Statistical physics of search algorithms

Rémi Monasson
ENS, Paris
Louis Pasteur Univ., Strasbourg

In collaboration with:

S. Cocco (Strasbourg)
L. Cugliandolo (Paris)
C. Deroulers (Paris)
G. Semerjian (Paris)

Lectures shortly available on http://www.lpt.ens.fr/~monasson

Trieste, september 2002

Content of the lectures

- *Today :*
  - Dynamical processes in physics …
    Algorithms for solving optimization problems
    Reminder on random 3-SAT
  - Analysis of random 3-SAT solving with a backtracking procedure

- *Tomorrow :*
  - Local search algorithms for random SAT
    - Gradient descent
    - Clause flip

Purpose: give technical ability to carry out calculations!

Dynamical processes in Physics

Configuration of variables $X=\{x_1,x_2,\ldots,x_N\}$ e.g. spins
Energy function H[X]
Equilibrium Probability Distribution $P_{gibbs}[X] = \exp(-H[X]/T) / Z$

Dynamics: *X at time T* ⇢ *Y at time T+1*

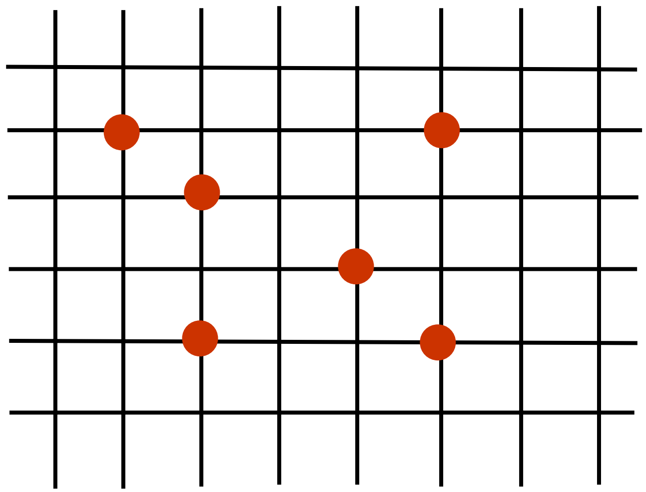
- $P[X,T=0] =$ d$_X$ or $1/2^N$
- $P[Y,T+1] =$ Ê$_X$ $M[Y,X]\ P[X,T]$ $\xrightarrow[\text{T ö ¶}]{}$ $P_{gibbs}[Y]$

e.g. Metropolis (one spin flip) $M[Y,X] = \begin{cases} 1 & \text{if } H[Y] \text{ § } H[X] \\ e^{-(H[Y]-H[X])/T} & \text{if } H[Y] \text{ ¥ } H[X] \end{cases}$

Convergence if T ö ¶, then N ö ¶

But what happens if N ö ¶, then T ö ¶ ?

Dynamical processes without Hamiltonian



Contact process:

- graph with vertices occupied or empty
- ocuppied becomes empty with prob. P
- empty becomes occupied with prob. Q×occup. neighbors

Evolution rules

- Large time limit: all empty!
- Non trivial if N infinite first
- Life time as a function of N (relaxation).

Combinatorial optimization algorithms

= set of evolution rules for an instance and/or a configuration of variables leading to a solution.

- Complete algorithms:

  tells you whether an instance of SAT has a solution or not (solves exactly a decision problem).

- Local search algorithms:

  looks for solutions through a sequence of small improvements of some initial configuration (cannot prove unsatisfiability).

- Probabilistic algorithms:

  Local search + upper bound on the probability of failure
  (one sided probabilistic algorithm)

Want to study various properties of algorithms, essentially average (or distribution) of running times.

A reminder on the Satisfiability of (random) Boolean constraints

(w or NOT x or y)
and
(NOT w or x or z)
and
(x or y or NOT z)

?

3-SAT NP-complete
(and >3)

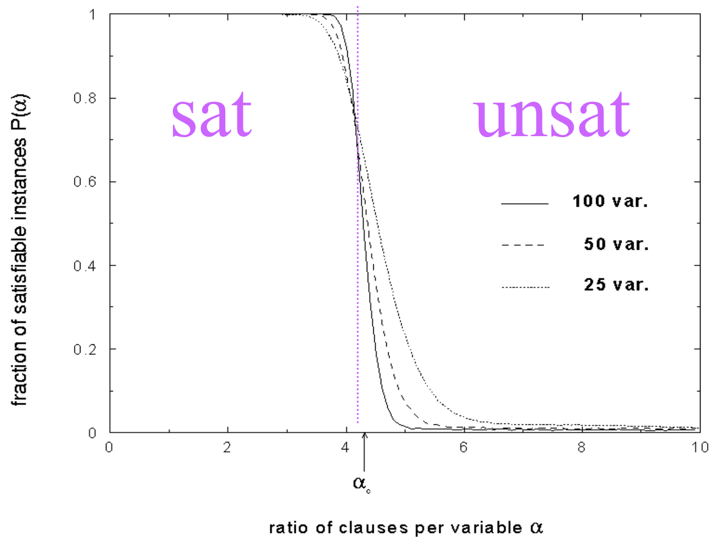
2-SAT P

$$\alpha = \frac{\text{nb. of clauses}}{\text{nb. of variables}}$$

Mitchell, Selman, Levesque ‘92
Crawford, Auton ‘93
Gent, Walsh ‘94

Chao, Franco ‘86, ‘90
Chvatal, Szmeredi ‘88

Phase transition and typical complexity

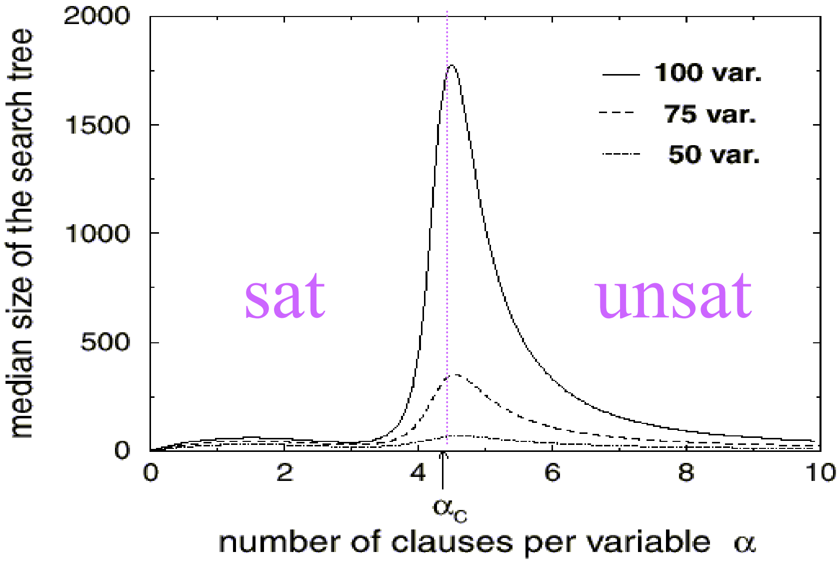


$\alpha_C \approx 4.3$

phase transition!

Rigorous results:

- $\alpha_C > 3.42$
- $\alpha_C < 4.51$
- transition region width $\rightarrow 0$



easy-hard-less hard pattern

Rigorous results:

- linear if $\alpha << \alpha_C$
- exponential if $\alpha > \alpha_C$
- "time" $< 1.5^N$

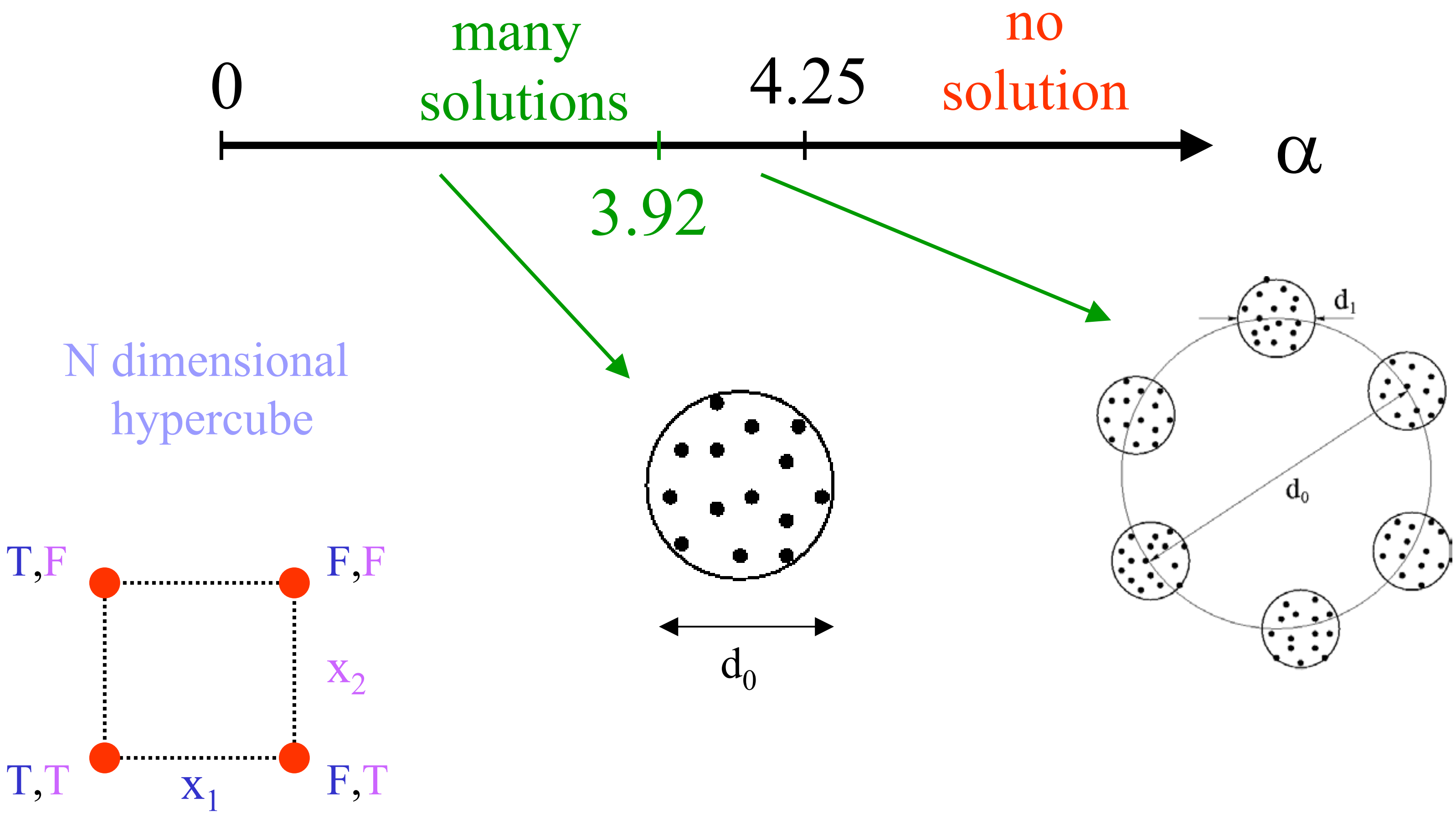
SAT,
a disordered
spin system
(at zero
temperature)

Dictionary

K-SAT	*statistical physics*
Boolean variable x = True , False	*Ising spin* $s_x = +1, -1$
Clauses	*Couplings and fields acting on spins*
Number of clauses violated by a logical configuration	*Energy E of the spins configuration*
some examples:	
2 SAT x or $\bar{y}$	$E = \frac{1}{4}(1-s_x)(1+s_y)$
(x or $\bar{y}$) and ($\bar{x}$ or z)	$E = \frac{1}{4}(1-s_x)(1+s_y) + \frac{1}{4}(1+s_x)(1-s_y)$
3 SAT x or $\bar{y}$ or z	$E = \frac{1}{8}(1-s_x)(1+s_y)(1-s_z)$
Minimal number of violated clauses	*Ground state energy*
The problem is satisfiable	*Ground state energy* = 0
The problem is not satisfiable	*Ground state energy* > 0

Cf. Lecture on Stat. Mech. by C. Borgs and J. Chayes.

Phase diagram of random 3-SAT



Cf. Lecture on Stat. Mech. Of Disordered Systems by M. Mezard.

Dynamics of the Davis-Putnam-Loveland-Logemann algorithm

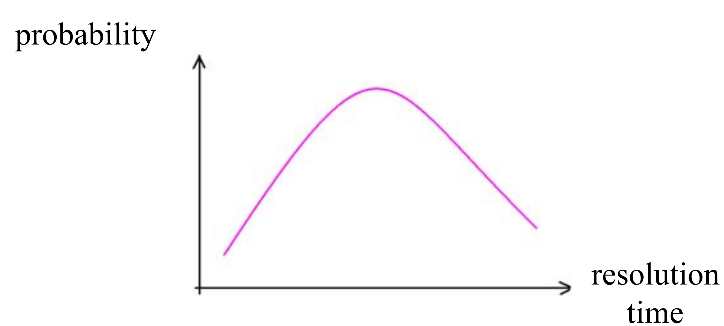
Rigorous analysis of search heuristics

Quantitative study of search trees (with backtracking)

$2^{N w}$

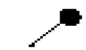
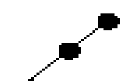
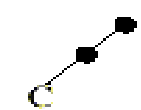
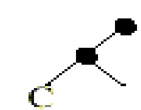
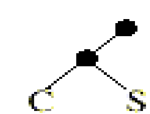
?

