A Study of Pure Random Walk on Random Satisfiability Problems with "physical" Methods

Guilhem Semerjian[1] and Rémi Monasson[1,2]

[1] CNRS-Laboratoire de Physique Théorique de l'ENS, 24 rue Lhomond, 75005 Paris, France.
[2] CNRS-Laboratoire de Physique Théorique, 3 rue de l'Université, 67000 Strasbourg, France.

Abstract. The performances of a local search procedure, the Pure Random Walk (PRW), for the satisfiability (SAT) problem is investigated with statistical physics methods. We identify and characterize a dynamical transition for the behavior of PRW algorithm on randomly drawn SAT instances where, as the ratio of clauses to variables is increased, the scaling of the solving time changes from being linear to exponential in the input size. A framework for calculating relevant quantities in the linear phase, in particular the average solving time, is introduced, along with an approximate study of the exponential phase.

1 Introduction

When faced with a computationally hard problem, such as the resolution of a satisfiability formula, one can choose between two main classes of algorithm. On the one hand, complete algorithms of the Davis-Putnam-Loveland-Logeman type performs an exhaustive, yet clever, search of the configuration space, providing either a solution of the formula or a proof of unsatisfiability. On the other hand, a large number of algorithms are based on local search ideas, and explore a part of the configuration space by improving locally the current assignment of variables at each step of their execution. The current theoretical understanding of these incomplete procedures is still far from being satisfactory though some important results were recently obtained [1, 2]. The present work, although based on non-rigorous methods issued from theoretical physics, may be of some help in this regard [3].

This paper is organized as follows. In Sec. 2 we define the local search procedure under study and recall the rigorous results previously obtained, then (Sec. 3) a dynamic transition is identified with numerical simulations. The next two sections (4 and 5) are devoted to the two regimes separated by this transition, and we conclude in Sec. 6.

2 Definitions and known results

We shall consider K-SAT formulas on N boolean variables drawn from the standard random ensemble [4]. M clauses are generated independently and in the same way. For each of them, a K-uplet of variables is chosen uniformly, each of which is negated or left unchanged with probability $1/2$. The resulting clause is the disjunction of these K literals. The formula is the conjunction of M clauses. We call $\alpha = M/N$ the ratio of clauses to variables. We concentrate on the asymptotic limit where M and N go to infinity with α kept fixed.

The Pure Random WalkSAT (PRWSAT) algorithm for solving K-SAT instances is defined by the following rules [5]. Choose randomly an assignment of the Boolean variables. If all clauses are satisfied, output "Satisfiable" and the current assignment (which is a solution). If not, choose randomly one of the unsatisfied clauses, one among the K variables of this clause, and reverse its state, from true to false or *vice versa*. Repeat this step until a solution has been found or a limit on the number of flips fixed beforehand has been reached. In the latter case, the output of the procedure is "Don't know": no certainty on the status of the formula is achieved.

Papadimitriou introduced the PRWSAT procedure for $K = 2$, and showed that it solves with high probability any satisfiable 2-SAT instance in a number of steps (flips) of the order of N^2[5]. Recently Schöning was able to prove the following very interesting result for 3-SAT[1]. Call 'trial' a run of PRWSAT consisting of the random choice of an initial configuration followed by $3 \times N$ steps of the procedure. If none of T successive trials on a given instance has been successful (has provided a solution), then the probability that this instance is satisfiable is lower than $\exp(-T \times (3/4)^N)$. This bound turns PRWSAT into a probabilistic algorithm: though it is not complete, the uncertainty on its output can be made as small as desired, with a complexity scaling exponentially with the size of the instance, $T \sim (4/3)^N$, up to non-exponential multiplicative factors. Further works improved on this scaling through a refined choice of the initial configuration [6, 7], the best result being, to our knowledge, $T \sim 1.3290^N$ [7].

The above bound is true for any instance. Restriction to special input distributions allows to derive further results. Alekhnovich and Ben-Sasson showed that instances drawn from the random 3-Satisfiability ensemble are solved in polynomial time with high probability when α is smaller than 1.63 [2]. It is remarkable that, despite the fact that the same clauses are seen various times in the course of the search, rigorous results on the operation of PRWSAT can be established.

3 Behavior of the algorithm

In this section, we briefly sketch the behavior of PRWSAT, as seen from numerical experiments [8] and the analysis of [9, 10]. It has been found that a dynamical threshold α_d separates two regimes:

- for $\alpha < \alpha_d \simeq 2.7$ for 3-SAT, the algorithm finds a solution very quickly, namely with a number of flips growing linearly with the number of variables N.
  Figure 1A shows the evolution of the amount of unsatisfied clauses during the execution of a single run of PRWSAT on an instance with ratio $\alpha = 2$

