Generating Hard But Solvable SAT Formulas

André Schumacher

T-79.7003 Research Course in Theoretical Computer Science

Laboratory for Theoretical Computer Science,

Helsinki University of Technology

Andre.Schumacher-''at''-tkk.fi

Outline

① The 3-SAT Problem

② Instance Generation with Hidden Satisfying Assignment

- Random Clause Generation (Uniform)
- Hidden Complementary Assignment
- 3-SAT Spin-Glass Model
- 3-XOR-SAT Spin-Glass Model

③ Conclusions

The 3-SAT Problem

➜ Given formula F over set of N boolean variables (in CNF)

$$\{x_i | i = 1, \ldots, N\}$$

➜ F consists of conjunction of M logical clauses

$$F = C_1 \wedge C_2 \wedge \ldots \wedge C_M, \quad C = \{C_\mu | \mu = 1, \ldots, M\}, \quad \alpha = M/N$$

➜ Each clause is disjunction of 3 *literals*

$$C_\mu = (l_\mu^1 \vee l_\mu^2 \vee l_\mu^3), \quad l_\mu^i = x_k, \bar{x}_k$$

➜ Exists assignment $x_i \mapsto \{\mathrm{true}, \mathrm{false}\}$ that satisfies F? (NP-complete)

➜ Complete (zChaff, Satz) vs. incomplete (WalkSAT, RRT, SP) solvers

➜ Problem: Generate test instances for solvers:
a) hard b) have known truth assignment c) easy and fast to generate

➜ Consider other applications: e.g. cryptography (one-way functions)

Instance Generation

- **First candidate:**

  Pick truth assignment A uniformly at random;
  foreach *clause* C_μ **do**
  pick indices i, j, k uniformly at random;
  foreach i, j, k **do**
  flip coin: negated or not;
  if A *evaluates* C_μ *to false* **then** discard C_μ;
  else accept;

- Potential problem: (hidden $A = \{x_1 \leftarrow \text{true}, x_2 \leftarrow \text{true}, x_3 \leftarrow \text{true}\}$)

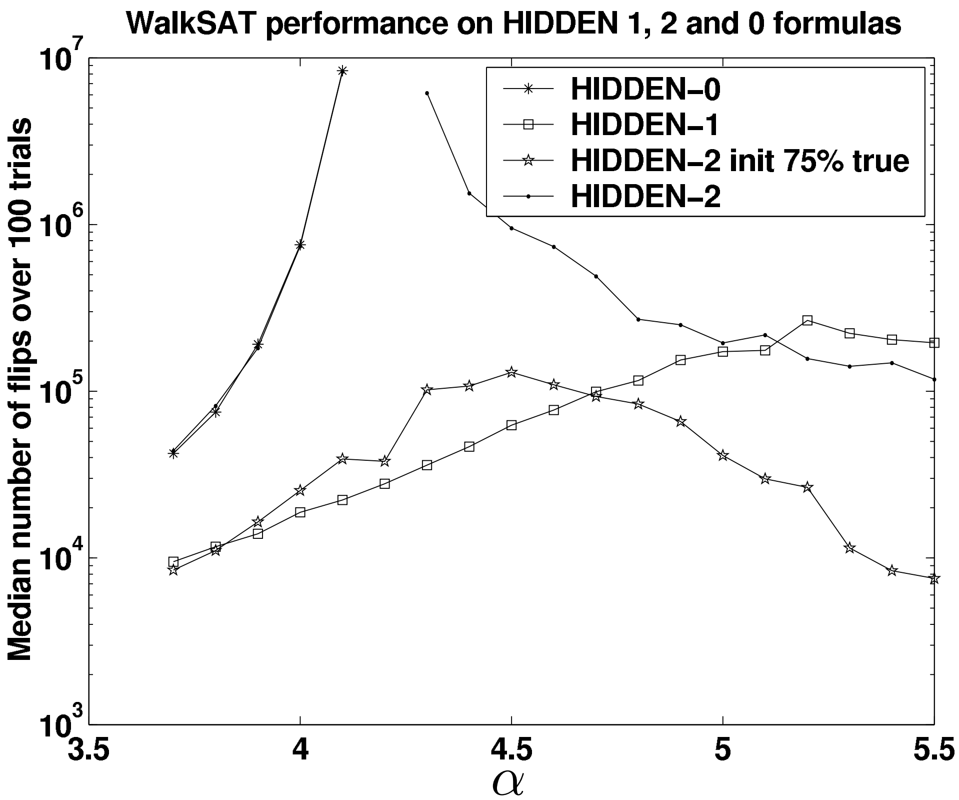
  $(x_1 \vee x_2 \vee x_3)$ $(x_1 \vee x_2 \vee \bar{x}_3)$ $(x_1 \vee \bar{x}_2 \vee x_3)$ $(x_1 \vee \bar{x}_2 \vee \bar{x}_3)$

