RICE UNIVERSITY

Random CNF-XOR Formulas

by

Jeffrey Dudek

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Science

Approved, Thesis Committee:

Moshe Vardi

Moshe Y. Vardi, Chair
Karen Ostrum George Distinguished
Service Professor in Computational
Engineering

S. Devika

Devika Subramanian
Professor of Computer Science and
Electrical Engineering

Philip Ernst

Philip A. Ernst
Assistant Professor of Statistics

Houston, Texas

April, 2017

ABSTRACT

Random CNF-XOR Formulas

by

Jeffrey Dudek

Boolean Satisfiability (SAT) is fundamental in many diverse areas such as artificial intelligence, formal verification, and biology. Recent universal-hashing based approaches to the problems of sampling and counting crucially depend on the runtime performance of specialized SAT solvers on formulas expressed as the conjunction of both k-CNF constraints and variable-width XOR constraints (known as CNF-XOR formulas), but random CNF-XOR formulas are unexplored in prior work.

In this work, we present the first study of random CNF-XOR formulas. We prove that a phase-transition in the satisfiability of random CNF-XOR formulas exists for $k = 2$ and (when the number of k-CNF constraints is small) for $k > 2$. We empirically demonstrate that a state-of-the-art SAT solver scales exponentially on random CNF-XOR formulas across many clause densities, peaking around the empirical phase-transition location. Finally, we prove that the solution space of random CNF-XOR formulas 'shatters' at *all* nonzero XOR-clause densities into well-separated components.

Acknowledgments

I would like to thank my advisor, Prof. Moshe Vardi, for his guidance and feedback throughout my research. I would also like to thank Prof. Devika Subramanian and Prof. Philip Ernst for agreeing to be on my committee, and for evaluating this thesis. Most of all, I would like to thank my collaborators, colleagues, family, and friends for their endless support.

Contents

Illustrations

Chapter 1

Introduction

The Boolean-Satisfaction Problem (SAT) is the problem of determining if a given propositional formula is satisfiable. SAT is one of the most fundamental problems in computer science, with a wide range of applications arising from diverse areas such as artificial intelligence, programming languages, formal verification and the like [1, 2]. The related problems of Constrained Counting and Constrained Sampling have found a wide range of applications in probabilistic reasoning, machine learning, statistical physics, verification, and other areas [3–6]. Given an input propositional formula, the problem of constrained counting is to count the number of satisfying assignments; the problem of constrained sampling is to generate satisfying assignments uniformly at random. Although these problems are computationally intractable in their exact form (in particular, they are #P-Complete) [7,8], recent hashing-based techniques [9–13] have emerged to compute approximate solutions to constrained counting and sampling problems. Unlike previous approaches to sampling and counting, hashing-based approaches provide strong theoretical approximation guarantees and scale to real-world instances involving formulas with *hundreds of thousands* of variables [13].

These hashing-based approaches for constrained sampling and counting employ SAT solvers to solve formulas naturally expressed as the conjunction of both CNF-clauses and XOR-clauses, known as CNF-XOR formulas [10]. In particular, the CNF-clauses are generated from the input to the constrained counting/sampling algorithm while the XOR-clauses are incrementally sampled by a stochastic process. While initial work [9] sampled XOR-clauses each with a fixed number of variables, recent work [10–13] obtained tighter approximations through the use of longer *variable-width* XOR-clauses, where the number of variables in each clause is stochastic. Although XOR formulas can be solved individually in polynomial time (using, e.g., Gaussian Elimination [14]), XOR formulas are empirically hard [15] for SAT solvers without equivalence reasoning or similar techniques. The rise of applications for CNF-XOR formulas has motivated the development of specialized CNF-XOR solvers, such as CryptoMiniSAT [16], that combine SAT-solving techniques with algebraic techniques and so can reason simultaneously about about both the CNF-clauses and XOR-clauses within a single CNF-XOR formula. Understanding the runtime of these specialized CNF-XOR solvers on CNF-XOR formulas is key to understanding the runtime behavior of hashing-based algorithms for constrained sampling and counting.

A large body of work explains the runtime performance of modern SAT solvers through analysis on random problems [17], motivated by the desire to study the "typical" hardness of SAT problems. Despite the abundance of prior work on the behavior of SAT solvers on random fixed-width (where each clause contains a fixed number of literals) CNF-formulas and on certain other classes of random formulas, no prior work considers the behavior of SAT solvers on CNF-XOR formulas. We believe that analysis of the behavior of SAT solvers on random CNF-XOR formulas is the first step towards demystifying the runtime behavior of specialized CNF-XOR SAT solvers and thus explaining the runtime behavior of hashing-based algorithms for constrained sampling and counting.

In the remainder of this chapter, we briefly review motivating factors and previous work on random SAT formulas.

1.1 Analysis of Random SAT Formulas

All known SAT-solving algorithms are exponential in the worst case. Nevertheless, state-of-the-art SAT solvers can now routinely find solutions to practical problems with millions of variables [18]. Instead of focusing on the hardness of the most difficult SAT formulas, random SAT formulas have historically been studied in an effort to understand the hardness of “typical” SAT formulas and thus better understand the practical performance of modern SAT-solving algorithms [19, 20]. Indeed, insights gleaned from the study of random SAT instances have led to several dramatic improvements in SAT-solving algorithms (e.g., the introduction of random-restarts [21]).

Of particular practical and theoretical interest is the analysis of random fixed-width CNF formulas, beginning in [22]. Early experiments [23–25] revealed a connection between the density of random CNF formulas and the runtime behavior of SAT solvers on such formulas. In particular, the runtime of SAT solvers (using DPLL and related algorithms) on random CNF formulas was shown to follow an *easy-hard-easy* pattern [25]: the runtime is low when the clause density is very low or very high and peaks around a fixed density, the location of which depends only on the clause width. This peak in SAT solver runtime is paired with a precipitous drop, believed to be a phase-transition, in the probability of satisfiability of random CNF formulas. Establishing this phase-transition in satisfiability analytically has been highly challenging [26], and it has been established only for for $k = 2$ [27, 28] and all large enough k [29].