and $N = 500$ variables. It is very important for what follows to stress that the curve is plotted in "reduced" units : the horizontal axis is the number of flips *divided by* M, that we shall call *time* from now on, the vertical one is the *fraction* of unsatisfied clauses, φ_0.
The curve shows a fast decrease from the initial value ($\varphi_0(t = 0) = 1/8$ in the large N limit independently of α) down to zero on a time scale $t_{sol} = O(1)$. Fluctuations of the curves plotted with this definition of the axes (in *reduced* units) are smaller and smaller as N grows. As a consequence the distribution of the solving time gets peaked around its mean value t_{sol}, an increasing function of α, when $N \to \infty$. This *relaxation* regime corresponds to the one studied by Alekhnovich and Ben-Sasson, and $\alpha_d > 1.63$ as expected [2].
- for instances in the $\alpha_d < \alpha < \alpha_c$ range, the initial relaxation taking place on $t = O(1)$ time scale is not sufficient to reach a solution (Figure 1B). The fraction φ_0 of unsat clauses then fluctuates around some plateau value for a very long time. On the plateau, the system seems to be trapped at a high level of unsatisfied clauses. The life time of this "trap", that is, the time it takes to escape from the trap and finally reach a solution is so huge that it is possible to define a (quasi) stationary probability distribution $p_N(\varphi_0)$ for the fraction φ_0 of unsat clauses on the plateau (Right inset of Figure 1B). The distribution of fractions is well peaked around some average value (height of the plateau), with left and right tails decreasing exponentially fast with N, $p_N(\varphi_0) \sim \exp(N\bar{\zeta}(\varphi_0))$ with $\bar{\zeta} \leq 0$. Eventually a large negative fluctuation will bring the system to a solution ($\varphi_0 = 0$). Assuming that these fluctuations are independent random events occurring with probability $p_N(0)$ on an interval of time of order 1, the solving time is a stochastic variable with exponential distribution[3]. Its average is, to leading exponential order, the inverse of the probability of finding a solution on the $O(1)$ time scale: $t_{sol} \sim \exp(N\zeta)$ with $\zeta = -\bar{\zeta}(0)$. Escape from the trap therefore takes place on exponentially large–in–N time scales, as confirmed by numerical simulations for different sizes. Schöning's result[1] can be interpreted as a lower bound to the probability $\bar{\zeta}(0) > \ln(3/4)$, true for any instance.

The fraction of unsatisfied clauses on the plateau reached by PRWSAT on the linear time scale is plotted on Figure 2. Notice that the "dynamic" critical value α_d above which the plateau appears (PRWSAT stops finding a solution in linear time) is strictly smaller than the "static" ratio $\alpha_c \simeq 4.3$, where formulas go from satisfiable with high probability to unsatisfiable with high probability. In the intermediate range $\alpha_d < \alpha < \alpha_c$, instances are almost surely satisfiable but PRWSAT needs an exponentially large time to prove so. Interestingly, α_d and α_c coincides for 2-SAT in agreement with Papadimitriou's result[5].

[3] This kind of distribution in local search algorithms has been largely discussed, see for instance [11]).

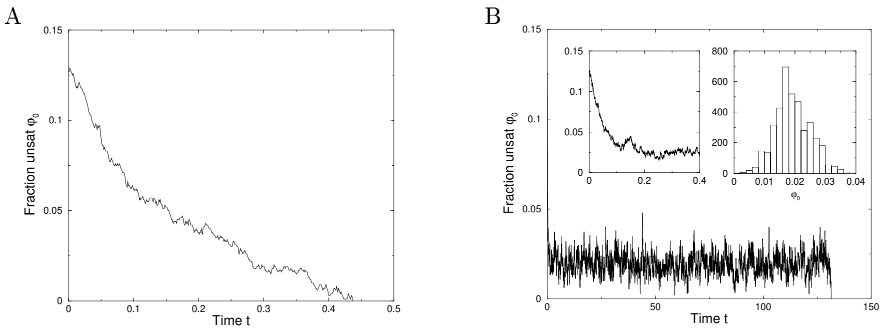

Fig. 1. Fraction φ_0 of unsatisfied clauses as a function of time t (number of flips over M) for two randomly drawn instances of 3-SAT with ratios $\alpha = 2$ (**A**) and $\alpha = 3$ (**B**) with $N = 500$ variables. Note the difference of time scales between the two figures. Insets of figure B: left: blow up of the initial relaxation of φ_0, taking place on the $O(1)$ time scale as in (**A**); right: histogram $p_{500}(\varphi_0)$ of the fluctuations of φ_0 on the plateau $1 \leq t \leq 130$.

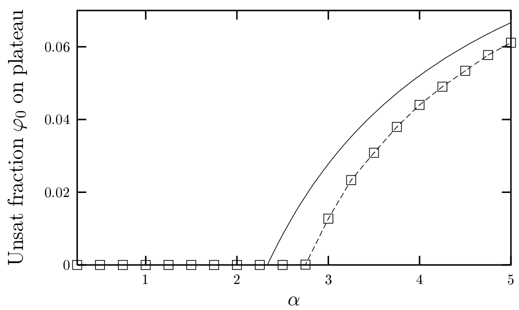

Fig. 2. Fraction φ_0 of unsatisfied clauses on the plateau as a function of the ratio α of clauses per variable. Symbols are the output of numerical experiments, and have been obtained through average of data from simulations at a given size N (nb. of variables) over 1,000 samples of 3-SAT, and extrapolation to infinite sizes (dashed line serves as a guide to the eye). The full line is the prediction of the approximation of Sec. 5.

4 Analysis of the linear phase

We present here a method suitable for the study of the low α phase. A detailed presentation of this method has been given in a general case in [12], and applied more specifically to this problem in [9].

