Phase Transitions and Local Search for k-Satisfiability

Pekka Orponen (joint work with many others)

Department of Information and Computer Science
Helsinki University of Technology TKK

Outline

- **Part I: Background – Combinatorial Phase Transitions**
- The satisfiability problem SAT
- The DPLL procedure
- “Where the Really Hard Problems Are?”
- Hard instances for 3-SAT & k-SAT
- Statistical mechanics of k-SAT
- **Part II: Local Search – Methods & Experiments**
- WalkSAT and related algorithms
- Focused Metropolis Search
- Whitening and local search
- **Part III: More Transitions**
- Focused Metropolis Search again
- The ChainSAT algorithm
- ChainSAT and whitening

I. The satisfiability problem SAT (1/2)

- INPUT: Boolean formula F composed of literals (variables and their negations) and connectives $\wedge$ (“and”) and $\vee$ (“or”).
- GOAL: Determine if F is satisfiable, i.e. if there is a truth assignment to the variables that makes F evaluate to true.
- E.g. the Boolean formula:

$$F = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_4)$$

  is satisfied by the truth assignment $\{x_1 = T, x_2 = T, x_3 = F, x_4 = T\}$, or briefly $x = (1,1,0,1)$ (and many other truth assignments too).
- The SAT problem is in general NP-complete, which means that all known complete decision algorithms require worst case exponential running time, in the length of the input formula F (Cook 1971, Levin 1973).

The satisfiability problem SAT (2/2)

- A Boolean formula F is in k-conjunctive normal form (k-cnf) if it is a conjunction ("and") of small disjunctions ("or"s), where each factor disjunction, or clause, contains exactly k literals.
- E.g.

$$F = (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_4)$$

  is a 3-cnf formula.
- The problem k-SAT is SAT restricted to k-cnf input formulas. In this formulation one often thinks of the formula as simply a set of clauses, each of which must be satisfied.
- Also k-SAT is NP-complete for all $k \geq 3$. 2-SAT has a nice graph-theoretic polynomial-time decision method (Aspvall, Plass & Tarjan 1979).

The DPLL (Davis-Putnam-Logemann-Loveland) procedure

A backtrack search method for testing satisfiability of a set of clauses Σ on variable set V. Basic outline:

- If Σ is empty, return “satisfiable”.
- If Σ contains an empty clause, return “unsatisfiable”.
- If Σ contains a unit clause $c = x^{\pm}$, assign to x a value which satisfies c, simplify the remaining clauses correspondingly, and call DPLL recursively.
- Otherwise select an unassigned $x \in V$, assign $x \leftarrow 1$, simplify Σ, and call DPLL recursively. If this call returns “satisfiable”, then return “satisfiable”; else assign $x \leftarrow 0$, simplify Σ, and call DPLL recursively again.

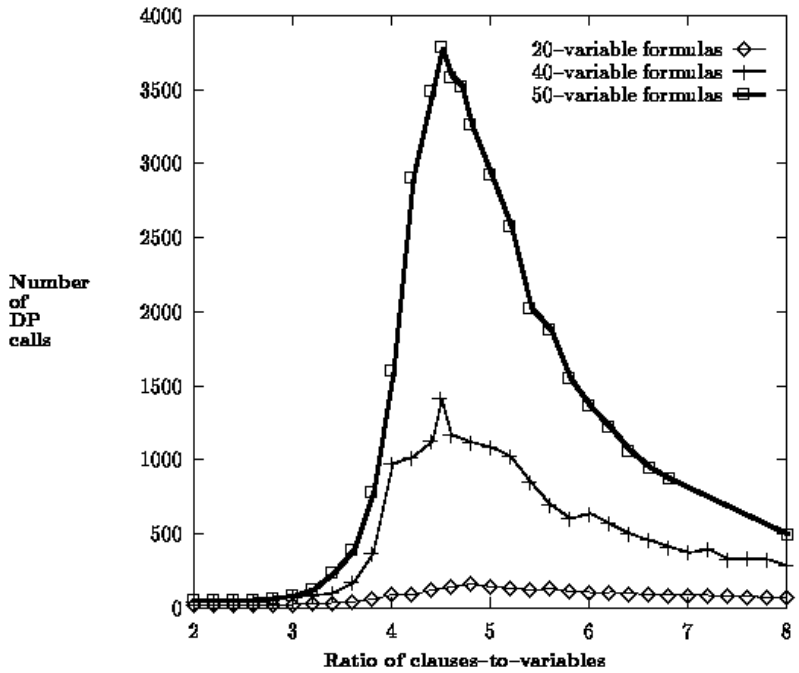
“Where the Really Hard Problems Are?”

- Cheeseman, Kanefsky & Taylor (1991)
- Many NP-complete problems, including satisfiability, can in many cases be solved reasonably well by heuristics or special-case methods.
- Where, then, are the (presumably) exponentially hard instances of these problems located? Could one tell ahead of time whether a given instance is likely to be hard? Could one learn something fundamental about the “reasons” for NP-completeness by focusing on the hard instances?
- Early studies: Yu & Anderson (1985), Hubermann & Hogg (1987), Cheeseman, Kanefsky & Taylor (1991), Mitchell, Selman & Levesque (1992), Kirkpatrick & Selman (1994).

Hard instances for 3-SAT (1/3)

- Mitchell, Selman & Levesque, AAAI-92
- Experiments on the behaviour of the DPLL procedure on randomly generated 3-cnf Boolean formulas.
- Distribution of test formulas:
  - $n =$ number of variables
  - $m = \alpha n$ randomly generated clauses of 3 literals, $2 \leq \alpha \leq 8$
- For sets of 500 formulas with $n = 20/40/50$ and various α, Mitchell et al. plotted the median number of recursive DPLL calls required for solution.

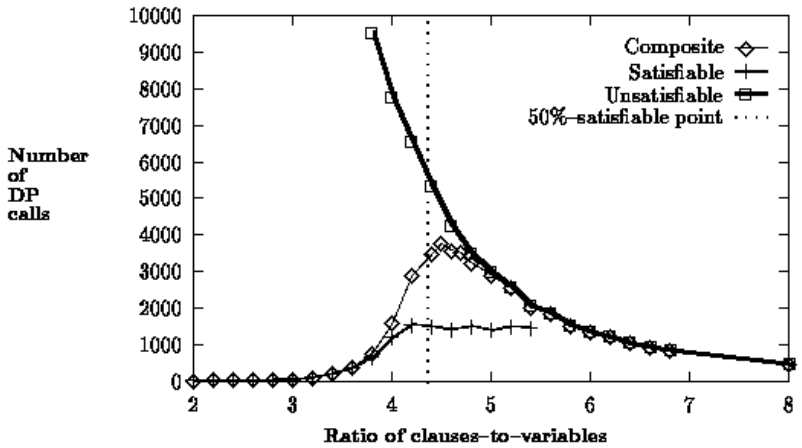
Hard instances for 3-SAT (2/3)



Results:

- A distinct peak in median running times at about clauses-to-variables ratio $\alpha \approx 4.5$.
- Peak gets more pronounced for increasing $n \Rightarrow$ well-defined “delta” distribution for infinite n?

