The Structure of Industrial SAT Instances

Tobias Friedrich Anton Krohmer Ralf Rothenberger Andrew M. Sutton
Hasso-Plattner-Institut, Potsdam

Motivation

Propositional satisfiability (SAT) – fundamental problem in CS.

Instance distributions: study properties of formulas drawn from a distribution.

- Most theoretical work is on the **uniform random model**
  - Easy to construct, easier to analyze
  - But structural properties are different from real-world formulas

Non-uniform distributions have been proposed:

- Random regular
- Geometric
- **Scale-free**

Goal: Utilize parallel computing power of the 1000 node cluster of the Future SOC Lab to:

1. Generate a massive set of large random non-uniform (scale-free) formulas and check their satisfiability & hardness
2. Compare a number of SAT solvers in their ability to determine bounds on the satisfiability threshold

Generating formulas uniformly

k-CNF formula over n Boolean variables $\{x_1, x_2, \dots, x_n\}$,

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m,$$

where each $C_i = (\ell_1 \vee \ell_2, \vee \cdots \vee \ell_k)$,

and each $\ell_j \in \{x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$.

Uniform random distribution $U_{n,m,k}$ – select each formula uniformly at random from the set of all well-formed k-CNF formulas on n variables and m clauses.

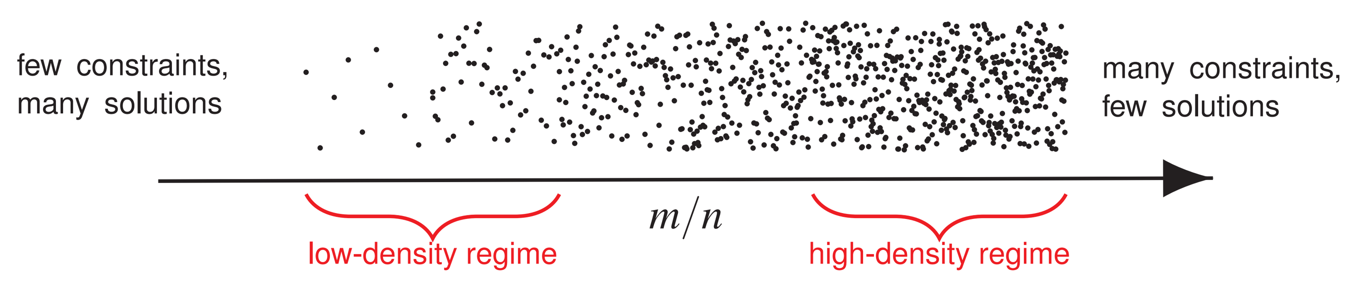
Constraint density and threshold phenomena

- ratio of clauses to variables m/n.
- **phase transition** from satisfiable to unsatisfiable

Satisfiability threshold conjecture: $\exists r_k \in \mathbb{R}$

$$\lim_{n\to\infty} \Pr\{F \text{ is satisfiable} \mid F \sim U_{n,m,k}\} = \begin{cases} 1 & m/n < r_k; \\ 0 & m/n > r_k. \end{cases}$$

- For classical DPLL solvers: **hard problems are near critical point**



Generating formulas non-uniformly

Observation: In many industrial instances the variable distribution follows a **power-law**.

Idea: Construct formulas at random, but with **power-law variable distributions**.

These formulas are called **scale-free** (underlying constraint graph is a scale-free network)

Define: for each $i \in \{1, 2, \dots, n\}$,

$$w_i := \left(\frac{n}{i}\right)^{\frac{1}{\beta-1}}, \qquad p_i := \frac{w_i}{\sum_{j=1}^{n} w_j}.$$

set of n weights → w_i; i-th variable probability → p_i; power-law exponent β

For each clause:

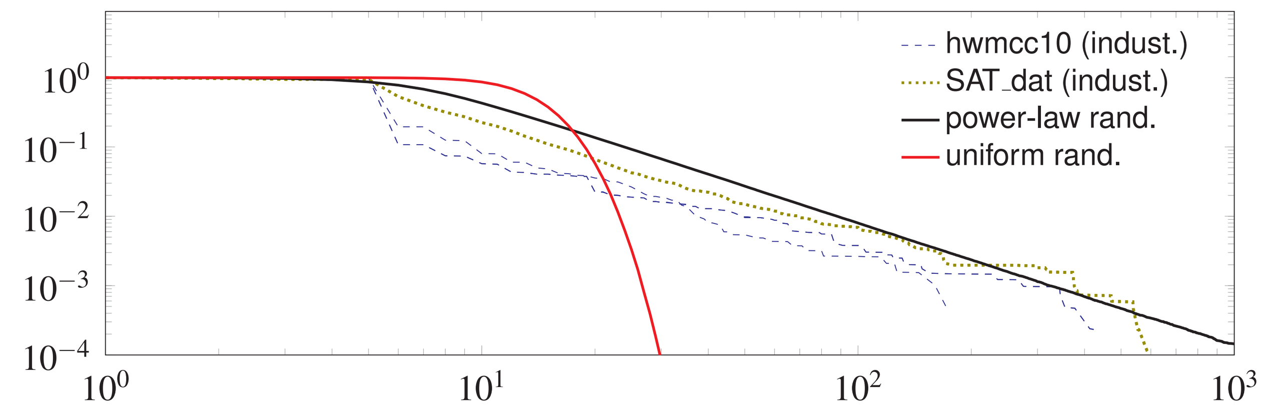
1. Select k variables independently at random from the distribution $\{p_i : i \in [n]\}$. Repeat until no variables coincide.
2. Negate each of the k variables independently at random with probability $1/2$.

Result: a random formula F with a variable occurrence (degree) distribution that follows a power-law with exponent β.

Degree distribution of industrial instances

Industrial instances arise in applications like software/hardware verification, automated planning and scheduling, circuit design.

300 instances from the main track of the SAT Race 2015 competition (`http://baldur.iti.kit.edu/sat-race-2015/`).



Cumulative variable occurrence distributions of two industrial categories from SAT Race 2015 compared to a random scale-free k-CNF formula ($n = 10^6$ variables, $m = 4.5 \times 10^6$ clauses, power law exponent $\beta = 2.75$) and a uniform random formula ($n = 10^6$, $m = 4.5 \times 10^6$).

Comparing SAT solvers

Unlike the uniform random model: the phase transition depends on both constraint density m/n and power-law exponent β

