The Hard Problems Are Almost Everywhere For Random CNF-XOR Formulas

Jeffrey M. Dudek, Kuldeep S. Meel, & Moshe Y. Vardi

Rice University

Motivation

- Approximate Model Counting/Sampling: Given a CNF formula *F*,
  - *Counting*: Estimate the number of solutions of *F*.
  - *Sampling*: Near-uniformly sample a solution of *F*.
  - Applications in probabilistic inference, automated reasoning, …
- Hashing-based sampling and counting algorithms reduce these problems to finding solutions to CNF-XOR Formulas (Boolean formulas with both CNF and XOR clauses).

  [Gomes *et al.* 2007], [Chakraborty *et al.*, 2013], [Ermon *et al.* 2013]
- **Goal**: Analyze the "behavior" of CNF-XOR formulas.
  - Which CNF-XOR Formulas are "hard" to solve (find a solution)?
  - **Empirical observation**: solving CNF-XOR formulas are hard in practice!

Prior Work: Which CNF Formulas are Hard?

A $\boldsymbol{k}$-CNF clause is the "or" of $\boldsymbol{k}$ variables, each possibly negated.

Ex: $X_1 \vee \neg X_5 \vee X_6$

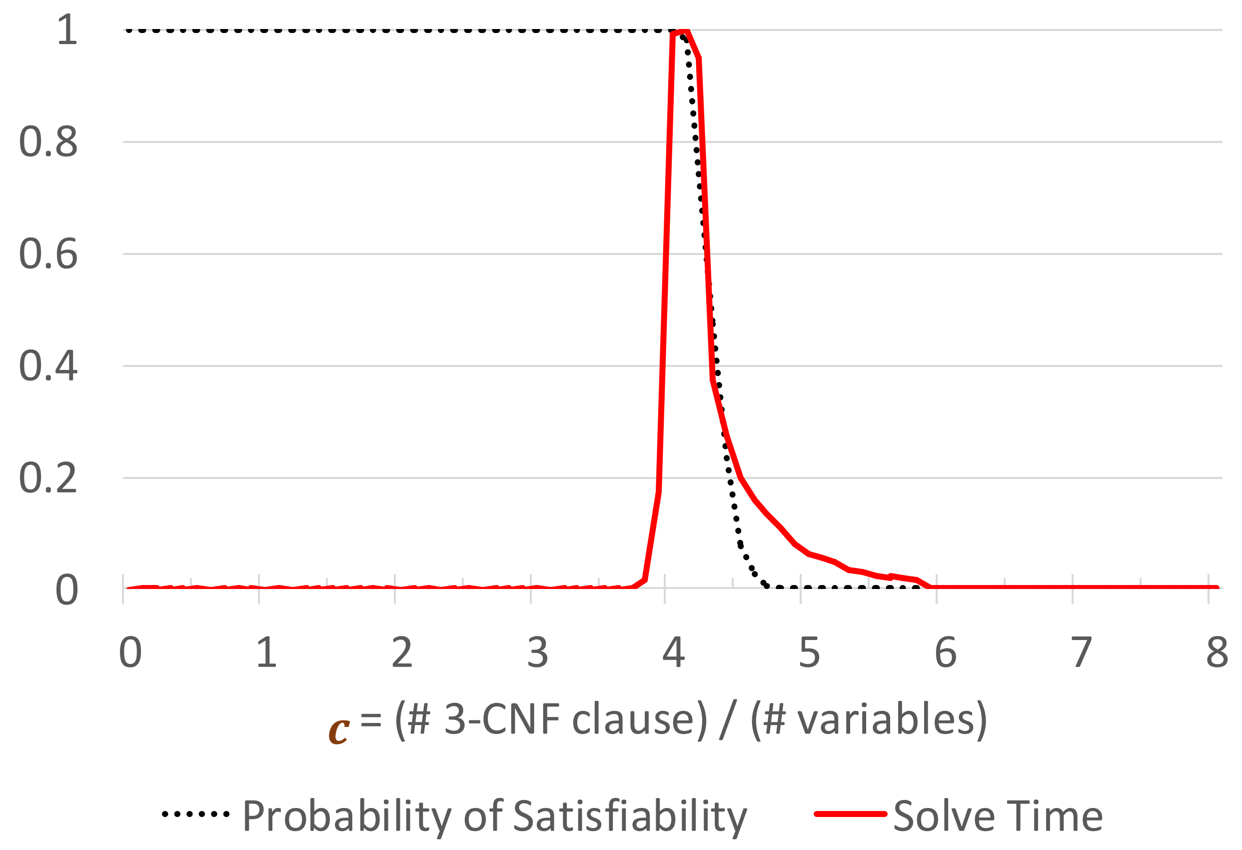
$\mathbf{CNF}_{\boldsymbol{k}}(\boldsymbol{n}, \boldsymbol{c})$ is a random formula with:

$\boldsymbol{n}$ variables.

$\boldsymbol{cn}$ independent, uniformly selected $\boldsymbol{k}$-CNF clauses.

For which parameters is $\mathrm{CNF}_{\boldsymbol{k}}(\boldsymbol{n}, \boldsymbol{c})$ hard to solve?

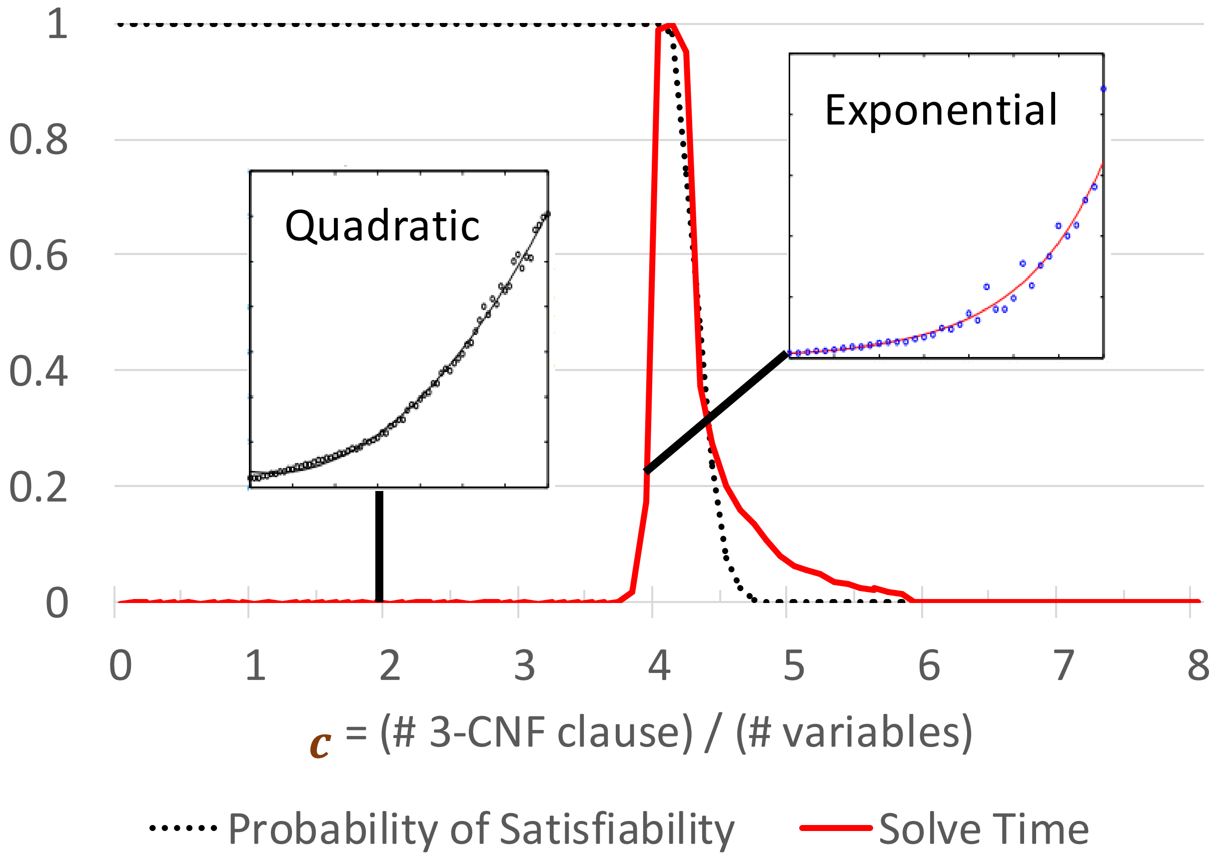
Median solve time of SAT Solver on $\mathrm{CNF}_3(200, \boldsymbol{c})$