Distribution of resolution times (left tail)



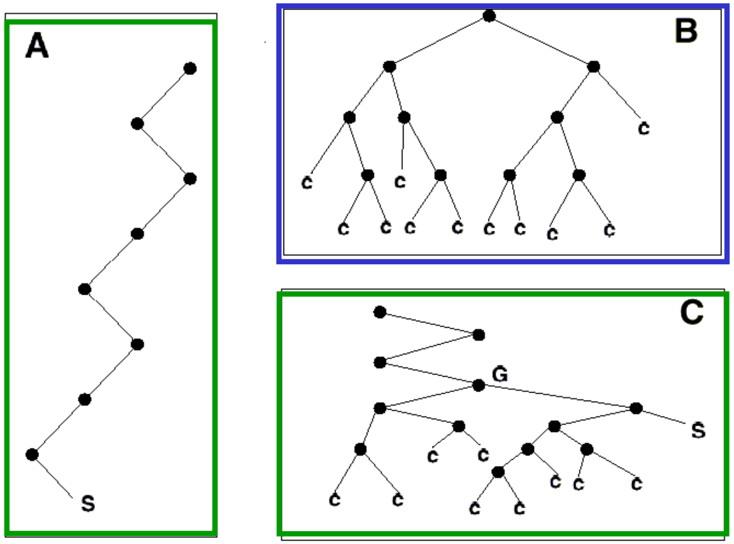
How to solve 3-SAT?

DPLL "Branch & bound" search algorithm

step	clauses	search tree
0	$w \vee \bar{x} \vee y$ $\bar{w} \vee x \vee z$ $\bar{w} \vee \bar{x} \vee \bar{y}$ $\bar{w} \vee \bar{x} \vee y$ $x \vee y \vee \bar{z}$	
1	split : $w = T$	
2	$x \vee z$ $\bar{x} \vee \bar{y}$ $\bar{x} \vee y$ $x \vee y \vee \bar{z}$	
3	split : $x = T$	
4	$\bar{y}$ y	
5	propagation : $y = F$, $y = T$ contradiction	
6	backtracking to stage 1 : $x = F$	
7	z $y \vee \bar{z}$	
8	propagation : $z = T$, $y = T$ solution : $w = T$, $x = F$, $y = T$, $z = T$	

Backtrack algorithm, search tree and heuristic

Davis-Putnam algorithm = heuristic + backtracking
↓
search tree



A satisfiable instance (easy)

B unsatisfiable instance (hard)

C satisfiable instance (hard)

Complexity = size of search tree (= number of nodes –or leaves-)

- Unit-Clause (UC): pick variable in 1-clause if any, or any unset variable
- Generalized unit-clause (GUC): pick variable in shortest clause
- Shortest Clause With Majority (SC): pick most frequent variable in 3-clauses

Trajectories and the 2+p-SAT problem

clauses with 3 var.

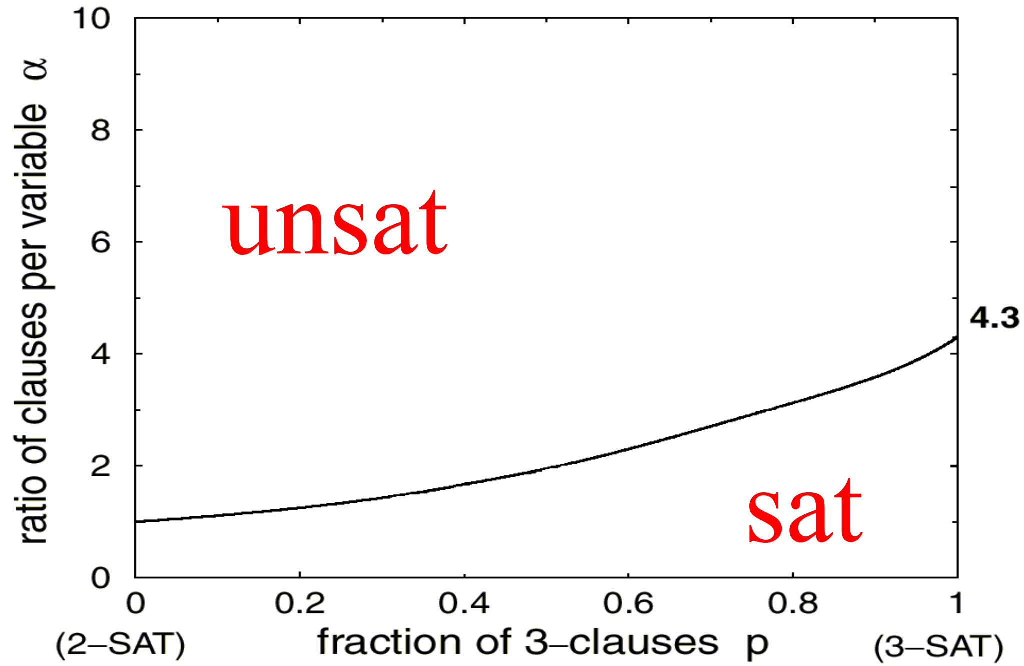
α

"dynamics" of the algorithm

clauses with 2 or 3 var.

α , p

phase diagram of the 2+p-SAT model



Analysis of the GUC heuristic (I)

NO BACKTRACKING !

Evolution rules:

1. If no clause left, output « Satisfiable »;
2. Otherwise, select randomly a shortest clause e.g. *(not y)*, and set one of its literals to True e.g. *y=F.*
3. Simplify remaining clauses;
   e.g. *(not y or z) is removed, (y or w or q)* ö *(w or q)*
3. If a unit clause with false literal appears e.g. *(y)*, output « Contradiction »;
   Else go to 2.

NB: Contradiction does not mean Unsatisfiable!

Analysis of the GUC heuristic (II)

Call $P^{(K)}[M;N]$ the measure over random K-SAT instances with M clauses and N variables.

Call P_T[instance] the measure over instances after T steps of the heuristic.

3-SAT instances $\xrightarrow[\text{heuristic}]{\text{GUC}}$ Mixed 1-,2-,3-SAT instances

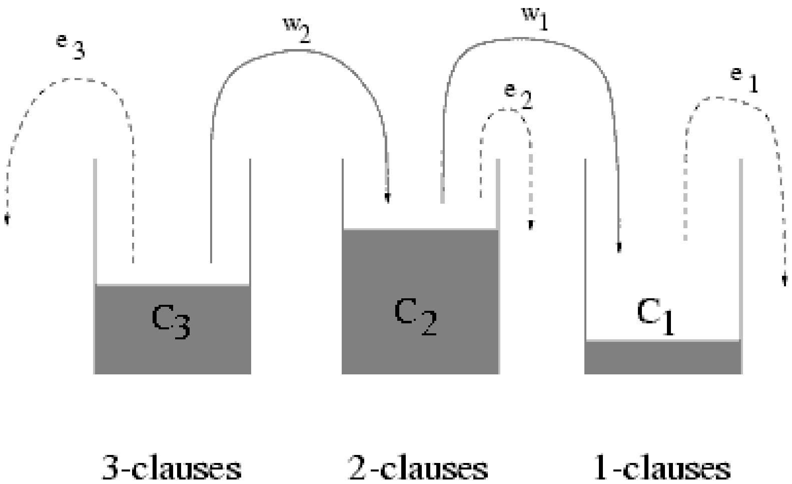
Initial measure
$P_{T=0}[instance] = P^{(3)}[M;N]$

Measure $P_T[instance]$?

Define the numbers $C_1(T)$, $C_2(T)$, $C_3(T)$ of 1-, 2-, 3-clauses after T steps;
Then:

$$\boxed{P_T[\text{instance} \mid C_1(T), C_2(T), C_3(T)] = P^{(1)}[C_1(T);N-T] \approx P^{(2)}[C_2(T);N-T] \approx P^{(3)}[C_3(T);N-T]}$$

Analysis of the GUC heuristic (III)



Flows of
clause populations

$$C_3(T+1) = C_3(T) - e_3(T) - w_2(T)$$
$$C_2(T+1) = C_2(T) - e_2(T) + w_2(T) - w_1(T)$$
$$C_1(T+1) = C_1(T) - e_1(T) + w_1(T) - 1$$

- Average values of e and w: $< e_3(T) > = < w_2(T) > = \dfrac{3}{2\,(N-T)} <C_3(T)> = O(1)$
- Concentration of clause densities:

$$C_j(T) = N\, c_j\left(\frac{T}{N}\right) + o(N) \qquad (j = 2, 3)$$

Reduced time
t=T/N

fl $< e_3(t) > = < w_2(t) > = \dfrac{3\ c_3(t)}{2\ (1-t)}$

Analysis of the GUC heuristic (IV)

Set of coupled Ordinary Differential Equations:

$$
\begin{aligned}
\frac{\mathrm{d}c_3(t)}{\mathrm{d}t} &= -\frac{3}{1-t}c_3(t) - \rho_3(t) \\
\frac{\mathrm{d}c_2(t)}{\mathrm{d}t} &= \frac{3}{2(1-t)}c_3(t) - \frac{2}{1-t}c_2(t) - \rho_2(t)
\end{aligned}
$$

where $r_j(t)$ is the probability that a literal is chosen from a j-clause at time t,

$$r_1(t) + r_2(t) + r_3(t) = 1$$

What are the values of these probabilities?

Analysis of the GUC heuristic (V)

Estimate the initial creation rate of 2-clauses: $< w_2(0) > = \dfrac{3\,a}{2}$

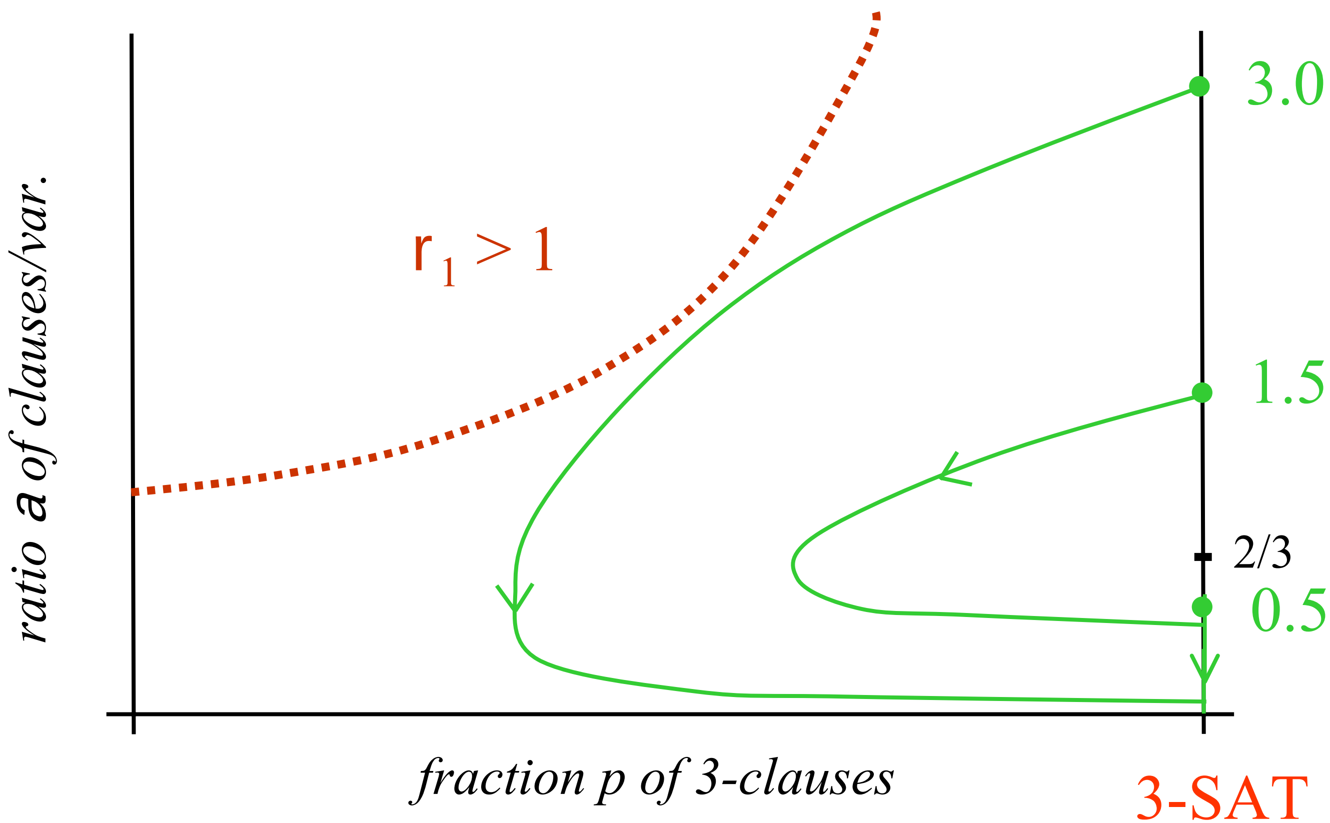
Case $a < 2/3$ $$\rho_1(t) = 0; \quad \rho_2(t) = \frac{3c_3(t)}{2(1-t)}; \quad \rho_3(t) = 1 - \rho_2(t)$$

Case $a > 2/3$ $$\rho_1(t) = \frac{c_2(t)}{1-t}\ ; \quad \rho_2(t) = 1 - \rho_1(t); \quad \rho_3(t) = 0$$

should always be smaller than unity!!!