Further analysis of the relationship between the clause density and SAT solver runtime revealed a more nuanced picture of the scaling behavior of SAT solvers on random fixed-width CNF instances. In particular, a secondary phase-transition was observed within the satisfiable region, where the median runtime transitions from polynomial to exponential in the number of variables for several SAT-solving algorithms [30]. Theoretical analysis of this phenomenon [31–33] has shown that the solution space of a random fixed-width CNF formula dramatically "shatters" at this secondary phase-transition into a solution space known to be difficult for a variety of SAT-solving algorithms [34, 35].

The goal of this thesis is to apply the techniques described above to random CNF-XOR formulas, a new class of random SAT formulas. Though this analysis, we hope to understand the behavior of specialized CNF-XOR solvers on typical CNF-XOR formulas and so explain the runtime behavior of hashing-based algorithms for constrained counting and sampling in practice.

1.2 Contributions

The main contribution of this thesis is a first study of random CNF-XOR formulas.

We define a class of random k-CNF-XOR formulas with both fixed-width CNF clauses (i.e., k-clauses) and variable-width XOR-clauses, motivated by hashing-based approaches to constrained sampling and counting.

We present the first study of phase-transition phenomenon in the satisfiability of random k-CNF-XOR formulas. In particular, we present experimental evidence for a k-CNF-XOR phase-transition that follows a linear trade-off between k-clauses and XOR-clauses. We prove that the k-CNF-XOR phase-transition exists when the ratio of k-clauses to variables is small. Notably, this fully characterizes the phase-transition when $k = 2$.

We present the first study of the runtime behavior of CNF-XOR solvers on random k-CNF-XOR formulas. In particular, we present experimental evidence that the runtime of the specialized CNF-XOR solver CryptoMiniSAT scales exponentially in the number of variables, even when both the CNF and XOR subformulas are separately solvable in polynomial time by CryptoMiniSAT. We show that the runtime peaks near the empirical phase-transition location for random k-CNF-XOR formulas. Moreover, we show that the solution space geometry of random CNF-XOR formulas is known to be difficult for many SAT-solving algorithms.

1.3 Organization

The remainder of this thesis is organized as follows.

Chapter 2 presents notation and describes related work both on random CNF formulas and on random XOR formulas. It formally defines the notion of a random k-CNF-XOR formula.

Chapter 3 discusses phase-transition phenomena in the satisfiability of random k-CNF-XOR formulas, covering both empirical evidence and theoretical analysis.

Chapter 4 discusses the runtime scaling behavior of a specialized SAT solver, CryptoMiniSAT, on random k-CNF-XOR formulas. It begins to explain the runtime scaling behavior (in the satisfiable region) through analysis of the solution-space geometry of random k-CNF-XOR formulas.

Finally, Chapter 5 summarizes the main contributions of this thesis and describes several possible future directions.

Chapter 2

Preliminaries

In this chapter, we introduce notations and preliminaries needed to present and understand our work. We begin with some basic notations.

2.1 Notations

Let $X = \{X_1, \cdots, X_n\}$ be a set of propositional variables and let F be a formula defined over X. A *satisfying assignment* or *solution* of F is an assignment of truth values to the variables in X such that F evaluates to true. The *solution space* of F is the set of all satisfying assignments. Let $\#F$ denote the number of satisfying assignments of F. We say that F is *satisfiable* (or *sat.*) if $\#F > 0$ and that F is *unsatisfiable* (or *unsat.*) if $\#F = 0$.

We describe the solution space of F using terminology from Achlioptas and Molloy [36]. Two satisfying assignments σ and τ of F are *d-connected*, for a real number d, if there exists a sequence of solutions $\sigma, \sigma', \cdots, \tau$ of F such that the Hamming distance of every two successive elements in the sequence is at most d. A subset S of the solution space of F is a *d-cluster* if every $\sigma, \tau \in S$ is d-connected. Two subsets S, S' of the solution space of F are *d-separated* if every pair $\sigma \in S$ and $\tau \in S'$ is not d-connected. Moreover, we say that F is d-separated if the Hamming distance between every pair of solutions of F is at least d.

If $g(n)$ is a function of n, we use $O(g(n))$ as shorthand for some function $g'(n) \in$

$O(g(n))$ and use $\Omega(g(n))$ as shorthand for some function $g''(n) \in \Omega(g(n))$ (where the choice of $g'(n)$ and $g''(n)$ is independent of n).

We use $\mathsf{Pr}\left[X\right]$ to denote the probability of event X. We say that an infinite sequence of random events $E_1, E_2, \cdots$ occurs *with high probability* (denoted, w.h.p.) if $\lim\limits_{n\to\infty} \mathsf{Pr}\left[E_n\right] = 1$. We use $\mathsf{E}\left[Y\right]$ to denote the expected value of a random variable Y.

2.2 Prior Work on k-CNF formulas

A *k-clause* (or *CNF-clause*) is the disjunction of k literals out of $\{X_1, \cdots, X_n\}$, with each variable possibly negated. For fixed positive integers k and n and a nonnegative real number r (known as the *k-clause density*), let the random variable $F_k(n, rn)$ denote the formula consisting of the conjunction of $\lceil rn \rceil$ k-clauses, each chosen uniformly and independently from all $\binom{n}{k}2^k$ possible k-clauses over n variables.

The early experiments on $F_k(n, rn)$ [23–25] led to the following conjecture:

Conjecture 1 (Satisfiability Phase-Transition Conjecture)**.** *For every integer $k \geq 2$, there is a critical ratio r_k such that:*

1. *If $r < r_k$, then $F_k(n, rn)$ is satisfiable w.h.p.*

2. *If $r > r_k$, then $F_k(n, rn)$ is unsatisfiable w.h.p.*

The Conjecture was quickly proved for $k = 2$, where $r_2 = 1$ [27, 28]. In recent work, Ding, Sly, and Sun established the Satisfiability Phase Transition Conjecture for all sufficiently large k [29]. The Conjecture has remained elusive for small values of $k \geq 3$, although values for these critical ratios r_k can be estimated experimentally (e.g., r_3 seems to be near 4.26) and predicted analytically using techniques from statistical physics [37].