[Cheeseman *et al.*, 1991]

Prior Work: Runtime Scaling of random k-CNF

Median solve time of SAT Solver on $\mathrm{CNF}_3(200, \boldsymbol{c})$

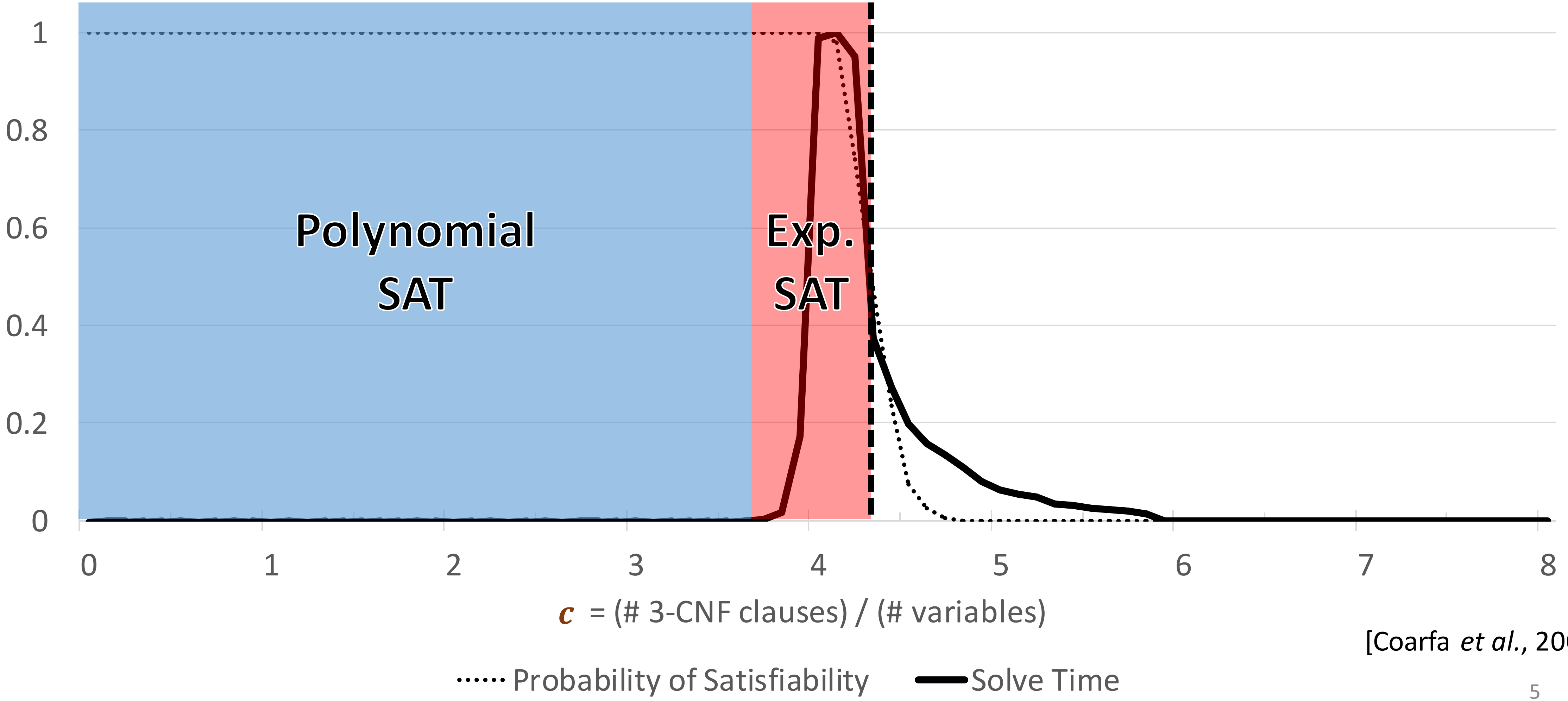


Runtime-Scaling Experiment [Coarfa *et al.*, 2003]

1. Fix a SAT Solver (GRASP).
2. Fix a value for $\boldsymbol{k}$ and $\boldsymbol{c}$ (i.e., all parameters except $\boldsymbol{n}$, the number of variables).