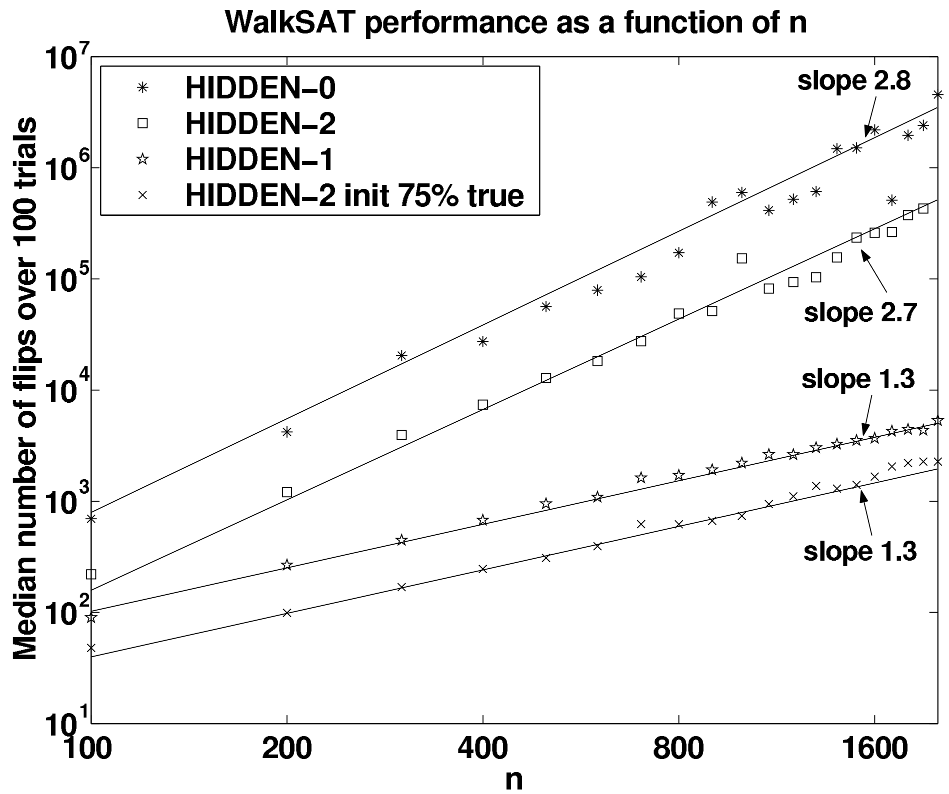
  $(\bar{x}_1 \vee x_2 \vee x_3)$ $(\bar{x}_1 \vee x_2 \vee \bar{x}_3)$ $(\bar{x}_1 \vee \bar{x}_2 \vee x_3)$ ~~$(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$~~

- Picking clauses uniformly at random introduces drift towards the correct assignment $\rightarrow$ formulas become “easier” (in some sense)

- Idea: Hide A and complementary $\bar{A}$ [Achlioptas, Jia, Moore, J. Artif. Intell. Res. (2005)]

RESULTS FOR WALKSAT





Plot for $\alpha = 4.25$

Spin-Glass Model for 3-SAT

➜ Map boolean $x_i = 0, 1$ to Ising spins $s_i = (-1)^{1-x_i} = -1, +1$; denote:

$$c_\mu^{(i)} = +1 \text{ if } x_i \text{ directly in } C_\mu, \qquad c_\mu^{(i)} = -1 \text{ if } x_i \text{ negated in } C_\mu$$

➜ Hamiltonian H defined to count unsatisfied clauses:

$$H(s) = C - \sum_{i=1}^{N} H_i s_i - \sum_{i<j} T_{ij} s_i s_j - \sum_{i<j<k} J_{ijk} s_i s_j s_k$$

➜ One obtains (collecting constant and first-order terms):

$$H(s) = \sum_{\mu=1}^{M} \frac{1}{8} \prod_{i=1}^{N} (1 - c_\mu^{(i)} s_i), \quad C = \frac{M}{8} = \frac{\alpha}{8} N, \quad H_i = \frac{1}{8} \sum_{\mu=1}^{M} c_\mu^{(i)}$$

➜ Collecting higher-order terms:

$$T_{ij} = -\frac{1}{8} \sum_{\mu} c_\mu^{(i)} c_\mu^{(j)}, \quad J_{ijk} = \frac{1}{8} \sum_{\mu} c_\mu^{(i)} c_\mu^{(j)} c_\mu^{(k)}$$

SPIN-GLASS MODEL FOR 3-SAT CONT.

- Idea: Generate clauses according to probability distribution, observe resulting spin-glass model (phase transition...) [Barthel et al., Phys. Rev. Lett. (2002)]

- Clause probabilities p_i s.t. $p_0 + 3p_1 + 3p_2 = 1$ (hiding $x_i = 1\ \forall i$):

$$
\begin{array}{llll}
p_0: & & (x_i \vee x_j \vee x_k) & \\
p_1: & (x_i \vee x_j \vee \bar{x}_k) & (x_i \vee \bar{x}_j \vee x_k) & (\bar{x}_i \vee x_j \vee x_k) \\
p_2: & (x_i \vee \bar{x}_j \vee \bar{x}_k) & (\bar{x}_i \vee x_j \vee \bar{x}_k) & (\bar{x}_i \vee \bar{x}_j \vee x_k)
\end{array}
$$

- Averages resulting for spin-glass model:

$$\Pr(x_i \text{ or } \bar{x}_i \text{ appears in clause } \mu) = \frac{\binom{N-1}{2}}{\binom{N}{3}} = \frac{3}{N}$$

$$\overline{H_i} = \frac{1}{8} M\ E(c_\mu^{(i)}) = \frac{3\alpha}{8}\left(p_0 \cdot 1 + 3p_1 \cdot \frac{2-1}{3} + 3p_2 \cdot \frac{1-2}{3}\right) = \frac{3\alpha}{8}\,(p_0 + p_1 - p_2)$$