4.1 Averages of additive quantities

The formulas studied here are constraints on N boolean variables x_i, from which $2N$ literals $x_i, \bar{x}_i$ are obtained. Given a formula F, two variables will be said *adjacent* if there is a clause of the formula in which both appears, irrespective of the fact that they are negated or not. Two variables will be said *connected* if there is a path of adjacent variables between them. These definitions allow to partition the variables of a formula F into sets of maximal connected components, and to break the formula into the independent sub-formulas F_r, one for each of the components, that we shall call *clusters* below. Note that a variable not appearing in any clause of F forms a cluster on its own.

A function Q from the set of formulas F to the set of real numbers will be said *additive* with respect to cluster decomposition if $Q(F) = \sum_r Q(F_r)$ for any F. We would like to compute the expectation $\mathbb{E}[Q]$ of Q with respect to the random K-SAT ensemble. One can decompose this average in two steps: the K variables belonging to a clause are chosen, and then the corresponding literals are negated or not. These two steps being independent we can perform the average in a convenient way.

Two clusters are said to have the same *shape* if they are isomorphic up to a relabeling of their variables, whether these variables are negated or not (see Fig. 3 for some examples). Distinct shapes of clusters will be labeled by an integer index $s \geq 0$. We denote L_s the number of variables in a cluster. Let us call $[Q]_s$ the average value of $Q(F_r)$ for a cluster of type s, the average being carried out on the choices of the negation or not of the literals in F_r only. $[Q]_s$ is independent of the labeling of the variables forming the cluster for the quantities we consider. Then the "intermediate" expectation of $Q(F)$, for a given choice of the M K-uplet of literals, can be written as $\sum_s n_s [Q]_s$, where n_s is the number of clusters of type s induced by the choice of the variables in the clauses.

To obtain $\mathbb{E}[Q]$, we are left with the calculation of $\mathbb{E}[n_s]$, the average number of type–s clusters in a random K-SAT formula. It is not hard to convince oneself that $\mathbb{E}[n_s] = N\, P_s / L_s$, where P_s is the probability that a given variable, say x_1, belongs to a cluster of type s. Then,

$$\frac{1}{N}\mathbb{E}[Q] = \sum_s \frac{1}{L_s} P_s [Q]_s \tag{1}$$

Next, we assume, that the limit (for large N) of Eqn. (1) exists, and that this limit, noted $q(\alpha, K)$, admits a power series expansion around $\alpha = 0$. Calculation of the terms in this expansion is made possible by the simplifications resulting from the large N limit. Indeed the underlying K-hypergraph undergoes a percolation phenomenon at $\alpha_p = 1/(K(K-1))$: for smaller values of

α, the clusters have size at most $O(\log N)$, and those of finite size that have a probability of order 1 are trees. In that case the expression for P_s simplifies to $P_s = (\alpha K!)^{m_s} \exp(-L_s \alpha K) V_s$, where m_s is the number of clauses of the cluster, $m_s = (L_s - 1)/(K - 1)$ for tree clusters, and V_s is a symmetry factor independent of α, equal to the number of distinct labellings of the variables of the cluster, divided by $(L_s - 1)!$. Taking into account all these facts, one realizes that the m-th term in the power series expansion of $q(\alpha, K)$ is a sum of $m + 1$ contributions, one for each of the tree clusters with a number of clauses lower or equal than m (one has to expand the exponentials in P_s to reorder the sum as a power series expansion in α). We now apply this method to two quantities of interest for the study of the PRWSAT algorithm.

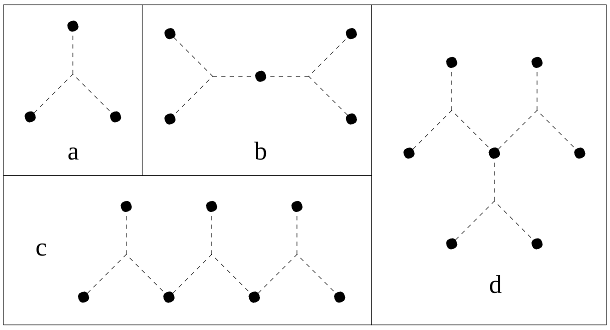

Fig. 3. Tree-like clusters contributing to the expansions, drawn for $K = 3$.

Type	L_s	$V_s(K!)^{m_s}/L_s$	$[T_{sol}]_s$
a	K	1	$\frac{1}{2^K}$
b	$2K - 1$	$\frac{K^2}{2}$	$\frac{1}{2^{2K}} \left[2^{K+1} + \frac{K+1}{K(K-1)} \right]$
c	$3K - 2$	$\frac{K^3(K-1)}{2}$	$\frac{1}{2^{3K}} \left[3 \cdot 2^{2K} + 2^{K+1} \frac{K+1}{K(K-1)} + \frac{4K^4+9K^3+9K^2+6K-4}{3K^2(K-1)(2K-1)(K^2-2)} \right]$
d	$3K - 2$	$\frac{K^3}{6}$	$\frac{1}{2^{3K}} \left[3 \cdot 2^{2K} + 2^K \frac{3(K+1)}{K(K-1)} - \frac{2K+1}{K^2(K-1)} \right]$

Table 1. Contributions to the cluster expansion for the solving time, see Figure 3.