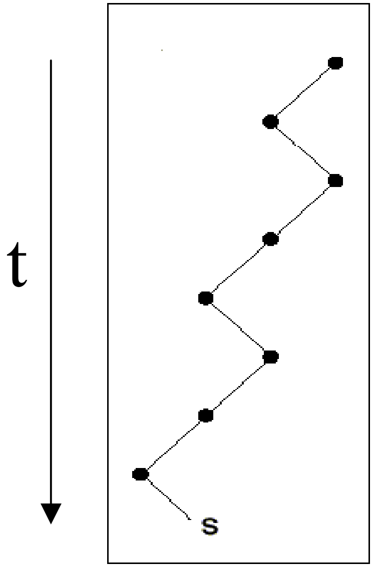
Analysis of the GUC heuristic (VI)

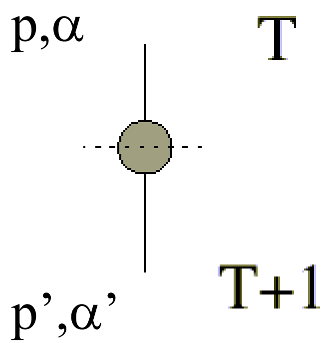
$$p(t) = \frac{c_3(t)}{c_2(t) + c_3(t)}, \qquad \alpha(t) = \frac{c_2(t) + c_3(t)}{1 - t}$$



OK if a < 3.003… (also sufficient condition)

Analysis of the GUC heuristic (VII)



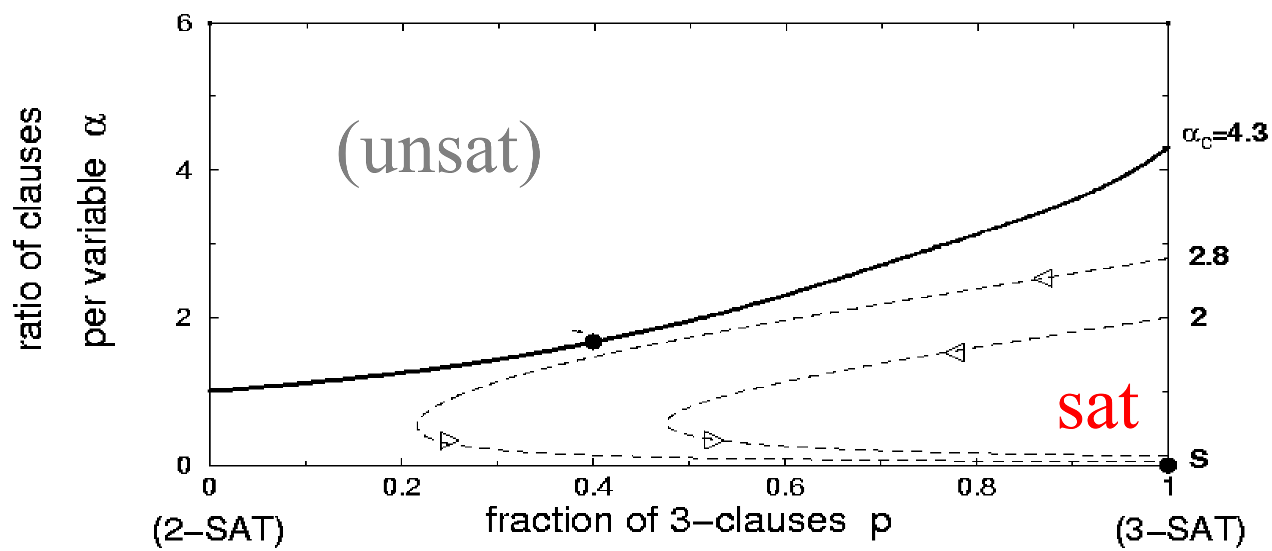


$$p(0)=1\ ,\ \alpha(0)=\alpha_0$$

$$\frac{dp}{dt}=F_p(p,\alpha,t)$$

$$\frac{d\alpha}{dt}=F_\alpha(p,\alpha,t)$$

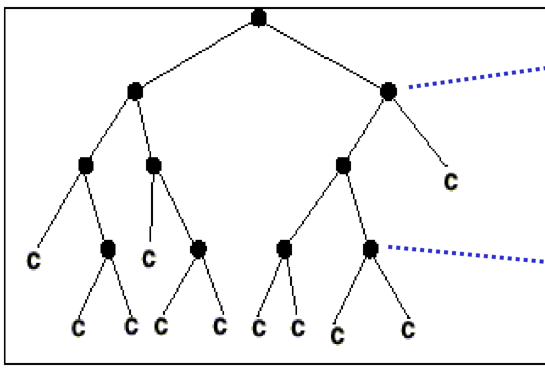
(ODE)



Complete search trees (I)

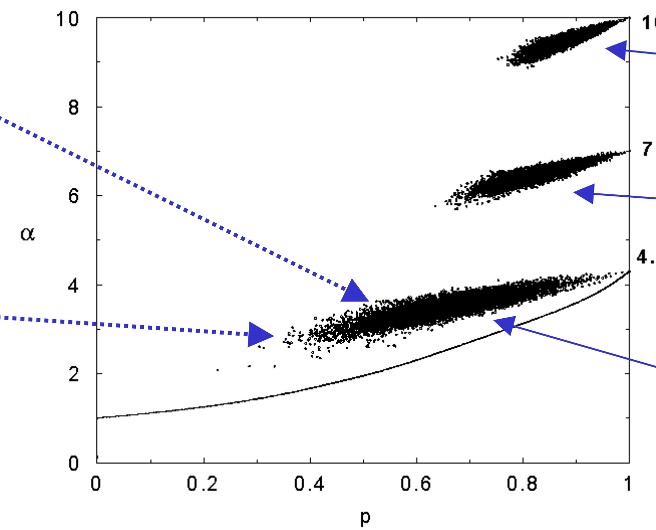
$a > a_C$

2+p-SAT phase diagram



p_1, a_1

p_2, a_2



N=300
6600 nodes

N=200
6300 nodes

N=120
7600 nodes

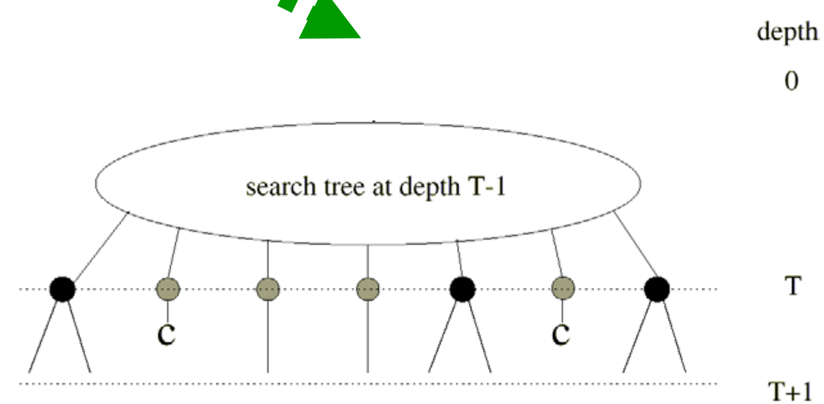
Proof of unsatisfiability

one branch: *$p(t)$, $\alpha(t)$* *ODE* → many branches: *$\omega\,(p,\alpha,t)$* *PDE*

Complete search trees (II)

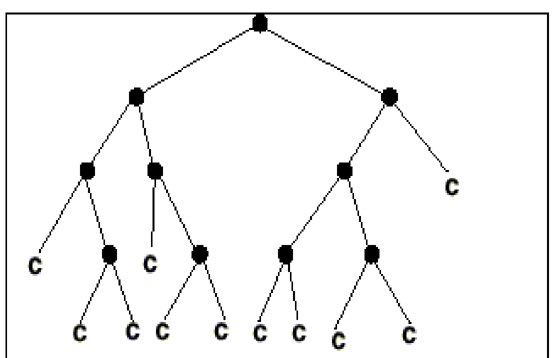
Instance to be solved

DPLL induces a non Markovian evolution of the search tree



Imaginary, parallel building up of the search tree

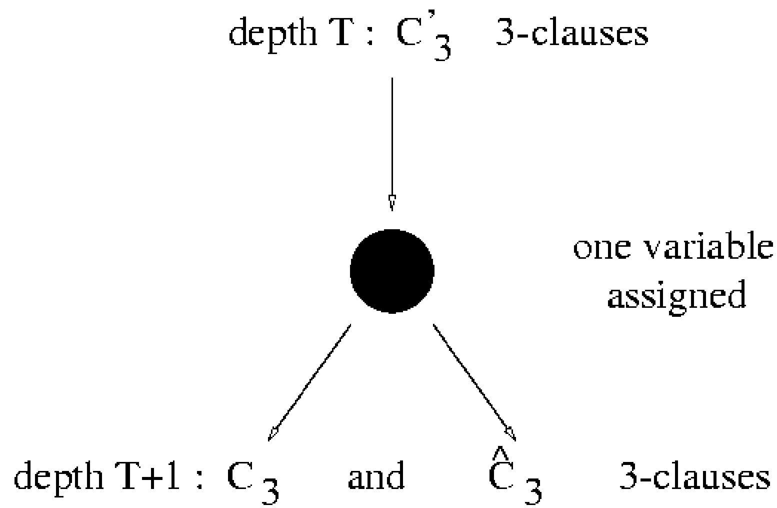
Final Search tree



Toy model of search tree (I)

Consider only
clauses with 3 variables

(Dont'care about empty clauses!)



• Obviously the total number of branches at depth T equals $B(T)=2^T$.

• Consider the number of branches B with C_3 3-clauses:

$$B(C_3; T+1) = \sum_{C_3'=0}^{\infty} K_{\mathrm{toy}}\,(C_3, C_3'; T)\, B(C_3'; T).$$

branching matrix

Toy model of search tree (II)

Branching matrix is simply (twice) a Binomial law:

$$K_{\text{toy}}\,(C_3, C_3'; T) = 2\binom{C_3'}{C_3' - C_3}\left(\frac{3}{N-T}\right)^{C_3'-C_3}\left(1-\frac{3}{N-T}\right)^{C_3}$$

Consider the action of the algorithm during the interval,

$$T_0 = t\,N \leq T \leq T_1 = (t+\epsilon)\,N$$

Then, clause densities are expected to change by O(ϵ), and

$$K_{\text{toy}}(C_3', C_3, t) = 2\,\chi\,(C_3' - C_3)\,e^{-m_3(t)}\,\frac{m_3(t)^{C_3'-C_3}}{(C_3'-C_3)!} + O(\epsilon)$$

where $\quad m_3(\text{t}) = \dfrac{3\,c_3(\text{t})}{2\,(1\text{-t})}$

Toy model of search tree (III)

During time interval $\mathcal{T} = T_1 - T_0 = \epsilon\, N$, branches numbers change according to

$$\boxed{B(C_3; t\, N + \mathcal{T}) = \sum_{C_3'=0}^{\infty} \left[K_{\mathrm{toy}}\right]^{\mathcal{T}} \, (C_3', C_3; t)\, B(C_3'; t\, N)}$$

- Diagonalization of the (Toeplitz) matrix K_{toy}:

Eigenvectors: $v(q_3, C_3) = \mathrm{e}^{\mathrm{i}\, q_3\, C_3} / \sqrt{2\pi}$ with $0 \le q_3 < 2\,\pi$
(wave number)

Eigenvalues: $\lambda_{q_3}(t) = 2\ \exp\left[m_3(t)\ \left(\mathrm{e}^{\mathrm{i}\, q_3} - 1\right) \right]$

- Exponential scaling Ansatz for the numbers of branches:

$$\lim_{N\to\infty} \frac{1}{N} \ln B(c_3\, N; t\, N) = \omega(c_3, t).$$

Toy model of search tree (IV)

Finally, changing the sum over clauses number into an integral through,

$$C_3' - C_3 = r_3\,\mathcal{T}$$

allows to obtain

$$\boxed{\begin{aligned}\exp\Big(N\,\omega(c_3, t+\epsilon)\Big) = \mathcal{T}\int_0^{c_3/\epsilon} \mathrm{d}r_3 \int_0^{2\pi} \frac{\mathrm{d}q_3}{2\pi} \\ \times \exp\Big[N\,\big(\epsilon\ln\lambda_{q_3}(t) - \mathrm{i}\,\epsilon\,r_3\,q_3 + \omega(c_3+\epsilon\,r_3, t)\big)\Big]\end{aligned}}$$

Now:

1. Estimate integrals through saddle-points;
2. Expand up to first order in e assuming w is once differentiable with respect to t and c_3.