- Similarly: $\overline{T_{ij}} = \frac{3\alpha}{4N}(-p_0 + p_1 + p_2)$, $\overline{J_{ijk}} = \frac{3\alpha}{4N^2}(p_0 - 3p_1 + 3p_2)$

EFFECT OF VARIOUS CHOICES FOR p_i

① Uniformly at random, rejecting clause $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$

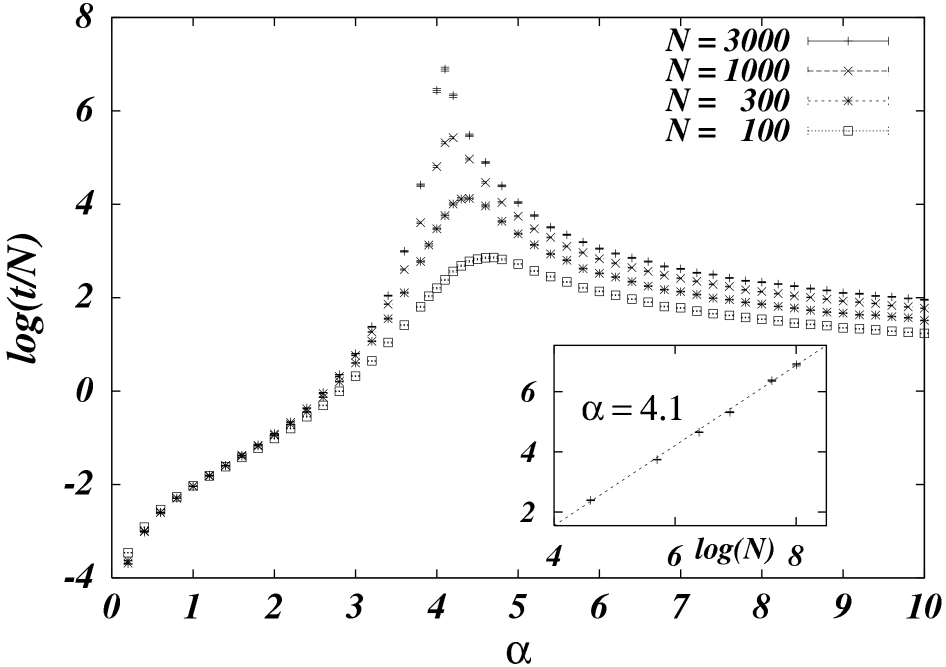
- ➜ $x_i = 1 \; \forall i$ is satisfying
- ➜ Case $p_0 = p_1 = p_2 = 1/7 \Rightarrow \overline{H_i} = \frac{3\alpha}{56}$
- ➜ Solvers (e.g. local solvers s.a. WalkSAT) guided by local field
- ➜ Therefore: Set $\overline{H_i} = \frac{3\alpha}{8}(p_0 + p_1 - p_2) = 0$
- ➜ Resulting restrictions for probabilities p_i:

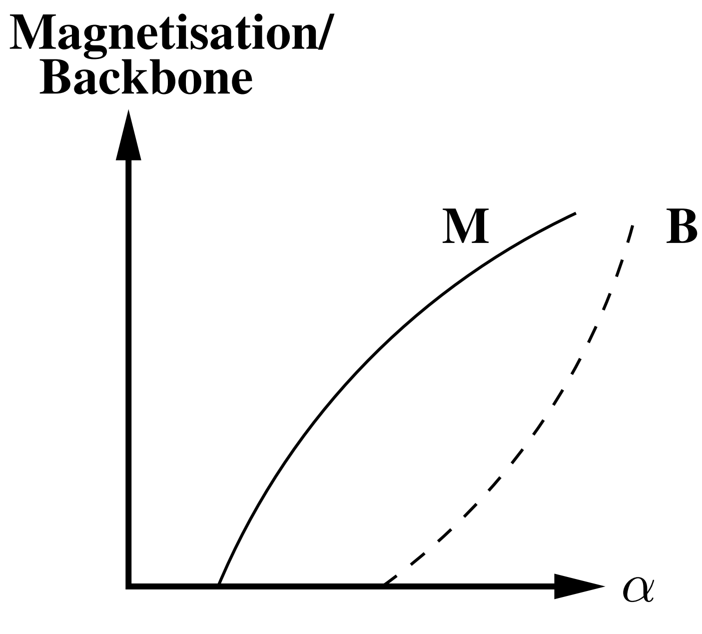
$$0 \leq p_0 \leq \frac{1}{4}, \quad p_1 = \frac{1 - 4p_0}{6}, \quad p_2 = \frac{1 + 2p_0}{6}$$

Effect of various choices for p_i (cont.)