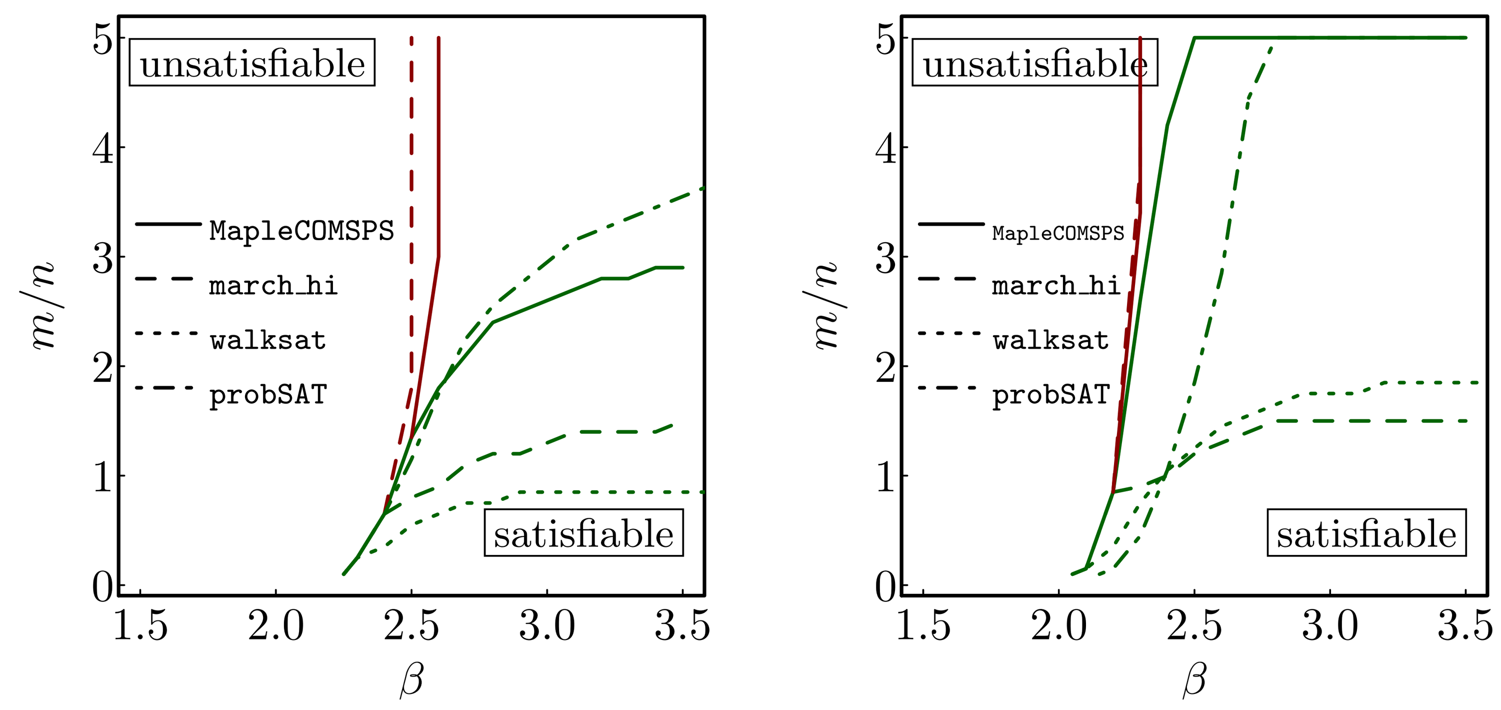
We consider four SAT solvers

- `march_hi` (DPLL solver using look-ahead)
- `MapleCOMSPS` (CDCL solver using machine learning techniques)
- `WalkSAT` (simple Stochastic Local Search)
- `ProbSAT` (probabilistic SLS algorithm)

To eliminate statistical fluctuations want n very large ($n = 10^6$).
Want a very high resolution to get a clear picture → many formulas

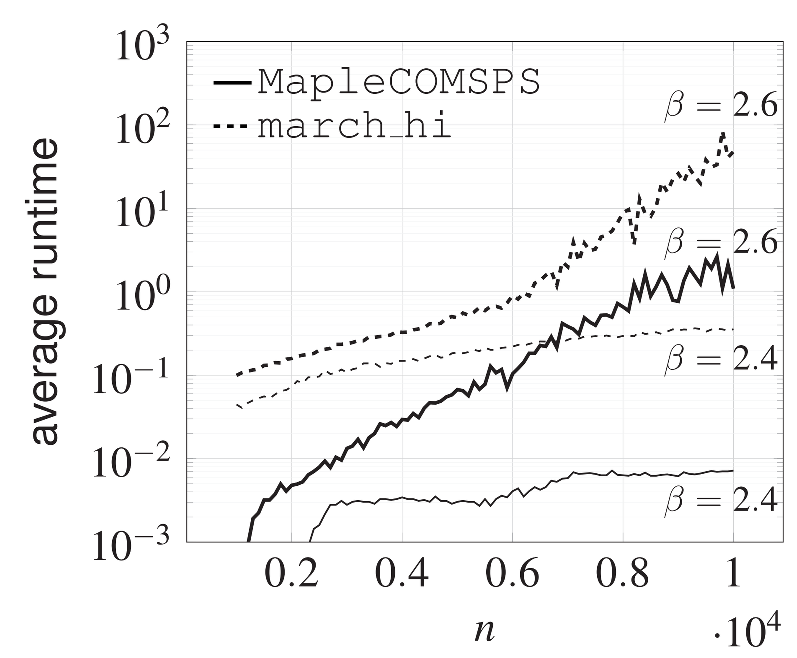
- Distribute many jobs over the cluster (using GNU Parallel).
- For each job:
  1. Generate set of random scale-free formulas
  2. Run each solver on each formula for at most 15 min (900s)
  3. If SAT cannot be decided within limit, mark formula **hard**.
- We generate 50 formulas for each power-law exponent $\beta = 1.5, 1.6, \dots, 3.5$ and each m such that $m/n = 1/10, \dots, 10$.
- Each solver provides upper/lower bounds on threshold

Empirical bounds on the phase transition



Comparison of threshold bounds proposed by four different solvers. As a function of β: the upper bound on density for the unsatisfiable phase is drawn in red; lower bound on density for satisfiable phase drawn in green. Results for $k = 3$ (left) and $k = 4$ (right)

Runtime scaling at the critical point



Scaling behavior of `MapleCOMSPS` and `march_hi` at fixed density $m/n = 2.28$ and $k = 3$ at critical point ($\beta = 2.6$) and below critical point ($\beta = 2.4$). Both solvers scale exponentially at the critical point. Slightly below the critical point, both solvers scale more efficiently with problem size.