4.2 Solving time

The number $T_{sol}(F, C)$ of flips needed by PRWSAT to find a solution is a random variable, with two different sources of randomness: formulas F are drawn randomly, and, for a given formula, the sequence C of choices made by the algorithm is random. Numerical experiments [8] indicate that, for low enough values of α, the distribution of $T_{sol}(F,C)$ gets peaked in the infinite size limit. We make the following

Conjecture: there exists a critical value of the ratio of clauses per variable, $\alpha_d(K)$, such that in the large size $N \to \infty$ limit, a randomly drawn formula with ratio $\alpha < \alpha_d(K)$ is almost surely solved by PRWSAT in linear time. More precisely, the solving time $t_{sol} = T_{sol}(F,C)/M$ is highly concentrated around some value $t_{sol}(\alpha, K)$,

$$\lim_{\epsilon \to 0^+} \lim_{N \to \infty} \text{Proba}\Big[t_{sol}(\alpha,K) - \epsilon < t_{sol} < t_{sol}(\alpha,K) + \epsilon \Big] = 1 \qquad (2)$$

where the probability is taken with respect to both choices of the formula F and variable flip sequence C. We also conjecture that $t_{sol}(\alpha, K)$ admits a power series expansion in α with a radius of convergence $\alpha_d(K)$.

Provided this conjecture is true, the coefficients of the expansion of t_{sol} in powers of α can be calculated along the lines exposed below. Consider a particular K-SAT formula. The number of flips needed by PRWSAT to solve it is a random variable, that can be written as a sum of random variables, one for each cluster: no matter in which order the different unsatisfied clauses are considered in the course of the algorithm, the total number of flips will be the sum of the flips inside each cluster. Thus, the average (with respect to the choices of the algorithm and the initial configuration) number of flips needed by PRWSAT to solve the formula is additive with respect to the cluster decomposition. One can then apply the generic strategy exposed before, with the quantity Q being the average number of flips to reach a solution. The last task remaining is the computation of the average time to solve the small clusters of finite size. Let us give a few examples. Clusters with no clauses clearly do not contribute to the sum, as their variables are never flipped. Consider now the case of a cluster made of a single clause. All but one initial configurations are already solutions, and the only unsatisfying configuration is corrected in one flip; the average number of steps needed to solve this cluster is then $1/2^K$. The same kind of enumeration can be done on larger sub–formulas. We performed this task for clusters with up to three clauses, with results summarized in Table 1. The expansion of t_{sol} reads,

$$\begin{aligned} t_{sol}(\alpha,K) = {} & \frac{1}{2^K} + \frac{K(K+1)}{K-1} \frac{1}{2^{2K+1}} \alpha \\ & + \frac{4K^6 + K^5 + 6K^3 - 10K^2 + 2K}{3(K-1)(2K-1)(K^2-2)} \frac{1}{2^{3K+1}} \alpha^2 + O(\alpha^3) \, . \end{aligned} \qquad (3)$$

This expression compares very well with the results of numerical simulations, see Fig. 4A. As is to be expected for a power-series expansion, the agreement is very good at low values of α and worsens when α increases. Calculation of higher order terms is very cumbersome, but could open the way to an approximate determination of the radius of convergence $\alpha_d(K)$.

4.3 Hamming distance

The Hamming distance Δ between the initial random configuration chosen by the algorithm and the solution found at the end of its execution is another quantity that can be studied in this expansion framework. Numerical experiments show again a concentration of Δ around its average value $N\,\delta(\alpha, K)$. In addition it is clearly an additive quantity. The expansion of δ can be carried out along the lines of Sec. 4.2, with the result

$$\delta(\alpha, K) = \frac{1}{2^K}\,\alpha + \frac{1}{2^{2K+1}}\frac{K(K-1)}{K+1}\,\alpha^2 + \tag{4}$$
$$\frac{1}{2^{3K+1}}\frac{8K^8 - 6K^7 - 33K^6 + 35K^5 + 58K^4 - 24K^3 - 48K^2 - 2K}{3(K+1)^2(4K^2-1)(K^2-2)}\,\alpha^3 + O(\alpha^4)$$

Comparison with numerical simulations is very satisfactory, see Fig. 4B.

It would be very interesting to understand the behavior of $\delta(\alpha, K)$ at the threshold α_d. As the number of flips (divided by M) needed to find a solution diverges in this limit, one could think that the solution found is completely decorrelated from the initial assignment, and that δ would reach $1/2$ at the threshold. This can however not be strictly the case: for any value of α, a finite fraction of the variables do not belong to any clause, and are thus not flipped during the run of the algorithm. Their values in the initial and final assignments are equal. Similarly, at most one out of the K variables of an isolated clause can take a different value in the two configurations. It would be interesting to understand which part of the formula decorrelates in the vicinity of α_d.