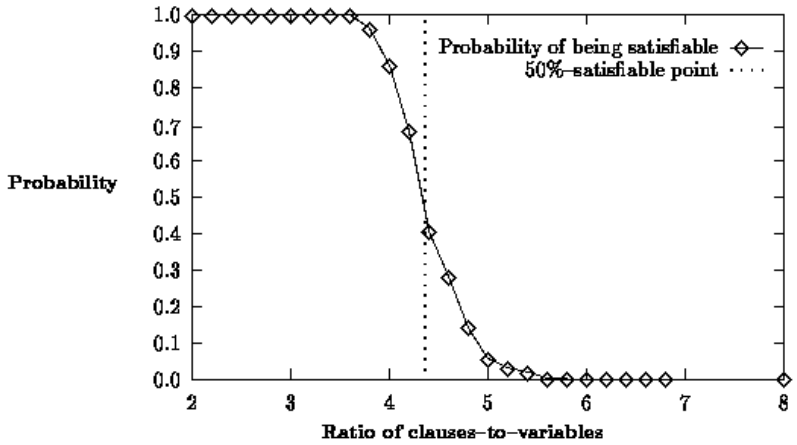
Hard instances for 3-SAT (3/3)



- The runtime peak seems to be located near the point where 50% of formulas are satisfiable.
- The peak seems to be caused by relatively short unsatisfiable formulas.

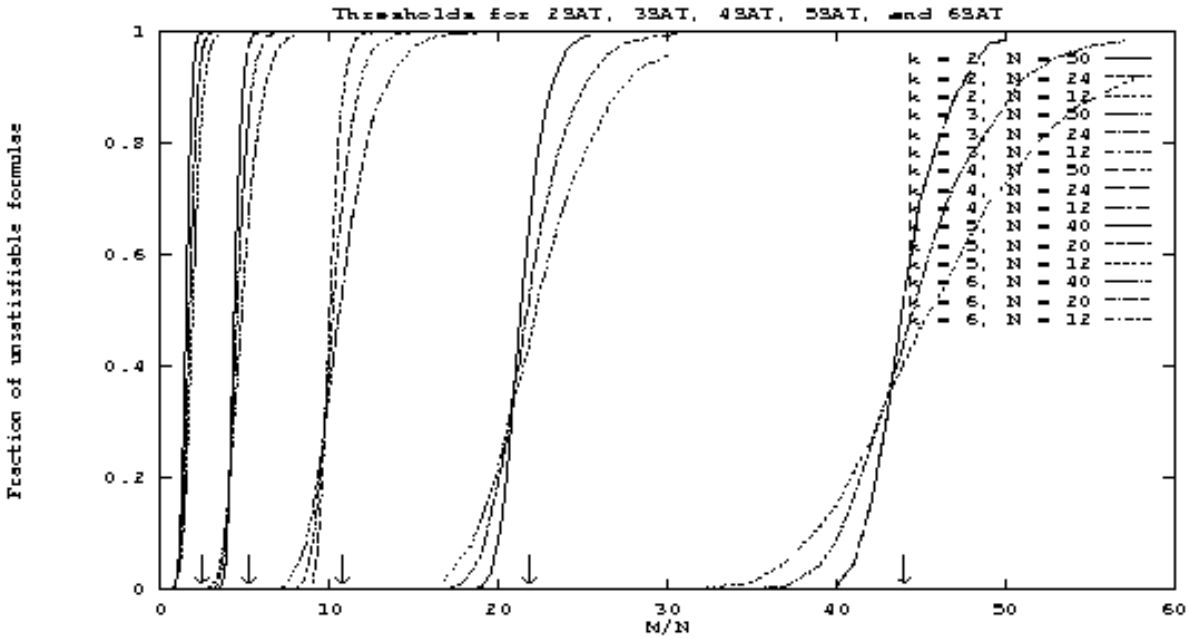
Question: Is the connection of the running time peak and the satifiability threshold a characteristic of the DPLL algorithm, or a (more or less) algorithm independent “universal” feature?

The satisfiability transition (1/2)



Mitchell et al. (1992): The “50% satisfiable” point or “satisfiability threshold” for 3-SAT seems to be located at $\alpha \approx 4.25$ for large n.

The satisfiability transition (2/2)



Kirkpatrick & Selman (1994):

- Similar experiments as above for k-SAT, $k = 2, \ldots, 6$, 10000 formulas per data point.
- The "satisfiability threshold" α_s shifts quickly to larger values of α for increasing k.

Statistical mechanics of k-SAT (1/3)

Kirkpatrick & Selman, *Science* 1994.

A spin glass model of a k-cnf formula:

- variables x_i $\sim$ spins with states ± 1
- clauses c $\sim$ k-wise interactions between spins
- truth assignment σ $\sim$ state of spin system
- Hamiltonian $H(\sigma)$ $\sim$ number of clauses unsatisfied by σ
- α_s $\sim$ critical connection density for "phase transition" from "satisfiable phase" to "unsatisfiable phase"

Statistical mechanics of k-SAT (2/3)

Tabulated estimates of α_s for various values of k via an annealing approximation (Kirkpatrick & Selman 1994), replica-symmetric calculation (Monasson & Zecchina 1997), and a recent "cavity method 1RSB" calculation (Mertens, Mézard & Zecchina 2006):

k	α_{ann}	α_{RS}	α_{1RSB}
2	2.41	1.00	-
3	5.19	4.60	4.267
4	10.74	?	9.931
5	21.83	?	21.117
6	44.01	?	43.37

Note: The precise value for $\alpha_s(2)$ is known to be 1 (Goerdt 1982, Chvátal & Reed 1982).

The estimate $\alpha_s(3) \approx 4.267$ has also been derived earlier (Braunstein, Mézard et al. 2002).

Statistical mechanics of k-SAT (3/3)

Analytically, it is known that:

- A sharp satisfiability threshold α_s exists for all $k \geq 2$ (Friedgut 1999).
- For $k = 2$, $\alpha_s = 1$ (Goerdt 1982, Chvátal & Reed 1982). Note that 2-SAT $\in$ P.
- For $k = 3$, $3.145 < \alpha_s < 4.506$ (lower bound due to Achlioptas 2000, upper bound to Dubois et al. 1999).
- For large k, $\alpha_s \sim (\ln 2) \cdot 2^k$ (Achlioptas & Moore 2002).

II. Local search

Naive, but surprisingly useful idea for (combinatorial) optimisation. Assume objective function $E = E(s)$ to be minimised. Then:

- Start with some randomly chosen feasible solution $s = s_0$.
- If value of $E(s)$ is not "good enough", search for some "neighbour" s' of s that satisfies $E(s') \lesssim E(s)$. If such an s' is found, set $s \leftarrow s'$ and repeat.
- If no improving neighbour is found, then either restart at new random $s = s_0$ or relax the neighbourhood condition [algorithm-dependent].

Good experiences for 3-SAT in the satisfiable region $\alpha < \alpha_s$: e.g. GSAT (Selman et al. 1992), WalkSAT (Selman et al. 1996).

GSAT

Gu 1992; Selman, Levesque & Mitchell 1992.

Denote by $E = E_F(s)$ the number of unsatisfied clauses in formula F under truth assignment s.

function GSAT(F):
$s \leftarrow$ initial truth assignment;
while flips $<$ max_flips **do**
    if s satisfies F **then return** s
    else
        find a variable x whose flipping in s causes
            largest decrease in $E(s)$ (if no decrease is
            possible, then smallest increase);
        $s \leftarrow (s$ with variable x flipped)
end while;
return s.

NoisyGSAT

Selman, Kautz & Cohen 1994.

