIME BEHAVIOR

The main idea of the analytical approach presented in this section is to characterize each variable only by the number of

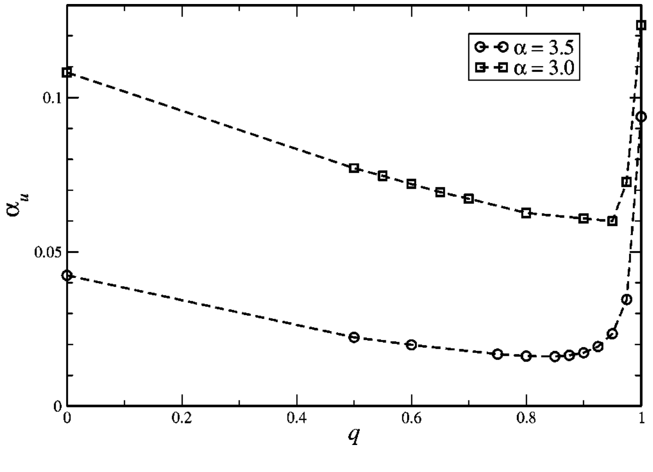

FIG. 5. 3-SAT: Plateau energy for $\alpha = 3.5$ and $\alpha = 3.0$ depending on the fraction q of greedy steps performed by the algorithm. The plateau energy is minimal for $q = 0.95$ (α=3.5) and $q = 0.85$ (α =3.0).

satisfied and unsatisfied clauses it is contained in. We subdivide the set of all N Boolean variables into subsets of $N_t(s,u)$ variables belonging to s satisfied and u unsatisfied clauses, for a randomly selected variable the numbers s and u are thus taken with probability $p_t(s,u)=N_t(s,u)/N$. The numbers $N_t(s,u)$ and thus also the probabilities $p_t(s,u)$ are changed by the action of walk-SAT, but for every single variable $s+u$ remains constant as it counts the total number of clauses containing this variable.

From these quantities we can, in particular, calculate the total number of unsatisfied clauses $N\alpha_u(t)$. Taking into account that by summing over variables every clause is counted K fold, we find

$$\alpha_u(t)=\frac{\langle u\rangle_t}{K}, \tag{2}$$

where $\langle\cdot\rangle_t=\Sigma_{s,u}(\cdot)p_t(s,u)$ denotes the average over the distribution p_t at MC time t.

The algorithm does not select variables according to $p_t(s,u)$, but selects first an unsatisfied clause C^* and then, according to the chosen heuristic (greedy or walk step), one of the variables v^* in C^* is flipped. The probability that variable v^* belongs to exactly s satisfied and u unsatisfied clauses is denoted by $p_t^{(flip)}(s,u)$, and can be calculated from $p_t(s,u)$ under the assumption of *independence of neighboring sites*, i.e., we assume that the joint distribution for three variables being in one unsatisfied clause factorizes. This assumption, which we will exploit more frequently, is the main approximation we apply in the analytical approach, and it allows us to describe the full dynamics in terms of $p_t(s,u)$. It is strictly valid only for the initial configuration of the dynamics, but as we will see below, it can give a good approximation also for larger times.

For a walk step, variable v^* is randomly selected in C^*. There are $uN_t(s,u)$ possibilities for selecting a v^* which appears in s satisfied and u unsatisfied clauses. By normalization, we thus find the following selection probability:

$$p_t^{(flip-walk)}(s,u)=\frac{up_t(s,u)}{\langle u\rangle_t}=:p_t^{(u)}(s,u). \tag{3}$$

For a greedy step, the only random choice is the selection of the unsatisfied clause C^*. Then, the variable v^* is selected which appears in the smallest number s of satisfied clauses among all K variables in C^*. If there is more than one variable with the same minimal s, then one of them is chosen randomly. Applying the independent-site assumption, and using the Heaviside function under the convention $\Theta(0)=1/2$, we find for $K=2$

$$\begin{aligned}p_t^{(flip-2-greedy)}(s,u)&=\sum_{s_1,u_1,s_2,u_2}p_t^{(u)}(s_1,u_1)p_t^{(u)}(s_2,u_2)[\delta_{(s_1,u_1),(s,u)}\Theta(s_2-s_1)\quad+\delta_{(s_2,u_2),(s,u)}\Theta(s_1-s_2)]\\&=2p_t^{(u)}(s,u)\sum_{s',u'}p_t^{(u)}(s',u')\Theta(s'-s)\\&=p_t^{(u)}(s,u)\left[2-\sum_{u'=0}^{\infty}\left(p_t^{(u)}(s,u')\quad+2\sum_{s'=0}^{s-1}p_t^{(u)}(s',u')\right)\right],\end{aligned} \tag{4}$$

and similarly for $K=3$

$$\begin{aligned}p_t^{(flip-3-greedy)}(s,u)&=3p_t^{(u)}(s,u)\sum_{s',u',s'',u''}p_t^{(u)}(s',u')p_t^{(u)}(s'',u'')[\Theta(s'-s)\Theta(s''-s)+1/12\delta_{s,s'}\delta_{s,s''}]\\&=3p_t^{(u)}(s,u)\left[1-\sum_{u'=0}^{\infty}\left(1/2p_t^{(u)}(s,u')+\sum_{s'=0}^{s-1}p_t^{(u)}(s',u')\right)\right]^2+1/4p_t^{(u)}(s,u)\left[\sum_{u'=0}^{\infty}p_t^{(u)}(s,u')\right]^2.\end{aligned} \tag{5}$$

Note that the contribution $\delta_{s,s'}\delta_{s,s''}/12$ is a correction term for the case that $s=s'=s''$ which results from the convention $\Theta(0)=1/2$.

For the full algorithms, these two different steps appear with probabilities q and $1-q$. The selection probability $p_t^{(flip)}(s,u)$ is thus given by the linear combination of the two cases,

$$p_t^{(flip)}(s,u)=qp_t^{(flip-walk)}(s,u)+(1-q)p_t^{(flip-K-greedy)}(s,u). \tag{6}$$

At this point, GSAT-like heuristics could also be included, e.g., by taking $p_t^{(flip)}(s,u)\sim u^{\gamma}p_t(s,u)$ with $\gamma>1$. This would guarantee a preferential selection of variables belonging to a high number of unsatisfied clauses. Here, we do not consider this additional possibility.

A. A Poissonian estimate for the pure walk dynamics

For a moment, we concentrate on the simplified case where the algorithms uses only walk steps, i.e., to $q=1$ [35]. We further assume that s and u are, for arbitrary times, distributed independently according to Poissonian distributions:

$$p_t(s,u)=e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^u}{s!u!}. \tag{7}$$

Again, this assumption is valid for $t=0$, whereas deviations appear for larger times. On an average each variable is contained in $K\alpha_s(t)=K(\alpha-\alpha_u(t))$ satisfied and $K\alpha_u(t)$ unsatisfied clauses. If we plug this ansatz into Eq. (3), we get for an algorithm without greedy steps

$$p_t^{(flip)}(s,u)=e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^{u-1}}{s!(u-1)!}, \tag{8}$$

which again is a product of Poissonian distributions of s and $u-1$. Hence, on average, the negated variable v^* is contained in $K\alpha_s(t)$ satisfied and $K\alpha_u(t)+1$ unsatisfied clauses.

1. Random K-XOR-SAT

We continue by first considering the analytically simpler case of K-XOR-SAT. There, by flipping v^*, all s satisfied clauses containing v^* become unsatisfied, whereas all u unsatisfied ones become satisfied. The *expected number of unsatisfied clauses* $N_t^{(u)}$ changes during one step as

$$\Delta N_t^{(u)}=-[K\alpha_u(t)+1]+K\alpha_s(t)=K\alpha-2K\alpha_u(t)-1. \tag{9}$$

Concentrating on the *average dynamics*, which is followed with probability approaching 1 in the thermodynamic limit $N\to\infty$, we have $N_t^{(u)}=N\alpha_u(t)$. Measuring the time t in MC sweeps, every algorithmic step contributes a $\Delta t=1/N$, and the difference on the left-hand side of Eq. (9) can be replaced by a time derivative (if $N\gg 1$),

$$\dot{\alpha}_u(t)=K\alpha-2K\alpha_u(t)-1. \tag{10}$$

If we solve this differential equation, we get for the the energy density of K-XOR-SAT

$$\alpha_u(t)=\frac{1}{2K}(K\alpha-1+Ce^{-2Kt}). \tag{11}$$

In the typical starting configuration half the clause are satisfied and half are not, i.e., $\alpha_u(0)=\alpha/2$. So, we finally get

$$\alpha_u(t)=\frac{1}{2K}(K\alpha-1+e^{-2Kt}). \tag{12}$$

In Fig. 6, the results for different α are compared with numerical simulations. For small times both curves coincide, because correlations have not yet builtup. Later the algorithm reaches a lower density of unsatisfied clauses than the Poissonian approximation would suggest.