② Uniformly at random, rejecting clauses $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$, $(x_i \vee x_j \vee x_k)$

- Both, $x_i = 1\ \forall i$ and $x_i = 0\ \forall i$, are satisfying
- Case $p_0 = 0, p_1 = p_2 = 1/6 \Rightarrow \overline{J_{ijk}} = \frac{3\alpha}{4N^2}(p_0 - 3p_1 + 3p_2) = 0$
- Seems more difficult, but WalkSAT shows avg. runtime $O(N^c)$





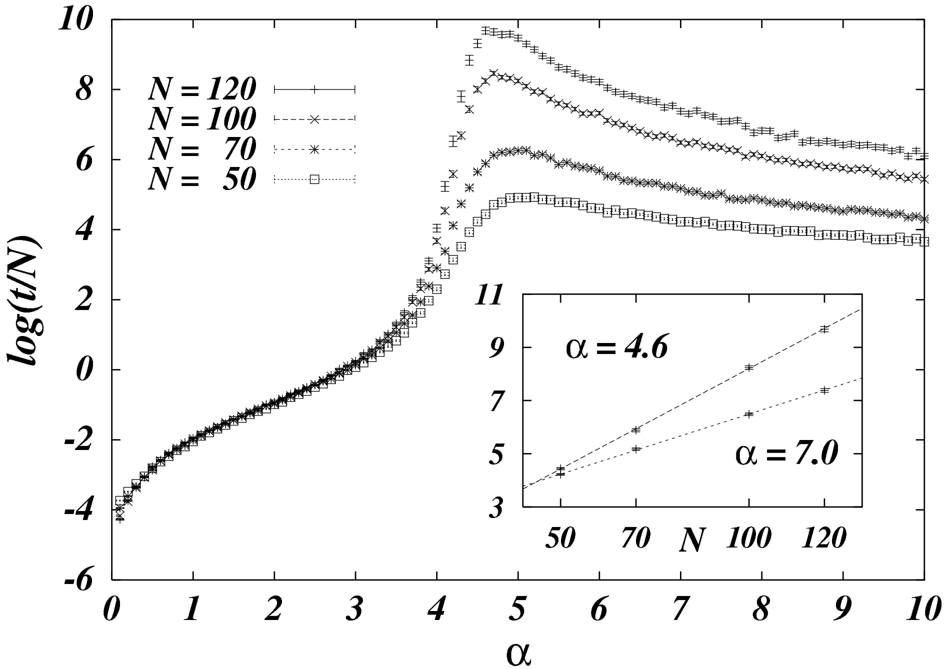
EFFECT OF VARIOUS CHOICES FOR p_i (CONT.)

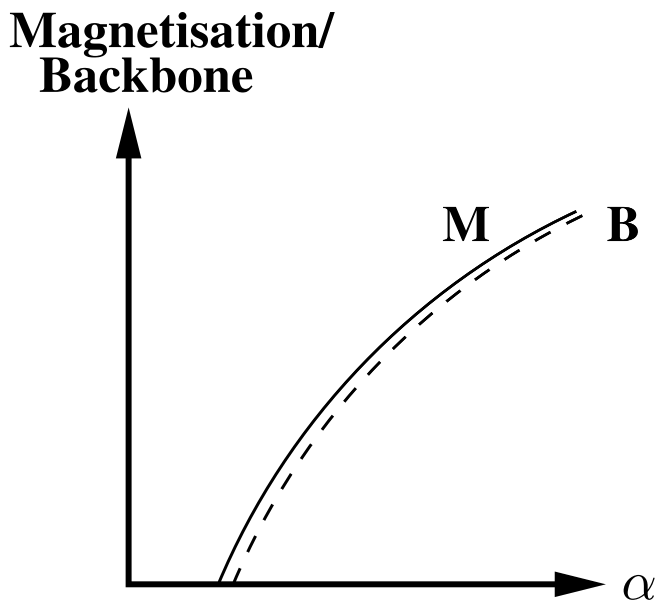
③ Random satisfiable 3-XOR-SAT

➜ Case $p_0 = p_2 = 1/4, p_1 = 0$

➜ Can be solved in $O(N^c)$ time (Gauss), but difficult for solvers

➜ WalkSAT shows exponential running time close to phase transition



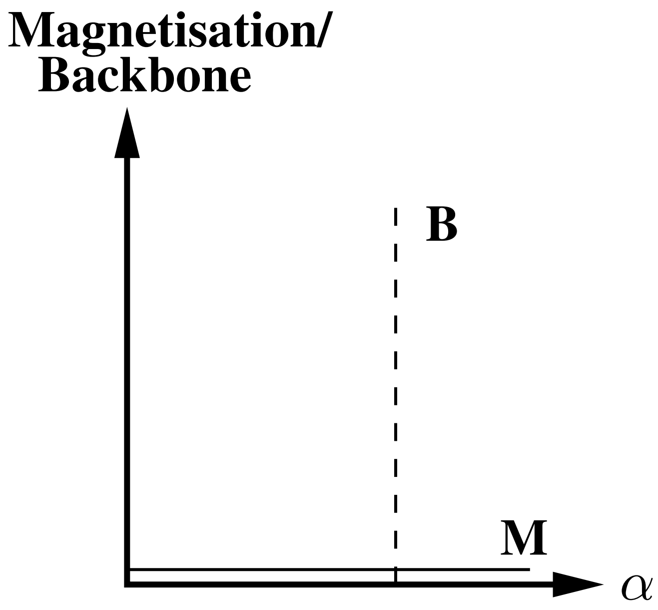


Effect of various choices for p_i (cont.)