Idea: Augment GSAT by a fraction p of random walk moves.

function NoisyGSAT(F,p):
$s \leftarrow$ initial truth assignment;
while flips $<$ max_flips **do**
    if s satisfies F **then return** s
    else
        with probability p, pick a variable x
            uniformly at random;
        with probability $(1-p)$, do basic GSAT move:
            find a variable x whose flipping causes
            largest decrease in $c(s)$ (if no decrease is
            possible, then smallest increase);
        $s \leftarrow (s$ with variable x flipped)
end while;
return s.

WalkSAT

Selman, Kautz & Cohen 1994/1996.

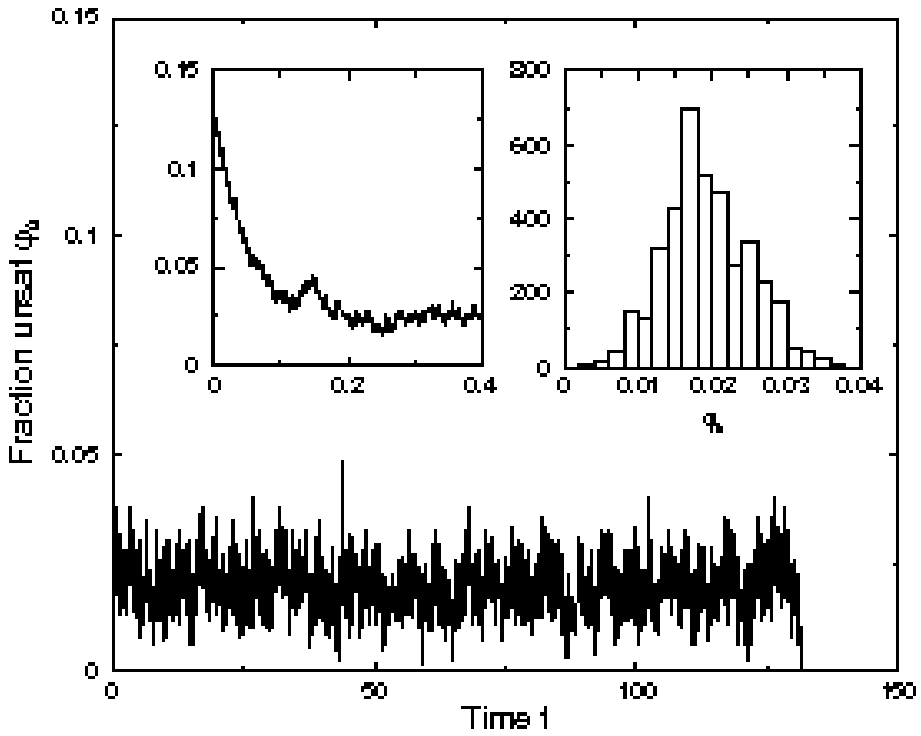
Idea: NoisyGSAT with the provision that the choice of flipped variables is always focused to the presently unsatisfied clauses.

function WalkSAT(F,p):
$s \leftarrow$ initial truth assignment;
while flips $<$ max_flips **do**
    if s satisfies F **then return** s **else**
        choose a random unsatisfied clause C in F;
        if some variables in C can be flipped without
            breaking any presently satisfied clauses,
            then pick one such variable x at random; else:
        with probability p, pick a variable x in C unif. at random;
        with probability $(1-p)$, do basic GSAT move:
            find a variable x in C whose flipping causes
            largest decrease in $c(s)$;
        $s \leftarrow (s$ with variable x flipped)
end while;
return s.

WalkSAT vs. NoisyGSAT

The focusing seems to be important: in the (unsystematic) experiments in Selman et al. (1996), WalkSAT outperforms NoisyGSAT by several orders of magnitude. Later experimental evidence by other authors corroborates this.

Dynamics of local search



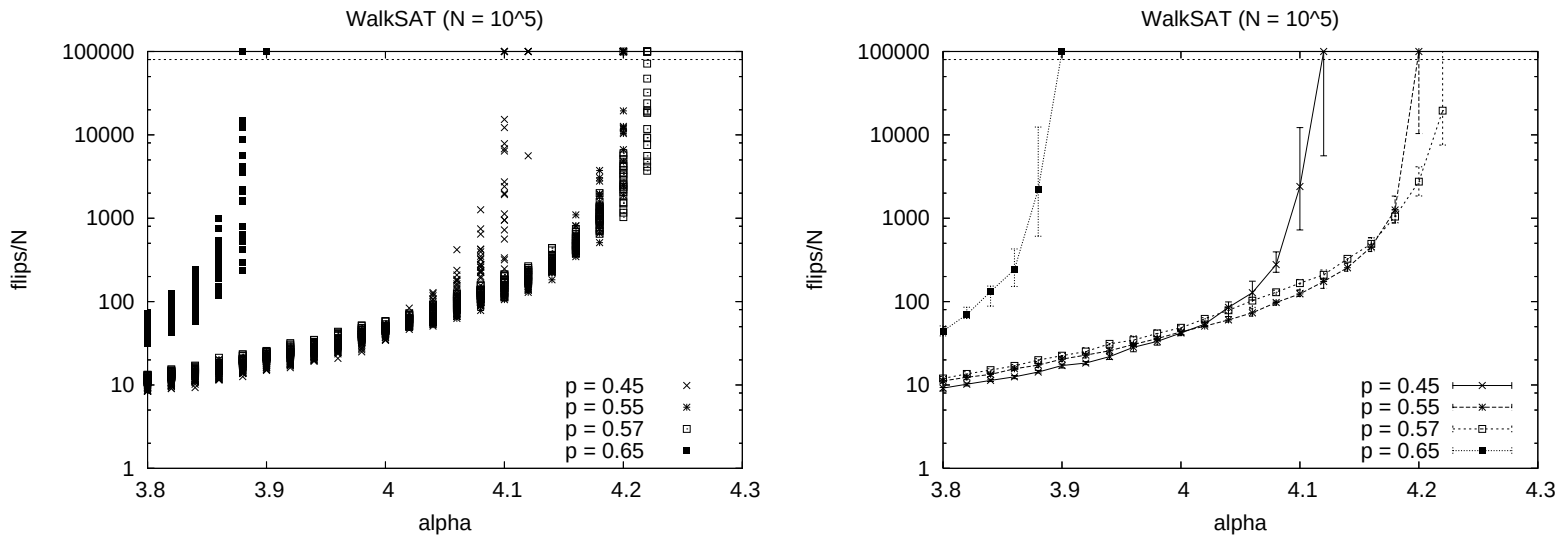
A WalkSAT run with $p = 1$ (“focused random walk”) on a randomly generated 3-SAT instance, $\alpha = 3$, $n = 500$: evolution in the fraction of unsatisfied clauses (Semerjian & Monasson 2003).

Results and conjectures ca. 2005

- Barthel, Hartmann & Weigt (2003), Semerjian & Monasson (2003): WalkSAT with $p = 1$ has a "dynamical phase transition" at $\alpha_{\mathrm{dyn}} \approx 2.7 - 2.8$. When $\alpha < \alpha_{\mathrm{dyn}}$, satisfying assignments are found in linear time per variable (i.e. in a total of cN "flips"), when $\alpha > \alpha_{\mathrm{dyn}}$ exponential time is required.
- Explanation: for $\alpha > \alpha_{\mathrm{dyn}}$ the search equilibrates at a nonzero energy level, and can only escape to a ground state through a large enough random fluctuation.
- Conjecture: all local search algorithms will have difficulties beyond a clustering transition of the solution space at $\alpha_c \approx 3.92$ (Mézard & Zecchina 2002).
- Observation & conjecture: Nevertheless WalkSAT seems to work in linear time at least up to the "1RSB stability transition" at $\alpha \approx 4.15$ (Aurell et al. 2004), but maybe not beyond that (Aurell, Montanari et al.)