Toy model of search tree (V)

We obtain:

$$\frac{\partial \omega}{\partial t}(c_3, t) = \max_{r_3, q_3} \left(\ln \lambda_{q_3}(t) - \mathrm{i}\, r_3\, q_3 + r_3 \frac{\partial \omega}{\partial c_3}(c_3, t) \right)$$

Saddle point over r_3: $q_3 = -\mathrm{i}\frac{\partial \omega}{\partial c_3}$

$$\boxed{\frac{\partial \omega}{\partial t}(c_3, t) = \ln 2 - \frac{3\, c_3}{1-t} + \frac{3\, c_3}{1-t}\; \exp\left[\frac{\partial \omega}{\partial c_3}(c_3, t)\right]}$$

Initial condition:

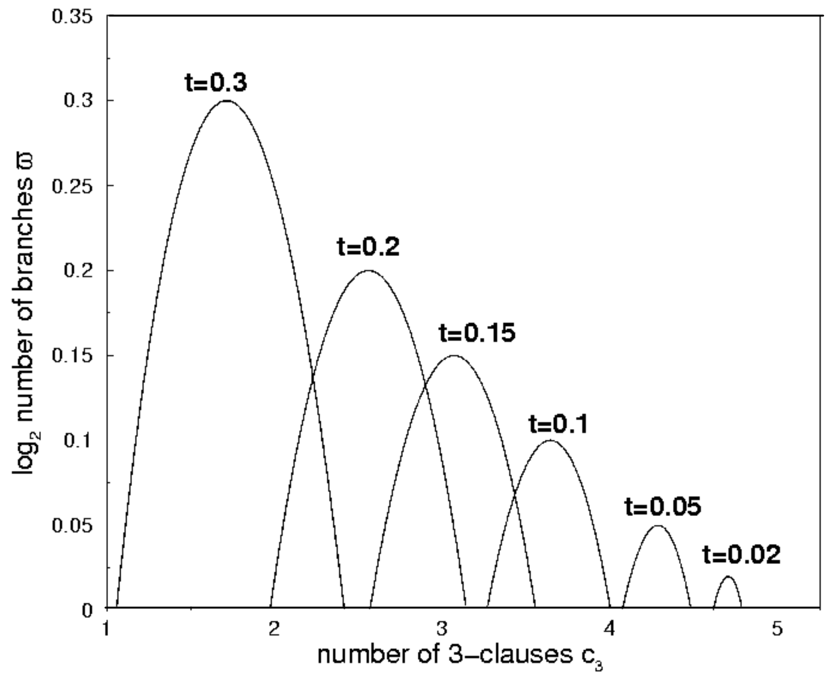
$$\boxed{\mathrm{w}(c_3\,, t{=}0) = \begin{cases} 0 \;\; \text{if } c_3 = \mathrm{a} \\ \text{-¶} \;\; \text{otherwise} \end{cases}}$$

Toy model of search tree (VI)

Solution:

$$\omega(c_3,t) = t\ \ln 2 + 3c_3\ \ln(1-t) + \alpha\ \ \ln\alpha \\ - c_3 \ln c_3 - (\alpha\ \ - c_3)\ln\left[\frac{\alpha\ \ - c_3}{1-(1-t)^3}\right]$$

Graphical representation



NB: max of $w(c_3\ , t)$ gives back most probable $c_3(t)$ for GUC heuristic trajectory.

Analysis of DPLL search tree (I)

Average number of branches with clause populations C_1, C_2, C_3

$$B(C_1, C_2, C_3; T+1) = \sum_{C'_1, C'_2, C'_3 = 0}^{\infty} K\left(C_1, C_2, C_3, C'_1, C'_2, C'_3; T\right) B(C'_1, C'_2, C'_3; T)$$

Branching matrix

$$\begin{aligned} K(C_1,C_2,C_3;C'_1,C'_2,C'_3;T) = & \binom{C'_3}{C'_3 - C_3} \left(\frac{3}{N-T}\right)^{C'_3 - C_3} \left(1 - \frac{3}{N-T}\right)^{C_3} \sum_{w_2=0}^{C'_3 - C_3} \left(\frac{1}{2}\right)^{C'_3 - C_3} \binom{C'_3 - C_3}{w_2} \\ & \times \Bigg\{ (1 - \delta_{C'_1}) \sum_{z_2=0}^{C'_2} \binom{C'_2}{z_2} \left(\frac{2}{N-T}\right)^{z_2} \left(1 - \frac{2}{N-T}\right)^{C'_2 - z_2} \sum_{w_1=0}^{z_2} \left(\frac{1}{2}\right)^{z_2} \binom{z_2}{w_1} \delta_{C_2 - C'_2 - w_2 + z_2}\, \delta_{C_1 - C'_1 - w_1 + 1} + \delta_{C'_1} \\ & \times \sum_{z_2=0}^{C'_2 - 1} \binom{C'_2 - 1}{z_2} \left(\frac{2}{N-T}\right)^{z_2} \left(1 - \frac{2}{N-T}\right)^{C'_2 - 1 - z_2} \sum_{w_1=0}^{z_2} \left(\frac{1}{2}\right)^{z_2} \binom{z_2}{w_1} \delta_{C_2 - C'_2 - w_2 + z_2 + 1} \left[\delta_{C_1 - w_1} + \delta_{C_1 - 1 - w_1}\right] \Bigg\}, \end{aligned}$$

- $B(C_1, C_2, C_3;T) \sim \exp[\, N\ w(c_2, c_3;t)\ ]$ where $c_i = C_i/N$, $t = T/N$
- Distribution of C_1 becomes stationary over $O(1)$ time scale

Analysis of DPLL search tree (II)

Eigenvectors of K: $$v_{q,q_2,q_3}(C_1',C_2',C_3') = \mathrm{e}^{\mathrm{i}(q_2 C_2' + q_3 C_3')}\,\tilde{v}_q(C_1)$$

$$\tilde{K}_{q_2}(C_1,C_1') = \sum_{z_2=0}^{\infty} \mathrm{e}^{-m_2}\,\frac{(m_2\,\mathrm{e}^{\mathrm{i}q_2})^{z_2}}{z_2!} \sum_{w_1=0}^{z_2} \binom{z_2}{w_1} \frac{1}{2^{z_2}}$$
$$\times \left\{ \delta_{C_1-C_1'-w_1+1}\,(1-\delta_{C_1'}) + \mathrm{e}^{\mathrm{i}\,q_2}(\delta_{C_1-w_1}+\delta_{C_1-1-w_1})\,\delta_{C_1'} \right\}$$

Effective matrix in (C_1,C_1’) sector

Analysis of DPLL search tree (III)

Generating function of Eigenvector components

$$V_q(x) = \sum_{C_1=0}^{\infty} \tilde{v}_q(C_1)\, x^{C_1} = V_q(0) \frac{L(x)\; N(x)}{\Lambda_q - L(x)}$$

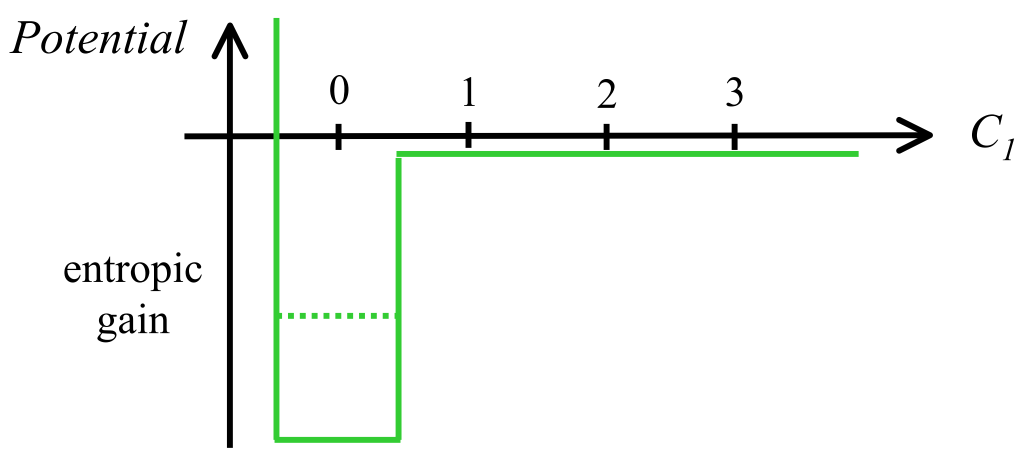
$$N(x) = \mathrm{e}^{-y_2}\,(x+x^2) - 1, \quad L(x) = \frac{1}{x}\,\exp\left(\frac{m_2}{2}\,\mathrm{e}^{-y_2}\,x\right)$$

Asymptotic behavior:

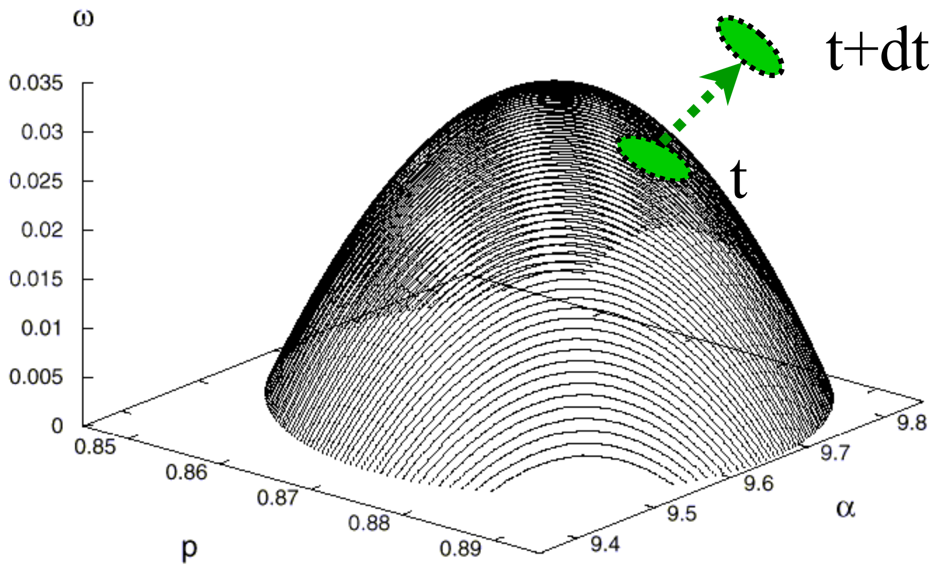
$$\tilde{v}_0(C_1) \sim \left(\frac{1}{R}\right)^{C_1} \longrightarrow V_0(x) \sim (R-x)^{-1}$$

Pole of V_0 located in R>1 (localized eigenvector), or R=1 (extended eigenvector)

Analogy with
1D-Quantum Mechanics
(discrete, non hermitean)



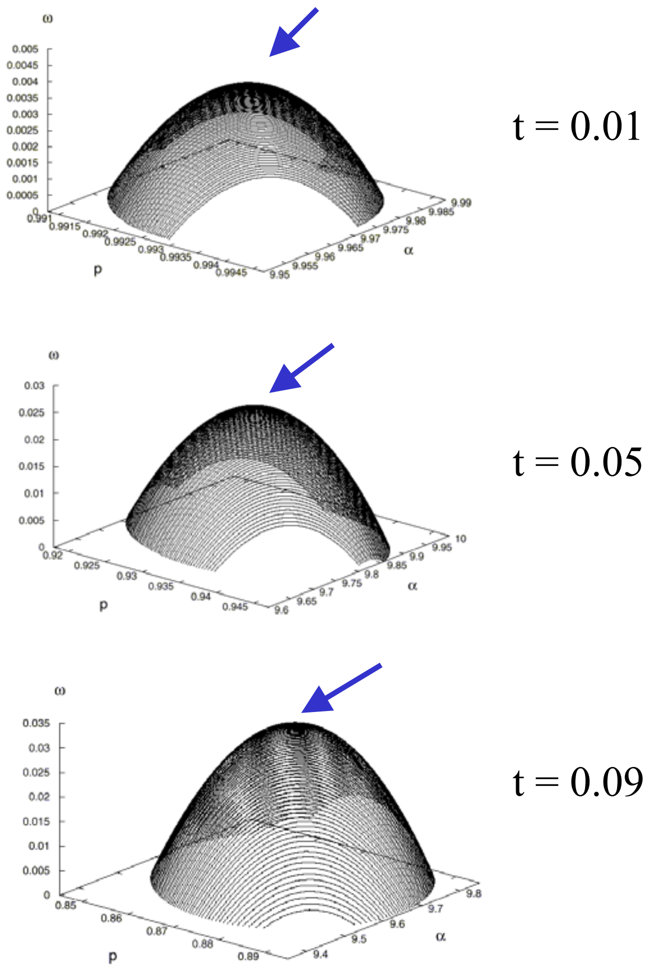
Analysis of DPLL search tree (IV)

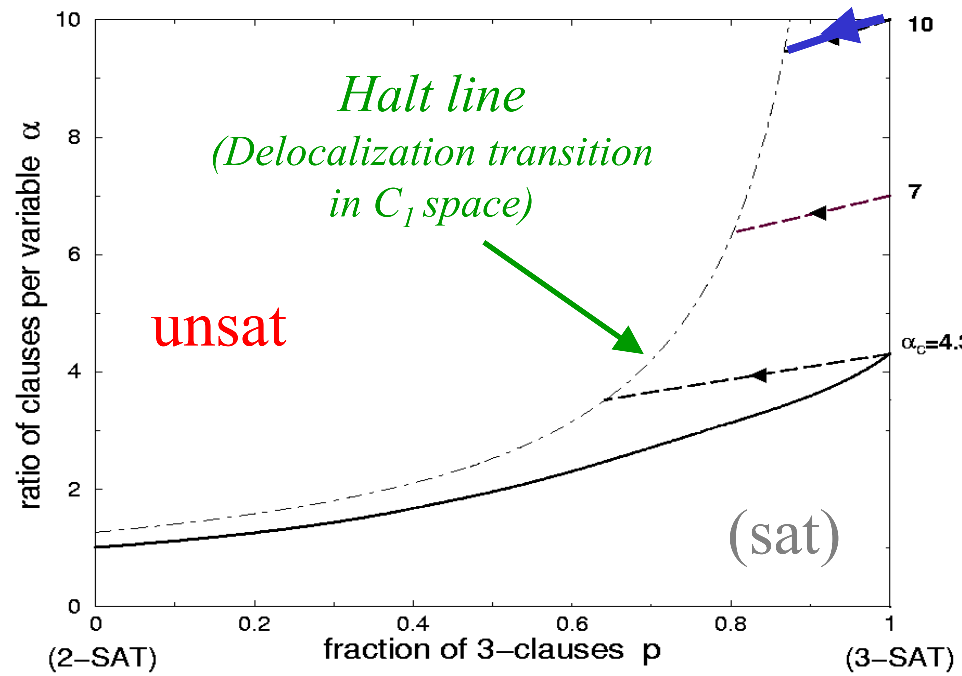


$$\frac{\partial \omega}{\partial t} = \mathcal{H}\left[p, \alpha, \frac{\partial \omega}{\partial p}, \frac{\partial \omega}{\partial \alpha}, t \right] \qquad \text{(PDE)}$$