④ Range $0.077 \lesssim p_0 < 1/4$

➜ Discontinuous appearance of backbone

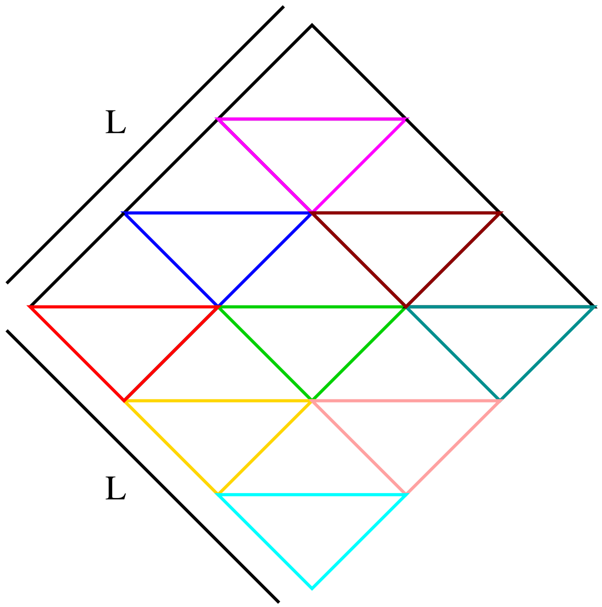
➜ Leads to hard instances

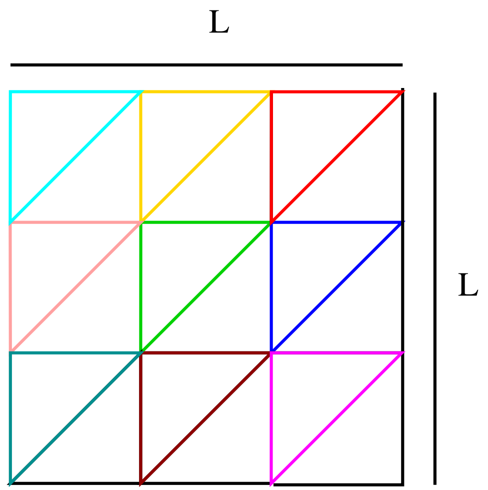


3-XOR-SAT

- Triangular lattice spin-glass model [Jia, Moore, Selman, Th. Appl. o. Sat. Testing (2005)]
- Nearest neighbour interaction and short loops $\rightarrow$ glassy
- Lattice: $L \times L$ rhombus with periodic boundary conditions
- Hamiltonian

$$H = \frac{1}{2} \sum_{i,j=0}^{L-1} s_{i,j} \cdot s_{i,j+1 \bmod L} \cdot s_{i+1 \bmod L, j}$$





3-XOR-SAT CONT.

➜ Rewriting in terms of booleans $x_{i,j} = \frac{1}{2}(s_{i,j} + 1)$ then (up to a constant)

$$H = \sum_{i,j=0}^{L-1} \left(\left(x_{i,j} + x_{i,j+1 \bmod L} + x_{i+1 \bmod L,j} \right) \bmod 2 \right)$$

➜ Corresponds to L^2 3-XOR-SAT clauses of the form

$$\overline{x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j}}$$
$$\equiv \left(\bar{x}_{i,j} \vee x_{i,j+1 \bmod L} \vee x_{i+1 \bmod L,j}\right) \wedge \left(x_{i,j} \vee \bar{x}_{i,j+1 \bmod L} \vee x_{i+1 \bmod L,j}\right) \wedge$$
$$\left(x_{i,j} \vee x_{i,j+1 \bmod L} \vee \bar{x}_{i+1 \bmod L,j}\right) \wedge \left(\bar{x}_{i,j} \vee \bar{x}_{i,j+1 \bmod L} \vee \bar{x}_{i+1 \bmod L,j}\right)$$

➜ 3-SAT formula, L^2 variables, $4L^2$ clauses, H counts unsatisfied clauses

➜ Satisfying: $x_{i,j} = 0 \; \forall i, j$ (unique if $L = 2^k$ [Newman, Moore, Phys. Rev. Lett., (1999)])

➜ Hidden assignment is arbitrary: define $y_{i,j} = x_{i,j} \oplus a_{i,j}$ (can be found e.g. by Gauss elimination mod 2)

