

On Dynamics of Stochastic Local Search

Aapo Nummenmaa

Laboratory of Computational Engineering

Slide 1

Helsinki University of Technology

Introduction: Aspects of Equilibrium and Nonequilibrium Phenomena

- Equilibration of a (physical, thermodynamic, statistical mechanics, dynamical) system could be characterised by the time-independent behaviour of some quantities of interest.
 - A system of pointlike masses interacting in a Newtonian way: in the absence of external forces, the center of mass moves with constant velocity.
 - A converged Markov chain: the states of the chain are distributed according to the stationary distribution.
 - Statistical mechanics: the probability distribution of states in equilibrium doesn't change:

$$p_i \propto \exp(-\beta E_i), \quad (1)$$

where E_i is the energy of the i :th state and β is the inverse temperature.

Slide 2

Slide 3

- The fluctuations in equilibrium are also of great interest (correlations, phase transitions *etc.*).
- What happens if a system is perturbed out of the equilibrium?
 - Since the equilibrium state is a stationary one, it might not be too far-fetched to think that the system simply returns to the equilibrium if left alone.
 - Usually this is the case (at least for small perturbations), but not always. Adding enough interactions and/or nonlinearities, even a structurally very simple system might show extremely complicated behaviour (strange attractors, chaos, metastable states *etc.*).
 - Quantifying **how** a system approaches the equilibrium (if there is an equilibrium state to begin with) seems therefore much more difficult a task than characterising the properties of the system when in equilibrium (which is not an easy task itself).
- These issues are exemplified by the following simple Metropolis sampling from

Laplacian and Gaussian distributions.

Slide 4

Slide 5

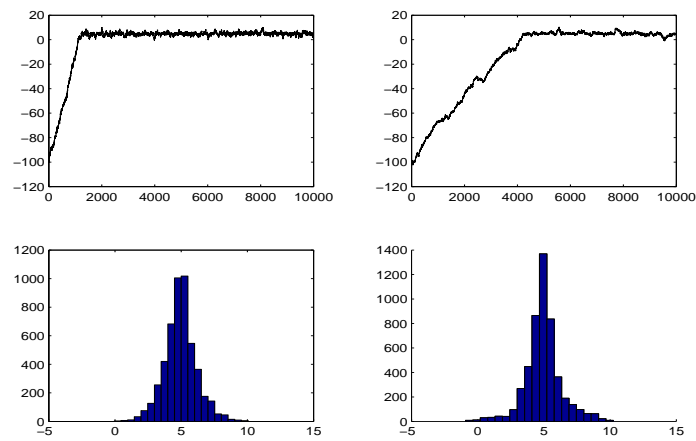


Figure 1: Sampling from Laplacian distribution with Metropolis update rule. Results are shown for two Gaussian proposal distributions with standard deviations of 0.5 (left) and 0.25 (right). 5000 last samples are used to draw the histograms.

Slide 6

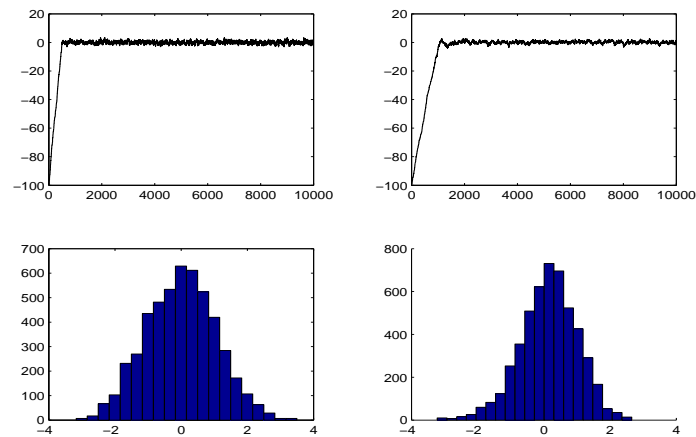


Figure 2: Sampling from Gaussian distribution with Metropolis update rule. Results are shown for two Gaussian proposal distributions with standard deviations of 0.5 (left) and 0.25 (right). 5000 last samples are used to draw the histograms.