+ moving frontier between alive (finite C_1) and dead branches (C_1=O(N))

Analysis of DPLL search tree (V)





$\alpha_0 = 10$

Comparison to numerical experiments

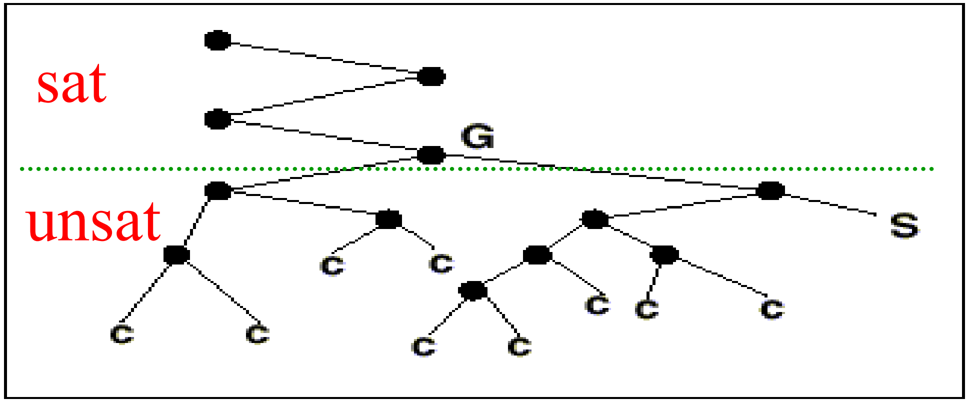
$$Q = 2^{N\omega}$$

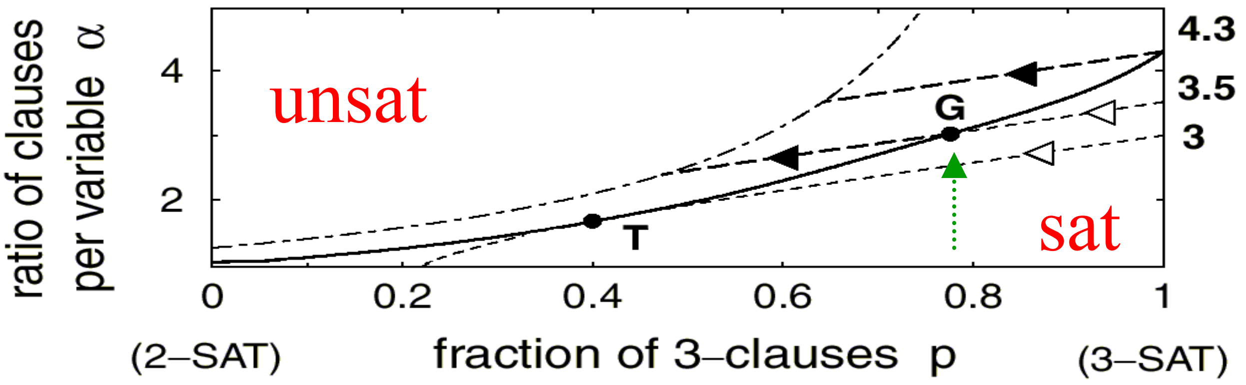
	Initial Ratio α_0	Experiments $\log_2 Q$	$\log_2 B$	Theory $\hat{\omega}$
unsat	20	0.0153 ± 0.0002	0.0151 ± 0.0001	0.0152
	15	0.0207 ± 0.0002	0.0206 ± 0.0001	0.0206
	10	0.0320 ± 0.0005	0.0317 ± 0.0002	0.0319
	7	0.0482 ± 0.0005	0.0477 ± 0.0005	0.0477
	4.3	0.089 ± 0.001	0.0895 ± 0.001	0.0875
sat	3.5	0.034 ± 0.003		0.035
	G	0.040 ± 0.002	0.041 ± 0.003	0.044
		(nodes)	(leaves)	

$$\omega = \frac{3+\sqrt{5}}{6\ln 2}\left[\ln\left(\frac{1+\sqrt{5}}{2}\right)\right]^2 \frac{1}{\alpha} \approx \frac{0.292}{\alpha}$$

Satisfiable, hard instances 3.003< α < 4.3

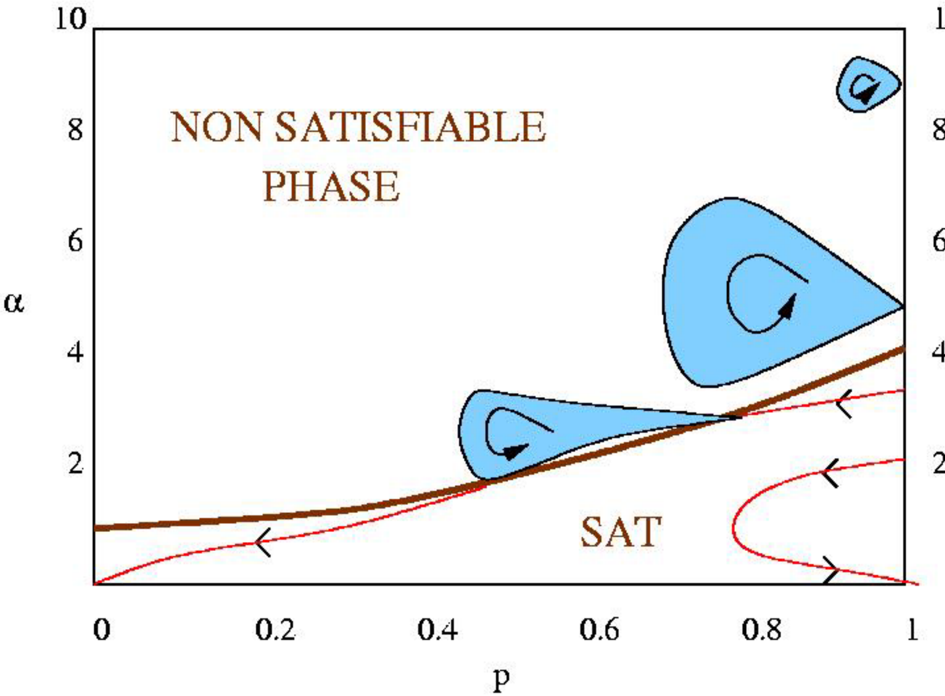
(which could made be easier?)

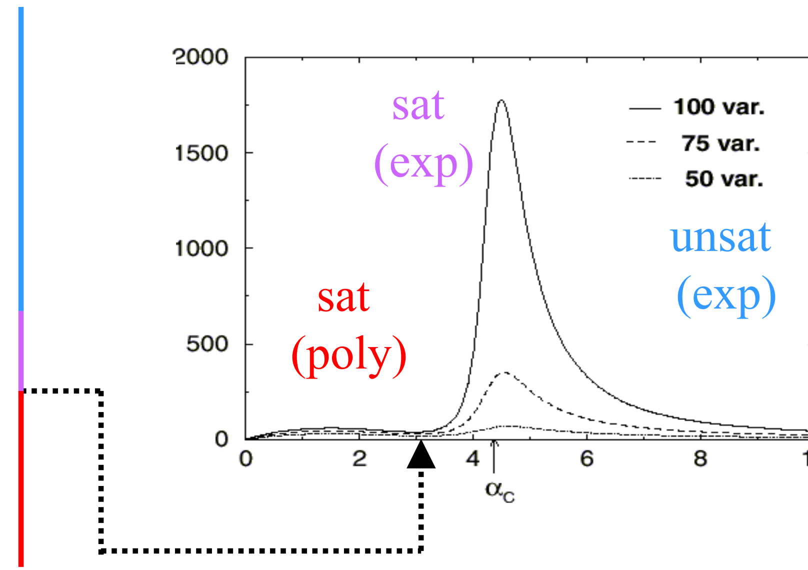




The complexity of 3-SAT solving is strongly affected by the phase transitions of 2+p-SAT .

The polynomial/exponentiel crossover





"dynamical" transition
(depends on the heuristic)

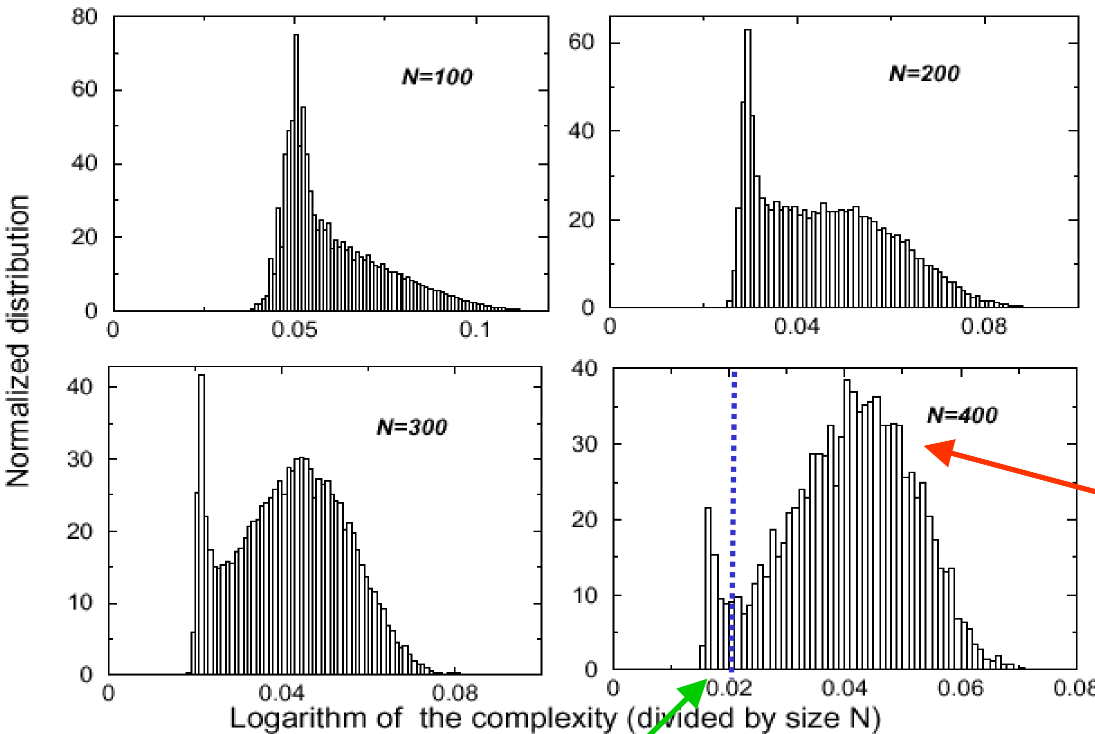
$$\begin{aligned} UC &: \quad 2.667 \\ GUC &: \quad 3.003 \\ SC &: \quad \approx 3 \end{aligned}$$

but

$$p_T = \frac{2}{5}, \alpha_T = \frac{5}{3}$$

T is largely heuristic independent (universality)

Fluctuations of complexity for finite instance size



Histograms of solving times

α=3.5

Exponential regime Complexity $= 2^{0.035\, N}$ + fluctuations !

Linear regime
Very rare! frequence $= 2^{-0.011\, N}$

Talks later this week by S. Cocco and A. Montanari

Bibliography

1. DPLL algorithm:
   M. Davis, H. Putnam, J. Assoc. Comput. Mach. 7, 201 (1960).
   M. Davis, G. Logemann, D. Loveland, Comm. ACM 5, 394 (1962).

2. Analysis of search heuristics:
   M. Chao, J. Franco, Inf. Sci. 51, 289 (1990).
   A. Frieze, S. Suen, J. Algorithms 20, 312 (1996).
   D. Achlioptas, Theor. Comp. Sci. 265, 159 (2001).
   A. Kaporis, L. Kirousis, E. Lalas, proceedings of SAT2002, 362 (2002).

3. The 2+p-SAT model:
   R. M., R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Nature 400, 133 (1999).
   D. Achlioptas, L. Kirousis, E. Kranakis, D. Krizanc, Theor. Comp. Sci. 265, 109 (2001).

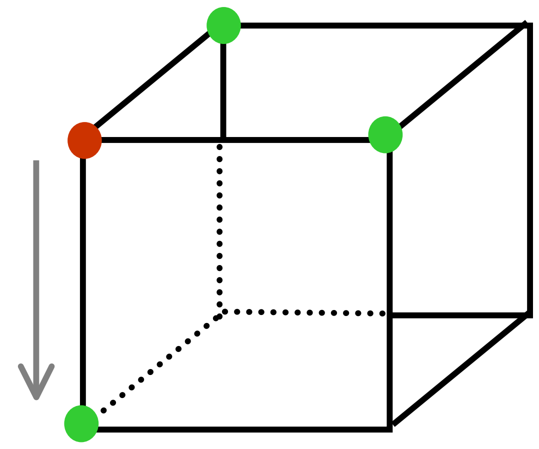
4. Analysis of DPLL search tree:
   V. Chvàtal, E. Szmeredi, Journal of the ACM 35, 759 (1988).
   P. Beame, R. Karp, T. Pitassi, M. Saks, STOC 98, 561 (1998).
   S. Cocco, R. M., Phys. Rev. Lett. 86, 1654 (2001); Eur. Phys. B 22, 505 (2001).
   D. Achlioptas, P. Beame, M. Molloy (2002).