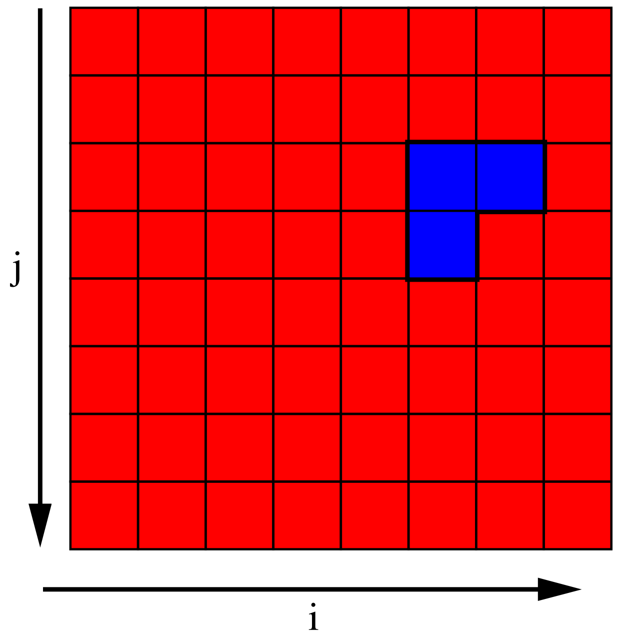
HARDNESS OF FORMULAS FOR SAT SOLVERS

- Satisfying assignment implies recurrence equation

  $$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

  $\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

- ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



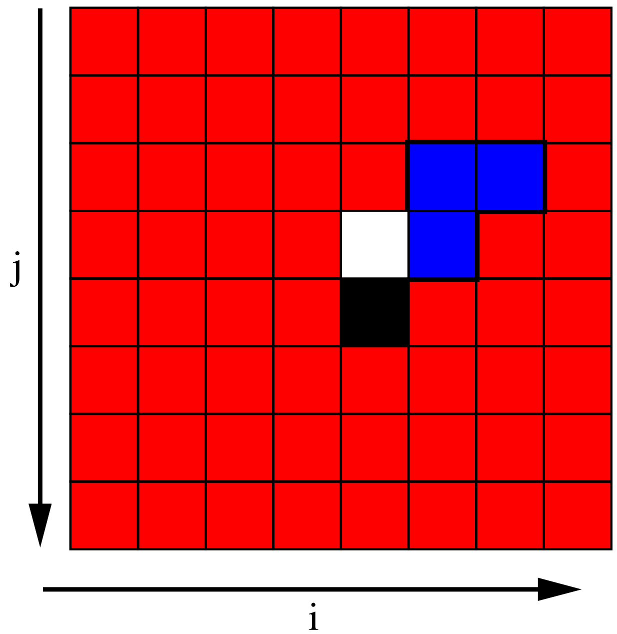
HARDNESS OF FORMULAS FOR SAT SOLVERS

➜ Satisfying assignment implies recurrence equation

$$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

$\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

➜ ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



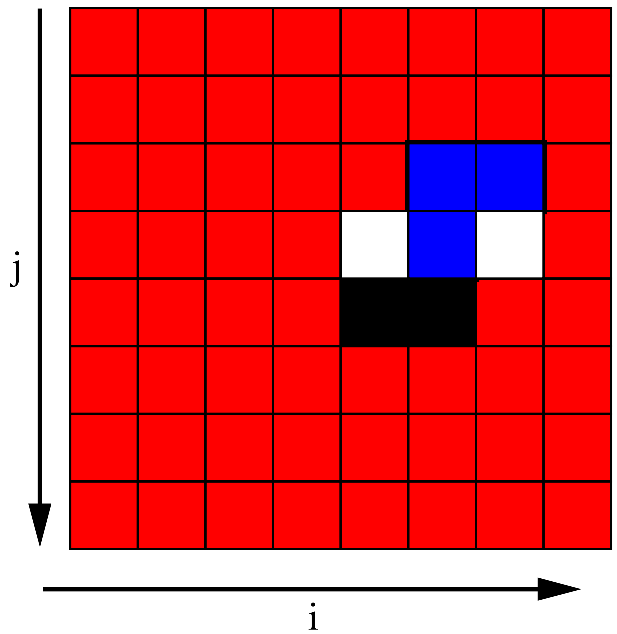
HARDNESS OF FORMULAS FOR SAT SOLVERS

➜ Satisfying assignment implies recurrence equation

$$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

$\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

➜ ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



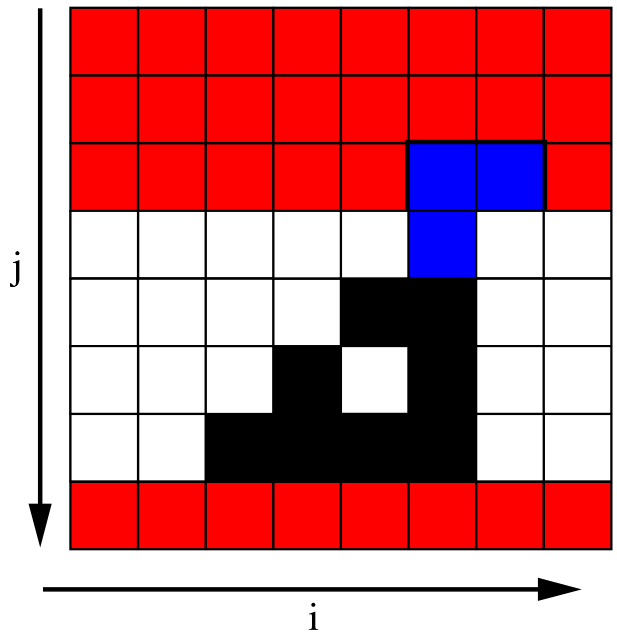
HARDNESS OF FORMULAS FOR SAT SOLVERS

- Satisfying assignment implies recurrence equation

  $$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

  $\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

- ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



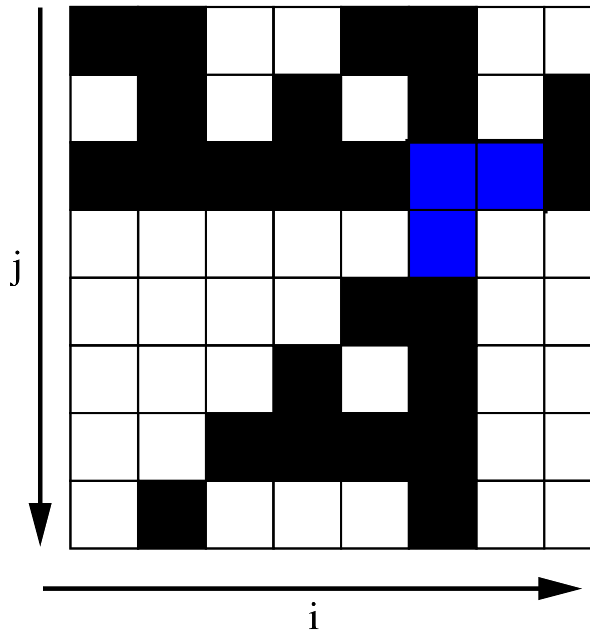
HARDNESS OF FORMULAS FOR SAT SOLVERS