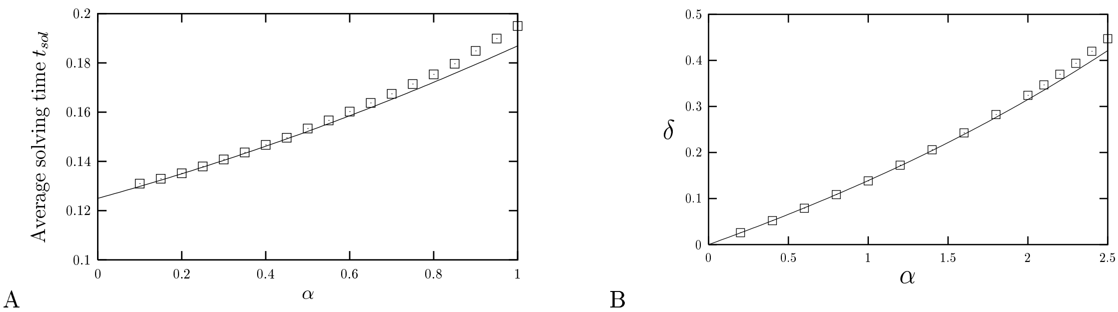

Fig. 4. A: Average solving time $t_{sol}(\alpha, 3)$ for PRWSAT on 3-SAT. Symbols: numerical simulations, averaged over $1,000$ runs for $N = 10,000$. Solid line: prediction from Eq. (3). B: Average Hamming distance between initial configuration and solution found $\delta(\alpha, 3)$ for PRWSAT on 3-SAT. Symbols: numerical simulations, averaged over 100 runs for $N = 5000$. Solid line: prediction from Eq. (4).

5 Analysis of the exponential phase

The method developed above cannot be used for the study of the regime $\alpha_d < \alpha < \alpha_c$, where formulas are satisfiable but the finding of a solution happens through a fluctuation of the number of unsatisfied clauses from its typical value. We now introduce an approximate method adapted to this situation.

5.1 The simplifying approximation

Let us denote by $\mathcal{C}$ an assignment of the Boolean variables. PRWSAT defines a Markov process on the set of configurations $\mathcal{C}$, of cardinality 2^N. Tracking the set of probabilities $\text{Prob}[\mathcal{C},T]$ of each configuration $\mathcal{C}$ as a function of the number of steps T of the algorithm is a formidable task. We will therefore consider a simplified version of the true PRWSAT Markov chain, which retains some essential features of the original process and is much easier to understand analytically. The crucial quantity to consider is the number of clauses unsatisfied by the current assignment of the boolean variable, $M_0(\mathcal{C})$. When $M_0(\mathcal{C})$ vanishes, the algorithm has found a solution and stops. The simplifying approximation that we shall make is the following:

Approximation: after T steps of PRWSAT, the probability that the Boolean configuration is $\mathcal{C}$ is a function of $M_0(\mathcal{C})$ only.

In other words, at each time step, configurations with the same number of unsatisfied clauses are equally likely. This statement cannot be true. Indeed, the Hamming distance between two configurations visited at step T and $T+k$ of the algorithm is at most k, a constraint not enforced in the approximate process. Strikingly, the results obtained within this approximation are much more sensible from a quantitative point of view that one could fear.

Within this simplifying approximation, we can write down the transition matrix A which determines the probability $\text{Prob}[M_0,T+1]$ that M_0 clauses are unsatisfied at step $T+1$ in terms of its counterpart $\text{Prob}[M_0,T]$ at step T,

$$\text{Prob}[M_0',T+1] = \sum_{M_0=0}^{M} A_{M_0'\,M_0}\,\text{Prob}[M_0,T] \quad , \tag{5}$$

$$A_{M_0'\,M_0} = \sum_{Z_u=0}^{M_0}\sum_{Z_s=0}^{M-M_0} \frac{Z_u N}{K M_0}\binom{M_0}{Z_u}\binom{M-M_0}{Z_s}\left(1-\frac{K}{N}\right)^{M_0-Z_u}\left(\frac{K}{N}\right)^{Z_u}\times$$
$$\left(1-\frac{K}{(2^K-1)\,N}\right)^{M-M_0-Z_s}\left(\frac{K}{(2^K-1)\,N}\right)^{Z_s}\mathbf{1}_{M_0'-M_0+Z_u-Z_s} \tag{6}$$

where $\mathbf{1}_X = 1$ if $X=0$, 0 otherwise. Z_u is the number of unsatisfied clauses which contains the variable to be flipped. All these clauses will become satisfied after the flip. The factor $Z_u N/(KM_0)$ represents the probability of flip of the variable. Z_s is the number of clauses satisfied prior to the flip and violated after. The origin of the Binomial laws is the following. Assume that the configuration violates M_0 clauses. In the absence of further information, the variable which is going to flip has probability K/N to be present in a given clause (there are $\binom{N}{K}$ possible K-uplets over N variables, and $\binom{N-1}{K-1}$ which include a given variable). Furthermore, a satisfied clause that contains the flipped variable has a probability $1/(2^K-1)$ to become unsatisfied later. Z_u (respectively Z_s) is thus drawn from a binomial law with parameter K/N (resp. $K/(N(2^K-1))$) , over M_0 (resp. $M-M_0$) tries.