5. Vertex covering of random graphs:
   Papers by A. Hartmann, M. Weigt: Phys. Rev. Lett. 86, 1658 (2001).

Local search algorithm

Configuration: set of N Boolean variables or spins $x_i = 0,1$
Goal: minimize some cost function over the x's.

Configuration space
for N=3



Local search algorithm:
1. Start from some (more or less) random configuration C (●);
2. Look at neighbors of C in Configuration Space (●), and their attached costs;
3. Move to one of these neighbors (⟶) with some transition rate, or remain in C.
4. Go to 2.

Gradient descent procedure

= Monte Carlo Algorithm at zero temperature

- The procedure:

  a. *Start from a random configuration, call E the number of unsatisfiable clause (energy).*
  b. *If E=0, STOP.*
  c. *If E>0, pick up a variable randomly; flip it if the corresponding energy E' is smaller or equal to E. Goto b.*

- Possible relationship between Replica results and MC method

- Example of 3-XORSAT

Random 3-XORSAT (I)

Logical XOR instead of OR = system of linear equations modulo 2

- N Boolean variables $x_1, x_2, x_3, \ldots, x_N$;
- M clauses (pick up randomly three variables and a second member equal to 0 or 1), e.g.

$$x_1 + x_3 + x_{12} = 0 \mod 2$$

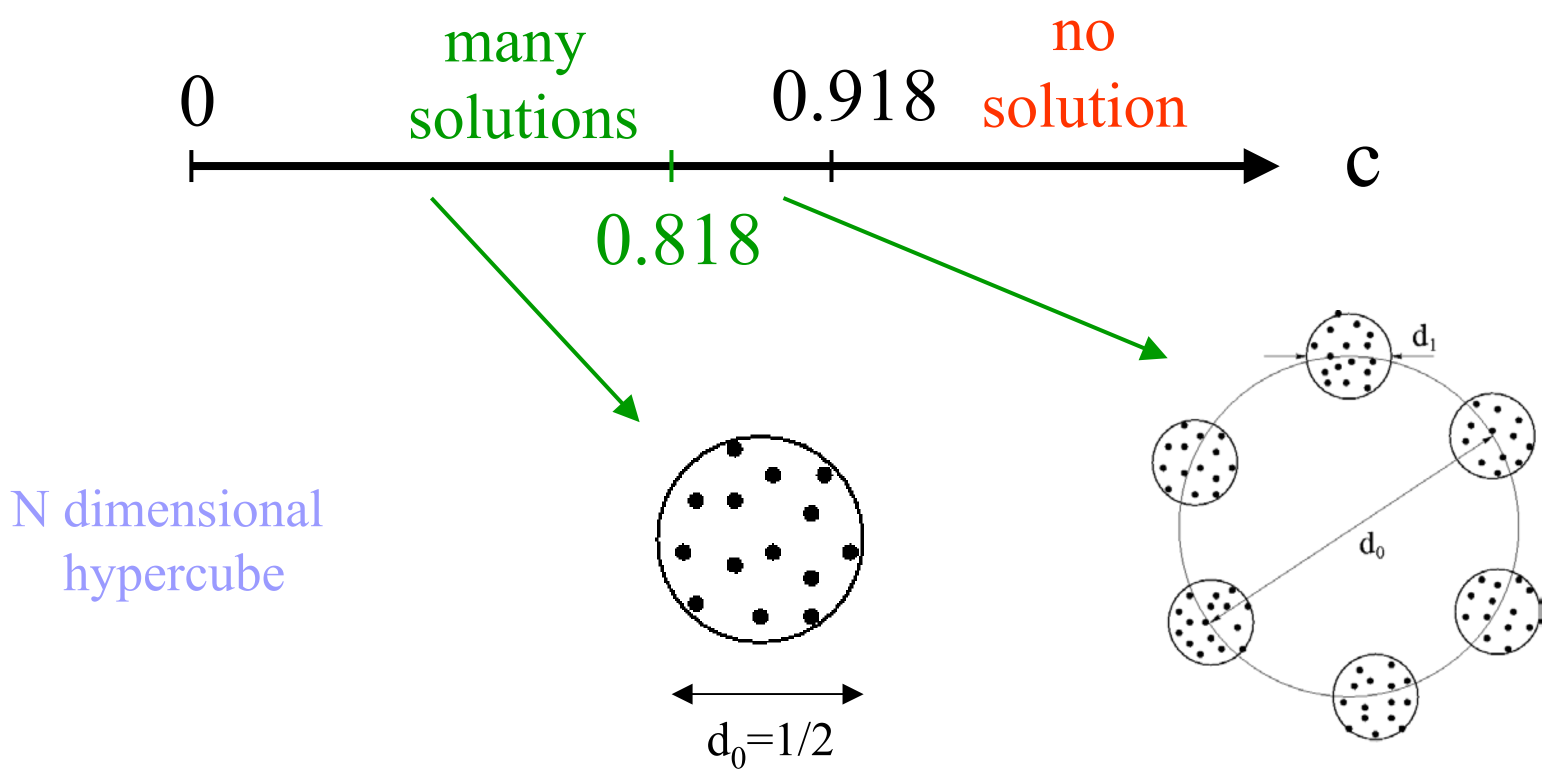
- repeat M times.

Are all clauses (equations) satisfiable together?

…. depends on ratio $c=M/N$!

Random 3-XORSAT (II)

Phase diagram derived from Replica method, then Rigorous analysis:

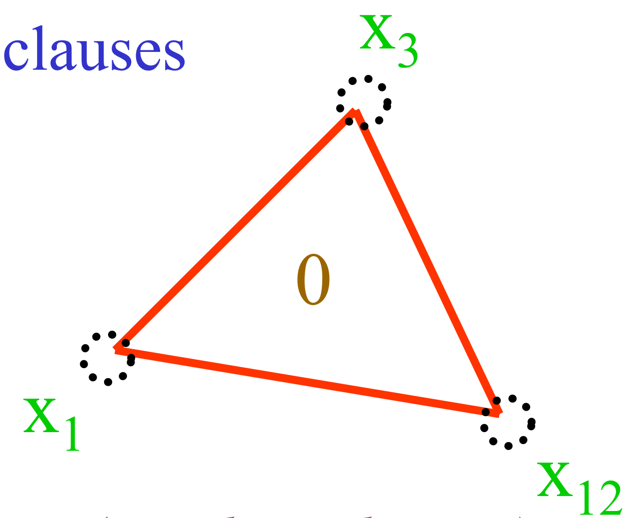


Hamming distance (fraction of
variables that differ) between solutions

Random 3-XORSAT (III)

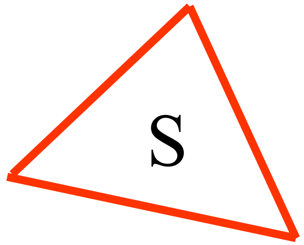
Graphical representation of clauses

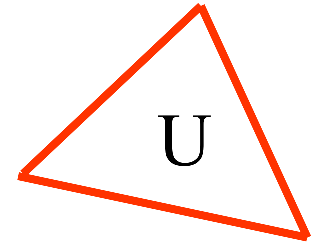
$$x_1 + x_3 + x_{12} = 0 \quad \longrightarrow$$



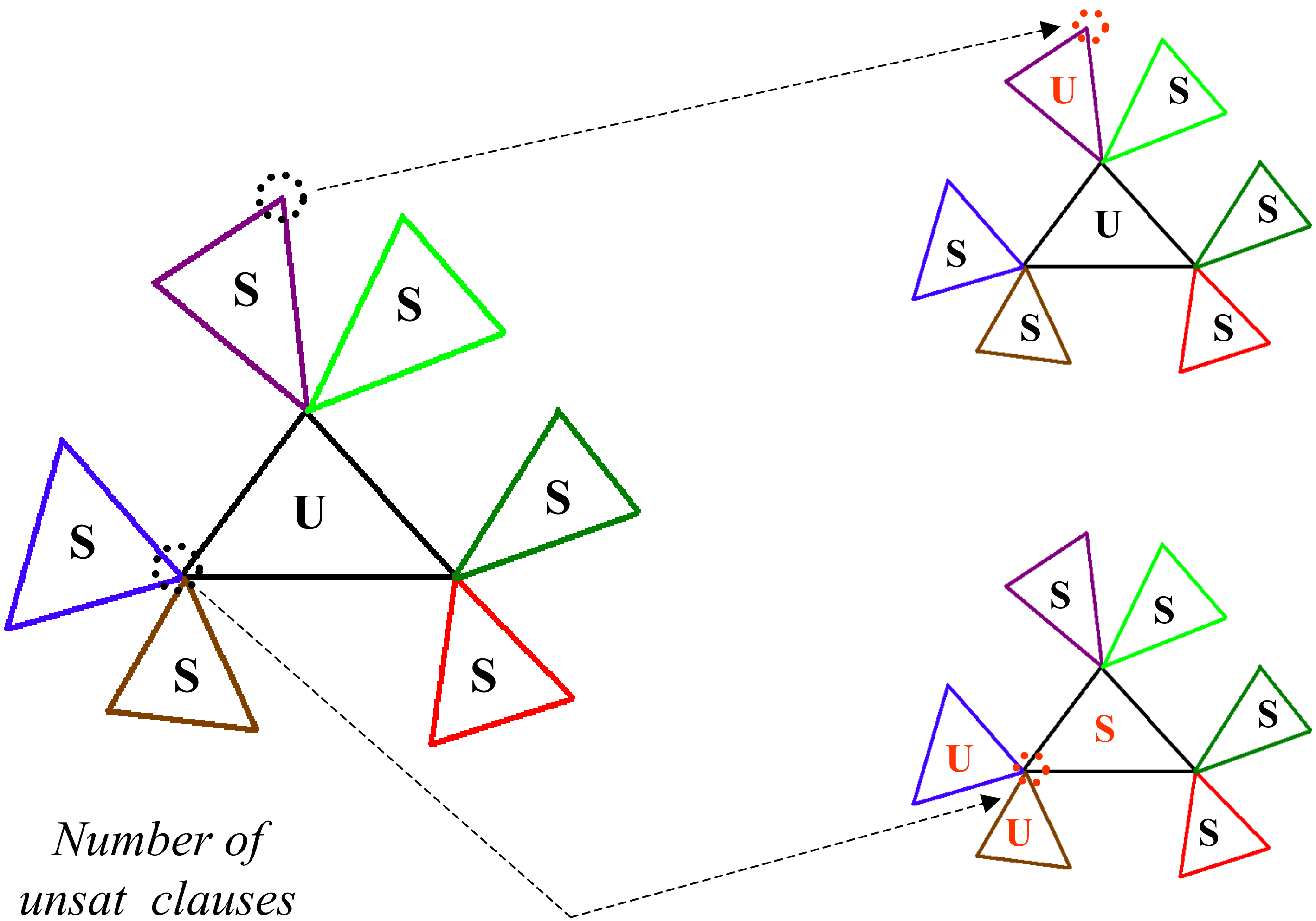
(triangle, or plaquette)

Depending on values of variables, the clause is Satisfied or Unsatisfied,





Blocked Islands (I)



Number of unsat clauses = 1

Unsat = 2

Unsat = 2

Blocked Islands (II)

Statement:

Blocked islands prevent the algorithm from reaching the ground state

Proof:

• *show that, with high probability, there is an extensive number*

$$\text{Islands} = \frac{729}{8} c^7 e^{-45c} N + o(N)$$

of islands, a finite fraction of which are blocked.

• *deduce a lower bound to the final energy reached by Gradient descent*

Energy ¥ q(c) . N

Blocked Islands (III)

NB: Fixed probability (not fixed number) statistical ensemble

Any triplet p of vertices carries a clause with prob. $\boxed{\mu = \frac{cN}{\binom{N}{3}} = \frac{6c}{N^2} + O\left(\frac{1}{N^3}\right)}$

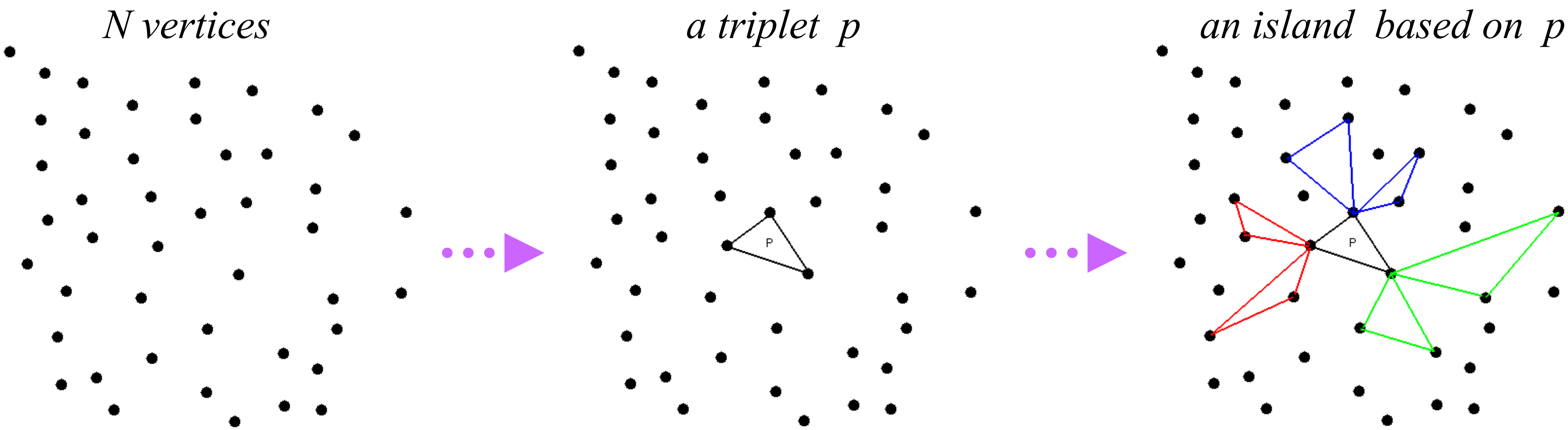
Draw randomly a set of clauses, i.e. a set of triangles joining vertices

Define $\boxed{\mathcal{I}_p = \begin{cases} 1 & \text{if triplet } p \text{ is the center of an island} \\ 0 & \text{if not} \end{cases}}$

Show that $$\mathcal{I} = \sum_p \mathcal{I}_p = \text{Number of islands}$$

is highly concentrated and calculate its typical value.