- Satisfying assignment implies recurrence equation

  $x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$

  $\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

- ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



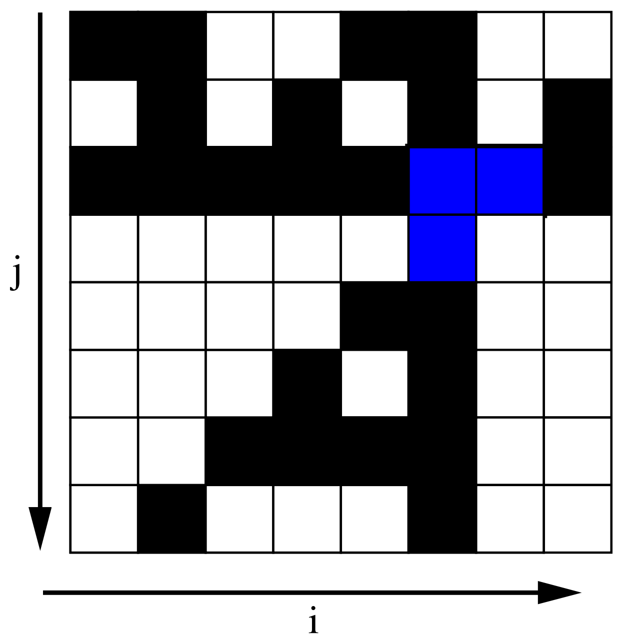
Hardness of Formulas for SAT Solvers

- ➜ Satisfying assignment implies recurrence equation

$$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

$\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

- ➜ ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



Analytical calculations show

- ➜ Hamming distance to sat. assignment: $L^{\log_2 3}$ (#ones)
- ➜ Energy barrier towards sat. assignment: $O(\log_2 L)$
- ➜ #local minima: $O(\kappa^{L^2})$, $\kappa \approx 1.395$ (*hard hexagon constant*)

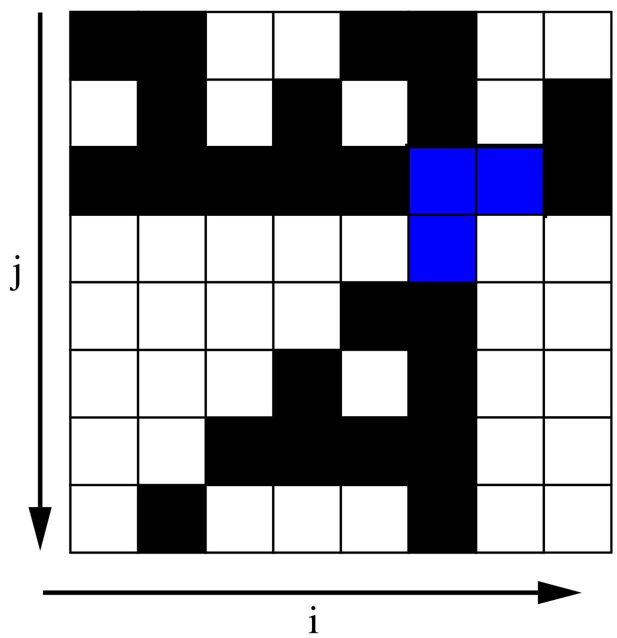
HARDNESS OF FORMULAS FOR SAT SOLVERS

➜ Satisfying assignment implies recurrence equation

$$x_{i,j} \oplus x_{i,j+1 \bmod L} \oplus x_{i+1 \bmod L,j} \equiv 0 \Rightarrow x_{i,j+1 \bmod L} = x_{i,j} \oplus x_{i+1 \bmod L,j}$$

$\Rightarrow$ Truth values are given by Pascal's triangle mod 2!

➜ ■: undetermined, ■: fixed to 1 (defect), ■: 1, □: 0



Analytical calculations show

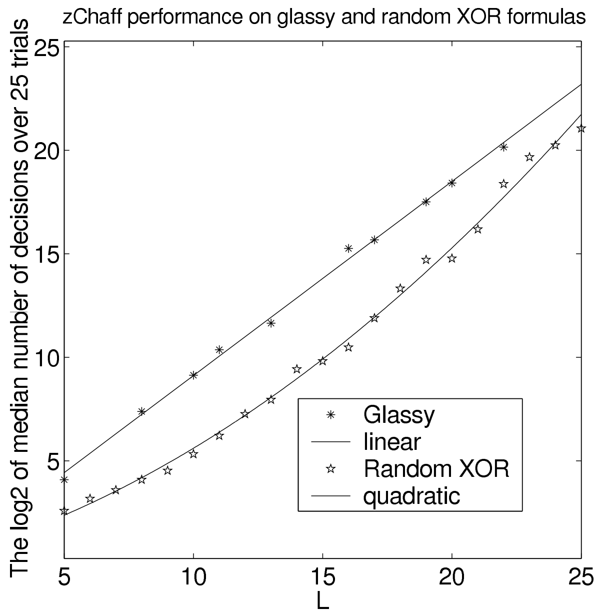
➜ Hamming distance to sat. assignment: $L^{\log_2 3}$ (#ones)