We also see that there are two different regimes. For small α, the energy decreases quickly to zero—reaching zero at finite MC times with nonzero slope. For larger α, the number of unsatisfied clauses first decreases, but then reaches a positive plateau value. Both regimes are separated by a dynamical threshold that is located at

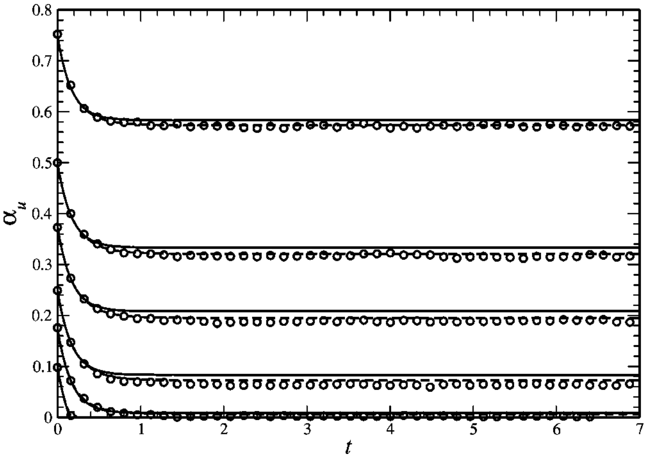

FIG. 6. 3-XOR-SAT: typical number of unsatisfied clauses (divided by N), as a function of the MC time t, for walk-SAT with *walk steps* only. Different ratios of α are shown; from top to bottom we have $\alpha=1.5$, 1, 0.75, 0.5, 0.35, 0.2. The dashed line is obtained by numerically integrating Eq. (20); the full line gives the Poissonian approximation. These results are compared to the evolution for a (random) single 3-XOR-SAT instance with $N=50\,000$, as given by the symbols.

$$\alpha_d=\frac{1}{K}. \tag{13}$$

In the special case $K=3$, we thus find $\alpha_d=1/3$ which coincides perfectly with our numerical findings. Note that for $\alpha<\alpha_d$, the algorithm thus constructs a satisfying assignment already after a linear number of algorithmic steps. Above α_d, the algorithm does not reach a solution in linear times with a probability tending to 1 in the large-N limit.

2. Random K-SAT

For random K-SAT, we can get a similar estimate. We have to take into account that now satisfied clauses do not necessarily become unsatisfied if a contained variable is inverted. For each K-SAT clause C, there is one unsatisfying and 2^K-1 possible satisfying assignments. The only case where the clause becomes unsatisfied by flipping a single variable v^* is the assignment where this variable is the only correct assigned variable in C. If we assume independent clauses this happens with probability $1/(2^K-1)$, so we get for the expected number of unsatisfied clauses

$$\Delta N_t^{(u)}=-[K\alpha_u(t)+1]+\frac{1}{2^K-1}K\alpha_s(t)$$
$$=\frac{K\alpha}{2^K-1}-\frac{2^KK}{2^K-1}\alpha_u(t)-1. \tag{14}$$

Going for $N\to\infty$ again to continuous-time quantities and differential equations, we find

$$\dot{\alpha}_u(t)=\frac{K\alpha}{2^K-1}-\frac{2^KK}{2^K-1}\alpha_u(t)-1, \tag{15}$$

with solution [the initial condition is given by $\alpha_u(0) = \alpha/2^K$]

$$\alpha_u(t) = \frac{1}{2^K K}(K\alpha + [2^K - 1][e^{-[2^K/(2^K-1)]t} - 1]), \quad (16)$$

cf. Fig. 8. For random K-SAT, we thus find the Poissonian estimate

$$\alpha_d = \frac{2^K - 1}{K}, \quad (17)$$

for the onset of exponential solution times. In the special case $K=3$, we get $\alpha_d = 7/3$ which is smaller than the numerical value 2.7.

B. Rate equation for the walk-SAT algorithm

We have seen that already a simple Poissonian approximation is able to qualitatively reproduce the behavior of walk-SAT for linear solution times, at least for a pure walk dynamics without greedy steps. There were, however, some systematic quantitative deviations, in particular, for the case of random K satisfiability. It is thus necessary to go beyond the simple Poissonian ansatz for $p_t(s,u)$, i.e., for the time-dependent fraction of Boolean variables belonging to exactly s satisfied and u unsatisfied clauses. Our aim is to work only with these quantities, i.e., we still have to keep the approximation that the joint distribution for variables within one clause factorizes. This approximation of independent neighboring variables was already used in the beginning of this section, when $p^{(flip)}(s,u)$ was derived, cf. Eqs. (3)–(6).

1. Random K-XOR-SAT

As above, we denote by $N_t(s,u) = Np_t(s,u)$ the expected number of variables that occur in exactly s satisfied and u unsatisfied clauses at step t. Our algorithm starts at $t=0$ and each step counts as Δt. We follow the procedure in Ref. [16] to describe the typical evolution of the algorithm.

A variable v^* with s^* satisfied and u^* unsatisfied clauses is flipped. This occurs with probability $p_t^{(flip)}(s^*,u^*)$. The three different processes contributing to $N_{t+\Delta t}(s,u)$ are the following:

(a) Contribution by v^*. The s^* satisfied clauses become unsatisfied, whereas the u^* unsatisfied clauses become satisfied. The number of variables characterized by s^* satisfied, u^* unsatisfied clauses is thus decreased by one, the number of variables in u^* satisfied and s^* unsatisfied clauses is increased by one. This means that the expected number of variables $N_t(s^*,u^*)$ is decreased by $p_t^{(flip)}(s^*,u^*)$, and $N_t(u^*,s^*)$ is increased by the same amount.

(b) Neighbors of v^* in previously satisfied clauses. The flipped variable v^* occurs, on an average, in $\langle s\rangle_t^{(flip)}$ previously satisfied clauses, where $\langle\cdot\rangle_t^{(flip)} = \Sigma_{s,u}(\cdot)p_t^{(flip)}(s,u)$. Since each clause contains K variables, and since random formulas are locally treelike, there are on an average $(K-1)\langle s\rangle_t^{(flip)}$ neighbors in previously satisfied clauses.

All these clauses become unsatisfied. This means that for each other variable contained in these satisfied clauses, the number of satisfied clauses goes down by one, the number of unsatisfied clauses is increased by one. Taking into account that, according to the assumption of independent neighbors, these belong to s satisfied and u unsatisfied clauses with probability $sp_t(s,u)/\langle s\rangle_t$, we conclude that $N_t(s,u)$ is, on an average, decreased by $(K-1)\langle s\rangle_t^{(flip)} sp(s,u)/\langle s\rangle_t$. One out of these s satisfied clauses is the one with the flipped variable v^*, so the decrease of $N_t(s,u)$ is now added to $N_t(s-1,u+1)$.

(c) Neighbors of v^* in previously unsatisfied clauses. Analogously to the discussion in the last item one gets contributions to $N_t(s,u)$ for variables v that occur together with v^* in unsatisfied clauses.