As a side remark, let us comment on the similitudes and differences between our approach and the one initiated by Schöning [1]. In Schöning's work, the quantity monitored during the execution of PRWSAT is the Hamming distance d between the current assignment $\mathcal{C}$ of the variables and an arbitrary solution of the formula. Rigorous bounds on the transition probabilities for the evolution of d between steps T and $T+1$ are then established. For typical random instances, the bound on the running time of the algorithm is far from being optimal : as we have seen before, for $\alpha < \alpha_d$ it is only linear, and for $\alpha > \alpha_d$ it is exponential but with a growth coefficient much lower than 4/3. This discrepancy is partly due to the fact that there are exponentially numerous solutions, whereas Schöning's bound takes into account only one of them. In a sense the quantity we focus on, that is, the number M_0 of unsatisfied clauses, bypasses this limitation. It would however be much more difficult to obtain rigorous bounds on the transition probabilities between different M_0, which can be true only for typical random instances. It is understood that the two approaches are of different nature, one interested in average-case (over an ensemble of formulas) and the other in worst-case (true for any formula) behaviour of the same probabilistic algorithm.

5.2 Average fraction of unsat clauses

Let us analyze the consequences of the above approximation on the average performances of the algorithm. We define the average fraction of unsat clauses at reduced time $t = T/M$ through

$$\varphi_0(t) = \frac{1}{M} \sum_{M_0=0}^{M} M_0 \ \mathrm{Prob}[M_0, T = tM] \quad . \tag{7}$$

Evolution equation (5) yields the following first-order linear differential equation,

$$\frac{d\varphi_0}{dt} = -1 + \frac{\alpha K}{2^K - 1} - \frac{\alpha K}{1 - 2^{-K}} \, \varphi_0 \ . \tag{8}$$

This equation, together with the initial condition $\varphi_0(0) = 2^{-K}$ (the initial configuration of the variables is random, thus each of the clauses is unsatisfied with probability 2^{-K}), can be easily solved. There appears a critical value of α,

$$\alpha_d(K) = \frac{2^K - 1}{K} \quad . \tag{9}$$

For $\alpha < \alpha_d(K)$, $\varphi_0(t)$ vanishes for a finite value of its argument, a solution is typically found in linear time. For larger values of α, φ_0 remains positive at all times, which means that typically a large formula will not be solved by PRWSAT, and that the fraction of unsat clauses on the plateau will be the limit when $t \to \infty$ of $\varphi_0(t)$. The predicted value for $K = 3$, $\alpha_d = 7/3$, is in good but not perfect agreement with the estimates from numerical simulations, around 2.7. The fraction of unsatisfied clauses on the plateau reads

$$\varphi_0(t \to \infty) = \frac{1}{2^K} \left(1 - \frac{\alpha_d(K)}{\alpha} \right) \quad , \tag{10}$$

and is shown on Fig. 2.

5.3 Large deviations and solving time

As explained above, when $\alpha > \alpha_d$, the system gets almost surely trapped in a portion of the configuration space with a non zero number of unsatisfied clauses. Numerical experiments indicate the existence of an exponentially small-in-N probability $\sim \exp(N\,\bar{\zeta}(\alpha))$ with $\bar{\zeta} < 0$ for the escape of the trap and the discovery of a solution in linear time[4].

We now make use of our approximation to derive an approximate expression for $\bar{\zeta}$. Contrary to the previous section, we now consider the large deviation of the process with respect to its typical behavior. This can be accessed through the study of the large deviation function $\pi(\varphi_0, t)$ of the fraction φ_0,

$$\pi(\varphi_0, t) = \lim_{N\to\infty} \frac{1}{N} \ln \mathrm{Prob}\big[M_0 = M\,\varphi_0, T = t\,M\big] \tag{11}$$

To compute the large deviation function π, we introduce the generating function of M_0 [13],

$$G[y, T] = \sum_{M_0=0}^{M} \mathrm{Prob}[M_0, T]\ \exp(y\,M_0) \quad , \tag{12}$$

where y is a real-valued number. G is expected to scale exponentially with N with a rate

$$g(y, t) \equiv \lim_{N\to\infty} \frac{1}{N} \ln G[y, T = t\,M] = \max_{\varphi_0}[\pi(\varphi_0, t) + \alpha\,y\,\varphi_0] \ , \tag{13}$$

equal to the Legendre transform of π from insertion of definition (11) into Eqn. (12). Using evolution equations (5,6), we obtain the following equation for g,

$$\frac{1}{\alpha}\frac{\partial g(y,t)}{\partial t} = -y + \frac{\alpha K}{2^K - 1}(e^y - 1) + K\left(e^{-y} - 1 - \frac{1}{2^K - 1}(e^y - 1)\right)\frac{\partial g(y,t)}{\partial y} \ . \tag{14}$$

[4] This statement is directly related to Schöning's bound on the number of "trials" necessary to reach a solution. In practice, PRWSAT is run once, until a solution is found, which happens after an exponentially large time. Intuitively, the two statements "Having an exponentially small probability $\sim \exp(N\,\bar{\zeta})$ of finding a solution in $O(N)$ steps" or "Finding a solution in $\sim \exp(-N\,\bar{\zeta})$ steps (up to a non-exponential multiplicative factor) with high probability" are equivalent.

along with the initial condition

$$g(y,0) = \alpha \ln \left(1 - \frac{1}{2^K} + \frac{e^y}{2^K} \right) \quad . \tag{15}$$

The average evolution studied in the previous section can be found again from the location of the maximum of π or, equivalently, from the derivative of g in $y = 0$: $\varphi_0(t) = (1/\alpha)\, \partial g/\partial y(0,t)$. The logarithm of the probability that a solution is reached after a $O(1)$ time ($O(N)$ flips) is given by

$$\bar{\zeta}(\alpha, K) = \lim_{t \to \infty} \pi(\varphi_0 = 0, t) = \int_0^{\tilde{y}(\alpha)} dy\, z(y,\alpha) \ , \tag{16}$$

where $z(y,\alpha) = (y - \alpha K(e^y - 1)/(2^K - 1))/(K(e^{-y} - 1 - (e^y - 1)/(2^K - 1)))$ and $\tilde{y}(\alpha)$ is the negative root of z.