- From these examples, the following remarks could be made:
 - The “path” to the equilibrium seems to be roughly linear independent of the target pdf (Laplacian vs. Gaussian).
 - The slope of the (linear) path depends on the details of the algorithm (width of the proposal distribution) and the starting point ($x = -100$ in this case).
 - The fluctuations in equilibrium (*i.e.* how fast the chain “scans” the stationary distribution) also depend on the details of the algorithm.
 - Would it be possible to quantitatively predict this behavior given the algorithmic details and the structure of the state-space (nonequilibrium dynamics)?
- In this presentation, an attempt is made to cast some light on these questions for a specific system (K -XOR)-SAT) and a specific stochastic algorithm (walk-SAT). All the results will be *a priori* characteristic to this system only. See the references for the original presentation(s) of these issues.

Slide 7

- NOTE: The nature of the K -XOR)-SAT/walk-SAT problem is such that the analogy with the simple MCMC-simulation example is far from perfect; it still might be helpful to keep in mind that we’re trying to understand (qualitatively & quantitatively) how the algorithm makes the system evolve in time (*i.e.* its dynamics). This issue is discussed further in the next section.

Slide 8

A Glimpse of K (-XOR)-SAT and walk-SAT

Slide 9

- Elements of a random K -satisfiability formula:
 - M logical clauses $\{C_\mu\}_{\mu=1,\dots,M}$ defined over N Boolean variables $\{x_i = 0, 1\}_{i=1,\dots,N}$, where $0 = \text{FALSE}$, $1 = \text{TRUE}$.
 - Every clause contains K randomly chosen Boolean variables that are connected by logical OR operations and appear negated with probability $1/2$; for example $C_\mu = (x_i \vee \bar{x}_j \vee x_k)$.
 - In the final formula all such clauses are connected by logical AND operations:

$$F = \bigwedge_{\mu=1}^M C_\mu. \quad (2)$$

- Such a formula is thus satisfied iff each of the clauses has a correct assignment for at least one variable.

Slide 10

- Some established facts about the K -SAT:
 - For $K = 2$ the problem is easy and a polynomial-time solving algorithm exists.
 - For $K \geq 3$ the problem is NP-complete and so it is expected that no efficient polynomial-time solvers for generic K -SAT formulas exist.
 - For $\alpha < 4.26$, $\alpha := M/N$, N sufficiently large, almost all 3-SAT formulas are found to be satisfiable. When $\alpha > 4.26$ all formulas are found to be unsatisfiable with probability one, as $N \rightarrow \infty$. This “phase transition” coincides with a strong peak in the algorithmic solution times of the complete solver algorithms.
 - A second phase transition for $K = 3$ occurs within the satisfiable phase when the solution space breaks from an exponentially large cluster into an exponential number of clusters at $\alpha = 3.92$.
- Similar, but analytically simpler model K -XOR-SAT:

- The variables that appear in the clauses are now connected with logical XOR operations (\oplus).
- Such a clause is then satisfied iff an odd number of variables is assigned correctly.
- This can be used to map the problem into a linear equation (mod 2), and thus solved in $\mathcal{O}(N^3)$ steps by a global algorithm (e.g. Gaussian elimination).
- However, if local algorithms are used, similar phenomena occur for 3-XOR-SAT as for 3-SAT: transition from sat-regime to unsat-regime at $\alpha = 0.981$; in regime $0.818 < \alpha < 0.981$ formulas are satisfiable a.s., but the solution space decays into an exponential number of clusters.

Slide 11

- The walk-SAT algorithm for solving SAT-problems:
 - 1) Assigning all N variables randomly; then there will be $\alpha_s N$ satisfied and $\alpha_u N = (\alpha - \alpha_s)N$ unsatisfied clauses.
 - 2) Select an unsatisfied clause C randomly and 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 in the least number of satisfied clauses (greedy step).