When the k-clause density is small (e.g. below $2^k \ln(k)/k$) there are algorithms that are known to solve $F_k(n, rn)$ with high probability in polynomial time [38]. No algorithm is known that can solve $F_k(n, rn)$ in polynomial time when the clause density is larger, even when $F_k(n, rn)$ is still expected to have exponentially many solutions [33]. The solution space of $F_k(n, rn)$ can also be characterized in the satisfiable region. In particular, for every $k \geq 8$ there exists some k-clause density r where w.h.p. $F_k(n, rn)$ is satisfiable and almost all of the solution space of $F_k(n, rn)$ can be partitioned into exponentially many $O(n)$-clusters such that each pair of clusters is $\Omega(n)$-separated [33]. This 'shattering' of the solution space into linearly separated clusters is known to be difficult for a variety of SAT algorithms [34, 35].

2.3 Prior Work on XOR formulas

An XOR-clause over n variables is the 'exclusive or' of either 0 or 1 together with a subset of the variables $X_1, \cdots, X_n$. An XOR-clause including 0 (respectively, 1) evaluates to true if and only if an odd (respectively, even) number of the included variables evaluate to true. For a fixed positive integer n and a nonnegative real number p, a *random XOR-clause with variable-probability* p is an XOR clause A chosen so that each X_i is included in A independently with probability p and 1 is included in A independently with probability 1/2. Note that all k-clauses contain *exactly* k variables, whereas the number of variables in an XOR-clause is not fixed; a random XOR-clause chosen with variable-probability p over n variables contains pn variables in expectation.

For a fixed positive integer n, a nonnegative real number s (known as the *XOR-clause density*), and a nonnegative real number p (known as the *XOR variable-probability*), let the random variable $Q^p(n, sn)$ denote the formula consisting of the conjunction of $\lceil sn \rceil$ XOR-clauses, with each clause an independently chosen random XOR-clause with variable-probability p. Creignou and Daude [39,40] proved a phase-transition in the satisfiability of $Q^{1/2}(n, sn)$: if $s < 1$ then $Q^{1/2}(n, sn)$ is satisfiable w.h.p., while if $s > 1$ then $Q^{1/2}(n, sn)$ is unsatisfiable w.h.p. The solution space geometry of $Q^p(n, sn)$ has not been characterized in prior work.

The random variable $Q^{1/2}(n, sn)$ matches the XOR-clauses used in several hashing-based constrained sampling and counting algorithms [10]. Recent work [12] has also made use of $Q^p(n, sn)$ with $p < 1/2$ for constrained sampling and counting algorithms.

There is a related model of random fixed-width ℓ-XOR formulas where every XOR-clause contains exactly ℓ variables. Creignou and Daudé [40] also proved the existence of a phase transition for random ℓ-XOR formulas (where each XOR-clause contains exactly ℓ literals), for $\ell \geq 1$, without specifying an exact location for the phase-transition. Dubois and Mandler [41] independently identified the location of a phase transition for random 3-XOR formulas. More recently, Pittel and Sorkin [42] identified the location of the phase-transition for ℓ-XOR formulas for $\ell > 3$. For ℓ-XOR formulas, w.h.p. the solution space can be partitioned into a set of $O(\log n)$-clusters such that each pair of clusters is $\Omega(n)$-separated [36,43].

2.4 Defining CNF-XOR formulas

A CNF-XOR formula (respectively, k-CNF-XOR formula) is the conjunction of some number of CNF-clauses (respectively, k-clauses) and XOR-clauses. For fixed positive integers k and n and fixed nonnegative real numbers r and s, let the random variable $\psi_k^p(n, rn, sn)$ denote the formula consisting of the conjunction of $\lceil rn \rceil$ k-clauses, each chosen uniformly and independently from all possible k-clauses over n variables, and $\lceil sn \rceil$ independently chosen XOR-clauses with variable-probability p. (The motivation for using fixed-width CNF-clauses and variable-width XOR-clauses comes from the hashing-based approaches to constrained sampling and counting discussed in Chapter 1.) Although random k-CNF formulas and XOR formulas have been well studied separately, no prior work considers the satisfiability of random mixed formulas arising from conjunctions of k-clauses and XOR-clauses.

2.5 Experimental Setup

In this thesis, we explore empirically the behavior of CNF-XOR solvers on randomly constructed k-CNF-XOR formulas. To do this, we built a prototype implementation in Python that employs the CryptoMiniSAT[*] [16] solver to check satisfiability of random k-CNF-XOR formulas. We chose CryptoMiniSAT because it is typically used in hashing-based approaches to sampling and counting due to its ability to handle the combination of k-clauses and XOR-clauses efficiently [44].

The objective of the experimental setup is to empirically determine the behavior of CryptoMiniSAT on checking satisfiability of $\psi_k^p(n, rn, sn)$ with respect to n (the number of variables), r (the k-clause density), s (the XOR-clause density), and p (the XOR variable-probability) for fixed k. In particular, we aim to estimate both (in Chapter 3) the probability that $\psi_k^p(n, rn, sn)$ is satisfiable and (in Chapter 4) the median solve time for CryptoMiniSAT on $\psi_k^p(n, rn, sn)$.

To uniformly choose a k-clause we uniformly selected without replacement k out of the variables $\{X_1, \cdots, X_n\}$. For each selected variable X_i, we include exactly one of the literals X_i or $\neg X_i$ in the k-clause, each with probability 1/2. The disjunction of these k literals is a uniformly chosen k-clause. To choose an XOR-clause with

[*] `http://www.msoos.org/cryptominisat4/`

variable-probability p, we include each variable of $\{X_1, \cdots, X_n\}$ with probability p in a set A of variables. We also include in A exactly one of 0 or 1, each with probability 1/2. The 'exclusive-or' of all elements of A is a random XOR-clause with variable-probability p.

To empirically estimate both the probability that $\psi_k^p(n, rn, sn)$ is satisfiable and the median runtime of CryptoMiniSAT on $\psi_k^p(n, rn, sn)$, we evaluated satisfiability, using CryptoMiniSAT, of 100 uniformly generated formulas of $\psi_k^p(n, rn, sn)$ by constructing the conjunction of $\lceil rn \rceil$ k-clauses and $\lceil sn \rceil$ XOR-clauses (with variable-probability p), with each clause chosen independently as described above. The solving of each formula was individually timed. The percentage of satisfiable formulas gives us an empirical estimate of $\mathsf{Pr}\left[\psi_k^p(n, rn, sn) \text{ is satisfiable}\right]$. The median runtime is an estimate for the median CryptoMiniSAT solve time on k-CNF-XOR formulas with parameters $(k,\, p,\, n,\, r,\, s)$.

