Advanced Topics in SAT-Solving

Part II: Theoretical Aspects

Carsten Sinz

Wilhelm-Schickard-Institut for Computer Science
University of Tübingen

28.09.2004

SAT is NP-complete

This means (consider SAT $= \{F$ in CNF $\mid F$ satisfiable$\}$):

1. SAT $\in$ NP, i.e. there is a non-deterministic Turing machine which recognizes the set of satisfiable formulae in polynomial time
   Alg.: non-deterministically build a solution vector (assignment); check whether the generated vector is a solution.

2. Every problem $P \in$ NP is polynom. reducible to SAT
   Idea of Cook's proof (1971): show that any such P can be computed by a Boolean circuit (i.e. a computer)

NP-completeness of SAT

- There is no known deterministic polynomial time algorithm to solve SAT (unless P=NP)
- An algorithm solving all SAT instances in polynomial time would imply P=NP
- Although there are SAT instances that require exponential time, subclasses of SAT may be tractable (i.e. solvable in polynomial time)
- 3-SAT is still in NP, even if all variables appear at most 3 times

Tractable Subclasses

Syntactically recognizable: (tractable: decidable in polytime)

- 2-SAT (only Krom-clauses): each clause contains at most two literals

- Horn-SAT (only Horn clauses): in each clause at most one literal is positive; Horn clauses usually written as implications with negative literals on the left:

  $(\neg x_1 \vee \cdots \vee \neg x_n \vee y)$ is written as $x_1 \wedge \cdots \wedge x_n \Rightarrow y$

- Trivial subclasses: no positive (negative) clauses [positive (negative) clause: all literals positive (negative)]; ...

Complexity: Upper Bounds

Deterministic 3-SAT algorithms:

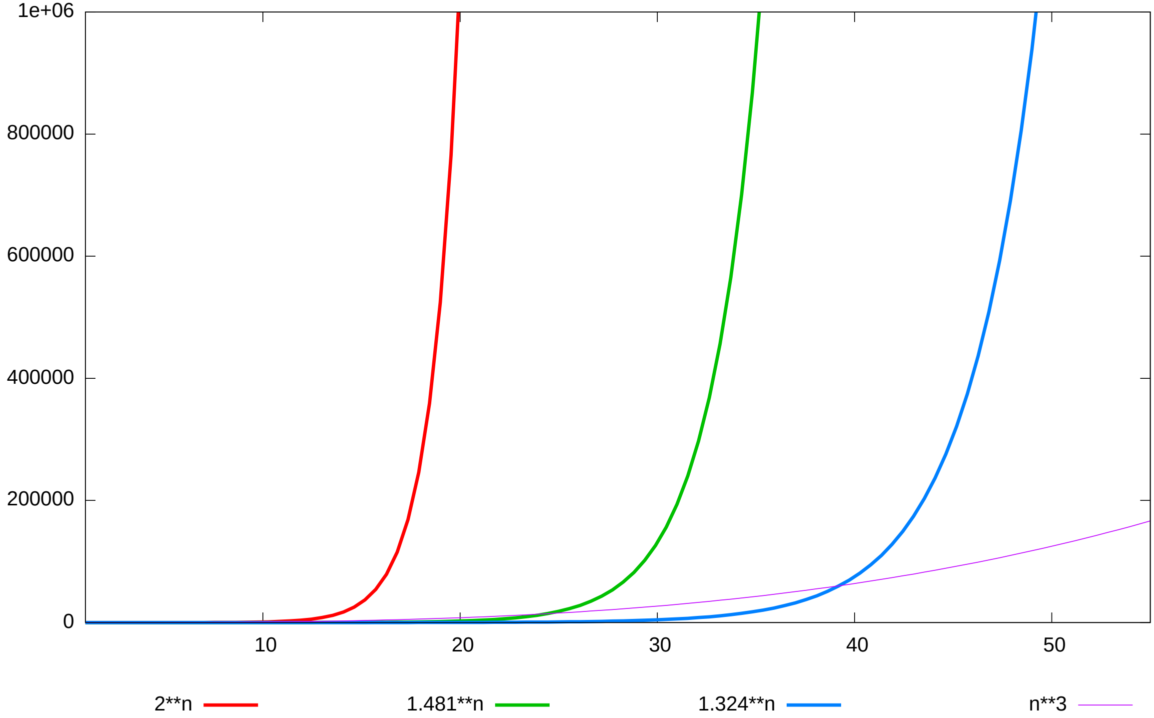
2^n	truth table method
1.618^n	Monien, Speckenmeyer (1985)
1.505^n	Kullmann (1999)
1.481^n	Dantsin, Goerdt, Hirsch, Schöning (2000)