- 3) Invert the current assignment of v^* . All clauses containing v^* that were unsatisfied become satisfied. Clauses containing v^* that were satisfied behave differently for K -SAT and K -XOR-SAT: For K -SAT a previously satisfied clause becomes unsatisfied iff v^* was the only correctly assigned variable in this clause. For K -XOR-SAT, every previously satisfied clause containing v^* becomes unsatisfied.

Slide 12

- 4) Repeat 2) and 3) until all clauses become satisfied or some upper limit on running time is reached.
- Comments about walk-SAT:
 - Walk-SAT isn't guaranteed to find a solution (in a finite time) even if the formula is satisfiable.
 - There are many variants of the greedy step: "select the variable in C leading

to minimal number of unsatisfied clauses (maximal gain)” or “select the variable in C minimizing the number of previously satisfied clauses that become unsatisfied (minimal negative gain)”

- Restarting the algorithm after $3N$ steps leads to exponential acceleration for pure random walk ($q = 1$) dynamics.
- The walk-SAT algorithm “induces” a stochastic process on the state-space $\{0, 1\}^N$ which is quite obviously a Markov chain. This is nevertheless not of the standard form, since it is not ergodic (probability going from a solution state to nonsolution state is zero, since the algorithm stops there). Thus the questions about stationary distributions, convergence etc. are somewhat ill-posed.
- In this presentation these nonequilibrium issues are treated analytically in a very hand-waving way (and mostly just for pure (random) walk-SAT and/or K -XOR-SAT).

Slide 13

Numerical Results by Figures

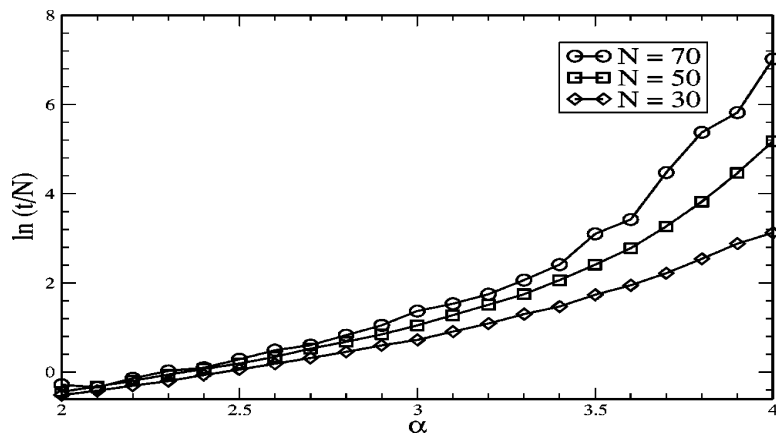
- We are looking for the time-behaviour of $\alpha_u(t)$ (number of unsatisfied clauses per variable under walk-SAT); this could be thought as an energy density for the system.
- NOTE: In the following, time is measured in MC sweeps (*i.e.* $\Delta t = 1/N$).
- Phenomenology is roughly speaking the following (3-SAT, $N \gg 1$, pure walk dynamics):
 - The algorithm starts with a significant fraction of unsatisfied clauses;
 $\alpha_u(0) = (M/8)/N = \alpha/8$ almost surely.
 - 1) For $\alpha < \alpha_d \approx 2.7$ (α_d = “dynamical threshold”), a solution is found after a finite number of MC sweeps (*i.e.* $\alpha_u(t)$ becomes zero at finite MC times).
 - 2) For $\alpha > \alpha_d$, the energy density $\alpha_u(t)$ initially decreases and equilibrates

Slide 14

to a nonzero plateau value. For larger times $\alpha_d(t)$ fluctuates around this plateau value, and reaches zero if the formula is satisfiable (such a fluctuation has an exponentially small probability).

- Introducing good heuristics (greedy steps) can make the plateau energy lower and hence the fluctuation needed to reach zero more feasible.
- For K -XOR-SAT ($K = 3, q = 1$) the behaviour is similar ($\alpha_d \approx 0.33$).