All experiments were run on a node within a high-performance computer cluster. These nodes contain 12-processor cores at 2.83 GHz each with 48 GB of RAM per node.

Chapter 3

Phase-Transition Phenomena

As introduced in Chapter 1, there is a deep connection between the runtime behavior of SAT solvers on random CNF formulas and on the probability that such random formulas are satisfiable. The key experimental findings [23–25] are: (1) as the density (ratio of clauses to variables) of random CNF instances increases, the probability of satisfiability decreases with a precipitous drop, believed to be a phase-transition, around the point where the probability of satisfiability is 0.5, and (2) instances at the phase-transition point are particularly challenging for DPLL-based SAT solvers. Indeed, phase-transition instances serve as a source of difficult benchmark problems in SAT competitions [45]. The connection between runtime performance and the satisfiability phase-transition has propelled the study of such phase-transition phenomena over the past two decades [19].

For random k-CNF formulas, where every clause contains exactly k literals, experiments suggest a specific phase-transition density (for example, density 4.26 for random 3-CNF formulas), but establishing this analytically has been highly challenging [26], and it has been established only for for $k = 2$ [27, 28] and all large enough k [29]. A phase-transition phenomenon has also been identified in random XOR formulas (with variable-probability $\frac{1}{2}$) at density 1 [39].

Despite the abundance of prior work on the phase-transition phenomenon in the satisfiability of random k-CNF formulas and random XOR formulas, no prior work considers the satisfiability of random k-CNF-XOR formulas. Since the phase-transition behavior of k-CNF constraints have been analyzed to explain runtime behavior of SAT solvers [20], analysis of the phase-transition phenomenon for random k-CNF-XOR formulas is the first step towards demystifying the runtime behavior of CNF-XOR solvers (e.g., CryptoMiniSAT [16]) and thus explaining the runtime behavior of hashing-based algorithms.

In the remainder of this chapter, we present the first study of phase-transition phenomenon in the satisfiability of random k-CNF-XOR formulas, henceforth referred to as the k-CNF-XOR phase-transition. In particular:

1. We present (in Section 3.2) experimental evidence for a k-CNF-XOR phase-transition following a linear trade-off between k-CNF clauses and XOR clauses.

2. We prove (in Section 3.3) that the k-CNF-XOR phase-transition exists when the ratio of k-CNF clauses to variables is small. This fully characterizes the phase-transition when $k = 2$.

3. We prove (in Section 3.3) upper and lower bounds on the location of the k-CNF-XOR phase-transition region.

4. We conjecture (in Section 3.4) that the exact location of a phase-transition for $k \geq 3$ follows the linear trade-off between k-CNF and XOR clauses seen experimentally.

3.1 Experimental Setup

We used the experimental setup described in Section 2.5 to explore empirically the satisfiability of random k-CNF-XOR formulas. In particular, the objective of the experimental setup is to empirically determine the behavior of $\mathsf{Pr}\left[\psi_k^p(n, rn, sn) \text{ is sat}\right]$

with respect to r and s, the k-clause and XOR-clause densities respectively, for fixed k, p, and n.

We ran 55 experiments with various values of k, p, and n. The value of p ranged over $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, independently of k and n. For $k = 2$, we ran experiments for $n \in \{25, 50, 100, 150\}$. For $k = 3$, we ran experiments for $n \in \{25, 50, 100\}$. For $k = 4$ and $k = 5$, we ran experiments for $n \in \{25, 50\}$. We were not able to run experiments for values of n significantly larger than those listed above: at some k-clause and XOR-clause densities, the run-time of CryptoMiniSAT scaled far beyond our computational capabilities.

In each experiment, the XOR-clause density s ranged from 0 to 1.2 in increments of 0.02. Since the location of phase-transition for $k-$CNF depends on k, the range of k-clause density r also depends on k. For $k = 3$, r ranged from from 0 to 6 in increments of 0.04; for $k = 5$, r ranged from 0 to 26 in increments of 0.43, and the like. For each assignment of values to k, p, r, s, and n, we used the experimental setup described in Section 2.5 to estimate the probability that $\psi_k^p(n, rn, sn)$ is satisfiable. All formulas were given a timout of 1000 seconds.

3.2 Experimental Results

We present scatter plots demonstrating the behavior of satisfiability of k-CNF-XOR formulas. We present here results only for the six experiments when $p \in \{0.2, 0.5\}$ and $(k, n) \in \{(2, 150), (3, 100), (5, 50)\}$. [*].

The plots for $k = 2, 3$ and 5 are shown in Figures 3.1 and 3.2, Figures 3.3 and 3.4, and Figures 3.5 and 3.6 (respectively, for $p = 0.5$ and $p = 0.2$).

[*] The data from all experiments is available at `http://www.cs.rice.edu/CS/Verification/Projects/CUSP/`

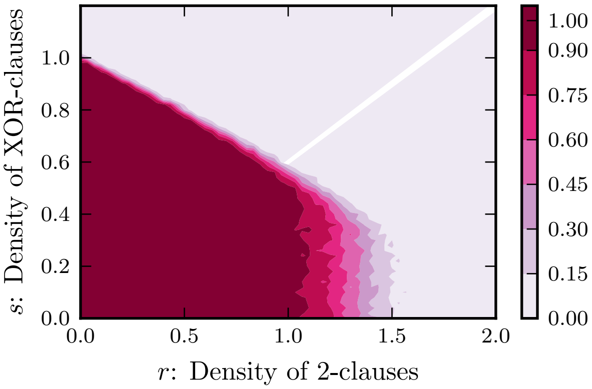


Figure 3.1 : Phase transition for 2-CNF-XOR formulas ($p = 0.5$)