Some WalkSAT experiments on (3-SAT)

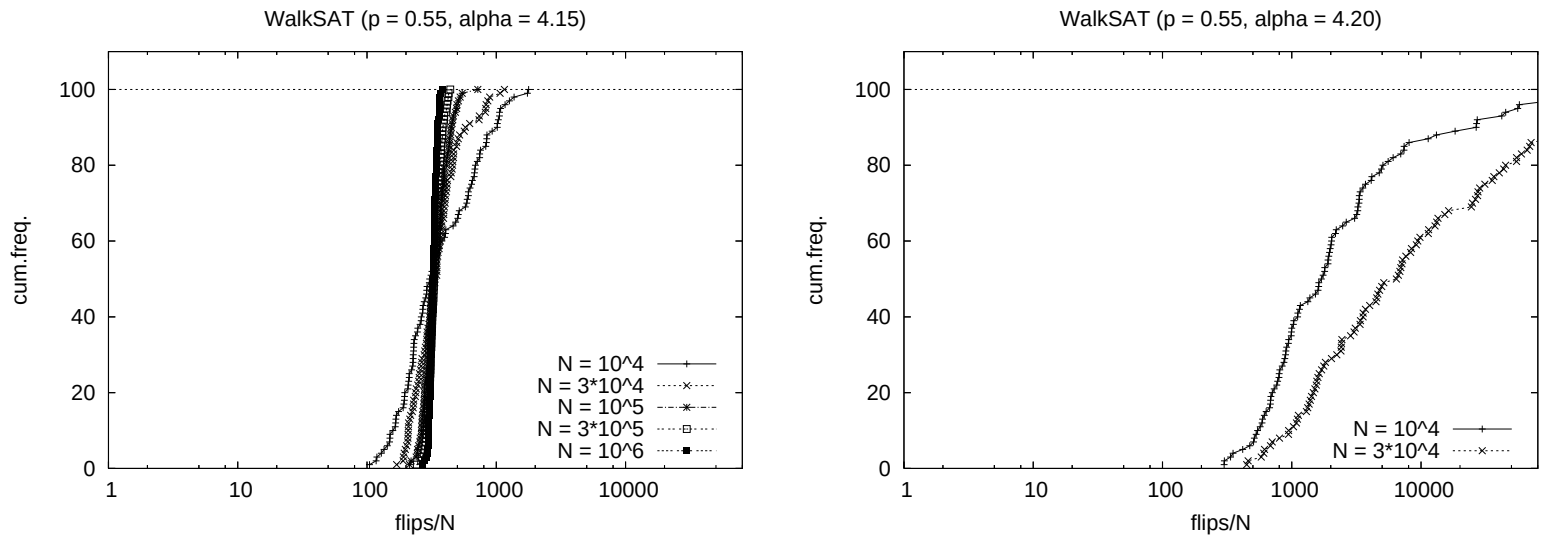
Seitz, Alava & Orponen (2005)



Normalised (flips/n) solution times for finding satisfying assignments using WalkSAT, $\alpha = 3.8 \ldots 4.3$.
Left: complete data; right: medians and quartiles.

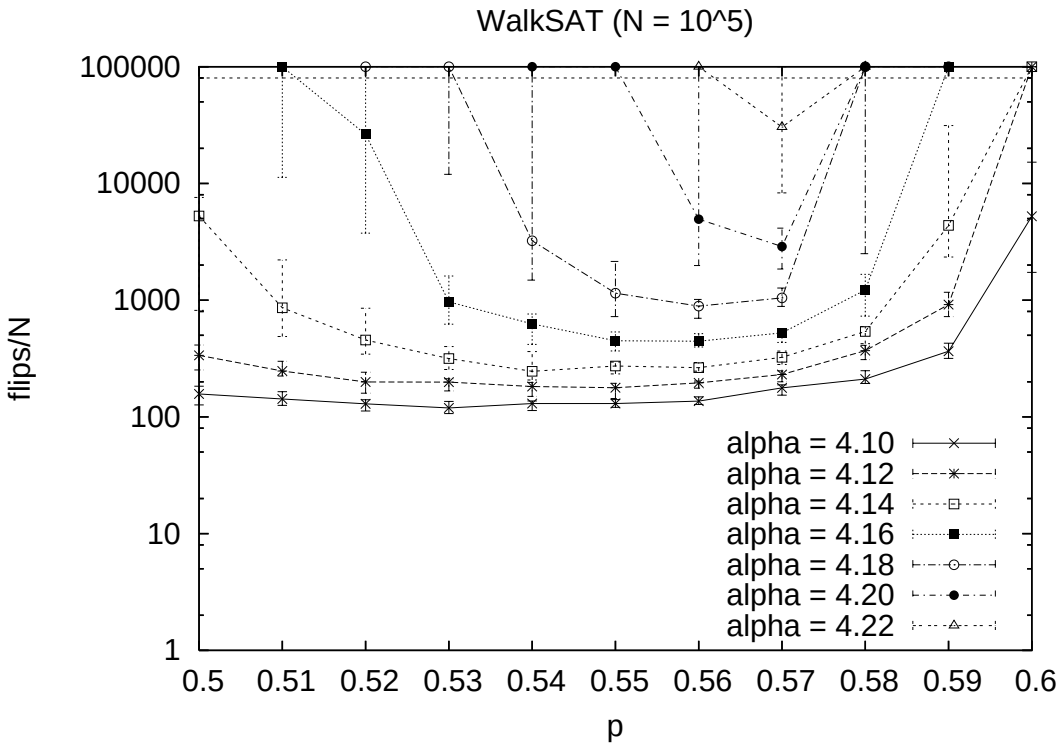
Data suggest linear solution times for $\alpha \gg \alpha_c \approx 3.92$.