Slide 15



Slide 16

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

Slide 17

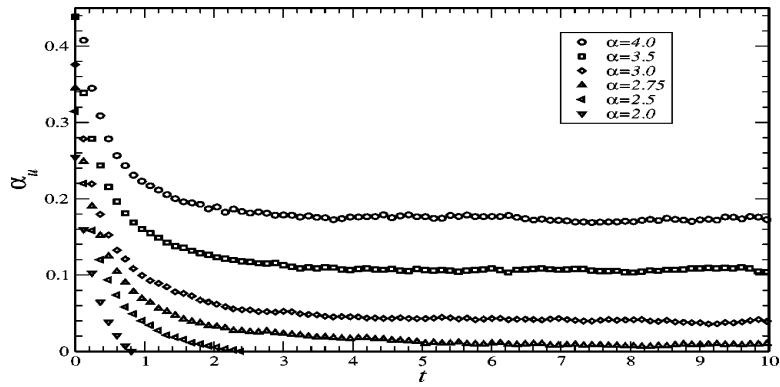


FIG. 2. 3-SAT: average number α_u of unsatisfied clauses per variable with sample size $N=50\,000$. Initially, this energy density quickly decreases. For $\alpha < \alpha_d \approx 2.7$, it becomes zero after a finite time, for larger α a nonzero plateau is reached.

Slide 18

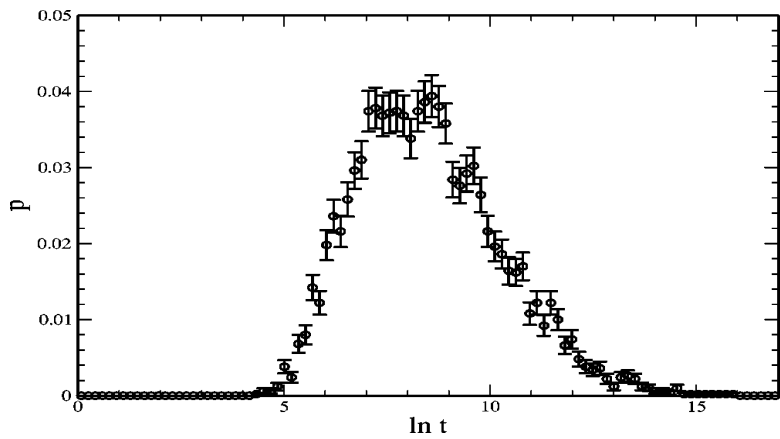


FIG. 3. 3-SAT: Histogram of the logarithm of the running times of walk-SAT without restarts for $\alpha=3.5$ and $N=100$.

Slide 19

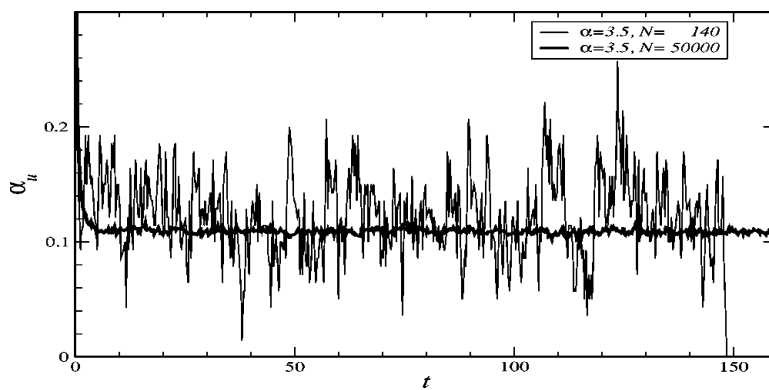


FIG. 4. After the initial decrease α_u fluctuates around its plateau value. Two different system sizes are shown. For the smaller one with $N=150$ a fluctuation after about 145 MC steps was large enough to reach a solution of the formula.