3. Incrementally increase $\boldsymbol{n}$. At each $\boldsymbol{n}$:
   T($\boldsymbol{n}$) = median runtime of solving CNF$_{\boldsymbol{k}}(\boldsymbol{n}, \mathbf{c})$
4. Compute the best-fit line to T($\boldsymbol{n}$)

Prior Work: Runtime Scaling of random k-CNF

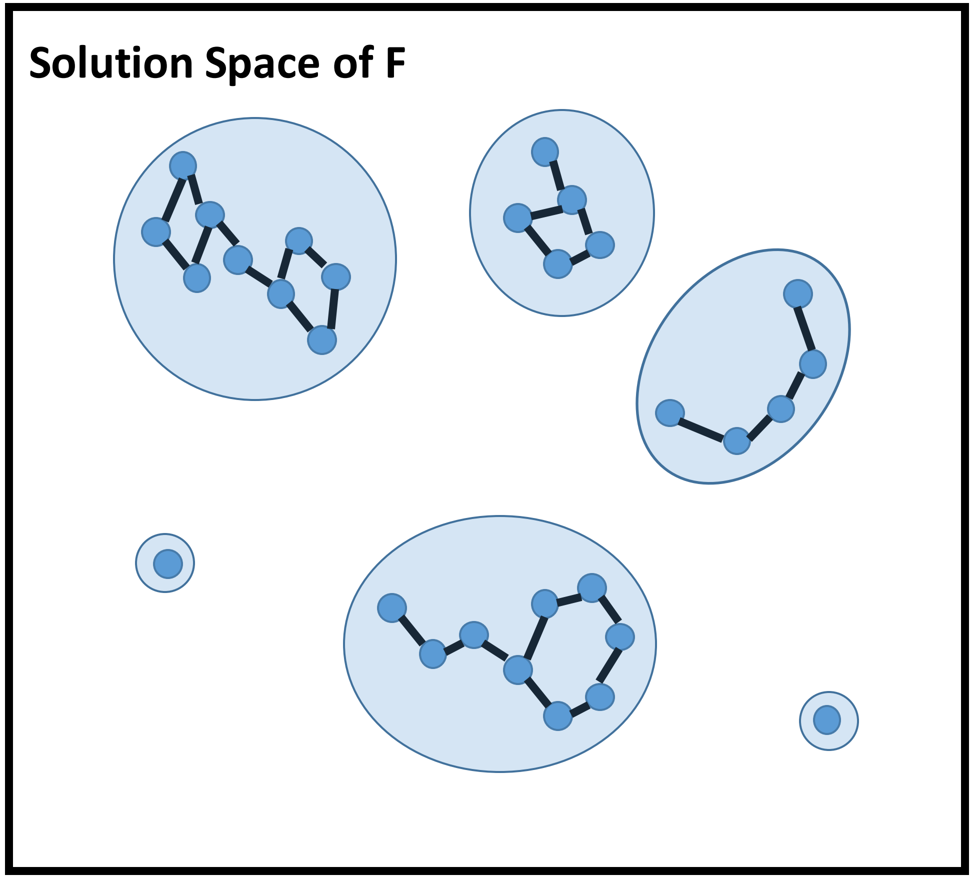
Average satisfiability and median solve time of SAT Solver (CDCL) on $\mathrm{CNF}_3(200, \boldsymbol{c})$