Predictions for $\bar{\zeta}$ in the $K = 3$ case are plotted in Fig. 5. They are compared to experimental measures of ζ, that is, the logarithm (divided by N) of the average solving times. It is expected on intuitive grounds exposed in Sec. 3 that ζ coincides with $-\bar{\zeta}$. Despite the roughness of our approximation, theoretical predictions are in qualitative agreement with numerical experiments.

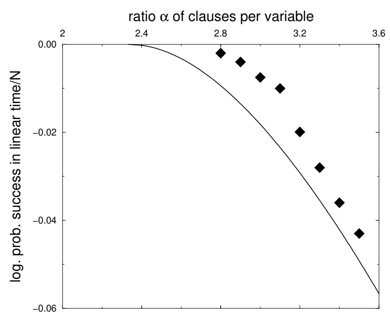

Fig. 5. Large deviations for the 3-SAT problem. The logarithm $\bar{\zeta}$ of the probability of successful resolution (over the linear in N time scale) is plotted as a function of the ratio α of clauses per variables. Prediction for $\bar{\zeta}(\alpha, 3)$ has been obtained within the approximation of Sec. 5. Diamonds corresponds to (minus) the logarithm ζ of the average resolution times (averaged over 2,000 to 10,000 samples depending on the values of α, N, divided by N and extrapolated to $N \to \infty$) obtained from numerical simulations. Error bars are of the order of the size of the diamond symbol. As expected, Schöning's bound $\bar{\zeta} \geq \ln(3/4) \simeq -0.288$, true for any satisfiable instance, lies below the estimate of $\bar{\zeta}$ computed for random 3-SAT instances for all ratios $\alpha < \alpha_c$.

A similar study of the behavior of PRWSAT on XORSAT problems has been also performed in [9, 10], with qualitatively similar conclusions: there exists a dynamic threshold α_d for the algorithm, smaller than the satisfiability threshold (known exactly in this case [14–16]). For lower values of α, the number of steps needed to find a solution is linear in the size of the formula, between α_d and α_c solutions are found on exponentially large time scales, through fluctuations around a plateau value for the number of unsatisfied clauses. In the XORSAT case, the agreement between numerical experiments and this approximate study (which predicts $\alpha_d = 1/K$) is quantitatively better and seems to improve with growing K.

6 Conclusion and perspectives

We have studied the dynamics of a simple local search procedure for the satisfaction of Boolean constraints, the Pure Random Walk algorithm. Use of complementary techniques (expansions and approximations) has allowed us to show that, for randomly drawn input instances, PRWSAT may have two qualitatively distinct behaviors. Instances with small ratios α of clauses per variable are almost surely solved in a time growing linearly with their size [2]. On the contrary, for ratios above some "dynamical" threshold α_d, the algorithm gets trapped for an exponentially large time in a region of the assignment space with a positive fraction of unsatisfied clauses. A solution is finally reached through a large "fluctuation" from this trap.

It would be of course of the greatest interest to confirm rigorously the results presented here. At first sight, the low α phase seems easier to tackle [2], due to the apparent concentration of the solving time in the large size limit and to the simplifications occurring in the neighborhood of $\alpha = 0$, as explained in Sec. 4. Another possible direction would be to consider the large K limit, some arguments suggesting that the very crude approximation of Sec. 5 becomes exact in this limit [9]. For "static" properties e.g. the value of the satisfiability threshold, it was recently rigorously shown that the simple "annealed" procedure of statistical physics becomes indeed exact in this limit [17], as was conjectured in [18]. Finally, a challenging problem would be to prove bounds on the transition probabilities for the number of unsatisfied clauses, in the spirit of Schöning's work [1], true w.h.p. on the random distribution of instances.

The PRWSAT local search procedure studied above is a particular case of a general algorithmic architecture known as WalkSAT [19]. In general, one still chooses randomly an unsatisfied clause at each step in an uniform way, but the choice of the variable to be flipped inside this clause is done following a specific heuristic. Many different strategies can, and have, been proposed (see for instance [20, 11]). One can classify them into two categories. On the one hand, some of them are "Markovian": the choice of the variable to be flipped requires only the knowledge of their current assignment. For instance, one can choose the variable whose flip will lead to the greatest decrease of the number of unsatisfied clauses (WSAT/G in the classification of [20]), or the one which belongs to the minimal number of clauses only satisfied by the variable to be flipped (WSAT/B). On the other hand, another family of heuristics introduces "memory effects": besides the current status of the boolean variables, one bookkeeps the *age* of each of them, namely the time at which it was flipped for the last time. Heuristics in this family (for instance TABU, NOVELTY and R-NOVELTY) use this age information to choose the variable to be flipped. Most of these heuristics have a tunable parameter that, roughly speaking, controls the amount of randomness in the execution of the algorithm. It has been often empirically observed that this parameter presents an optimum value [20, 11]. With the picture of fluctuations around a plateau in mind, the claims of [20] become clearer: it is intuitive that to reach a solution quickly, one would have to favor at the same time a low value of the plateau height, and a large intensity of the fluctuations around it. A compromise must be found between these two quantities to have an optimal tuning of the algorithm. The expansion method for the average solving time in the linear regime can be easily extended to the "Markovian" heuristics. One could thus justify the existence of an optimal tuning, at least in the small α regime.