WalkSAT linear scaling



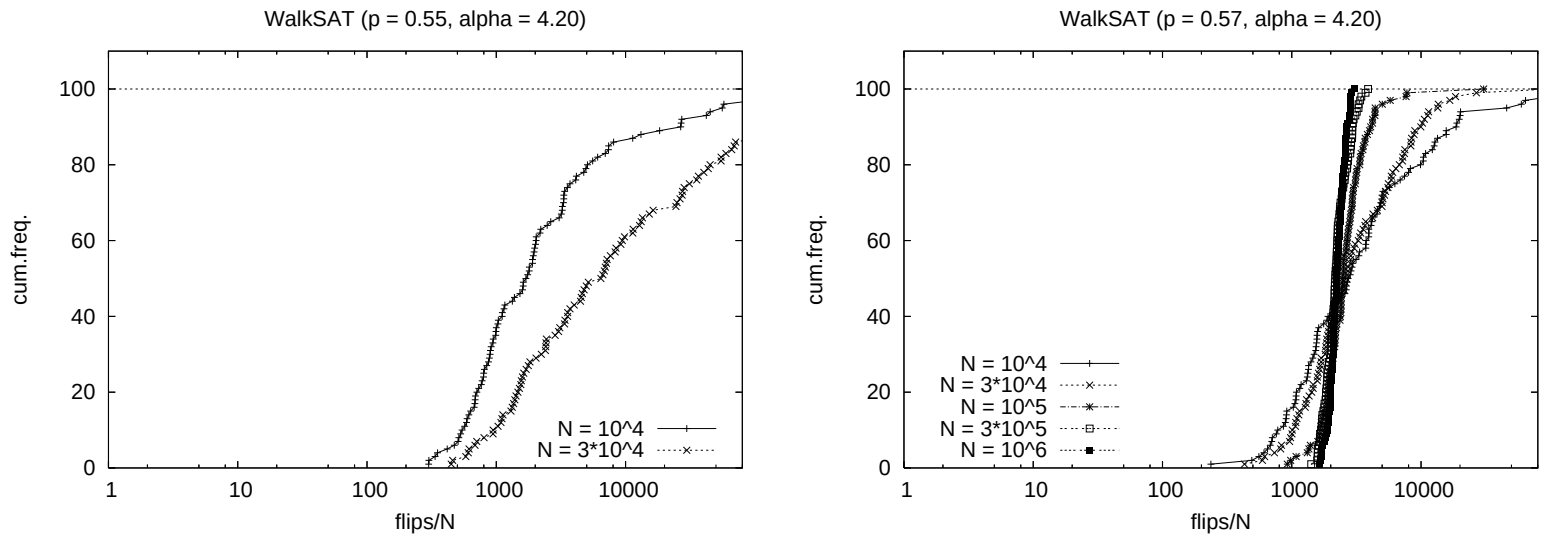
Cumulative solution time distributions for WalkSAT with $p = 0.55$.

WalkSAT optimal noise level?



Normalised solution times for WalkSAT with $p = 0.50 \ldots 0.60$, $\alpha = 4.10 \ldots 4.22$.

WalkSAT sensitivity to noise



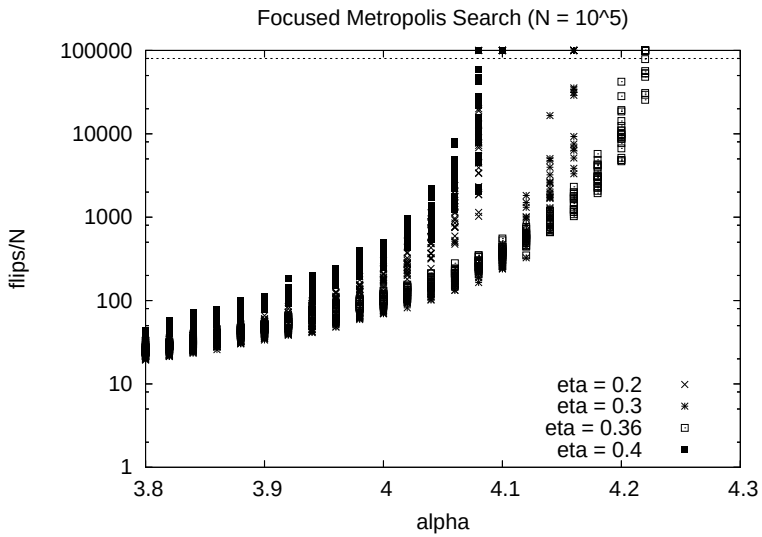
Cumulative solution time distributions for WalkSAT at $\alpha = 4.20$ with $p = 0.55$ and $p = 0.57$.

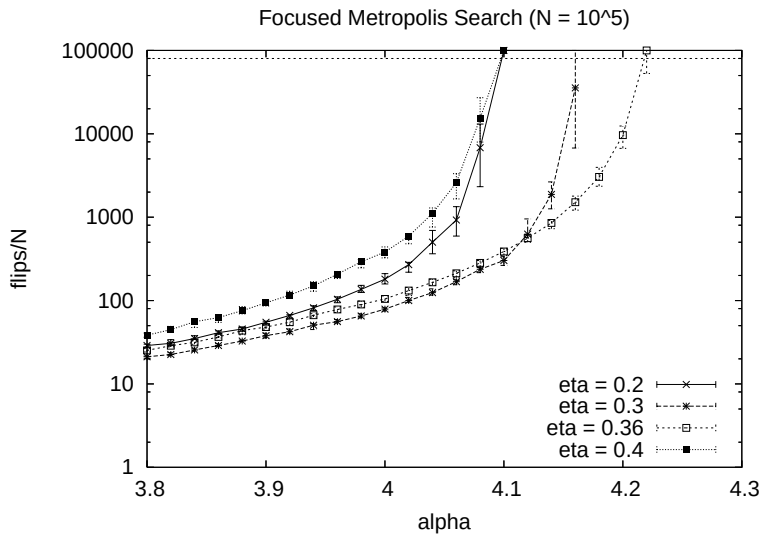
Focused Metropolis Search

Arguably the most natural focused local search algorithm. Variable flip acceptance probabilities determined by a parameter η, $0 \leq \eta \leq 1$.

function FMS(F,η):
$s \leftarrow$ initial truth assignment;
while flips $<$ max_flips **do**
    if s satisfies F **then return** s **else**
        choose a random unsatisfied clause C in F;
        choose a variable x in C at random;
        let $x' \leftarrow \mathrm{flip}(x)$, $s' \leftarrow s[x'/x]$;
        if $E(s') \leq E(s)$ **then** flip x, **else**
            flip x with prob. $\eta^{(E(s')-E(s))}$.

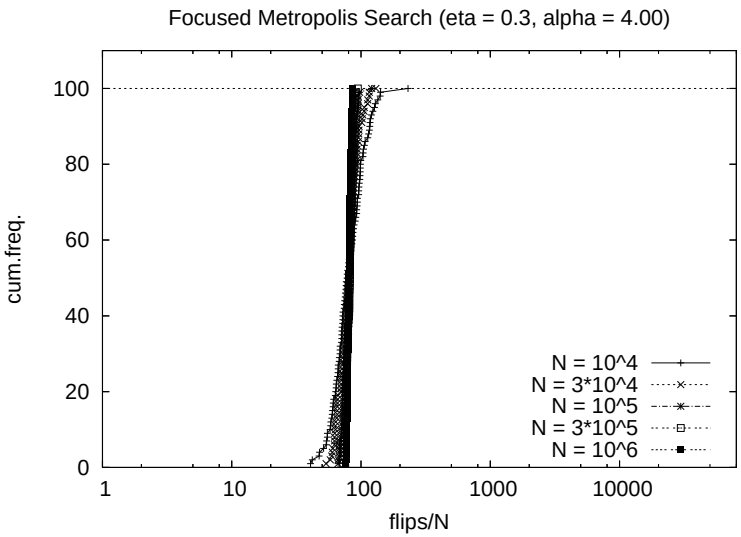
FMS experiments (3-SAT)

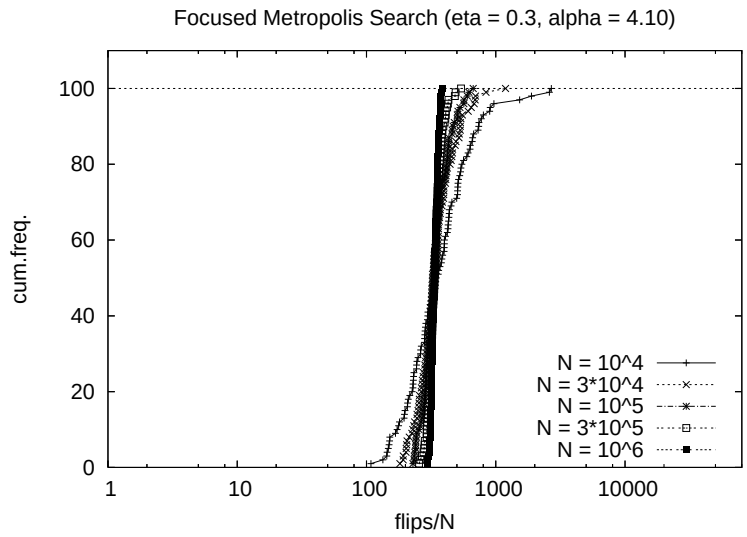




Normalised solution times for FMS, $\alpha = 3.8 \ldots 4.3$.
Left: complete data; right: medians and quartiles.