Slide 20

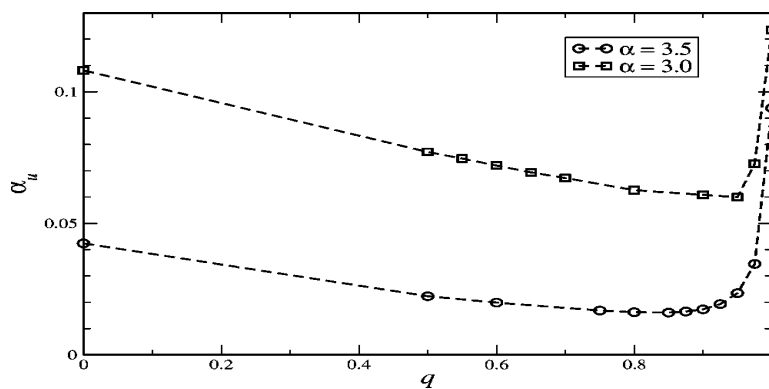


FIG. 5. 3-SAT: Plateau energy for $\alpha=3.5$ and $\alpha=3.0$ depending on the fraction q of greedy steps performed by the algorithm. The plateau energy is minimal for $q=0.95$ ($\alpha=3.5$) and $q=0.85$ ($\alpha=3.0$).

Rate-Equation Approach: Notations

- The main idea: characterize each variable only by the number of satisfied and unsatisfied clauses it is contained in.
- Hence subdivide the set of all N Boolean variables at time t into subsets of $N_t(s, u)$ variables belonging to s satisfied and u unsatisfied clauses.

Slide 21

- Then at time t randomly selected variable is in the set characterized by s, u with probability $p_t(s, u) := N_t(s, u)/N$.
- This probability $p_t(s, u)$ is changed in the course of walk-SAT, but for all variables $s + u$ remains constant because it counts the total number of clauses (M).
- One can then compute the total (expected) number of unsatisfied clauses

$N\alpha_u(t)$:

$$\alpha_u(t) = \frac{\langle u \rangle_t}{K}, \quad (3)$$

where $\langle \cdot \rangle_t = \sum_{s,u} (\cdot) p_t(s, u)$ and the factor K comes from the fact that when summing over variables each clause is counted K times.

Slide 22

- The walk-SAT algorithm doesn't select variables according to $p_t(s, t)$ but chooses randomly an unsatisfied clause C^* and flips one of its variables v^* according to a chosen heuristic (walk or greedy step).
- The probability that this variable v^* belongs to exactly s satisfied and u unsatisfied clauses is denoted by $p_t^{(flip)}(s, u)$ and can be calculated from $p_t(s, u)$ assuming the [independency of neighboring sites](#) (i.e. the joint distribution of K variables being ^a in one unsatisfied clause factorizes).

^aThe original article says "three variables being"...

- This assumption fails and becomes an approximation for other than the initial configurations, but it allows a description of the walk-SAT dynamics in terms of $p_t(s, u)$.
- For a walk step, variable v^* is randomly selected from a random unsatisfied clause C^* . Given s and u there appears to be $uN_t(s, u)$ possibilities for doing this (assuming independent neighbors?).

Slide 23

- By normalization, one obtains

$$p_t^{(flip-walk)}(s, u) = \frac{u p_t(s, u)}{\langle u \rangle_t} =: p_t^{(u)}(s, u). \quad (4)$$

- For greedy steps, an expression for $p_t^{(flip-K-greedy)}(s, u)$ can be derived in terms of $p_t^{(u)}(s, u)$, but these expressions are rather cumbersome.
- For $K = 2$ we have

$$p_t^{(flip-2-greedy)}(s, u) = \sum_{s_1, u_1, s_2, u_2} p_t^{(u)}(s_1, u_1) p_t^{(u)}(s_2, u_2) \times [\delta_{(s_1, u_1), (s, u)} \Theta(s_2 - s_1) + \delta_{(s_2, u_2), (s, u)} \Theta(s_1 - s_2)],$$

where δ is the Kronecker delta and Θ is the Heaviside function with $\Theta(0) = 1/2$.

Slide 24

- Generally then,

$$p_t^{(flip)}(s, u) = q p_t^{(flip-walk)}(s, u) + (1 - q) p_t^{(flip-K-greedy)}(s, u), \quad (5)$$

but the main idea of the final analysis doesn't depend (fortunately) on the details of the flipping probability.