Further studies of these heuristics is necessary; the first point to investigate is the generality of the scenario presented here, with the appearance of a plateau above some ratio α_d and resolution happening through fluctuations. Next, the dependence of α_d with the the features of the local search heuristic (randomness parameter, ...) should be clarified. This question is all the more of interest that some local search algorithms seem to be able to solve random formulas in a linear number of flips up to, or very close to the sat/unsat threshold α_c [21].

Acknowledgements: We thank S. Cocco, L. Cugliandolo, C. Deroulers, O. Dubois, C. Girit and A. Montanari for numerous and stimulating discussions. Partial financial support was provided by the French Ministry of Research through the ACI Jeunes Chercheurs "Algorithmes d'optimisation et systèmes quantiques désordonnés".

References

1. U. Schöning, A Probabilistic algorithm for k-SAT based on limited local search and restart, *Algorithmica* **32**, 615-623 (2002).
2. M. Alekhnovich, E. Ben-Sasson, Analysis of the Random Walk Algorithm on Random 3-CNFs, preprint (2002).
3. S. Cocco, A. Montanari, R. Monasson, G. Semerjian, Approximate analysis of search algorithms with "physical" methods, *cs.CC/0302003*.
4. D. Mitchell, B. Selman, H. Levesque, Hard and Easy Distributions of SAT Problems, *Proc. of the Tenth Natl. Conf. on Artificial Intelligence (AAAI-92)*, 440-446, The AAAI Press / MIT Press, Cambridge, MA (1992).
5. C.H. Papadimitriou, On Selecting a Satisfying Truth Assignment, *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, 163-169 (1991).
6. T. Hofmeister, U. Schöning, R. Schuler and O. Watanabe, A probabilistic 3-SAT algorithm further improved, *Proceedings of 19th Symposium on Theoretical Aspects of Computer Science*, *LNCS* 2285, 192-203 (2002).
7. S. Baumer and R. Schuler, Improving a probabilistic 3-SAT algorithm by dynamic search and independant clause pairs, in this volume.

8. A.J. Parkes, Scaling Properties of Pure Random Walk on Random 3-SAT, *Lecture Notes in Computer Science* **2470**, 708 (2002).
9. G. Semerjian, R. Monasson, Relaxation and metastability in a local search procedure for the random satisfiability problem, *Phys. Rev. E* **67**, 066103 (2003).
10. W. Barthel, A.K. Hartmann, M. Weigt, Solving satisfiability problems by fluctuations: The dynamics of stochastic local search algorithms, *Phys. Rev. E* **67**, 066104 (2003).
11. H. H. Hoos, T. Stützle, Local Search Algorithms for SAT: An Empirical Evaluation, *J. Automated reasoning* **24**, 421 (2000).
12. G. Semerjian, L.F. Cugliandolo, Cluster expansions in dilute systems: applications to satisfiability problems and spin glasses, *Phys. Rev. E* **64**, 036115 (2001).
13. S. Cocco, R. Monasson, Restarts and exponential acceleration of random 3-SAT instances resolutions: a large deviation analysis of the Davis-Putnam-Loveland-Logemann algorithm, to appear in *Annals of Mathematics and Artificial Intelligence* (2003)
14. O. Dubois, J. Mandler, The 3-XORSAT threshold, *Proc. of the 43rd annual IEEE symposium on Foundations of Computer Science*, Vancouver, 769–778 (2002).
15. S. Cocco, O. Dubois, J. Mandler, R. Monasson, Rigorous decimation-based construction of ground pure states for spin glass models on random lattices, *Phys. Rev. Lett.* **90**, 047205 (2003).
16. M. Mézard, F. Ricci-Tersenghi, R. Zecchina, Alternative solutions to diluted p-spin models and XORSAT problems, *J. Stat. Phys.* **111**, 505 (2003).
17. D. Achlioptas, Y. Peres, The Threshold for Random k-SAT is $2^k(\ln 2 + o(1))$, preprint (2002).
18. R. Monasson, R. Zecchina, Statistical mechanics of the random K-satisfiability model, *Phys. Rev. E* **56**, 1357 (1997).
19. B. Selman, H. Kautz and B. Cohen, Noise Strategies for Improving Local Search, *Proc. AAAI-94*, Seattle, WA, 337-343 (1994).
20. D. McAllester, B. Selman, H. Kautz, Evidence for Invariants in Local Search, *Proc. AAAI-97*, Providence, RI (1997).
21. S. Seitz, P. Orponen, An efficient local search method for random 3-Satisfiability, preprint (2003).