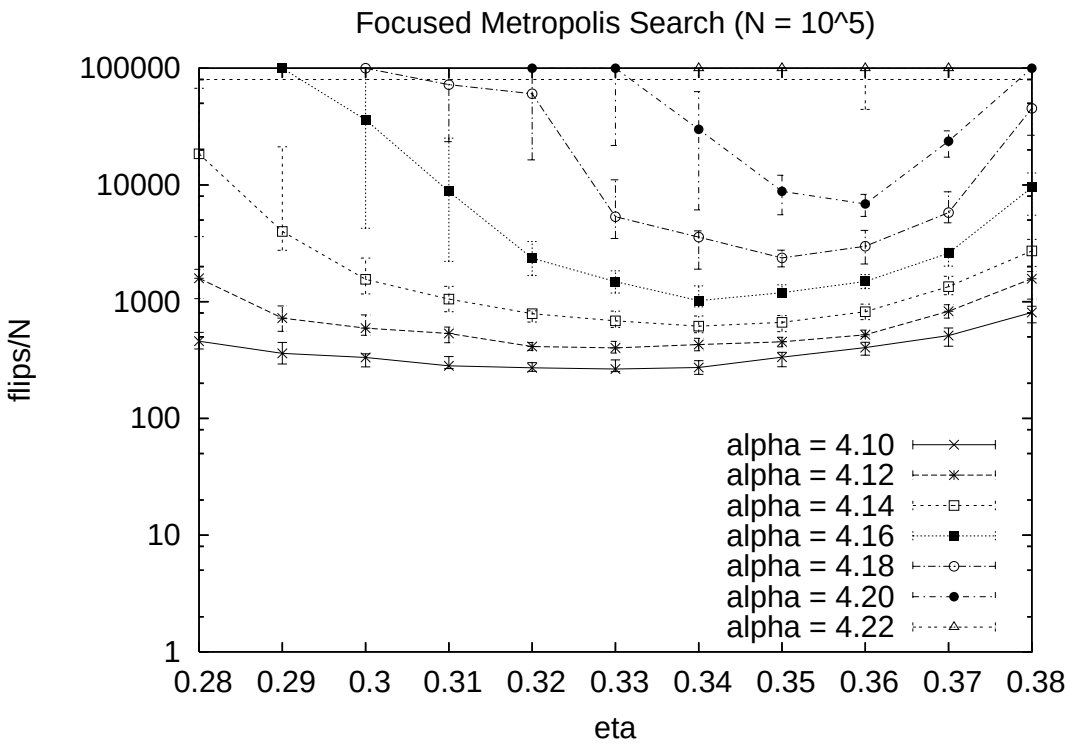
FMS linear scaling





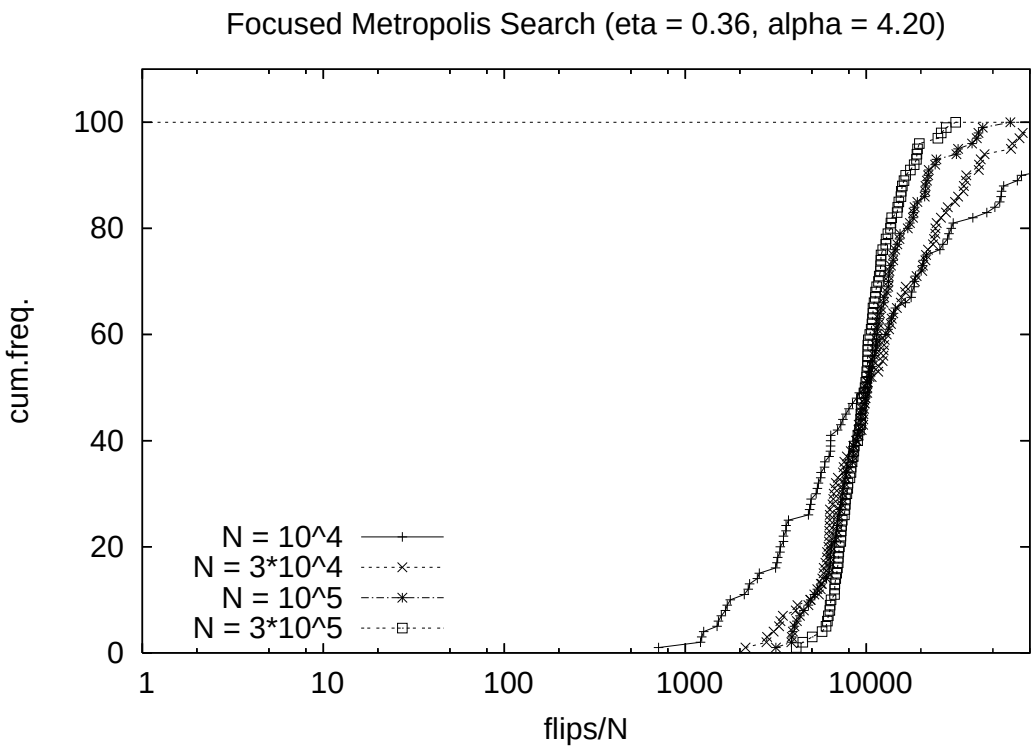
Cumulative solution time distributions for FMS with $\eta = 0.3$.

FMS optimal acceptance ratio?



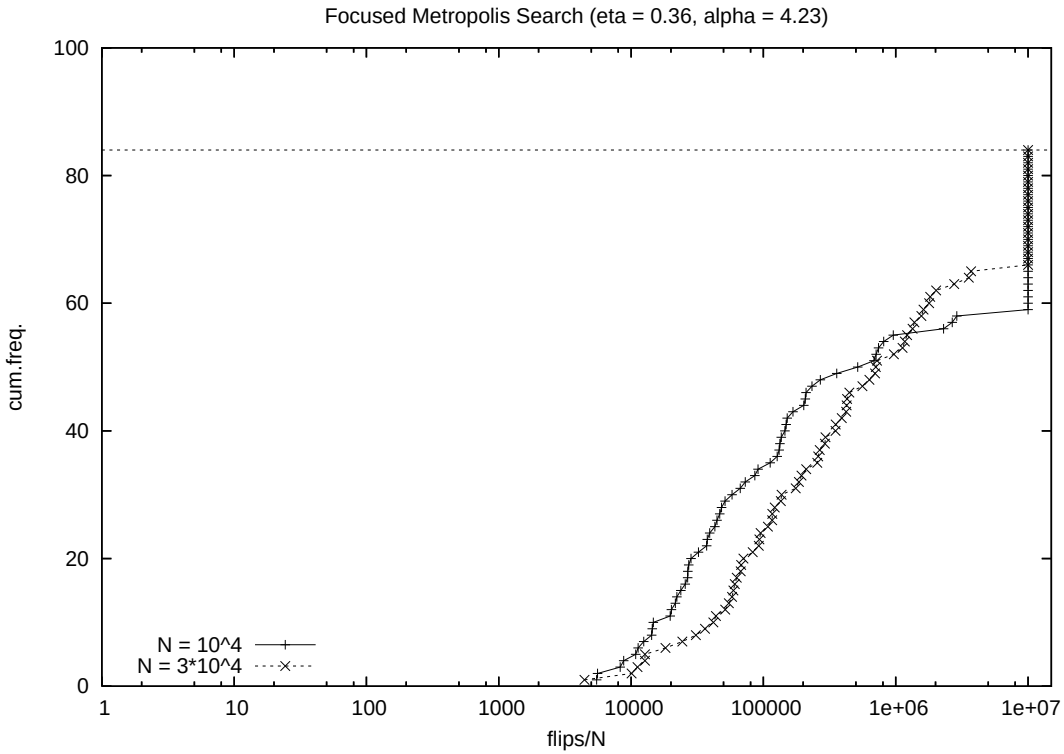
Normalised solution times for FMS with $\eta = 0.28 \ldots 0.38$, $\alpha = 4.10 \ldots 4.22$.