Probabilistic 3-SAT algorithms:

1.587^n	Paturi, Pudlák, Zane (1997)
1.330^n	Hofmeister, Schöning, Schuler, Watanabe (2001)
1.329^n	Baumer, Schuler (2002)
1.324^n	Iwama, Tamaki (2003)

(from Schöning: Algorithms for the SAT Problem)

Upper Bounds: Runtime Comparison



Lower Bounds

Why are lower bounds of interest?
'Optimality' / limits of algorithms; P-NP problem

(Lower bounds are generally harder to obtain than upper bounds)

Results depend on algorithm: e.g. Resolution needs at least $f(n)$ steps to solve problem $X(n)$

Haken's result (1985): Any resolution proof of PHP$^n_{n-1}$ is of length $2^{\Omega(n)}$ (for sufficiently large n)

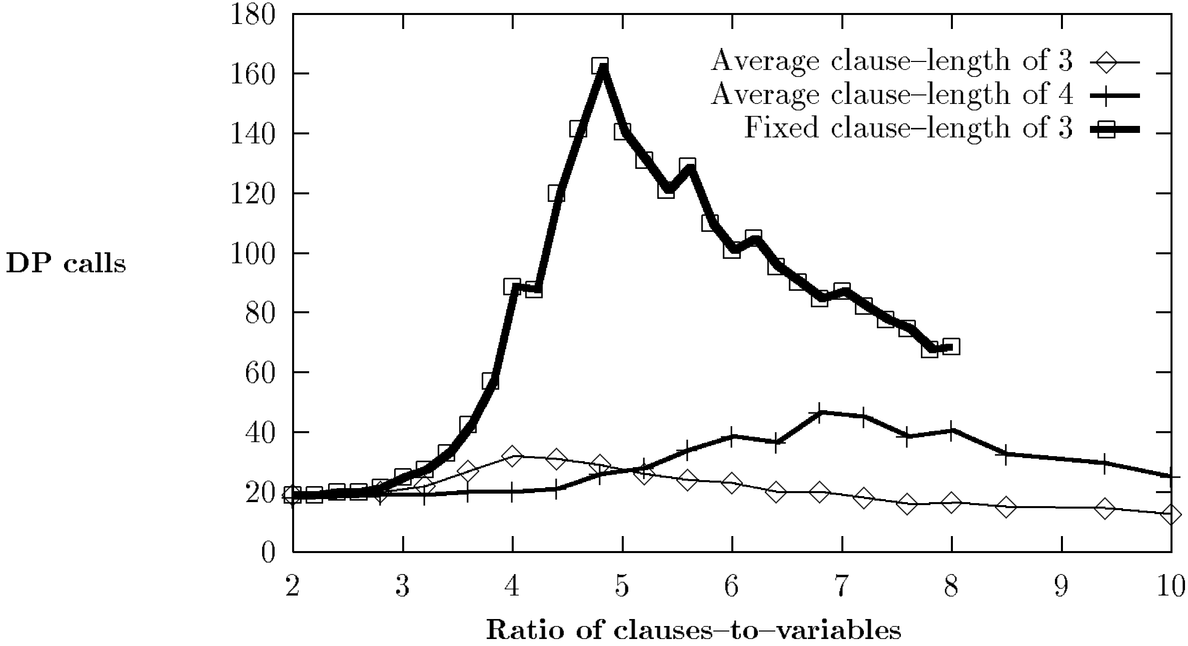
(PHP$^n_{n-1}$: pigeon hole formula :– n pigeons cannot sit in $n-1$ holes)

Random SAT Problems

Variable clause size model: To generate a clause C, scan through all variables and add variable x with fixed probability p to the clause; negate variable with probability $1/2$.
$\rightsquigarrow$ Shown to be solvable in polynomial average time.