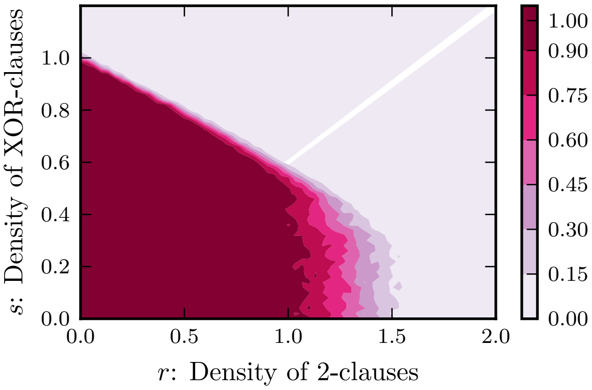


Figure 3.2 : Phase transition for 2-CNF-XOR formulas ($p = 0.2$)

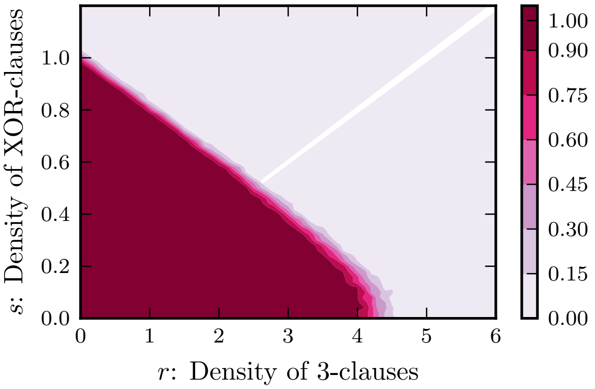


Figure 3.3 : Phase transition for 3-CNF-XOR formulas ($p = 0.5$)

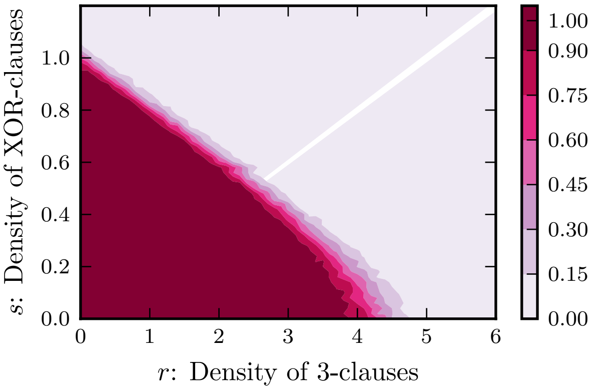


Figure 3.4 : Phase transition for 3-CNF-XOR formulas ($p = 0.2$)

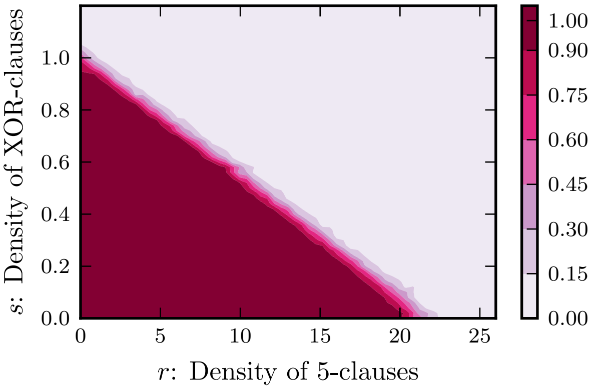


Figure 3.5 : Phase transition for 5-CNF-XOR formulas ($p = 0.5$)

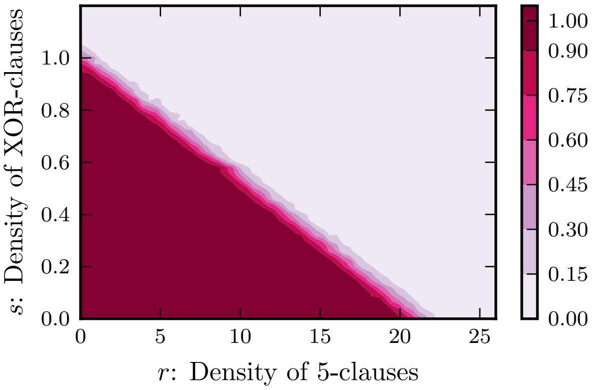


Figure 3.6 : Phase transition for 5-CNF-XOR formulas ($p = 0.2$)

Each figure is a 2D plot, representing the observed probability that $\psi_k^p(n, rn, sn)$ is satisfiable as the density of k-clauses r and the density of XOR-clauses s varies. The x-axis indicates the density of k-clauses r. The y-axis indicates the density of XOR-clauses s. The dark (respectively, light) regions represent clause densities where almost all (respectively, no) sampled formulas were satisfiable.

Note that $\psi_k^p(n, rn, sn)$ consists only of XOR clauses when $r = 0$. Examining the figures along the line $r = 0$ the phase-transition location is around ($r = 0$, $s = 1$), which matches previous theoretical results on the phase-transition for XOR formulas [39]. Likewise, $\psi_k^p(n, rn, 0) = F_k(n, rn)$ and, by examining the figures along the line $s = 0$, we observe phase-transition locations that match previous studies on the phase-transition for k-CNF formulas for $k = 2, 3,$ and 5 [19]. Note that the phase-transition we observe for 2-CNF formulas is slightly above the true location at $s = 1$ [27,28]; the correct phase-transition point for 2-CNF formulas is observed only when the number of variables is above 4096 [46].

In all the plots, we observe a large triangular region where the probability that $\psi_k^p(n, rn, sn)$ is satisfiable is nearly 1. We likewise observe a separate region where the observed probability that $\psi_k^p(n, rn, sn)$ is satisfiable is nearly 0. More surprisingly, the shared boundary between the two regions for large areas of the plots seems to be a constant-slope line. A closer examination of this line at the bottom-right corners of the figures for $k = 2$ and $k = 3$, where the k-clause density is large, reveals that the line appears to “kink” and abruptly change slope. We discuss this further in Section 3.4.

Notice that in all cases the plots for $p = 0.5$ and for $p = 0.2$, for identical values of n and k, are nearly identical. This suggests that the location of the phase-transition is independent of the XOR variable-probability.