[Coarfa *et al.*, 2003]

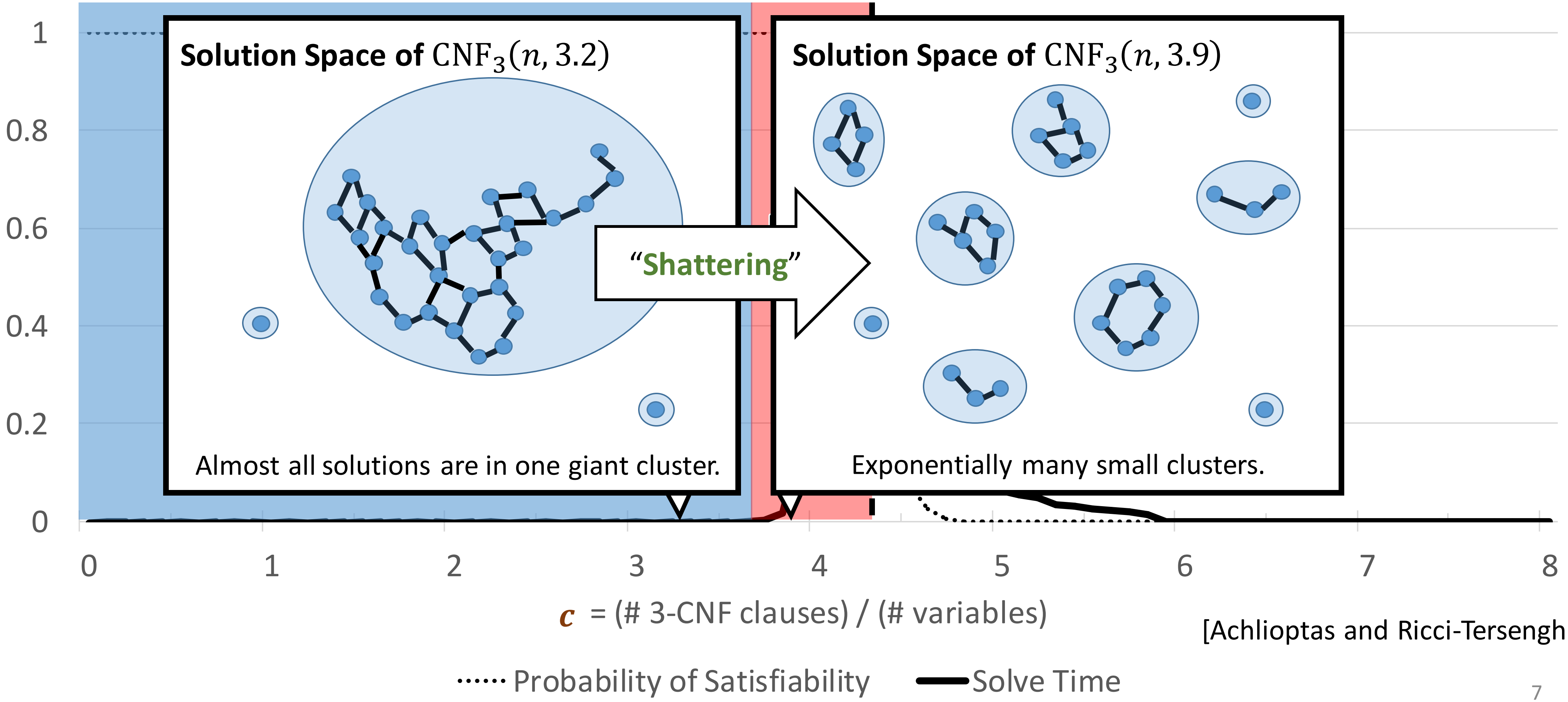
Solution-Space Geometry [Achlioptas and Ricci-Tersenghi, 2011]

Draw an edge between two solutions if they differ on no more than δn variables.