Interlude: the Poissonian Ansatz

- Let us suppose $q = 1$ and that the variables s and u are independently distributed in a Poissonian way for all times t :

$$p_t(s, u) = e^{-K\alpha} \frac{[K\alpha_s(t)]^s [K\alpha_u(t)]^u}{s!u!}. \quad (6)$$

Slide 25

- This is strictly valid only at time $t = 0$ and deviations appear at larger times.
- Thus on average each variable is contained in $K\alpha_s(t) = K(\alpha - \alpha_u(t))$ satisfied and $K\alpha_u(t)$ unsatisfied clauses.
- Plugging the Poissonian ansatz to the equation (4) one gets

$$p_t^{(flip)}(s, u) = e^{-K\alpha} \frac{[K\alpha_s(t)]^s [K\alpha_u(t)]^{u-1}}{s!(u-1)!}, \quad (7)$$

which is again a Poissonian distribution of s and $u - 1$.

- Hence, on average the flipped variable is contained in $K\alpha_s(t)$ satisfied and $K\alpha_u(t) + 1$ unsatisfied clauses.
- Let us apply this to the case of K -XOR-SAT: by flipping a variable v^* all s satisfied clauses containing v^* become unsatisfied whereas all u unsatisfied ones become satisfied.

Slide 26

- The expected number of unsatisfied clauses N_t^u changes during one step as

$$\Delta N_t^u = -[K\alpha_u(t) + 1] + K\alpha_s(t) = K\alpha - 2K\alpha_u(t) - 1. \quad (8)$$

- Going to the large N limit (where fluctuations become negligible) we have $N_t^u = N\alpha_u(t)$. Measuring time in MC-time $\Delta t = 1/N$

and replacing the difference by derivative one gets

$$\dot{\alpha}_u(t) = K\alpha - 2K\alpha_u(t) - 1. \quad (9)$$

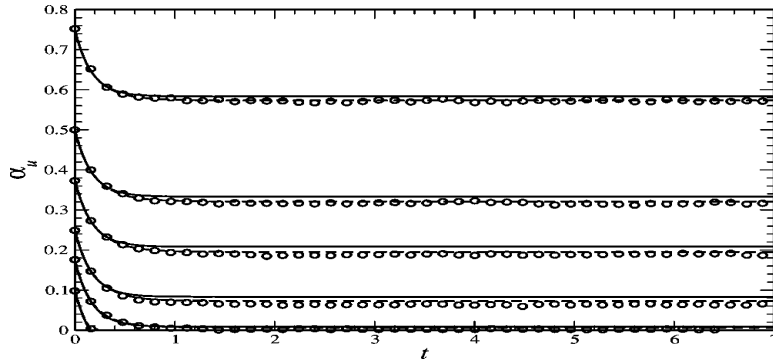
- Integrating this differential equation yields

$$\alpha_u(t) = \frac{1}{2K}(K\alpha - 1 + Ce^{-2Kt}). \quad (10)$$

Slide 27

- Assuming that in a typical starting configuration half of the clauses are satisfied and half are not (*i.e.* $\alpha_u(0) = \alpha/2$) one gets

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



Slide 28

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.

- From this simple Poissonian approximation one gets $\alpha_d = 1/K$. For $K = 3$ this value 0.333 . . . agrees perfectly with numerical simulations.

Slide 29

Rate-Equation for K -XOR-SAT

- There are some systematic deviations from the Poissonian behaviour and so it is necessary to go beyond the Poissonian approximation (especially if greedy steps are included).
- However, the assumption of independent neighboring sites is still built-in to the analysis.
- Only the K -XOR-SAT-case is considered here.
- Reflect the following: (I quote Barthele *et al.*) “ 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 time t ”.
- At first glance this seems to be serious abuse of notation that is (unfortunately)

Slide 30

very typical for physicists.