Combining these processes, we get an evolution equation for the expected numbers $N_t(s,u)$ of variables appearing in exactly s satisfied and u unsatisfied clauses at time t:

$$\begin{aligned} N_{t+\Delta t}(s,u) = {} & N_t(s,u) - p_t^{(flip)}(s,u) + p_t^{(flip)}(u,s) + (K-1) \\ & \times \langle s\rangle_t^{(flip)}\left(-\frac{sp_t(s,u)}{\langle s\rangle_t} \right. \\ & \left. + \frac{(s+1)p_t(s+1,u-1)}{\langle s\rangle_t} \right) + (K-1)\langle u\rangle_t^{(flip)} \\ & \times \left(-\frac{up_t(s,u)}{\langle u\rangle_t} + \frac{(u+1)p_t(s-1,u+1)}{\langle u\rangle_t} \right). \end{aligned} \quad (18)$$

Setting again $\Delta t = 1/N$ and replacing differences by derivatives in the thermodynamic limit,

$$\begin{aligned} N_{t+\Delta t}(s,u) - N_t(s,u) & = N[p_{t+\Delta t}(s,u) - p_t(s,u)] \\ & = \frac{p_{t+\Delta t}(s,u) - p_t(s,u)}{\Delta t} \to \frac{d}{dt}p_t(s,u), \end{aligned} \quad (19)$$

we get a set of differential equations for $p_t(s,u)$,

$$\begin{aligned} \dot{p}_t(s,u) = {} & -p_t^{(flip)}(s,u) + p_t^{(flip)}(u,s) + (K-1)\langle s\rangle_t^{(flip)} \\ & \times \left(-\frac{sp_t(s,u)}{\langle s\rangle_t} + \frac{(s+1)p_t(s+1,u-1)}{\langle s\rangle_t} \right) + (K-1) \\ & \times \langle u\rangle_t^{(flip)}\left(-\frac{up_t(s,u)}{\langle u\rangle_t} + \frac{(u+1)p_t(s-1,u+1)}{\langle u\rangle_t} \right). \end{aligned} \quad (20)$$

In the typical initial configuration, the probability of a clause to be unsatisfied is 1/2 and so $p_0(s,u)$ is given by Eq. (7) with $\alpha_s(t) = \alpha_u(t) = 1/2$.

By numerical integration of Eq. (20), we can find the typical trajectory for an algorithm with given $p_t^{(flip)}$. The results for an algorithms without greedy steps [e.g., $p_t^{(flip)}(s,u) = p_t^{(u)}(s,u)$] for different values of the ratio $\alpha = M/N$ are shown in Fig. 6. They are compared with numeri-

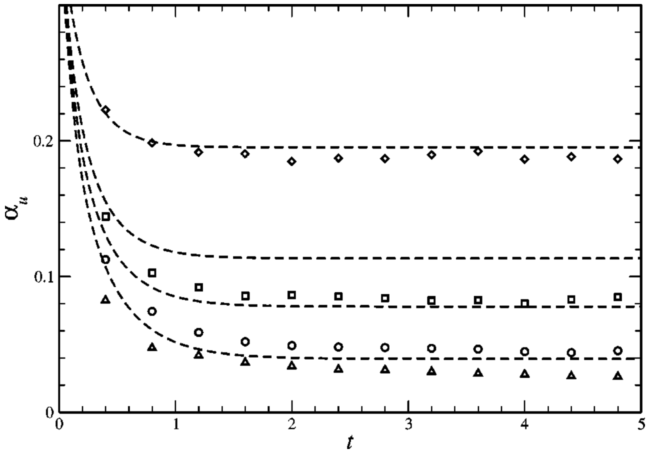

FIG. 7. 3-XOR-SAT: influence of greedy steps on the behavior of the energy density at $\alpha=0.75$. As above the dashed line is obtained by numerically integrating Eq. (20) after plugging in Eq. (6) and using $N=1/\Delta t=50\,000$. The dotted line shows the evolution of a (random) single run of the algorithm with $N=50\,000$. From top to bottom, we have $q=0$ (i.e., no greedy steps), $q=0.5$, $q=0.7$, $q=0.9$. The energy plateau decreases with q but due to correlations the integrated equation does not fit the numerical data.

cal data obtained from single runs of the algorithm on a large single, randomly selected sample formula. As we can see the assumption of independent variables is suitable to describe the behavior of the algorithm in this model. We also see that the dynamical threshold α_d, which marks the onset of exponential solution times, is again given by 1/3.

When analyzing the algorithm including a fraction of greedy steps, we see that the assumption of independent variables is indeed very crucial. In Fig. 7, we show the result of the numerical integration now using $p^{(flip)}(s,u)=qp^{(flip-walk)}(s,u)+(1-q)p_t^{(flip-3-greedy)}(s,u)$ as given by Eq. (6). In this case, the flipping probability of a variable depends on its neighbors, too, and correlations between neighbors appear naturally. This explains why the ansatz does not give a good quantitative approximation when greedy steps are included.

2. Random K-SAT

Let us now consider the slightly more involved case of random K-SAT. As already discussed in the context of the Poissonian approximation, we have to take into account that flipping a variable does not necessarily unsatisfy all previously satisfied clauses the variable is contained in. We assume again that the probability of such a clause to become unsatisfied is clause and time independently given by its naive average $\mu=1/(2^K-1)$. Similar to XOR-SAT, we get three contributions to $N_{t+\Delta t}(s,u)$, one coming from the flipped variable itself, two from neighbors in previously satisfied (unsatisfied) clauses.

(a) If the flipped variable v^* appears in exactly s^* satisfied and u^* unsatisfied clauses than, as in XOR-SAT, $N_t(s^*,u^*)$ is decreased by one. This happens with probability $p_t^{(flip)}(s^*,u^*)$. By flipping v^*, all u^* previously unsatisfied clauses become satisfied. Out of the s^* previously satisfied clauses, a random number k remains satisfied, s^*-k become unsatisfied, i.e., $N_t(u^*+k,s^*-k)$ is increased by one. There are $\binom{s^*}{k}$ possibilities for selecting these k clauses, each one appearing with probability $\mu^{s^*-k}(1-\mu)^k$. The total contribution by v^* is obtained by summing over all possible values of k.

(b) The contributions from neighbors of the flipped variable are similar to XOR-SAT. The only difference is that the average number of neighboring variables on satisfied clauses becoming unsatisfied is now $\mu(K-1)\langle s\rangle_t^{(flip)}$.

Combining all contributions, we derive a set of differential equations for the probability distribution of the variables:

$$\dot{p}_t(s,u)=-p_t^{(flip)}(s,u)+\left(\frac{1}{2^K-1}\right)^u\sum_{k=0}^{s}\binom{u+k}{k}\left(1-\frac{1}{2^K-1}\right)^k p_t^{(flip)}(u+k,s-k)+\frac{K-1}{2^K-1}\langle s\rangle_t^{(flip)}\times\left(-\frac{sp_t(s,u)}{\langle s\rangle_t}+\frac{(s+1)p_t(s+1,u-1)}{\langle s\rangle_t}\right)+(K-1)\times\langle u\rangle_t^{(flip)}\left(-\frac{up_t(s,u)}{\langle u\rangle_t}+\frac{(u+1)p_t(s-1,u+1)}{\langle u\rangle_t}\right). \tag{21}$$

Also these equations have to be solved numerically. The results for the most interesting case $K=3$ [i.e., $q_u=1/(2^K-1)=1/7$] for different values of α are shown in Fig. 8. Even if they are quantitatively much more accurate than the Poissonian approximation, there are some systematic deviations compared to the direct numerical simulations. The curves match the simulation results for small times. Then correlations between neighboring variables buildup, violating our basic assumption. However, for larger times both curves match again, because the same distribution $p_t(s,u)$ is reached. This can be seen in the histogram in Fig. 9: At $t=1.4$, the distributions $p_t(s,u)$ as derived from the rate equations or evaluated numerically are different, while after $t=6$ they again have almost converged to the same distribution.

This observation allows for a precise determination of the dynamical threshold α_d which marks the transition from typically linear to exponential algorithmic solution times needed by walk-SAT; the transition is defined by the point where the expected energy density $\alpha_u(t)$ asymptotically does not decrease to zero any more. In Fig. 10, one can see that, for $K=3$, this happens at $\alpha_d\simeq 2.71$.

As already observed for XOR-SAT, the influence of greedy steps cannot be reproduced very well. In Fig. 11, we show results for three different q at $\alpha=3.5$. The energy density obtained by assuming independent variables gives a too low-energy density. For $\alpha=0.9$, it even decreases to zero at finite times, contrary to our numerical results (cf. Sec. III A). We therefore conclude that the independent-neighbor approximation is only suitable for the case without greedy steps, where less correlations can be builtup.