FMS optimal acceptance ratio cont'd



Cumulative solution time distributions for FMS with $\eta = 0.36$, $\alpha = 4.20$.

FMS optimal acceptance ratio cont'd



Cumulative solution time distributions for FMS with $\eta = 0.36$, $\alpha = 4.23$.

Whitening (1/2)

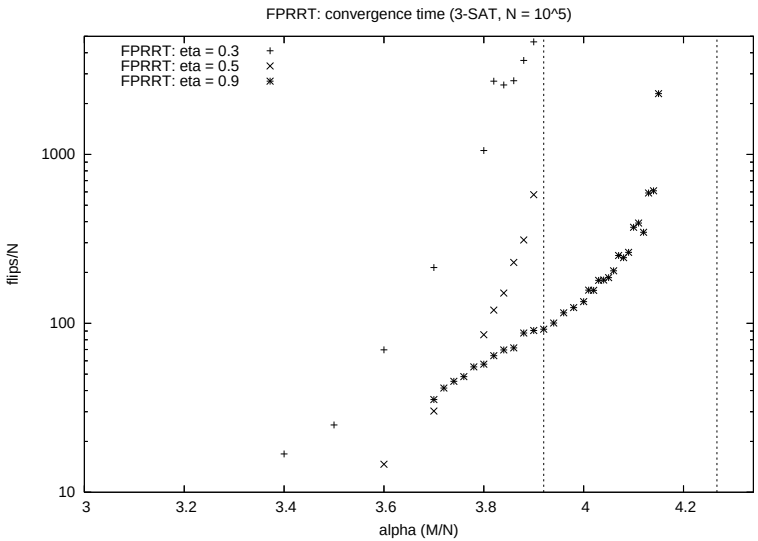
Technique introduced by Parisi, Braunstein, Zecchina et al. to determine the "frozen" variables (spins, degrees of freedom) in a given configuration.

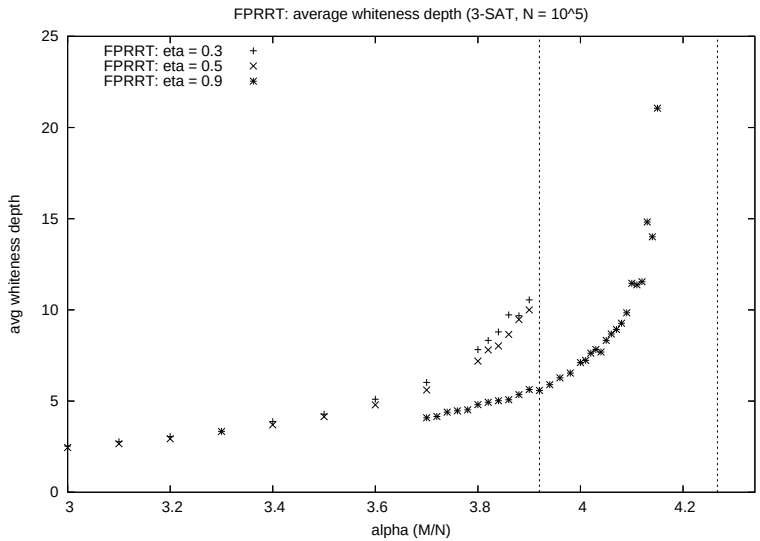
A variable is *frozen* in truth assignment s to 3-SAT formula F, unless determined *white* by the following process:

function WHITENING(F, s):
mark all clauses white except those that
    have exactly one true literal;
loop:
    mark all variables white except those that
        appear as the unique satisfying literals
        in non-white clauses;
    halt, if all variables are white (full whitening);
    halt, if no new variables became white (core found);
    mark those clauses white that contain
        at least one white variable.

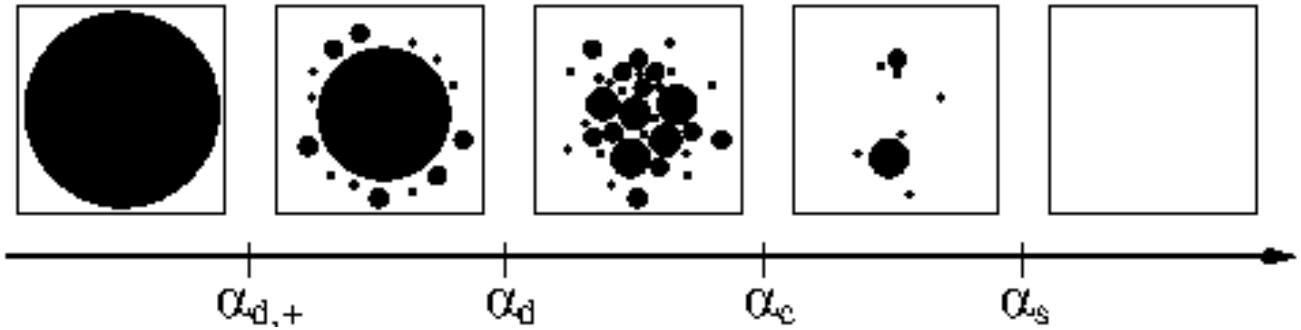
Whitening (2/2)

The “whiteness” of solutions seems to have many connections to the behaviour of local search algorithms. Consider e.g. the following plots of runtimes of one local search algorithm variant (FRRT) vs. the “average whiteness depth” of the solutions found by it:





III. More transitions

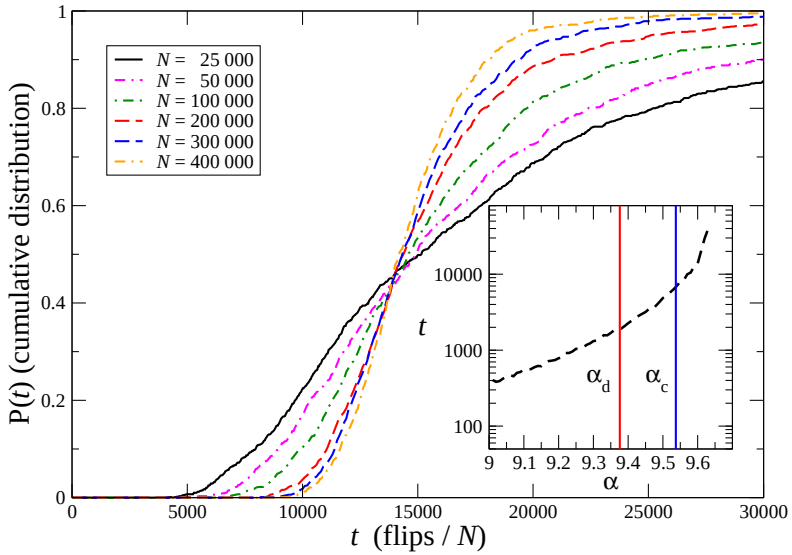


Recently, the picture of the solution space of k-SAT problems has been further refined by the introduction of dynamic and condensation transitions α_d, α_c. (The basic clustering transition is here denoted α_{d+}.)