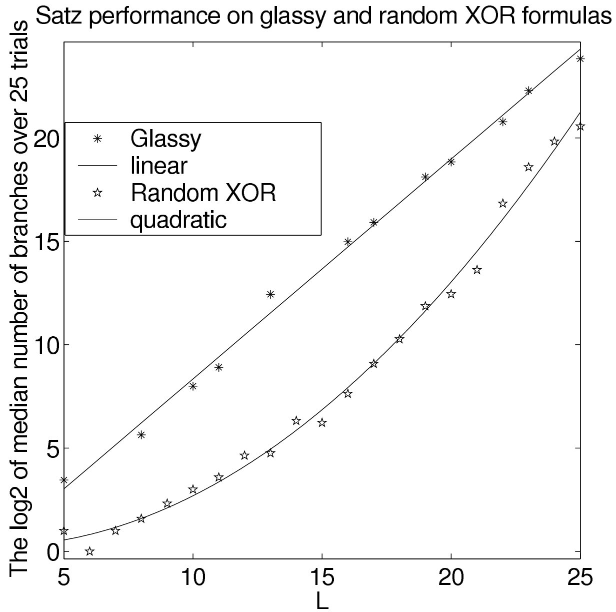
➜ Energy barrier towards sat. assignment: $O(\log_2 L)$

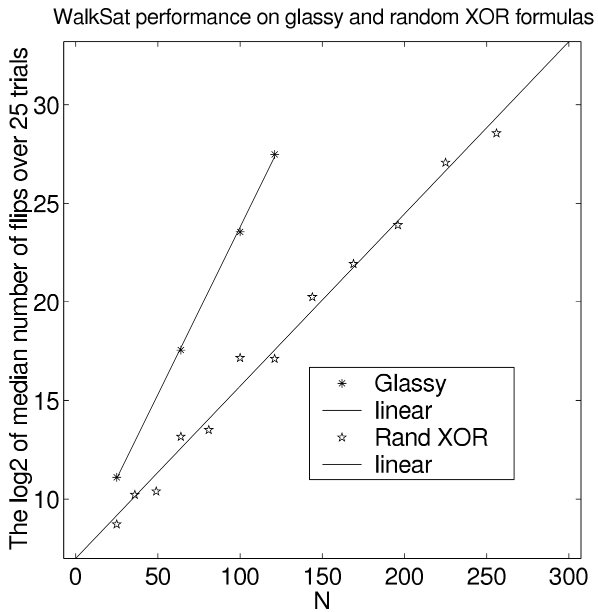
➜ #local minima: $O(\kappa^{L^2})$, $\kappa \approx 1.395$ (*hard hexagon constant*)

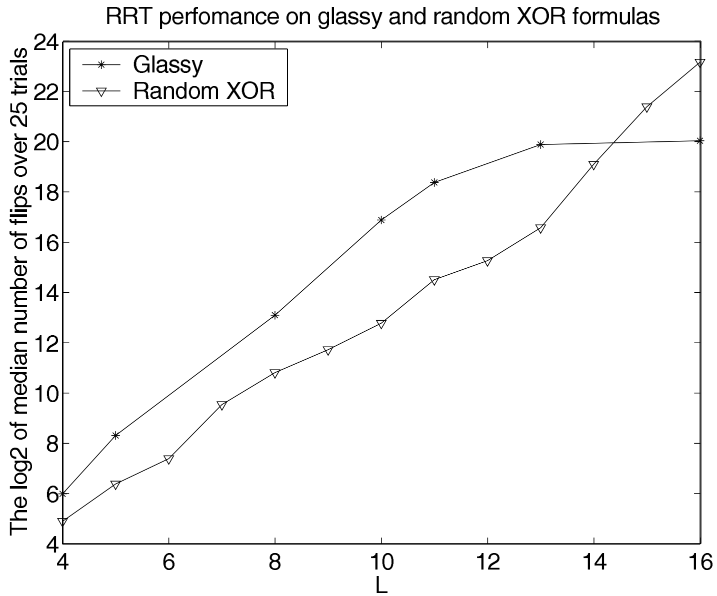
$\Rightarrow$ hard for local algorithms!

EXPERIMENTAL RESULTS [1]









Conclusions

- ➜ Generation of 3-SAT instances important for evaluation of solvers
- ➜ Not all “random” generation schemes result in hard formulas
- ➜ Parameterised generation of instances can reveal interesting aspects of problem structure
- ➜ Instances can be difficult for local solvers even though efficiently solvable by other means (XOR-SAT)
- ➜ Wide range of other generation methods, e.g. regular XOR-SAT based on 3-regular constraint graphs [Haanpää, Järvisalo, Kaski, and Niemelä, Journal on Satisfiability, Boolean Modeling and Computation (2006)]

COMMENTS, QUESTIONS?

Thank you for your attention.

References

[1] Jia, H., Moore, C., Selman, B.: From spin glasses to hard satisfiable formulas. In: Theory and Applications of Satisfiability Testing. Volume 3542/2005., Springer Berlin / Heidelberg (2005) 199–210

[2] Barthel, W., Hartmann, A.K., Leone, M., Ricci-Tersenghi, F., Weigt, M., Zecchina, R.: Hiding solutions in random satisfiability problems: A statistical mechanics approach. Phys. Rev. Lett. **88**(18) (2002) 188701

[3] Newman, M., Moore, C.: Glassy dynamics in an exactly solvable spin model. Phys. Rev. Lett. **60** (1999) 5068–5072

[4] Achlioptas, D., Jia, H., Moore, C.: Hiding satisfying assignments: Two are better than one. J. Artif. Intell. Res. (JAIR) **24** (2005) 623–639