Solution-Space Geometry of Random k-CNF

Average satisfiability and median solve time of SAT Solver (CDCL) on $\mathrm{CNF}_3(200, c)$



[Achlioptas and Ricci-Tersenghi, 2011]

Random p-XOR Formulas [Goerdt, 1996]

An **XOR clause** is the "sum mod 2" of variables, set equal to 0 or 1.

- **Ex:** $X_1 + X_4 + X_5 + X_6 = 0 \pmod 2$

$\mathrm{XOR}^{p}(\boldsymbol{n}, \boldsymbol{x})$ is a random formula with:

- $\boldsymbol{n}$ variables.
- $\boldsymbol{xn}$ independent XOR clauses, where each clause is randomly sampled by including each variable independently with probability $\boldsymbol{p}$.

(Exactly as in hashing-based sampling and counting algorithms)

Which XOR Formulas are hard to solve?

XOR Formulas are all easy (polynomial) with Gaussian Elimination.

Combining CNF and XOR Together

- **Definition:** A **CNF-XOR Formula** is a formula where each clause is either a CNF clause or an XOR clause.
- **Definition:** $\mathbf{CNFXOR}_k^p(n, c, x)$:= $\mathrm{CNF}_k(n, c) \land \mathrm{XOR}^p(n, x)$

  (A random formula with n variables, cn k-CNF clauses, and xn p-XOR clauses)

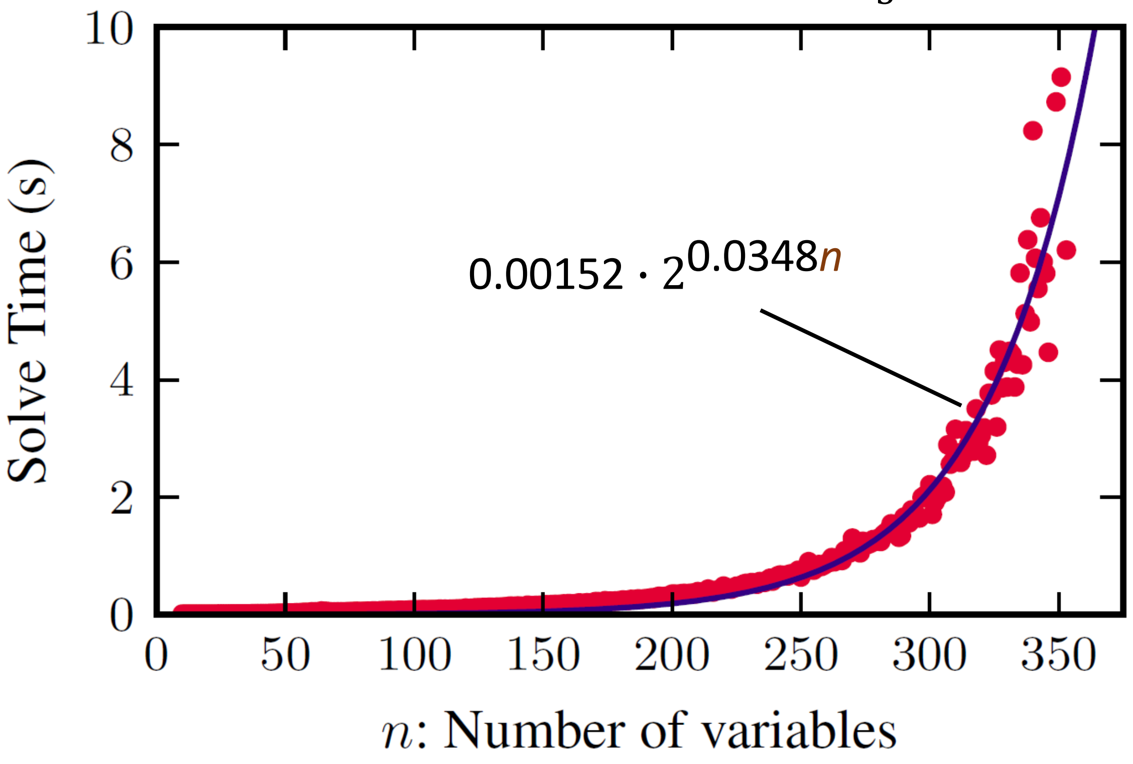
- **Goal**: Analyze the "behavior" of CNF-XOR formulas.
  - [**Dudek *et al.***, 2016] There is a phase-transition in the satisfiability of $\mathbf{CNFXOR}_k^{1/2}(n, c, x)$.
  - In this work, we analyze the runtime scaling behavior of SAT Solvers on $\mathbf{CNFXOR}_k^p(n, c, x)$.