Blocked Islands (IV)



$$E[I_p] = \frac{(N-3)(N-4)(N-5)(N-6)}{8} \times \frac{(N-7)(N-8)(N-9)(N-10)}{8}$$

$$\times \frac{(N-11)(N-12)(N-13)(N-14)}{8}$$

$$\times \mu^A (1-\mu)^B$$

with $A = 7$, $B = \underbrace{\binom{N}{3}}_{\text{total \# triplets}} - \underbrace{\binom{N-15}{3}}_{\text{\# triplets external to the island}} - 7$

Blocked Islands (V)

Average number of islands per vertex:

$$\lim_{N\to\infty} \frac{1}{N} E[\mathcal{I}] = \lim_{N\to\infty} \frac{1}{N} \binom{N}{3} E[\mathcal{I}_p] = \boxed{\frac{729}{8}\, c^7 \;\exp(-\,45\, c)}$$

Concentration:

E[x]

consider $x = \mathcal{I}/N$

Chebyshev's inequality: for all e > 0, $\mathrm{Prob}\left(\left|x - E[x]\right| \geq \epsilon\right) \leq \frac{\mathrm{Var}[x]}{\epsilon^2}$

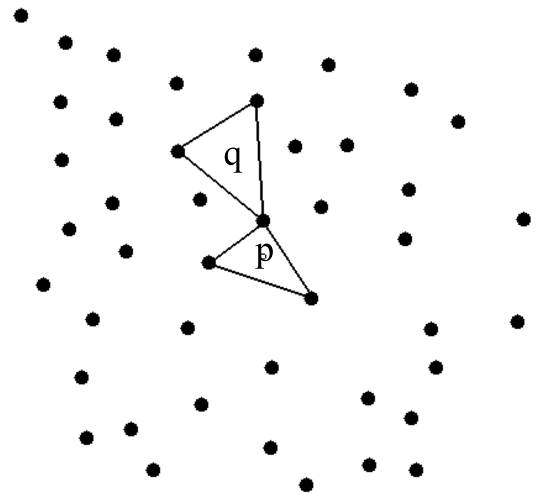
show that Var[x] tends to 0 as N tends to infinity …

Blocked Islands (VI)

Need to calculate second moment:

$$E[\mathcal{I}^2] = \sum_{p,q} E[\mathcal{I}_p \mathcal{I}_q]$$

Depends only upon the number *l* of common vertices between *p* and *q* = 0, 1, 2 or 3



$$\begin{aligned}
E_{\ell=3} &= E[\mathcal{I}_p^2] = E[\mathcal{I}_p] = \frac{3^7}{2^2}\, c^7\, e^{-45\,c}\, N^{-2} + O(N^{-3}) \\
E_{\ell=2} &= E_{\ell=1} = 0 \\
E_{\ell=0} &= \frac{(N-6)\ldots(N-17)}{2^9}\,\frac{(N-18)\ldots(N-29)}{2^9}\mu^{14}\,(1-\mu)^{\binom{N}{3}-\binom{N-30}{3}-14} \\
&= \frac{3^{14}}{2^4}\, c^{14}\, e^{-90\,c}\, N^{-4} + O(N^{-5})
\end{aligned}$$

Blocked Islands (VII)

$$
\begin{aligned}
E[\mathcal{I}^2] &= \binom{N}{3} E_{\ell=3} + \binom{N}{3}\binom{N-3}{3} E_{\ell=0} \\
&= E[\mathcal{I}]^2 + O(N)
\end{aligned}
$$

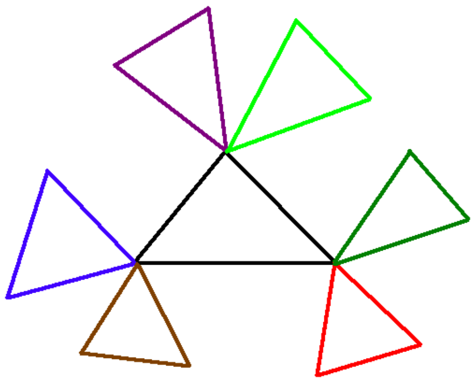
î $\lim_{N\to\infty} \mathrm{Var}[x] = 0$

Thus, with high probability, *x* does not deviate from the expectation value,

$$\mathrm{E}[x] = \frac{729}{8} c^7 \exp(-45\,c)$$

Blocked Islands (VIII)

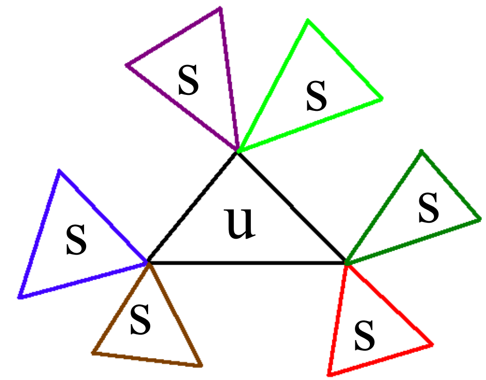
island



with prob.

$$\frac{1}{2^7}$$

blocked island



Thus with high probability, there is an number

$$BI = \frac{729}{1024}\, c^7\, e^{-45c}\, N + o(N)$$

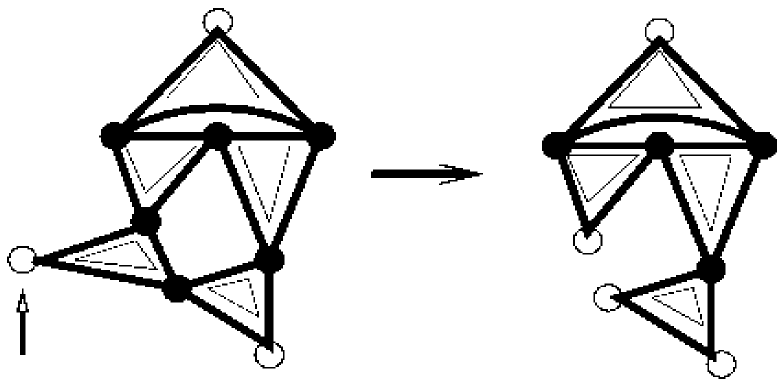
of blocked islands, each with an energy equal to unity.
This gives a lower bound to the final energy reached by Gradient descent

Energy ¥ BI

Comments on Gradient descent and XORSAT

- 3-XORSAT easy to solve (linear algebra problem)

- Easy to solve up to c_d=0.818 with a polynomial algorithm looking at vertex degrees (non local)



- Cannot be solved by gradient descent, even when looking Q steps ahead for any finite Q!

  BUT bound is very small: q(c,Q) N < q(c,1) N < 1.5 10^{-9} N

  cannot be seen for sizes < 1 billion spins

  Necessary condition for convergence: q(c,Q) N ö 0, i.e. Q ~ log N

- What happens for other energy functions e.g. ferromagnets, random 3-SAT?

Bibliography

1. Local search algorithm for SAT:
   J. Gu, P.W. Purdom, J. Franco, B. Wah. DIMACS Series on Discrete Mathematics and Theoretical Computer Science 35:19-151 (1997).
2. XORSAT, definition and bounds:
   N. Creignou, H. Daude, Discrete Applied Mathematics 96-97, 41 (1999)
   N. Creignou, H. Daude, O. Dubois, preprint (2001).
3. XORSAT with replicas:
   F. Ricci-Tersenghi, M. Weigt, R. Zecchina, Phys. Rev. E 63, 026702 (2001)
   S. Franz, M. Leone, F. Ricci-Tersenghi, R. Zecchina, Phys. Rev. Lett. 87, 127209 (2001).
4. XORSAT, mathematical analysis of thresholds:
   O. Dubois, J. Mandler, to appear in FOCS'02.
   S. Cocco, O. Dubois, J. Mandler, R. M., cond-mat/0206239
   M. Mezard, F. Ricci-Tersenghi, R. Zecchina, cond-mat/0207140
5. Gradient descent and spin models:
   R. Melin, J.C. Angles d'Auriac, P. Chandra, B. Doucot, J. Phys. A 29, 5773 (1996)
   D.S. Dean, Eur. Phys. J B 15, 493 (2000)
   P. Svenson, Phys. Rev. E 64, 036122 (2001)
   O. Häggström, Physica A 310, 275 (2002)
   V. Spirin, P.L. Krapivsky, S. Redner, Phys. Rev. E 65, 016119 (2001)

Unsatisfied clause flip procedure

- The clause flip procedure
- Exact results for 2-SAT and 3-SAT
  upper bounds on resolution times valid for worst case
- Mean Field approach
  approximate analysis of average performances for 3-SAT
- Conclusion

Clause flip procedure

1. Pick up an initial (random) configuration of variables;
2. If all clauses are satisfied, STOP.
3. Else, choose randomly one unsat clause and flip one of its attached variables; Goto 2.

Example:

set of clauses

$$
\begin{array}{lcl}
(\text{NOT } x_1 \text{ or } x_2 \text{ or } x_3) & \longrightarrow & F \\
(x_1 \text{ or } x_2 \text{ or NOT } x_4) & \longrightarrow & T \\
(\text{NOT } x_1 \text{ or } x_2 \text{ or } x_4) & \longrightarrow & T
\end{array}
$$

1. random configuration x_1=T, x_2=F, x_3=F, x_4=T
2. pick up first clause and flip one variable it contains

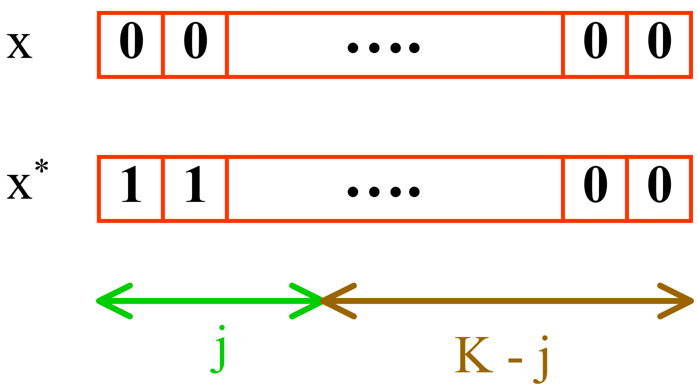
   e.g. x_2=F ö x_2=T then all clauses are satisfied

   e.g. x_1=T ö x_1=F then clause 1 is satisfied, but clause 2 is not any longer.

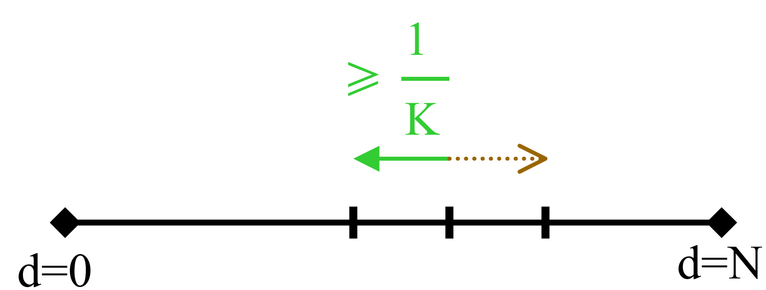
Energy is not guaranteed to decrease!

Exact results for 2-Sat and 3-Sat

Suppose the formula is satisfiable, and call $(x^*)_i$ a solution. Consider a clause not satisfied by x,



$$d=j \quad \text{ö} \quad d' = \begin{cases} j+1 & \text{with prob. } (K-j)/K \\ j-1 & \text{with prob. } j/K \end{cases}$$



2-SAT: free 1-D diffusion, resolution time growing as N^2 at most.