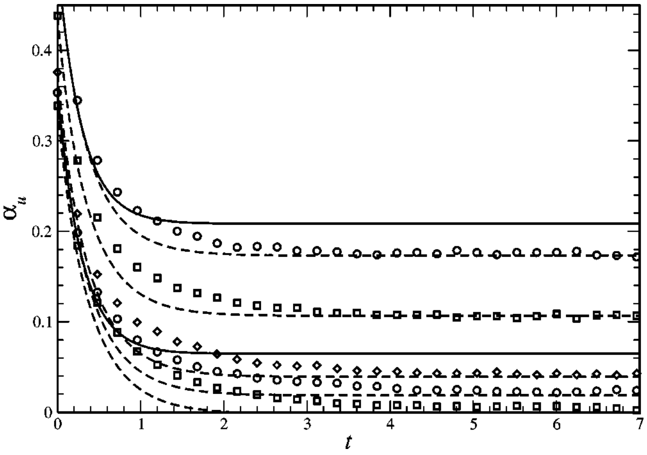

FIG. 8. 3-SAT: Running time of the walk-SAT algorithm with *walk steps* only. Different ratios of α are shown; from top to bottom, we have $\alpha=4.0,3.5,3.0,2.85,2.7$. The dashed line is obtained by integrating Eq. (21) with $N=1/\Delta t=50\,000$. The symbols show the evolution of a (random) single run of the algorithm with $N=50\,000$. The solid line shows the analytical solution (12) of the Markov equation assuming a Poissonian distribution $p_t(s,u)$ for all times t; for clarity only $\alpha=4.0$ and 2.85 are depicted.

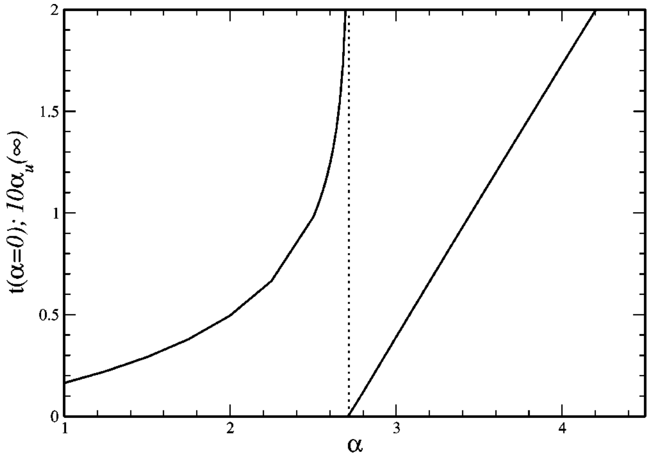

FIG. 10. 3-SAT: the left curve shows the (linear) solution time after which the expected energy density $\alpha_u(t)$ (from rate equations) reaches zero, as a function of α. This time diverges logarithmically at α_d. For larger α, a nonzero energy plateau is found, which is shown in the right curve.

V. LARGE DEVIATIONS AND THE EXPONENTIAL-TIME BEHAVIOR

In the last section, we have characterized the *typical linear-time behavior* of walk-SAT on satisfiable, randomly generated K-SAT and K-XOR-SAT formulas. We have, within some approximation assuming independent neighbors, calculated the trajectory that is followed by the system in terms of the probabilities $p_t(s,u)$ that a randomly selected variable belongs to exactly s satisfiable and u unsatisfiable clauses. "Typical" behavior means in this context that the trajectory is followed with probability tending to one in the thermodynamic limit $N\rightarrow\infty$.

We have seen that there exists some (model-dependent) dynamical threshold α_d, below which the algorithm reaches zero energy, i.e., a solution of the SAT formula, after linear time. Above α_d, the typical trajectory, however, shows a fast equilibration towards a nonzero plateau value $\alpha_u(t\rightarrow\infty)$. The walk-SAT algorithm is no longer able to construct a solution in linear time, i.e., we expect the solution times to become exponentially large. The approach of Sec. IV thus fails to describe the final descent of the energy to zero.

In Sec. III, we have seen that, for smaller system sizes, the number of unsatisfied clauses fluctuates around its expected value. Eventually, these fluctuations become large enough that the system by chance hits a solution—fluctuations are the way walk-SAT finally succeeds constructing a solution. However, we expect these fluctuations to be exponentially rare, i.e., we have to wait almost surely an exponentially long time to really touch a solution.

This section is dedicated to characterizing these fluctuations, or, more precisely, to calculating the probability $\mathbb{P}(\alpha_u(0)\rightarrow\alpha_u(t_f)=0)$ that the system reaches $\alpha_u(t_f)=0$ after some finite time t_f, under the condition that the system

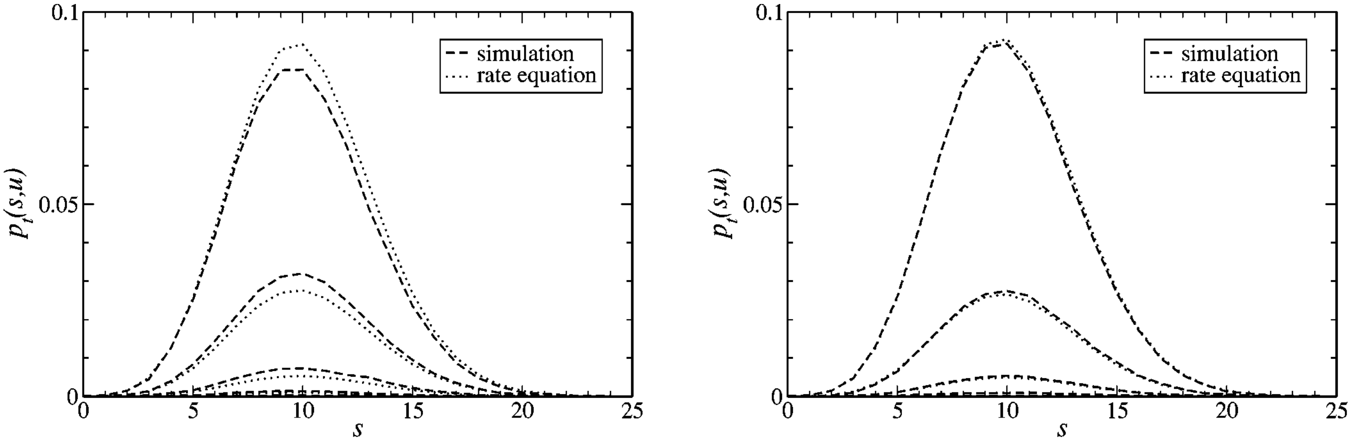

FIG. 9. 3-SAT: distributions $p_t(s,u)$ for $t=1.4$ (left) and $t=6$. The results are shown as a function of s; the different curves correspond to $u=0$, 1, 2, 3 (from top to bottom). One can see that numerical and analytical results differ for $t=1.4$, whereas they are very close for larger times corresponding to the energy plateau.

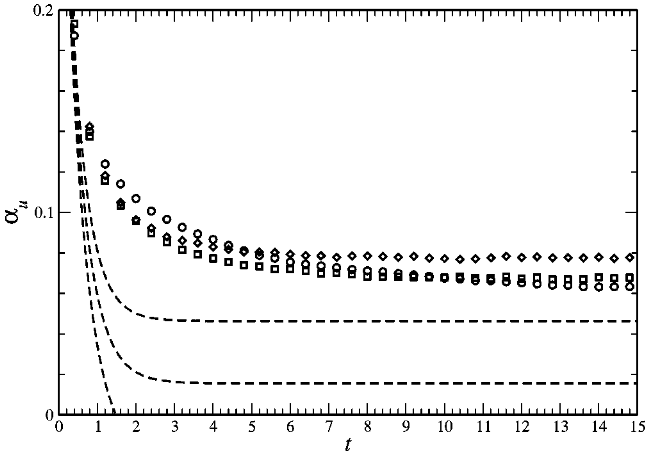

FIG. 11. 3-SAT: Influence of greedy steps at $\alpha=3.50$. As above the dashed line is obtained by numerically integrating Eq. (20) after plugging in Eq. (6). From top to bottom, we have $q=0.5$, $q=0.7$, $q=0.9$. The symbols show simulation data for the evolution of a single run of the algorithm with $N=500\,000$.

started initially with some $\alpha_u(0)$. This probability gives all important information about the dominant exponential contribution to the typical running times $t_{sol}\simeq e^{N\tau}$ beyond α_d.