Our Work: Runtime Scaling of CNF-XOR Formulas

Runtime-Scaling Experiment

1. Fix a SAT Solver (CryptoMiniSAT).
2. Fix a value for $\boldsymbol{k}$, $\boldsymbol{p}$, $\boldsymbol{c}$, and $\boldsymbol{x}$ (i.e., all parameters except $\boldsymbol{n}$, the number of variables).
3. Incrementally increase $\boldsymbol{n}$. At each $\boldsymbol{n}$:
   T($\boldsymbol{n}$) = median runtime of solving $\mathrm{CNFXOR}_{\boldsymbol{k}}^{\boldsymbol{p}}(\boldsymbol{n}, \boldsymbol{c}, \boldsymbol{x})$
4. Compute the best-fit line to T($\boldsymbol{n}$)

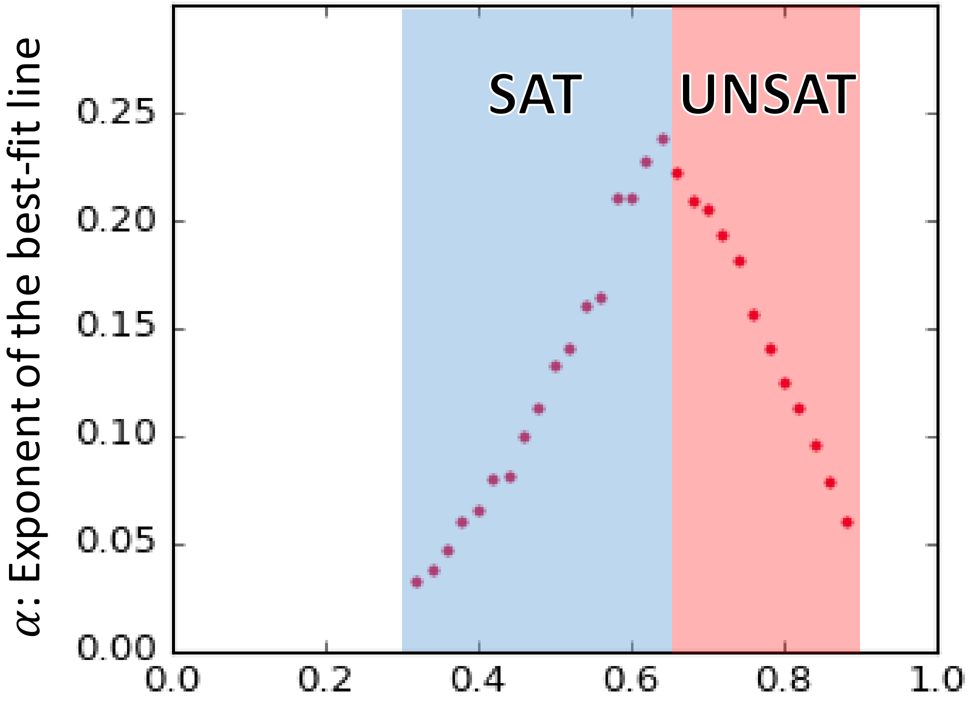
T($\boldsymbol{n}$) = Median Solve Time of $\mathrm{CNFXOR}_3^{1/2}(\boldsymbol{n}, 2, 0.3)$



How does the Runtime Scaling change?

Q: How does the runtime scaling change when x changes?