3-SAT: probability of success $> (1/2)^{d^*}/O(N^{1/2}) > (3/4)^N / O(N^{1/2})$
thus resolution time growing as $(4/3)^N$
recently: 1.329… instead of 1.333…

Mean field analysis (I)

Parameters:

N = number of variables
M = number of clauses = a N
K = number of variables in each clause (K-SAT)

- Define $M_j(T)$ the number of clauses satisfied by j literals after T clause flips with j = 0, 1,…, K.
  ($M_0(T)$ is the number of unsatisfied clauses)

- Normalization condition: $$\sum_{j=0}^{K} M_j(T) = M$$

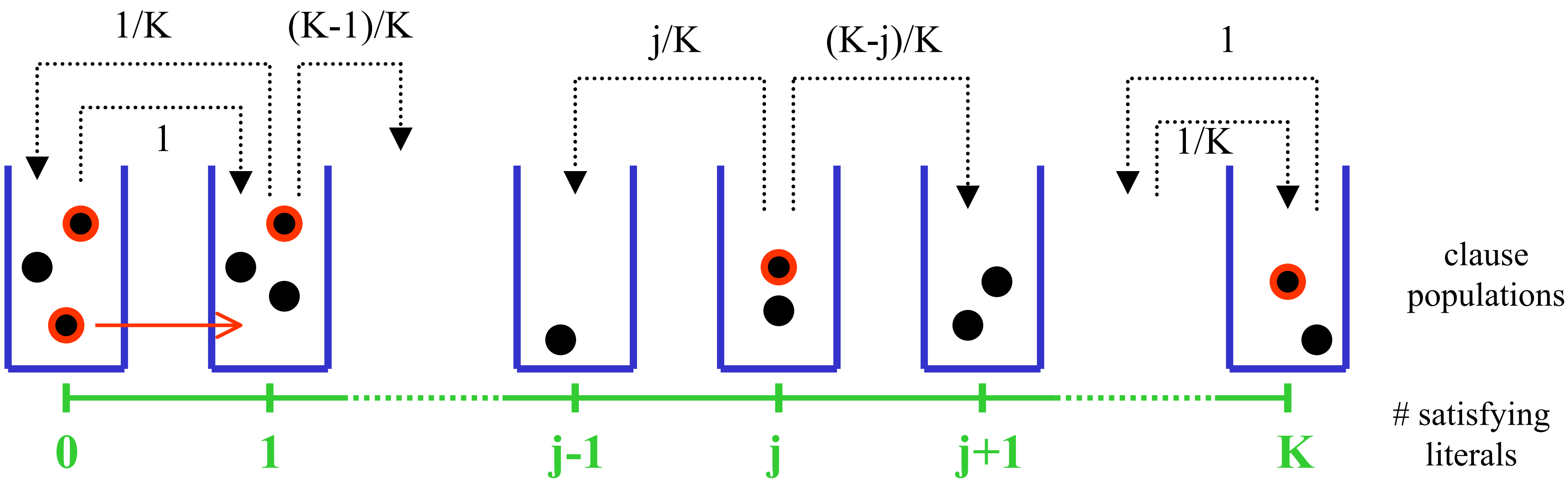
- Define fractions of j-clauses: $$\phi_j(t) = M_j(T)/M$$

Question: evolution equations for the fractions when T Ø T+1 ?

Mean field analysis (II)

Initial conditions for the fractions of j-clauses: $$\phi_j(0) = \frac{1}{2^K}\binom{K}{j}$$

After a new clause flip, i.e. from T Ø T+1:



$$M_j(T+1) - M_j(T) = \frac{K}{N}\Big((K-j+1)/K \;\; M_{j-1}(T) + (j+1)/K \;\; M_{j+1}(T) \Big) - d_j + d_{j-1}$$

Mean field analysis (III)

DT = 1 leads to DM_j = q(1). Thus, fractions F_j vary by q(1) on time scale DT = q(M). Define reduced time: t = T/M.

Evolution equation:

$$\frac{d\vec{\phi}}{dt} = \vec{v} + \alpha\, K\, W_r.\vec{\phi}$$

with

$$\vec{v} = \begin{pmatrix} -1 \\ 1 \\ 0 \\ .. \\ 0 \end{pmatrix} , \qquad W_r = \begin{pmatrix} -1 & \frac{1}{K} & 0 & 0 & 0 & .. & 0 \\ 1 & -1 & \frac{2}{K} & 0 & 0 & .. & 0 \\ 0 & \frac{K-1}{K} & -1 & \frac{3}{K} & 0 & .. & 0 \\ .. & .. & .. & .. & .. & .. & ... \\ 0 & 0 & 0 & 0 & .. & & -1 \end{pmatrix}$$

NB: normalization of fractions comes from nullity of matrix elements summed over columns.

Mean field analysis (IV)

Introduction of the generating function:

$$\varphi(x,t) = \sum_{j=0}^{K} \phi_j(t)\, x^j \qquad \text{with} \qquad \varphi(x,0) = \left(\frac{1+x}{2}\right)^K$$

$$\boxed{\frac{\partial \varphi}{\partial t}(x,t) = -1 + x + \alpha K(x-1)\varphi(x,t) + \alpha(1-x^2)\frac{\partial \varphi}{\partial x}(x,t)}$$

1. *Homogeneous equation:* $\varphi_0(x,t) = -\dfrac{1}{\alpha K}$

$$\psi(x,t) = \varphi(x,t) - \varphi_0(x,t)$$

$$\frac{\partial \psi}{\partial t}(x,t) = \alpha K(x-1)\psi(x,t) + \alpha(1-x^2)\frac{\partial \psi}{\partial x}(x,t)$$

First order linear Partial Differential Equation

Mean field analysis (V)

2. Wave propagation …

$$\chi(x,t) = (1+x)^K \;\psi(x,t)$$

$$\frac{\partial \chi}{\partial t}(x,t) = \alpha(1-x^2)\frac{\partial \chi}{\partial x}(x,t)$$

$$\chi(x,t) = \Omega\left(\frac{1}{\alpha}\,\tanh^{-1}(x) + t\right)$$

Initial conditions allow to determine the wave envelope

$$\Omega(u) = \frac{1}{2^K} + \frac{1}{\alpha K}\,[1+\tanh(\alpha u)]$$

Solution

$$\boxed{\varphi(x,t) = \left(\frac{1+x}{2}\right)^K + \frac{1}{\alpha K}\left\{\left[\frac{1+x\tanh(\alpha t)}{1+\tanh(\alpha t)}\right]^K - 1\right\}}$$

Mean field analysis (VI)

Existence of a critical value of the ratio clause/variable

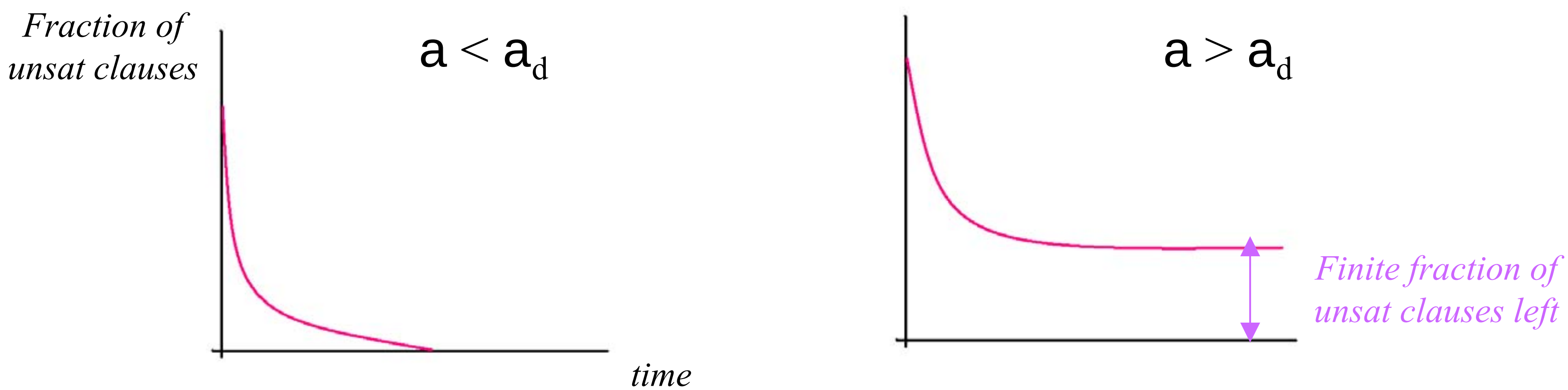
$$\boxed{\alpha_d(K) = \frac{2^K - 1}{K}}$$

- If $\alpha < \alpha_d$, then the procedure stops after a finite time with no unsat clause left.
- If $\alpha > \alpha_d$, then the procedure never stops, and a finite fraction of clauses remain unsatisfied.

$$\begin{aligned}
\phi_0^* &= \frac{1}{2^K} - \frac{1}{\alpha K}\left(1 - \frac{1}{2^K}\right) = \frac{1}{2^K}\left(1 - \frac{\alpha_d(K)}{\alpha}\right) \\
\phi_j^* &= \left(1 + \frac{1}{\alpha K}\right)\frac{1}{2^K}\binom{K}{j} \qquad .
\end{aligned}$$

Comments on the analysis

• Numerical simulations confirm scenario is qualitatively correct …

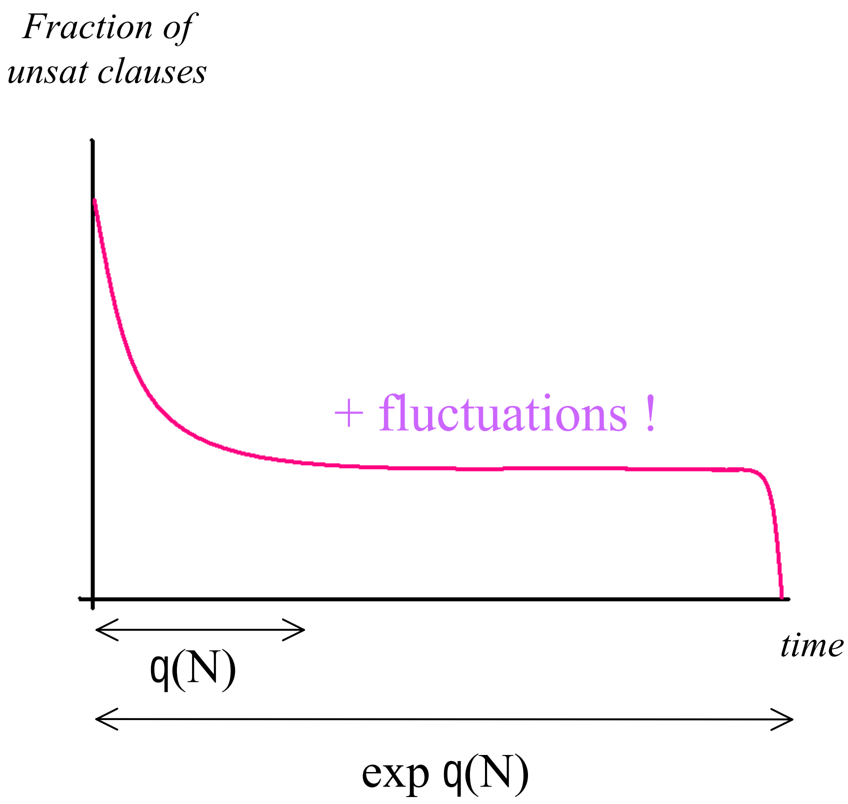


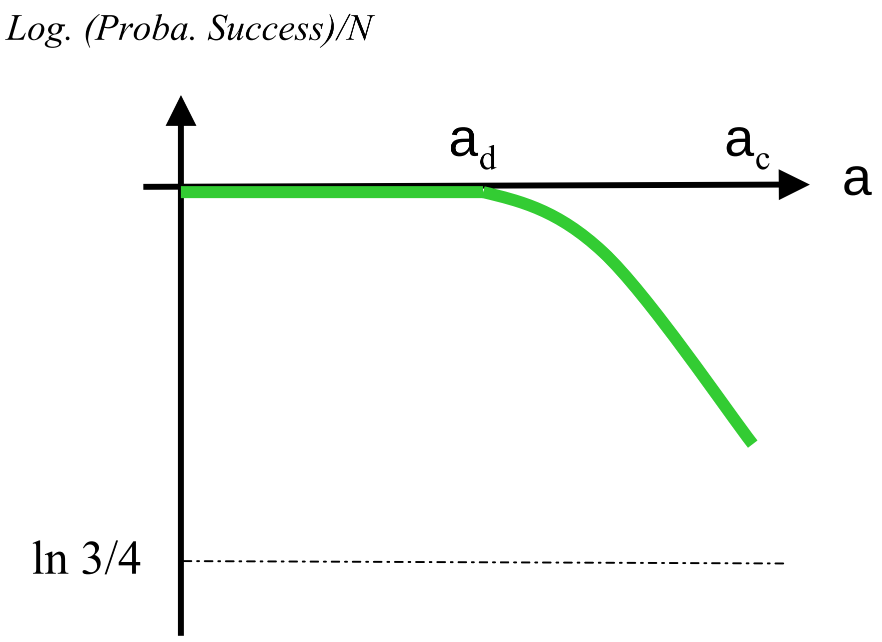
… but numerical estimates for a_d are larger than theoretical predictions.

• Quantitative agreement seems to improve for larger K or a in agreement with theoretical argument by G. Semerjian (2001).

• Behavior at very large times (exponential in N) for $a > a_d$ …

Rare deviations





Also work by
W. Barthel and M. Weigt

Bibliography

1. Local search algorithm for SAT:
   B. Selman, H. Kautz, B. Cohen, Noise strategies for improving local search, Proceedings AAAI-94, 337-343 (1994).
   J. Gu, P.W. Purdom, J. Franco, B. Wah. Algorithms for the satisfiability problem: a survey. DIMACS Series on Discrete Mathematics and Theoretical Computer Science 35:19-151 (1997).

2. Exacts results for 2-SAT:
   C. Papadimitriou, On selecting a truth assignment, FOCS'91, 163-169 (1991).

3. Exact results for 3-SAT:
   U. Schöning, A probabilistic algorithm for k-SAT and constraint satisfaction problems, FOCS'99, 410-414 (1999)
   U. Schöning, New worst case bounds on K-SAT, SAT2002, Cincinnati (2002); available from http://gauss.ececs.uc.edu/Conferences/SAT2002/sat2002list.html

Conclusion

Few examples of solving algorithms which can be studied using statistical mechanics tools
= *more or less legitimate extension of mathematical studies*

- Average properties of algorithms for random distribution of instances.

- Can we obtain results true for any instance?