3.3 Establishing a Phase-Transition

The experimental results presented in Section 3.2 empirically demonstrate the existence of a k-CNF-XOR phase-transition. Theorem 3.1 shows that the k-CNF-XOR phase-transition exists when the density of k-clauses is small. In particular, the function $\phi_k(r)$ (defined in Lemma 3.3) gives the location of a phase-transition between a region of satisfiability and a region of unsatisfiability in random k-CNF-XOR formulas. Moreover, this location is independent of p.

Theorem 3.1 (k-CNF-XOR Phase-Transition Theorem)**.** *Let $k \geq 2$. There is a function $\phi_k(r)$, a constant $\alpha_k \geq 1$, and a countable set of real numbers $\mathcal{C}_k$ (all defined in Lemma 3.3) such that for all $p \in (0, 1/2]$, $r \in [0, \alpha_k) \backslash \mathcal{C}_k$, and $s \geq 0$:*

(a). If $s < \phi_k(r)$, then w.h.p. $\psi_k^p(n, rn, sn)$ is satisfiable.

(b). If $s > \phi_k(r)$, then w.h.p. $\psi_k^p(n, rn, sn)$ is unsatisfiable.

Proof. Part (a) follows directly from Lemma 3.9. Part (b) follows directly from Lemma 3.14. The proofs of these lemmas are presented in Sections 3.3.1 and 3.3.2 respectively. □

$\phi_k(r)$ is the *free-entropy density* of k-CNF, drawing on concepts from spin-glass theory [47]. From the expression for $\phi_k(r)$ in Lemma 3.3, it is easily verified that $\phi_k(0) = 1$ and that $\phi_k(r)$ is a monotonically decreasing function of r. Thus when the k-clause density (r) is 0, Theorem 3.1 says that an XOR-clause density of 1 is a phase-transition for XOR-formulas (independently of the XOR variable-probability p), matching previously known results for $p = 1/2$ [39]. As the k-clause density increases, $\phi(r)$ is decreasing and so the XOR-clause density required to reach the phase-transition decreases.

Theorem 3.1 fully characterizes the random satisfiability of $\psi_k^p(n, rn, sn)$ when $r < 1$. In the case $k = 2$, prior results on random 2-CNF satisfiability characterize the rest of the region. If $r > 1$, then $F_2(n, rn)$ is unsatisfiable w.h.p. [27, 28] and so the 2-clauses within $\psi_2^p(n, rn, sn)$ are unsatisfiable w.h.p. without considering the XOR-clauses. Therefore $\psi_2^p(n, rn, sn)$ is unsatisfiable w.h.p. if $r > 1$. This, together with Theorem 3.1, proves that $\phi_2(r)$ is the complete location of the 2-CNF-XOR phase-transition.

Moreover, Lemma 3.4 shows that $\alpha_k \geq (1 - o_k(1)) \cdot 2^k \ln(k)/k$ (where $o_k(1)$ denotes a term that converges to 0 as $k \to \infty$) and so Theorem 3.1 shows that a phase-transition exists until near $r = 2^k \ln(k)/k$ for sufficiently large k.

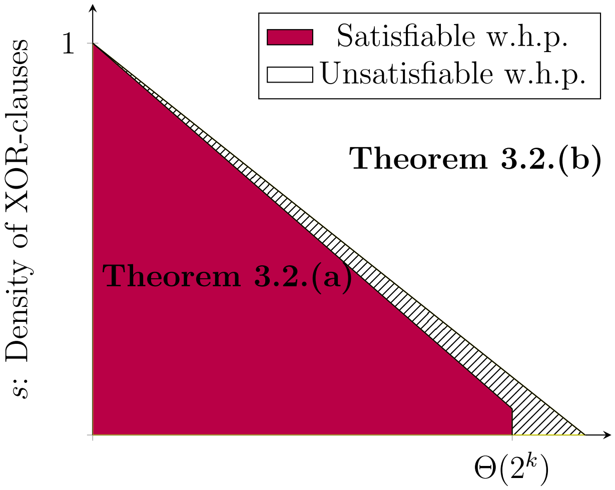
For small $k \geq 3$, the region $r < 1$ characterized by Theorem 3.1 is only a small portion of the region where the subset of k-clauses remains satisfiable. Moreover, the location of the phase-transition $\phi_k(r)$ given by Theorem 3.1 is difficult to compute directly. Theorem 3.2 gives explicit lower and upper bounds on the location of a phase-transition region.

Theorem 3.2. *Let $k \geq 3$. There is a function $\Lambda_b(k, r)$ (defined in Lemma 3.5) such that for all $p \in (0, 1/2]$, $s \geq 0$ and $r \geq 0$:*

(a). If $s < \frac{1}{2}\log_2(\Lambda_b(k, r))$ and $r < 2^k \ln(2) - \frac{1}{2}((k+1)\ln(2) + 3)$, then w.h.p. $\psi_k^p(n, rn, sn)$ is satisfiable.

(b). If $s > r \log_2(1 - 2^{-k}) + 1$, then w.h.p. $\psi_k^p(n, rn, sn)$ is unsatisfiable.

Proof. Part (a) follows directly from Lemma 3.10. Part (b) follows directly from Lemma 3.15. The proofs of these lemmas are presented in Sections 3.3.1 and 3.3.2 respectively. □



r: Density of k-clauses

Figure 3.7 : Satisfiability of $\psi_k^p(n, rn, sn)$ as $n \to \infty$

Both the upper bound $r \log_2(1 - 2^{-k}) + 1$ and (using the expression for $\Lambda_b(k, r)$ in Lemma 3.5) the lower bound $\frac{1}{2} \log_2(\Lambda_b(k, r))$ are linear in r. When the k-clause density r is 0, Theorem 3.2 agrees with Theorem 3.1. As the k-clause density increases past $\Theta(2^k)$, Theorem 3.2 no longer gives a lower bound on the location of a possible phase-transition.

3.3.1 A Proof of the Lower Bound

We now establish Theorem 3.1.(a) and Theorem 3.2.(a), which follow directly from Lemma 3.9 and Lemma 3.10 respectively.

The key idea in the proof of these lemmas is to decompose $\psi_k^p(n, rn, sn)$ into independently generated k-CNF and XOR formulas, so that $\psi_k^p(n, rn, sn) = F_k(n, rn) \wedge Q^p(n, sn)$. We can then bound the number of solutions to $F_k(n, rn)$ from below with high probability and bound from below the probability that $F_k(n, rn)$ remains satisfiable after including XOR-clauses on top of $F_k(n, rn)$.