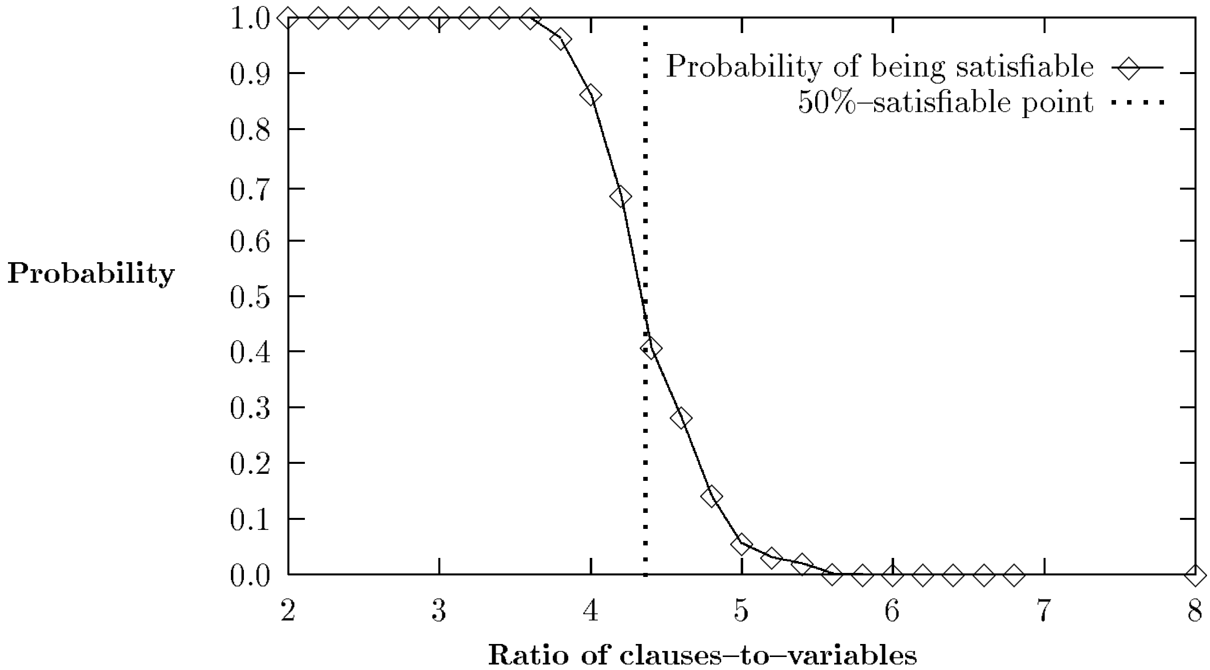
Fixed clause size model: To generate a k-clause, select k variables uniformly at random from V, negate each variable with probability $1/2$.
$\rightsquigarrow$ Produces hard problems (dependent on clause/variable ratio.