(a) For *walk*-SAT *without restarts*, we start from a typical initial condition $\alpha_u(0)=1/2^K$ (for *K*-SAT) resp. 1/2 (for *K*-XOR-SAT), and we wait until the system reaches $\alpha_u(t)=0$. This does not happen for finite times, i.e., the solution time is given by the exponent

$$\tau\simeq-\lim_{t_f\to\infty}\lim_{N\to\infty}\frac{1}{N}\ln\mathbb{P}(\alpha_u(0)\to\alpha_u(t_f)=0). \qquad (22)$$

The solution time is thus, in general, exponentially large in *N*. Note that the order of limits in the above expression is relevant, there t_f measures only a finite MC time scale. With interchanged limits, the right-hand side would vanish, since the algorithm finds a solution after exponential times t_f with probability 1.

(b) For *walk*-SAT *with restarts*, the situation changes slightly. Let us assume that the algorithm stops every t_fN walk-SAT iterations and reinitializes the variables randomly. In this case, we have to take into account two distinct rare events: First, the starting point may be close to a solution, i.e., $\alpha_u(0)$ is atypically small. This happens with probability $\rho(\alpha_u(o))\sim e^{Ns(\alpha_u(0))}$, where $s(\alpha_u(0))$ is the microcanonical entropy for the energy density $\alpha_u(0)$. $\rho(\alpha_u(0))$ tends to one for the typical starting point discussed in the previous item, and becomes exponentially rare for smaller initial energies. This may be balanced by the fact that finding a solution after some given time t_f becomes more probable for smaller initial energies. From the probability of finding a solution after a single restart, $\rho(\alpha_u(0))\max_{0\leq t\leq t_f}\mathbb{P}(\alpha_u(0)\to\alpha_u(t)=0)$, we can readoff the number of restarts $t_{sol}=e^{N\tau}$ needed to find a solution with high probability:

$$\tau\simeq-\max_{\alpha_u(0)}\lim_{N\to\infty}\frac{1}{N}\ln[\rho(\alpha_u(0))\max_{0\leq t\leq t_f}\mathbb{P}(\alpha_u(0)\to\alpha_u(t)=0)]. \qquad (23)$$

Our aim is thus to calculate the large-deviation functional determining $\mathbb{P}(\alpha_u(0)\to\alpha_u(t_f)=0)$. As we will see, this can be done only within the Poissonian approximation, i.e., throughout this section, we assume

$$p_t(s,u)=e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^u}{s!u!}, \qquad (24)$$

with $\alpha_u(t)+\alpha_s(t)=\alpha$ being time independent.

A. Random *K*-XOR-SAT

Here, we discuss only an algorithm without greedy steps, where the above approximation works reasonably well. Therefore, $p_t^{(flip)}(s,u)$ is given by Eq. 8. The number of unsatisfied clauses in a formula at time *t* is given by $N\alpha_u(t)$. This number changes by $\Delta e=s-u$ in the next step if a variable with *u* unsatisfied and *s* satisfied clauses is flipped. The probability $P(\Delta e)$ of a given energy shift Δe in a single step is consequently given by

$$\begin{aligned}P_t(\Delta e)&=\sum_{s,u=0}^{\infty}p_t^{(flip)}(s,u)\,\delta_{\Delta e,s-u}\\&=\sum_{s,u=0}^{\infty}\frac{up_t(s,u)}{K\alpha_u(t)}\,\delta_{\Delta e,s-u}\\&=\sum_{s,u=0}^{\infty}e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^u}{s!u!}\,\frac{u}{K\alpha_u(t)}\,\delta_{\Delta e,s-u}\\&=\sum_{s,u'=0}^{\infty}e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^{u'}}{s!u'!}\,\delta_{\Delta e,s-u'-1}.\end{aligned} \qquad (25)$$

The probability $P_{\Delta T,t}(\Delta E)$ of a change of the number of unsatisfied clauses by ΔE after ΔT steps is given by the convolution of the single-step probabilities. For $\Delta T=N\delta t$ with small $\delta t\ll 1$, the energy density $\alpha_u(t)$ and thus $P_t(\Delta e)$ are almost time independent, so we get in Fourier space

$$\begin{aligned}\hat{P}_{\Delta T,t}(l)&=(\hat{P}_t(l))^{\Delta T}\\&=\Bigg(\sum_{s,u=0}^{\infty}e^{-K\alpha}\frac{[K\alpha_s(t)]^s[K\alpha_u(t)]^u}{s!u!}\\&\qquad\times\exp\{-il(s-u-1)\}\Bigg)^{\Delta T}\\&=[\exp\{-K\alpha+K\alpha_s(t)e^{-il}+K\alpha_u(t)e^{il}+il\}]^{\Delta T}.\end{aligned} \qquad (26)$$

Switching again to intensive quantities, we have $\Delta E=N\dot{\alpha}_u(t)\,\delta t$ and thus

$$
\begin{aligned}
P_{\Delta T,t}(\Delta E)&=\int\frac{dl}{2\pi}e^{il\Delta E}(\hat{P}_t(l))^{\Delta T}\\
&=\int\frac{dl}{2\pi}\exp\{N\delta t[il\dot{\alpha}_u(t)-K\alpha\\
&\qquad+K(\alpha-\alpha_u(t))e^{-il}+K\alpha_u(t)e^{il}+il]\},
\end{aligned}\qquad(27)
$$

as the probability of the algorithm for getting from energy density $\alpha_u=E/N$ at time t to energy density $(E-\Delta E)/N$ at time $t+\Delta T$. To calculate the transition probability between $\alpha_u(0)$ and arbitrary $\alpha_u(t_f)$ after linear time t_fN, we write t_f as a composition of many small intervals δt. We then get that the transition probability by integrating over all possible paths $\alpha_u(t)$ going from $\alpha_u(0)$ to $\alpha_u(t_f)$. By this step also the conjugate variable l becomes a time-dependent function $l(t)$,

$$
\begin{aligned}
\mathbb{P}(\alpha_u(0)\to\alpha_u(t_f))&=\int_{\alpha_u(0)}^{\alpha_u(t_f)}\mathcal{D}\alpha_u(t)\int\mathcal{D}l(t)\\
&\quad\times\exp\left\{-N\int_0^t\delta t\mathcal{L}(l(t),\alpha_u(t),\dot{\alpha}_u(t))\right\},
\end{aligned}\qquad(28)
$$

where the Lagrangian $\mathcal{L}$ is given by

$$
\begin{aligned}
\mathcal{L}(l(t),\alpha_u(t),\dot{\alpha}_u(t))&=-il(t)\dot{\alpha}_u(t)+K\alpha-K(\alpha\\
&\quad-\alpha_u(t))e^{-il(t)}-K\alpha_u(t)e^{il(t)}-il(t).
\end{aligned}\qquad(29)
$$

We can replace the integral by its saddle point in the thermodynamic limit. Since $l(t)$ is not a dynamic variable [$\dot{l}(t)$ does not appear in the Lagrangian], we find

$$
0=\frac{\partial\mathcal{L}}{\partial l}=i\dot{\alpha}_u(t)-iK(\alpha-\alpha_u(t))e^{-il(t)}+iK\alpha_u(t)e^{il(t)}+i.\qquad(30)
$$

The saddle point in $\alpha_u(t)$ is given by the Euler-Lagrange equation

$$
0=\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\alpha}_u}-\frac{\partial\mathcal{L}}{\partial\alpha_u}=i\dot{l}(t)+Ke^{-il(t)}-Ke^{il(t)}.\qquad(31)
$$

We are, in particular, interested in trajectories leading to a solution of the formula, i.e., trajectories starting at some $\alpha_u(0)$ and going to $\alpha_u(t_f)=0$ after some given final time t_f. This results in a set of two coupled first-order nonlinear differential equations for $\alpha_u(t)$ and $l(t)$ with two boundary conditions given for $\alpha_u(t)$, and none for $l(t)$. By substituting $\kappa(t)=e^{il(t)}$ the equations read

$$
\begin{aligned}
\dot{\alpha}_u(t)&=-1-K\alpha_u(t)\kappa(t)+K\frac{\alpha-\alpha_u(t)}{\kappa(t)},\\
\dot{\kappa}(t)&=K\kappa^2(t)-K.
\end{aligned}\qquad(32)
$$

A trivial solution of the second equation, $\kappa(t)\equiv1$ leads to $\dot{\alpha}_u(t)=-1-K\alpha_u(t)+K(\alpha-\alpha_u(t))$, which is exactly the equation for the typical trajectory given by Eq. 10. Indeed, we have $\mathcal{L}(\kappa\equiv1,\alpha_u,\dot{\alpha}_u)\equiv0$, so this trajectory has probability 1 in the thermodynamic limit.