The following three lemmas achieve the first of the two tasks. The first, Lemma 3.3, gives a tight bound on $\#F_k(n, rn)$ for small k-clause densities.

Lemma 3.3. *Let $k \geq 2$ and let α_k be the supremum of*

$$\{r : \exists \delta > 0 \textit{ s.t. } \mathsf{Pr}\left[F_k(n, rn) \textit{ is unsat.}\right] \leq O(1/(\log n)^{1+\delta})\}.$$

Then $\alpha_k \geq 1$. Furthermore, there exists a countable set of real numbers $\mathcal{C}_k$ such that for all $r \in [0, \alpha_k) \backslash \mathcal{C}_k$:

- *(a). The sequence $\frac{1}{n}\mathsf{E}\left[\log_2(\#F_k(n, rn)) \mid F_k(n, rn) \textit{ is sat.}\right]$ converges to a limit as $n \to \infty$. Let $\phi_k(r)$ be this limit.*
- *(b). For all $\epsilon > 0$, w.h.p. $(2^{\phi_k(r)-\epsilon})^n \leq \#F_k(n, rn)$.*
- *(c). For all $\epsilon > 0$, w.h.p. $(2^{\phi_k(r)+\epsilon})^n \geq \#F_k(n, rn)$.*

Proof. These proofs are given in [48]. $\alpha_k \geq 1$ is given as **Remark 2**. Part (a) is given as **Theorem 3**. Parts (b) and (c) are given as **Theorem 1**. □

We abuse notation to let $\phi_k(r)$ denote the limit of the sequence in Lemma 3.3.(a) for all $r > 0$, although a priori this sequence may not converge for $r \geq \alpha_k$. Later work refined the value of α_k in Lemma 3.3 for sufficiently large k and so extended the tight bound on $\#F_k(n, rn)$. In particular, Lemma 3.4 implies that $\alpha_k \geq (1 - o_k(1)) \cdot 2^k \ln(k)/k$.

Lemma 3.4. *Let $k \geq 2$. For all $r \geq 0$, if $r \leq (1 - o_k(1)) \cdot 2^k \ln(k)/k$ then $\mathsf{Pr}\left[F_k(n, rn) \textit{ is sat.}\right] \geq 1 - O(1/n)$.*

Proof. The proof of this is given as **Theorem 1.3** of [49]. □

It is difficult to compute $\phi_k(r)$ directly. Instead, Lemma 3.5 provides a weaker but explicit lower bound on $\#F_k(n, rn)$.

Lemma 3.5. *Let $k \geq 3$, $\epsilon > 0$, and $r \geq 0$. Let β_k be the smallest positive solution to $\beta_k(2-\beta_k)^{k-1} = 1$ and define $\Lambda_b(k,r) = 4(((1-\beta_k/2)^k - 2^{-k})^2/(1-\beta_k)^k)^r$.*

If $r < 2^k \ln(2) - \frac{1}{2}((k+1)\ln(2)+3)$, then w.h.p. $\frac{1}{2}(\Lambda_b(k,r)-\epsilon)^{n/2} \leq \#F_k(n,rn)$.

Proof. The proof of this is given on page 264 of [33] within Section 6 (**Proof of Theorem 6**); the definition of $\Lambda_b(k,r)$ is given as equation (20). □

The following two lemmas bound from below the probability that a formula H (in Lemma 3.8 we take $H = F_k(n,rn)$) remains satisfiable after including XOR-clauses on top of H. We first state this bound using notation from [12].

Lemma 3.6. *Let $\epsilon \in (0,\frac{1}{2})$, $c \geq 0$, $\alpha' \in (0,1)$, $f \in (\frac{\log m}{m}(3.6 - \frac{5}{4}\log_2(\alpha')), 1/2]$, $m = \alpha' n$, and let H be a formula defined over $\{X_1, \cdots, X_n\}$. If $\#H = 2^{m+c}$ and $\epsilon < \frac{2^c-1}{2(2^c+1)}$, then $\mathsf{Pr}\left[H \wedge Q^f(n,m) \text{ is satisfiable}\right] \geq \frac{1}{2} + \epsilon$.*

Proof. The proof of this is given in [12] within **Proof of Theorem 2 (part 2)**. □

The following lemma restates Lemma 3.6 using more convenient notation. A similar result (and proof through Chebyshev's inequality) when $p = \frac{1}{2}$ appears in [9].

Lemma 3.7. *Let $p \in (0,1/2]$, and $s \geq 0$. Then there exists some integer $N_{s,p} > 0$ s.t. for all $\alpha \geq 1$, $n \geq N_{s,p}$ and all formulas H defined over $\{X_1, \cdots, X_n\}$, we have*

$$\mathsf{Pr}\left[H \wedge Q^p(n,sn) \text{ is satisfiable} \mid \#H \geq 2^{\lceil sn \rceil + \alpha}\right] \geq \frac{1}{2} + \frac{2^\alpha - 1}{2(2^\alpha+1)}.$$

Proof. If $s = 0$, take $N_{s,p} = 1$ and notice that $H \wedge Q^p(n,0) = H$ for all formulas H. The statement follows easily. If $s \geq 1$, notice that $2^{\lceil sn \rceil + \alpha} \geq 2^n$ and $\#H \leq 2^n$ and so the statement is vacuously true. Otherwise, $s \in (0,1)$. In this case, $\lim\limits_{n\to\infty} \frac{\log(sn)}{sn}(3.6 - \frac{5}{4}\log_2(s)) = 0$ and so there exists some $N_{s,p} > 0$ such that $p > \frac{\log(sn)}{sn}(3.6 - \frac{5}{4}\log_2(s))$ for all $n \geq N_{s,p}$.