Exponent in time of solving $\mathrm{CNFXOR}_3^{1/2}(n, 2, x)$



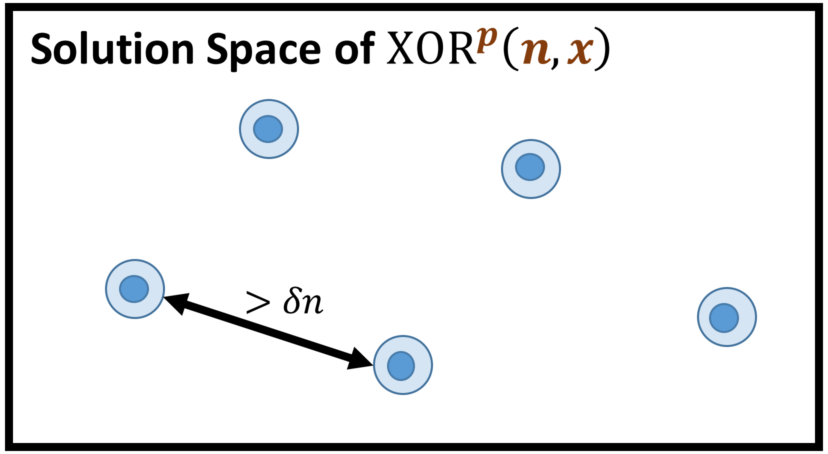
x : (# XOR clauses) / (# variables)

A: The exponent peaks near the phase-transition location.

Theory: Shattering the CNF-XOR Solution Space

- What does the solution-space geometry of random CNF-XOR formulas look like?

- What does the solution-space geometry of random XOR formulas look like?
- **Theorem**: For all $p \in (0, {}^1\!/_2]$, $x \in (0,1)$, the solution space of $\mathrm{XOR}^p(n, x)$ is shattered.



- **Corollary:** The solution space of random CNF-XOR formulas is *always* shattered.

Conclusion: CNF-XOR formulas are hard almost everywhere.

Prior Work: SAT Solvers scale exponentially on random k-CNF formulas when the solution-space shatters.

Our Contribution: "When do SAT solvers scale exponentially on CNF-XOR formulas?"

- CryptoMiniSAT scales exponentially for many parameters; no polynomial region.
- The CNF-XOR solution-space always shatters.
- Explain empirical observations on solving CNF-XOR formulas.

Future Work: Why are unsatisfiable CNF-XOR formulas hard?

Thanks!

Citations

- [Achlioptas and Ricci-Tersenghi, 2011] D. Achlioptas and F. Ricci-Tersenghi. On the Solution-Space Geometry of Random Constraint Satisfaction Problems. In *Random Structures & Algorithms*, 2011.
- [Chakraborty *et al.* 2013] S. Chakraborty, K. S. Meel, and M. Y. Vardi. A scalable and nearly uniform generator of SAT witnesses. In *Proc. of CAV*, pages 608–623, 2013.
- [Cheeseman *et al.,* 1991] P. Cheeseman, B. Kanefsky, and W. M. Taylor. Where the really hard problems are. In *Proc. of IJCAI*, 1991.
- [Coarfa *et al.*, 2003] C. Coarfa, D. D. Demopoulus, A. S. M. Aguirre, D. Subramanian, and M. Y. Vardi. Random 3-SAT: The plot thickens. In *Constraints*, 2003.
- [Dudek *et al.,* 2016] J. M. Dudek, K. S. Meel, and M. Y. Vardi. Combining the k-CNF and XOR Phase-Transitions. In *Proc. of IJCAI,* 2016.
- [Ermon *et al.* 2013] S. Ermon, C. P. Gomes, A. Sabharwal, and B. Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proc. of ICML*, 2013.
- [Goerdt, 1996] A. Goerdt. A threshold for unsatisfiability. In *Journal of Computer and System Sciences*, 1996.
- [Gomes *et al.* 2007] C.P. Gomes, A. Sabharwal, and B. Selman. Near-Uniform sampling of combinatorial spaces using XOR constraints. In *Proc. of NIPS*, 2007.