This solution is, however, not stable since we have $\dot{\kappa}(t)<0$ for $\kappa(0)<1$ and $\dot{\kappa}(t)>0$ for $\kappa(0)>1$, i.e., the trajectory deviates from the typical one once κ deviates from 1. We can, however, solve the equations for XOR-SAT in this Poissonian approximation and get

$$
\kappa(t)=\frac{1+Ae^{2Kt}}{1-Ae^{2Kt}},
$$

$$
\begin{aligned}
\alpha_u(t)&=\alpha_u(0)e^{-2Kt}\frac{1-A^2e^{4Kt}}{1-A^2}+\int_0^t d\tau\Bigg(-1\\
&\quad+K\alpha\frac{1-Ae^{2Kt}}{1+Ae^{2Kt}}\Bigg)e^{-2K(t-\tau)}\frac{1-A^2e^{4Kt}}{1-A^2e^{4K\tau}}.
\end{aligned}\qquad(33)
$$

In principle, also the integrals in the second expression can be carried out analytically, but we failed to find a compact representation of the result. The solution still contains the unknown constant A which has to be adjusted to meet the final condition $\alpha_u(t_f)=0$. We have observed that A is slightly smaller than e^{-2Kt_f}, but it is easier to determine $t_f(A)$ than its inverse $A(t_f)$.

The trajectories show an interesting behavior, cf. Fig. 12; after about 1 MC sweep the energy reaches a plateau independent of the starting energy density $\alpha_u(t)$. The plateau value is the same as given by the typical trajectory and almost independent of the time t_f where the solution is found. The energy drops to 0 suddenly about 1 MC sweep before t_f. This is similar to the qualitative picture, we observed numerically in Sec. III: The system first equilibrates and then, by means of an exponentially improbable fluctuation, reaches zero energy, cf. Fig. 4. The fluctuations that are present in the numerical data cannot be seen in the analytical curve due to the fact that the latter one gives an average over all possible trajectories under the condition that $\alpha_u=0$ is reached for the first time at t_f, so only the very last fluctuation leading to the solution is common to all possible numerical trajectories.

To calculate the probability that the algorithm, starting at some $\alpha_u(0)$, finds a solution after time t_f, we have to calculate the action

$$
\begin{aligned}
S(\mathcal{L}(\kappa(t),\alpha_u(t),\dot{\alpha}_u(t)))&=\int_0^{t_f}dt\mathcal{L}(\kappa(t),\alpha_u(t),\dot{\alpha}_u(t))\\
&=\int_0^{t_f}dt\Bigg(-\ln[\kappa(t)][\dot{\alpha}_u(t)+1]\\
&\quad+K\alpha-K\frac{\alpha-\alpha_u(t)}{\kappa(t)}\\
&\quad-K\alpha_u(t)\kappa(t)\Bigg),
\end{aligned}\qquad(34)
$$

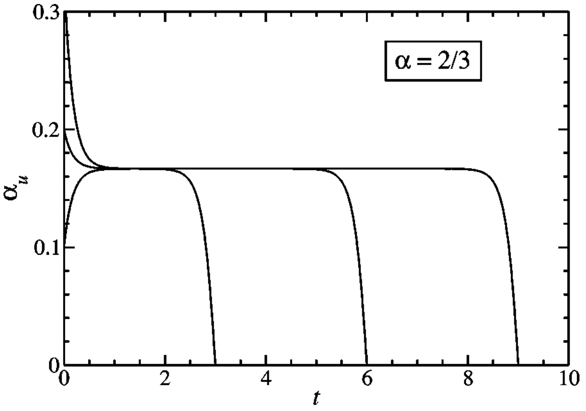

FIG. 12. 3-XOR-SAT at $\alpha=2/3$: energy densities $\alpha_u(t)$ for various initial conditions $\alpha_u(0)$ and solution times t_f. The system first equilibrates to a plateau being independent of the initial condition, and finally solves the SAT formula by a macroscopic fluctuation.

using solution (33). The evaluation is simplified by plugging in the saddle-point equations in order to eliminate $\dot{\alpha}_u(t)$,

$$S(\mathcal{L}(\kappa(t),\alpha_u(t),\dot{\alpha}_u(t))) = K\int_0^{t_f} dt\Bigg(-\ln[\kappa(t)]\bigg[-\alpha_u(t)\kappa(t)+\frac{\alpha-\alpha_u(t)}{\kappa(t)}\bigg] + \alpha - \frac{\alpha-\alpha_u(t)}{\kappa(t)} - \alpha_u(t)\kappa(t)\Bigg). \tag{35}$$

The results are shown in Fig. 13 for different values of the initial condition $\alpha_u(0)$ and different solution times. For the typical initial condition $\alpha_u(0)=\alpha/2$, we find a monotonically decreasing function that has practically reached its asymptotic value for $t_f>1$. From Eq. 28, the probability that the algorithm finds a solution is given by

$$\mathbb{P}(\alpha_u(0)\rightarrow\alpha_u(t_f)=0)=\exp\{-NS\}, \tag{36}$$

and the typical solution time of the algorithm *without restarts* is given by Eq. (22),

$$t_{sol}=\lim_{t_f\rightarrow\infty} e^{NS}. \tag{37}$$

We also observe that, for smaller than typical $\alpha_u(0)$, the action shows a pronounced minimum for small solution times. This minimum corresponds to trajectories that start close to a solution [small $\alpha_u(0)$] and go more or less directly to this solution (small t_f). As discussed in the beginning of this section, it may be possible that the algorithm can profit from this by using random restarts. Taking the entropy as calculated in Ref. [26], we, however, find that the minimum in S is overcompensated by the small entropy of low-energy starting configurations, cf. the inset of Fig. 13, where $\tau(\alpha_u(0),t_f)=-1/N\ \ln[\rho(\alpha_u(0))\mathbb{P}]$ is presented. The minimum of τ is still found for the typical starting configuration and it is related to the typical running time by $t_{sol}=\min e^{N\tau[\alpha_u(0),t_f]}$. Here, it coincides with the solution time without restarts.

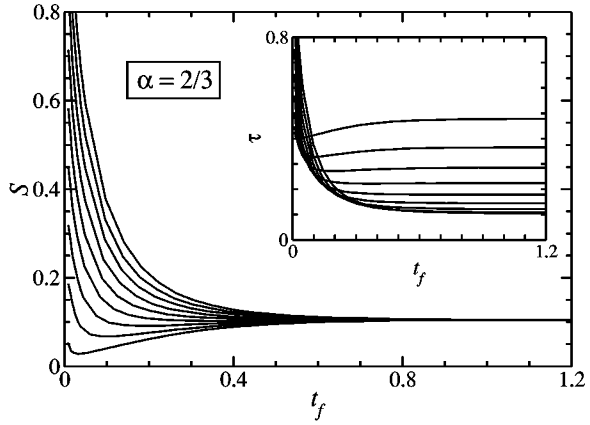

FIG. 13. 3-XOR-SAT at $\alpha=2/3$: action S as a function of the resolution time t_f, for initial conditions $\alpha_u(0)=0.02$, 0.06, 0.1, . . . ,0.34, from bottom to top. The inset shows the logarithm of the predicted solution time for the same values $\alpha_u(0)$, but now from top to bottom.

In Fig. 14, the resulting solution time is compared to numerical data obtained using the algorithm with random restarts after $3N$ iterations. Due to the exponential behavior only small systems up to $N=70$ could be investigated in the full satisfiable region. The resulting running times seem to be much smaller than the analytical predictions. There are, however, huge finite size effects. In the inset, we show numerical data for $\alpha=0.4$ and 0.42, where the exponent is still small enough that systems up to $N=1000$ can be easily solved. It is obvious that even from such large systems the asymptotic running time cannot be reasonably estimated. On the other hand, the qualitative behavior is well represented by the analytical curve, in particular the sublinear slope close to the threshold. The analytical curve suggests $\ln t_{sol}\sim\sqrt{\alpha-\alpha_d}$. Another interesting observation is that, at the SAT-UNSAT threshold $\alpha_c=0.918$, the analytical prediction for the solution-time exponent is 0.249, which is smaller but quite close to Schöning's rigorous upper worst-case bound $\ln(4/3)\simeq 0.288$.