Comparison: Fixed vs. Variable Clause Size

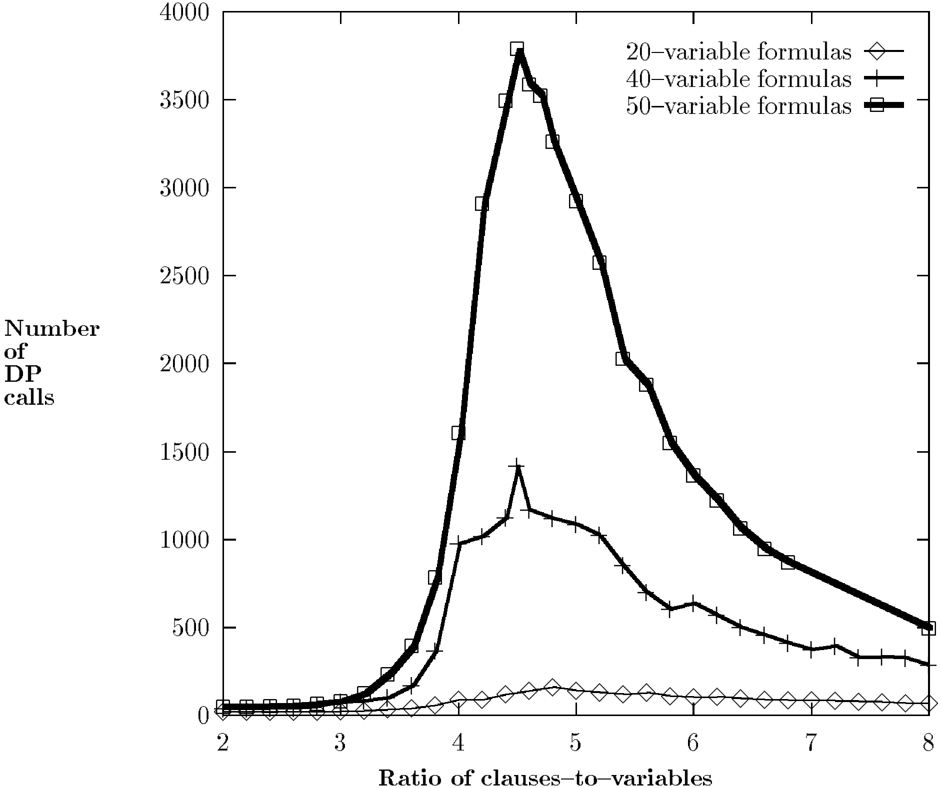


(Figures on this and next slides from: Mitchell, Selman, Levesque (1992))

Random 3-SAT: Threshold Phenomena I



Random 3-SAT: Threshold Phenomena II



Random k-SAT

1. Hardest problems are at phase transition point (transition from satisfiable to unsatisfiable)

2. Easy-hard-easy pattern at increasing clause/var.-ratio
   Intuition:
   At low ratios: few clauses (constraints), many satisfying assignments
   At high ratios: many constraints, inconsistencies easily detected

3. Phase transition points (Mertens, Mézard, Zecchina 2003):

k	phase transition point (m/n)
3	4.26675 ± 0.00015
4	9.331
5	21.117

$(2+p)$-SAT: Complexity

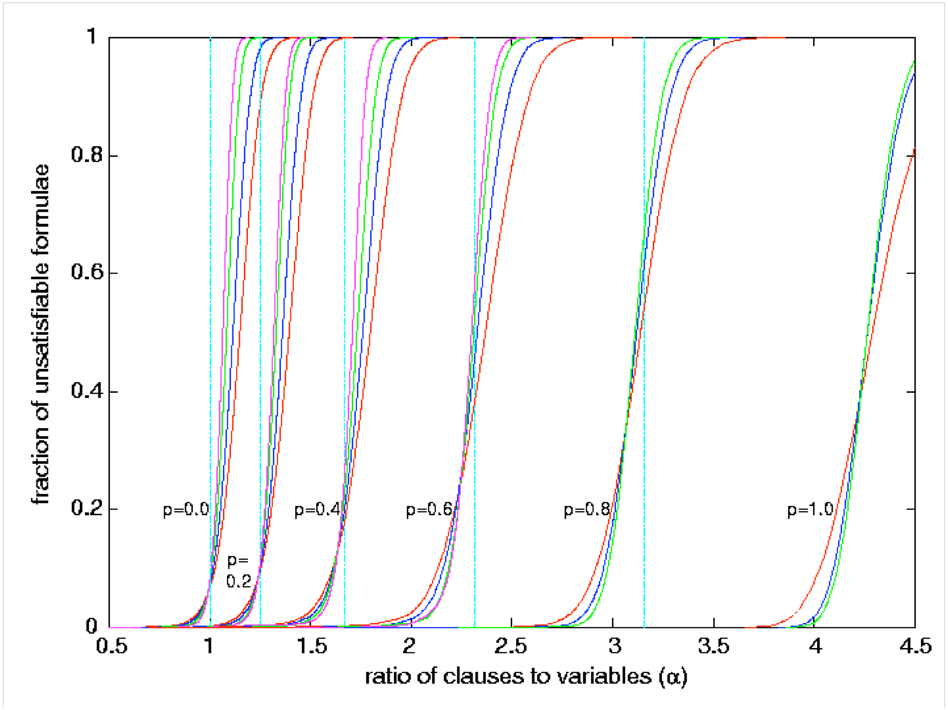
$(2+p)$-SAT: mixture of random 2- and 3-clauses; p: fraction of 3-clauses (p=0: 2-SAT, p=1: 3-SAT)

Motivation: Where is the borderline between P and NP?

Phase transition phenomenon: similar to random k-SAT: low clause/variable ratio $\rightsquigarrow$ almost always satisfiable, high clause/variable ratio $\rightsquigarrow$ almost always unsatisfiable

Experimental result: for $p < \approx 0.41$, $(2+p)$-SAT essentially behaves like 2-SAT (i.e. tractable)

$(2+p)$-SAT: Phase Transition



(Figures on this and next slide from Selman's ISAT'99 talk)

$(2+p)$-SAT: Computational Cost