- This abuse was done also when discussing the Poissonian ansatz.
- In my mind the resolution is following: when going to the very large N limit, the probability distribution $p_t(s, u)$ becomes very highly peaked around its mean value w.r.t time t (i.e. the history of the walk-SAT), and so we can say loosely that "on the average" this and that happen and forget that $p_t(s, u)$ in fact depends on the specific "run" of walk-SAT.

Slide 31

- Anyways, a variable v^* with s^* satisfied and u^* unsatisfied clauses is flipped at time t ; this happens with probability $p_t^{(flip)}(s^*, u^*)$.
- Contributions to $N_{t+\Delta t}(s, u)$ come from the following three processes (My explanation differs slightly from Barthel's):
 - 1) **The contribution from v^* itself** : The flipped variable belongs to s satisfied and u unsatisfied clauses with probability $p_t^{(flip)}(s, u)$. Hence $N_t(s, u)$

reduces (on the average) by amount $1 \cdot p_t^{(flip)}(s, u)$. Now recall that for K -XOR-SAT all the unsatisfied clauses containing v^* become satisfied and vice versa. The flipped variable belongs to u **satisfied** and s **unsatisfied** clauses with probability $p_t^{(flip)}(u, s)$. This adds to $N_t(s, u)$ (again on the average) an amount of $1 \cdot p_t^{(flip)}(u, s)$.

- 2) **Neighbors of v^* in previously satisfied clauses**: The flipped variable occurs (on an average) in $\langle s \rangle_t^{(flip)}$ previously satisfied clauses, where $\langle \cdot \rangle_t^{(flip)} = \sum_{s,u} (\cdot) p_t^{(flip)}(s, u)$. Since each clause contains K variables, and since random formulas are locally treelike there are (on an average) $(K - 1)\langle s \rangle_t^{(flip)}$ such neighbors in previously unsatisfied clauses. All these clauses become unsatisfied. Therefore, for these variables the number of satisfied clauses they are contained in goes down by one (and the number of unsatisfied clauses increases by one). According to the "independent neighbors"-assumption, these variables belong to s satisfied and u unsatisfied clauses with probability $sp_t(s, u)/\langle s \rangle_t$. Therefore, this process subtracts (on

Slide 32

an average) an amount of $(K - 1)\langle s \rangle_t^{(flip)} s p_t(s, u) / \langle s \rangle_t$ from $N_t(s, u)$. Now, however, these variables don't vanish from the system but are added to $N_t(s - 1, u + 1)$ (or alternatively: by the same token, $N_{t+\Delta t}(s, u)$ receives a positive contribution from $N_t(s + 1, u - 1)$).

- 3) **Neighbors of v^* belonging to previously unsatisfied clauses:** The contributions are analogous to those of the previous discussion.

Slide 33

- Combining the contributions from the previous discussion, one gets

$$\begin{aligned}
 N_{t+\Delta t}(s, u) &= N_t(s, u) - p_t^{(flip)}(s, u) + p_t^{(flip)}(u, s) \\
 &+ (K - 1)\langle s \rangle_t^{(flip)} \left(-\frac{s p_t(s, u)}{\langle s \rangle_t} + \frac{(s+1)p_t(s+1, u-1)}{\langle s \rangle_t} \right) \\
 &+ (K - 1)\langle u \rangle_t^{(flip)} \left(-\frac{u p_t(s, u)}{\langle u \rangle_t} + \frac{(u+1)p_t(s-1, u+1)}{\langle u \rangle_t} \right).
 \end{aligned} \tag{12}$$

- Setting $\Delta t = 1/N$ and replacing differences by differentials for large N one

obtains

$$\begin{aligned}
 N_{t+\Delta t}(s, u) - N_t(s, u) &= N(p_{t+\Delta t}(s, u) - p_t(s, u)) \\
 &= \frac{p_{t+\Delta t}(s, u) - p_t(s, u)}{\Delta t} \rightarrow \frac{d}{dt} p_t(s, u).
 \end{aligned} \tag{13}$$

And so we get (finally) the set of ordinary differential equations governing the dynamics of $p_t(s, u)$:

Slide 34

$$\begin{aligned}
 \dot{p}_t(s, u) &= -p_t^{(flip)}(s, u) + p_t^{(flip)}(u, s) \\
 &+ (K - 1)\langle s \rangle_t^{(flip)} \left(-\frac{s p_t(s, u)}{\langle s \rangle_t} + \frac{(s+1)p_t(s+1, u-1)}{\langle s \rangle_t} \right) \\
 &+ (K - 1)\langle u \rangle_t^{(flip)} \left(-\frac{u p_t(s, u)}{\langle u \rangle_t} + \frac{(u+1)p_t(s-1, u+1)}{\langle u \rangle_t} \right).
 \end{aligned} \tag{14}$$

- Given the initial distribution $p_0(s, u)$ (e.g. the Poisson distribution with $\alpha_s(0) = \alpha_u(0) = 1/2$), this set of ODE's can be solved numerically.

Slide 35

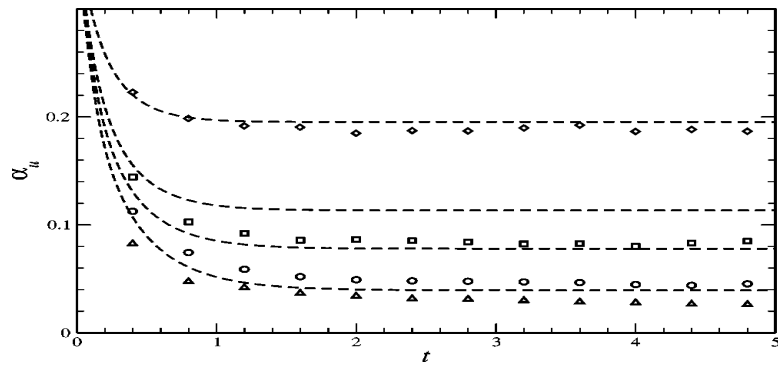


FIG. 7. 3-XOR-SAT: influence of greedy steps on the behavior of the energy density at $\alpha=0.75$. As above the dashed line is obtained by numerically integrating Eq. (20) after plugging in Eq. (6) and using $N=1/\Delta t=50\,000$. The dotted line shows the evolution of a (random) single run of the 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.

References and Discussion

- References are:
 - G. Semerjian, R. Monasson: *Relaxation and metastability in a local search procedure for the random satisfiability problem*. Physical Review E 67 (2003).
 - W. Barthel, A. K. Hartmann, M. Weigt: *Solving satisfiability problems by fluctuations: The dynamics of stochastic local search algorithms*. Physical Review E 67 (2003).
- Both papers deal with the same topic; this presentation followed Barthel's approach.
- The reason for this choice is that while Semerjian's approach is more rigorous and mathematically elegant it requires some knowledge of physics for understanding it.

Slide 36

- They develop a sort of a “quantum-mechanical” formalism for the system, and write the time evolution of the system in terms of a (stochastic) evolution operator, and the evolution equation is then solved by a sort of perturbative expansion of the evolution operator. According to the French tradition enough blind spots are left in the analysis in order to lead the reader sufficiently astray.
- Semerjian & Monasson study the pure walk-SAT only (no greedy steps).

Slide 37

- Barthel’s paper is more readable, but at the cost of being extremely heuristic.
- The Magic Words “On the Average” are said once in a while never really explicating what exactly is ment by this average (average over time-course of walk-SAT for a given formula, the randomness of initial formula, both, some other randomness, the average w.r.t. Poisson distribution...).
- I found myself wondering would it even be possible to present this analysis more

rigorously using just probability theory and stochastic processes / stochastic differential equations.

- It seems that equation (14) should apply to other local search algorithms as well, provided that one can obtain an expression for $p_t^{(flip)}(s, u)$ for that algorithm (in terms of $p_t(s, u)$ of course) and that independent neighbors -assumption *etc.* is sufficiently valid.

Slide 38

- Barthel *et al.* also present an interesting computation of $\mathbf{P}(\alpha_u(0) \rightarrow \alpha_u(t_f) = 0)$ for the Poissonian approximation but it’s not presented here.