B. Random *K*-SAT

The same type of analysis can be done for the case of random K-SAT. The main difference is, as mentioned already in Sec. IV, that a satisfied clause does not necessarily become unsatisfied when one of its variables is flipped. This happens only if the clause is satisfied only by the variable to be flipped, which is one of the 2^K-1 satisfying assignments to this clause. We again use the assumption that the variables in one clause are uncorrelated and assume that a clause becomes unsatisfied with probability $\mu=1/(2^K-1)$. In analogy to the discussion above, we conclude that the probability that a variable flip leads to a given energy change Δe is given by

$$P_t(\Delta e)=\sum_{s,u=0}^{\infty} p_t^{(flip)}(s,u)\sum_{k=0}^{s}\binom{s}{k}\mu^k(1-\mu)^{s-k}\delta_{\Delta e,k-u}, \tag{38}$$

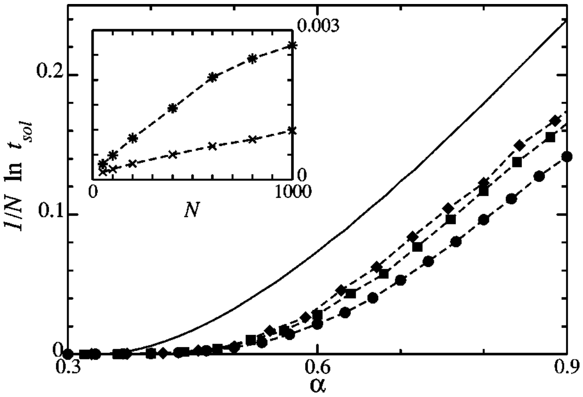

FIG. 14. 3-XOR-SAT: solution time t_{sol} for Schöning's algorithm (only walk steps, random restarts after $t_f N=3N$ steps) measured as the number of restarts, as a function of α. The analytical result is given by the full line. Numerical data for $N=30$, 50, 70 (dots, squares, diamonds, lines are guides to the eyes only) seem to indicate much smaller solution times. The inset shows, however, that there are huge finite size effects for $\alpha=0.4,0.42$ (crosses, stars). The analytical estimates for the corresponding solution times are $\ln(t_{sol})/N \simeq 0.0061, 0.0099$.

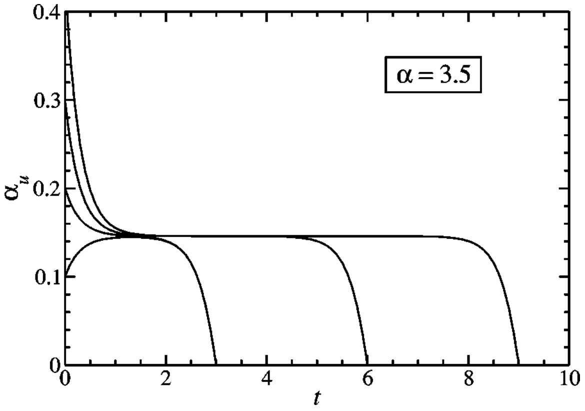

FIG. 15. 3-SAT at $\alpha=3.5$: energy densities $\alpha_u(t)$ for various initial conditions $\alpha_u(0)$ and solution times t_f. The system first equilibrates to a plateau being independent on the initial condition, and finally solves the SAT formula by a macroscopic fluctuation.

where k sums over all possible numbers of clauses that become unsatisfied in the considered algorithmic step. Concentrating again on the pure walk algorithm without greedy steps, i.e., on $q=1$, we can go through the same procedure as for K-XOR-SAT. The transition probability from some initial to some final density of unsatisfied clauses is, in the Poissonian approximation (24) given by the path integral

$$\mathbb{P}(\alpha_u(0)\rightarrow\alpha_u(t_f)) = \int_{\alpha_u(0)}^{\alpha_u(t_f)} \mathcal{D}\alpha_u(t) \int \mathcal{D}\kappa(t) \times \exp\left\{ -N\int_0^t \delta t \mathcal{L}(\kappa(t),\alpha_u(t),\dot{\alpha}_u(t))\right\}, \tag{39}$$

with the modified Lagrangian

$$\mathcal{L}(\kappa(t),\alpha_u(t),\dot{\alpha}_u(t)) = -[1+\dot{\alpha}_u(t)]\ln[\kappa(t)] + K\alpha - K[\alpha - \alpha_u(t)]\left(1-\mu+\frac{\mu}{\kappa(t)}\right) - K\alpha_u(t)\kappa(t). \tag{40}$$

The saddle-point equations are given by

$$\dot{\alpha}_u(t) = -1 - K\alpha_u(t)\kappa(t) + K\mu\frac{\alpha-\alpha_u(t)}{\kappa(t)},$$

$$\dot{\kappa}(t) = K\kappa^2(t) - K(1-\mu)\kappa(t) - K\mu. \tag{41}$$

Their solution dominates, for $N\rightarrow\infty$, the path integral (39) and is given by the generalization of Eqs. (33):

$$\kappa(t) = \frac{1+\mu A e^{2Kt}}{1-Ae^{2Kt}},$$

$$\alpha_u(t) = \alpha_u(0)e^{-(1+\mu)Kt}\frac{1-Ae^{(1+\mu)Kt}}{1-A}\,\frac{1+\mu Ae^{(1+\mu)Kt}}{1+\mu A} + \int_0^t d\tau\left(-1+\mu K\alpha\frac{1-Ae^{(1+\mu)Kt}}{1+\mu Ae^{(1+\mu)Kt}}\right)e^{-(1+\mu)K(t-\tau)} \times \frac{1-Ae^{(1+\mu)Kt}}{1-Ae^{(1+\mu)K\tau}}\,\frac{1+\mu Ae^{(1+\mu)Kt}}{1+\mu Ae^{(1+\mu)K\tau}}. \tag{42}$$

The results for the typical trajectories leading to a solution after some given final time t_f are presented in Fig. (15). They show the same qualitative behavior like K-XOR-SAT with a slightly slower convergence towards the equilibrium due to the reduced exponential factor $e^{-(1+\mu)Kt}$. Also the action calculated for the trajectories shows a similar behavior like for K-XOR-SAT, cf. Fig. 16. The exponentially dominant contribution to the typical solution time is again given by $t_{sol} \sim \lim_{t_f\rightarrow\infty} e^{NS(t_f)}$.

In Fig. 17, we finally compare the predicted typical solution time with numerical simulations. Close to the dynamical threshold, the numerical running times are much smaller, which can be explained already by the fact that the Poissonian approximation underestimates α_d. For larger α, the numerical data cross the analytical approximation, but both stay well below Schöning's bound. This is to be expected, since there is an exponential number of possible solutions, while Schöning assumes only the existence of a single one. Note that the solution times are exponentially smaller for 3-SAT than for random 3-XOR-SAT.

VI. CONCLUSION AND OUTLOOK

In this paper, we have presented an approximate analytical approach to describe the dynamical behavior of a class of

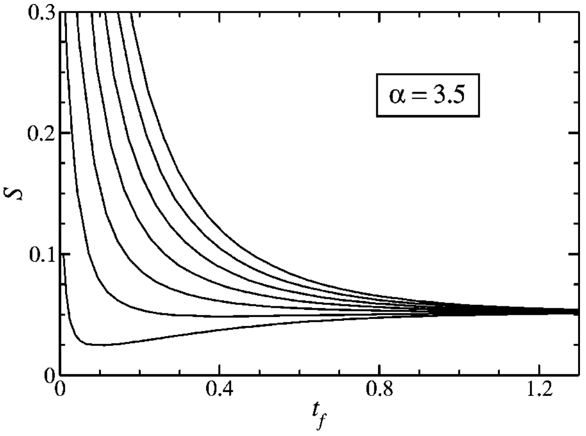

FIG. 16. 3-SAT at $\alpha = 3.5$: action S as a function of the resolution time t_f, for initial conditions $\alpha_u(0)/\alpha = 0.1, 0.3, 0.5, \ldots, 1.3$, from bottom to top.