Predicted values for these transitions are as follows (Mertens et al. 2006, Krzakala et al. 2007):

k	α_{d+}	α_d	α_c	α_s
3	3.927	3.86(?)	3.86(?)	4.267
4	8.297	9.38	9.547	9.931
5	16.12	19.16	20.80	21.117
6	30.50	36.53	43.08	43.37

FMS vs. transitions



Cumulative distributions of FMS solution times on random 4-SAT instances at $\alpha = 9.6$. Vertical axis indicates the fraction of 1001 random instances solved within a given running time, measured in flips / N on the horizontal axis. The "temperature" parameter of FMS is set to $\eta = 0.293$.

Inset: Scaling of the algorithm with increasing α at $N = 100000$.

The ChainSAT algorithm (1/2)

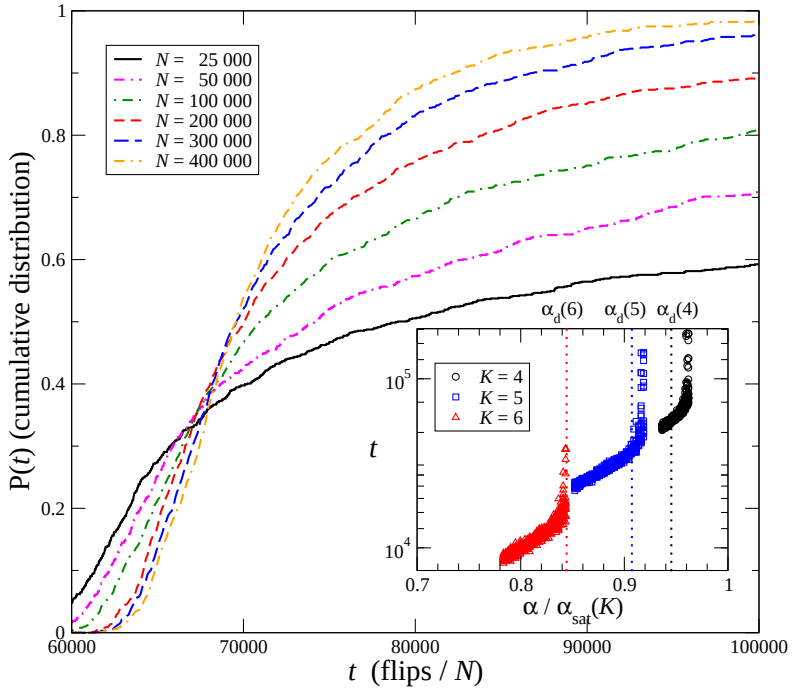
Local search method that never makes upwards moves in the energy landscape.

Idea: If the algorithm is not able to find a energy-decreasing (actually energy-nonincreasing) move, it moves along a critically satisfied variable-constraint-variable path (each constraint satisfied by a single variable) until an opportunity to decrease energy is found. (Note similarity to the whitening process.)

The ChainSAT algorithm (2/2)

function ChainSAT(F,p_1, p_2):
$s \leftarrow$ initial truth assignment;
chaining $\leftarrow$ FALSE;
while s does not satisfy F **do**
    if not chaining **then**
        choose a random unsatisfied clause C in F;
        choose a variable x in C at random;
    $\Delta E \leftarrow$ change in number of unsatisfied clauses if x is
            flipped in s;
    if $\Delta E = 0$ **then**
        flip x in s; chaining $\leftarrow$ FALSE;
    if $\Delta E < 0$ **then with probability** p_1
        flip x in s; chaining $\leftarrow$ FALSE;
    if $\Delta E > 0$ **then with probability** $1 - p_2$
        choose a random clause C satisfied only by x;
        choose a variable $x' \neq x$ in C at random;
        $x \leftarrow x'$;
        chaining $\leftarrow$ TRUE.

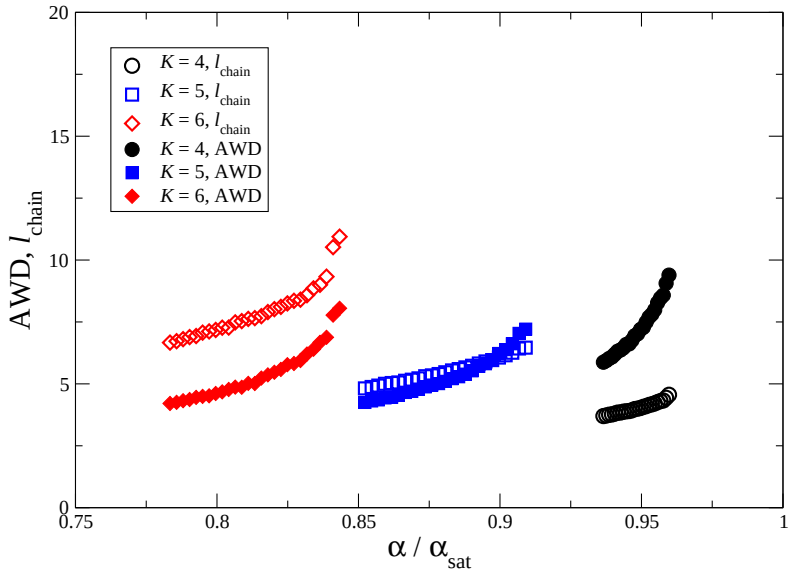
ChainSAT vs. transitions



Cumulative distributions of ChainSAT solution times on random 4-SAT instances at $\alpha = 9.55$. Vertical axis indicates the fraction of 1001 random instances solved within a given running time, measured in flips / N on the horizontal axis.

Inset: Scaling of the algorithm for $k = 4, 5, 6$ at $N = 100000$ with increasing α; the values of $\alpha(k)$ in the horizontal axis have been normalized with $\alpha_s(k)$.

ChainSAT and whitening



Average chain length in ChainSAT and the "average whiteness depth" of the solutions found in random k-SAT for $k = 4, 5, 6$. Each plotted value is the average over 21 random instances. The values of $\alpha(k)$ in the horizontal axis have been normalized with $\alpha_s(k)$. ChainSAT parameters are set to $p_1 = p_2 = 0.0001$ ($K = 4$), 0.0002 ($K = 5$), and 0.0005 ($K = 6$).

References

S. Seitz, M. Alava, P. Orponen: Focused local search algorithms for random 3-satisfiability. *J. Stat. Mech.* 2005, P06006.

M. Alava, J. Ardelius, E. Aurell, P. Kaski, S. Krishnamurthy, P. Orponen, S. Seitz: Circumspect descent prevails in solving random constraint satisfaction problems. Technical report cs.DS/0711.4902, arXiv.org (Nov 2007).