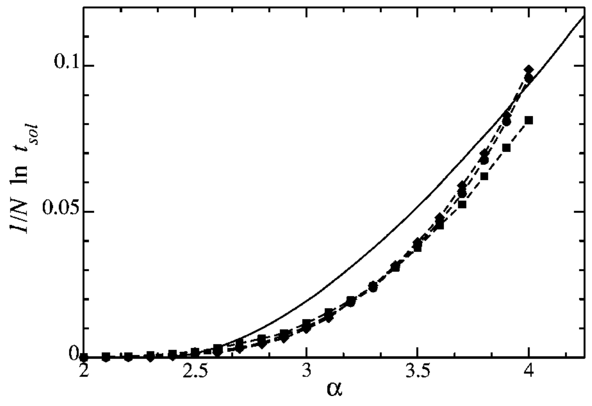

FIG. 17. 3-SAT: solution time t_{sol} for Schöning's algorithm (only walk steps, random restarts after $t_f N = 3N$ steps) measured as the number of restarts, as a function of α. The analytical result is given by the full line. Numerical data for $N = 30$, 50, 70 (squares, dots, diamonds) this time cross the analytical prediction. Note that the solution times are smaller than for 3-XOR-SAT.

stochastic local search algorithms applied to random K-satisfiability and K-XOR-satisfiability problems. We have seen that there are two distinct dynamical phases.

(a) For clause-to-variable ratio $\alpha < \alpha_d$ (with α_d being algorithm and problem dependent), the algorithm is able to solve almost all instances in linear time. In this regime, the dynamics was studied using a simple rate-equation approach which was able to capture the most important features of the average trajectory taken by the system under the action of the algorithm.

(b) For $\alpha > \alpha_d$, typical solution times were found to scale exponentially with the system size given by the number of variables N. This behavior could be understood analytically using a functional-integral approach to evaluate the probability of large deviations from the typical trajectory. We found the following behavior: The system equilibrates very fast to a nonzero plateau in the number of unsatisfied clauses. Then the system only fluctuates around this plateau. This goes on until an exponentially improbable macroscopic fluctuation towards one of the solutions appear, and the algorithm stops. The small probability of these fluctuations explains the exponentially large waiting times until a satisfying assignment is reached.

For the exponential-time regime, only a Poissonian approximation was used. In principle, it would be possible to go beyond this ansatz using the full distribution $p_t(s,u)$ of vertices with s satisfied and u unsatisfied clauses. Following the same scheme as in the Poissonian approach, we reach a system of the first-order differential equations for all $p_t(s,u)$ and their conjugate parameters $\kappa_t(s,u)$. Being nonlinear, it is far from obvious how to construct an analytical solution. But also the numerical integration of these equations is a hard problem: For the $p_t(s,u)$ there are initial and final conditions, whereas the $\kappa_t(s,u)$ have no boundary condition at all. The question if it is possible to follow this improved approach is still under investigation.

Another possible extension of this work concerns the application of different heuristics such as GSAT that was discussed in the second section. The analytical approach can serve as a basis for evaluating the relative performance of different heuristics and, as a consequence of the insight gained, also as a step towards a systematic improvement of stochastic local search.

A third point that remains open is the question that how far the solution space structure influences the performance of walk-SAT. As discussed in the beginning of the paper, random K-SAT and random K-XOR-SAT undergo a clustering transition deep inside the satisfiable phase. Below this transition, all solutions are collected in one huge cluster, above, an exponential number of such clusters exists. The clustering transition is also connected to a proliferation of metastable states which are expected to cause problems for any local algorithm. However, in our approach to the walk-SAT dynamics, we do not see any sign of a direct impact of this transition on the performance of the algorithms under consideration. The onset of exponential solution times is found to be inside the unclustered phase. It thus remains an open problem whether the clustering transition can be approached by using improved heuristic criteria.

Note added. Recently, we noticed that a complementary study of the walk-SAT algorithm was carried out independently by Semerjian and Monasson [23].

ACKNOWLEDGMENTS

We are grateful to R. Zecchina for helpful discussions. We also thank R. Monasson and G. Semerjian for communicating their results [23] prior to publication. W.B. and A.K.H. obtained financial support from the DFG (Deutsche Forschungsgemeinschaft) under Grant No. Zi 209/6-1. A.K.H. was also partly funded by the VolkswagenStiftung within the program ‘‘Nachwuchsgruppen an Universitäten.’’

[1] Special issue of Artif. Intel. **81** (1996).
[2] Special issue of Theor. Comput. Sci. **265** (2001).
[3] R. Monasson and R. Zecchina, Phys. Rev. Lett. **75**, 2432 (1995); Phys. Rev. E **56**, 1357 (1997).
[4] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Nature (London) **400**, 133 (1999).
[5] G. Biroli, R. Monasson, and M. Weigt, Eur. Phys. J. B **14**, 551 (2000).
[6] M. Mézard, G. Parisi, and R. Zecchina, Science (Washington, DC, U.S.) **297**, 812 (2002).
[7] M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002).
[8] S. Mertens, Phys. Rev. Lett. **81**, 4281 (1998).
[9] S. Mertens, Phys. Rev. Lett. **84**, 1347 (2000).
[10] M. Weigt and A.K. Hartmann, Phys. Rev. Lett. **84**, 6118 (2000).
[11] A.K. Hartmann and M. Weigt, Theor. Comput. Sci. **265**, 199 (2001).
[12] M. Weigt and A.K. Hartmann, Phys. Rev. E **63**, 056127 (2001).
[13] R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **89**, 268701 (2002).
[14] D. Achlioptas, Theor. Comput. Sci. **265**, 159 (2001).
[15] J. Franco, Theor. Comput. Sci. **265**, 147 (2001).
[16] M. Weigt, Eur. Phys. J. B **28**, 369 (2002).
[17] S. Cocco and R. Monasson, Phys. Rev. Lett. **86**, 1654 (2001).
[18] M. Weigt and A.K. Hartmann, Phys. Rev. Lett. **86**, 1658 (2001).
[19] S. Cocco and R. Monasson, Eur. Phys. J. B **22**, 505 (2001).
[20] A. Montanari and R. Zecchina, Phys. Rev. Lett. **88**, 178701 (2002).
[21] S. Cocco and R. Monasson, Phys. Rev. E **66**, 037101 (2002).
[22] B. Selman, H. Kautz, and B. Cohen, in *Cliques, Coloring and Satisfiability*, edited by D.S. Johnson and M.A. Trick, DIMACS Series Vol. 26 (AMS, Providence, 1996).
[23] G. Semerjian and R. Monasson, Phys. Rev. E **67**, 066103 (2003).
[24] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, New York, 1979).
[25] B. Selman and S. Kirkpatrick, Science (Washington, DC, U.S.) **264**, 1297 (1994).
[26] F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, Phys. Rev. E **63**, 026702 (2001).
[27] S. Cocco, O. Dubois, J. Mandler, and R. Monasson, Phys. Rev. Lett. (to be published).
[28] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina, e-print cond-mat/0207140.
[29] S. Franz, M. Mézard, F. Ricci-Tersenghi, M. Weigt, and R. Zecchina, Europhys. Lett. **55**, 465 (2001).
[30] See the SATLIB at http://www.satlib.org/
[31] B. Selman, H. Levesque, and D. Mitchell, in *Proceedings of the 10th National Conference on Artificial Intelligence*, edited by W.R. Swartout (MIT Press, Cambridge, MA, 1992), Vol. 440.
[32] H.H. Hoos and T. Stützle, J. Automated Reasoning **24**, 421 (2000).
[33] U. Schöning, Algorithmica **32**, 615 (2002).
[34] E. Dantsin, A. Goerdt, E.A. Hirsch, and U. Schöning, in *Proceedings of the International Colloques on Automata, Languages and Programming, 2000*, edited by U. Montanari, J.D.P. Rolim, and E. Welzl (Springer, New York, 2000).
[35] As we will see later on, the case $q<1$ is not well described by the independent-neighbor approximation. This is also true for the stronger Poissonian approximation.