Now, let H be an arbitrary boolean formula defined over $\{X_1, \cdots, X_n\}$ such that $\#H \geq 2^{\lceil sn \rceil + \alpha}$. Let $c = \log_2(\#H) - sn$ so that $c \geq \alpha$ and $\#H = 2^{sn+c}$. Consider an arbitrary $\beta > 0$. It follows from Lemma 3.6 (with $\epsilon = \frac{2^c - 1 - \beta}{2(2^c+1)}$, $\alpha' = s$, $f = p$, and c and H as defined above) that $\mathsf{Pr}\left[H \wedge Q^p(n, sn) \text{ is satisfiable}\right] \geq \frac{1}{2} + \frac{2^c - 1 - \beta}{2(2^c+1)}$. Since $c \geq \alpha$ and $\frac{2^x - 1 - \beta}{2(2^x+1)}$ is an increasing function of x, it follows that $\mathsf{Pr}\left[H \wedge Q^p(n, sn) \text{ is satisfiable}\right] \geq \frac{1}{2} + \frac{2^\alpha - 1 - \beta}{2(2^\alpha+1)}$. Finally, we take the limit of this inequality as $\beta \to 0$ to get that $\mathsf{Pr}\left[H \wedge Q^p(n, sn) \text{ is satisfiable}\right] \geq \frac{1}{2} + \frac{2^\alpha - 1}{2(2^\alpha+1)}$ as desired. □

Using the key behavior of XOR-clauses described in Lemma 3.7, we can transform lower bounds (w.h.p.) on the number of solutions to $F_k(n, rn)$ into lower bounds on the location of a possible k-CNF-XOR phase-transition.

Lemma 3.8. *Let $k \geq 2$, $p \in (0, 1/2]$, $s \geq 0$, and $r \geq 0$. Let B_1, B_2, $\cdots$ be an infinite convergent sequence of positive real numbers such that $B_i^n \leq \#F_k(n, rn)$ occurs w.h.p. for all $i \geq 1$. If $s < \log_2(\lim_{i\to\infty} B_i)$, then w.h.p. $\psi_k^p(n, rn, sn)$ is satisfiable.*

Proof. For all integers $n \geq 0$, let the event E_n denote the event when $\psi_k^p(n, rn, sn)$ is satisfiable. We would like to show that $\mathsf{Pr}\left[E_n\right]$ converges to 1 as $n \to \infty$.

The general idea of the proof follows. We first decompose $\psi_k^p(n, rn, sn)$ into $\psi_k^p(n, rn, sn) = F_k(n, rn) \wedge Q^p(n, sn)$. Let the event L_n denote the event when the number of solutions of $F_k(n, rn)$ is bounded from below (by a lower bound to be specified later). We show that L_n occurs w.h.p.. Next, we use Lemma 3.7 to bound from below the probability that $F_k(n, rn) \wedge Q^p(n, sn)$ remains satisfiable given that $F_k(n, rn)$ has enough solutions; we use this to show that $\mathsf{Pr}\left[E_n \mid L_n\right]$ converges to 1 as $n \to \infty$. Finally, we combine these results to prove that $\mathsf{Pr}\left[E_n\right]$ converges to 1.

Since $2^s < \lim_{i\to\infty} B_i$, there is some integer $i \geq 1$ such that $2^s < B_i$. Define the event L_n as the event when $\#F_k(n, rn) \geq B_i^n$. Then L_n occurs w.h.p. by hypothesis.

Next, we show that $\mathsf{Pr}\left[E_n \mid L_n\right]$ converges to 1. Choose $\delta > 0$ and $N > N_{s,p}$ (with $N_{s,p}$ from Lemma 3.7) such that $2^{s+\delta+1/N} < B_i$; we can always find sufficiently small δ and sufficiently large N such that this holds. Since we are concerned only with the behavior of $\mathsf{Pr}\left[E_n \mid L_n\right]$ in the limit, we can restrict our attention only to large enough n. In particular, consider $n > 2N$. Then we get that $2^{sn+\delta n+2} < B_i^n$ and so $2^{\lceil sn \rceil+\delta n+1} < B_i^n$. Let $\alpha = \delta n + 1$, so that $2^{\lceil sn \rceil+\alpha} \leq B_i^n$. Then Lemma 3.7 says that $\mathsf{Pr}\left[E_n \mid L_n\right] \geq \frac{1}{2} + \frac{2^{\delta n+1}-1}{2(2^{\delta n+1}+1)}$. Since $\frac{2^x-1}{2(2^x+1)}$ converges to $\frac{1}{2}$ as $x \to \infty$, we have that $\frac{1}{2} + \frac{2^{\delta n+1}-2}{2(2^{\delta n+1}+1)}$ converges to 1 as $n \to \infty$. Thus $\mathsf{Pr}\left[E_n \mid L_n\right]$ must also converge to 1.

Thus both $\mathsf{Pr}\left[E_n \mid L_n\right]$ and $\mathsf{Pr}\left[L_n\right]$ converge to 1 as $n \to \infty$. Since $\mathsf{Pr}\left[E_n \cap L_n\right] = \mathsf{Pr}\left[E_n \mid L_n\right] \cdot \mathsf{Pr}\left[L_n\right]$, this implies that $\mathsf{Pr}\left[E_n \cap L_n\right]$ also converges to 1. Finally, since $\mathsf{Pr}\left[E_n \cap L_n\right] \leq \mathsf{Pr}\left[E_n\right] \leq 1$, this implies that $\mathsf{Pr}\left[E_n\right]$ converges to 1. □

Finally, it remains only to use Lemma 3.8 to obtain bounds on the k-CNF-XOR phase-transition. The tight lower bound on $\#F_k(n, rn)$ from Lemma 3.3.(b) corresponds to a tight lower bound on the location of the phase-transition.

Lemma 3.9. *Let $k \geq 2$, and let α_k, $\mathcal{C}_k$, and $\phi_k(r)$ be as defined in Lemma 3.3. For all $p \in (0, 1/2]$, $r \in [0, \alpha_k) \backslash \mathcal{C}_k$ and $s \in [0, \phi_k(r))$, $\psi_k^p(n, rn, sn)$ is satisfiable w.h.p..*

Proof. Let $B_i = 2^{\phi_k(r)-1/i}$. By Lemma 3.3.(b), $B_i^n \leq \#F_k(n, rn)$ w.h.p. for all $i \geq 1$. Furthermore, $\lim_{i\to\infty} B_i = 2^{\phi_k(r)}$ and so $s < \log_2(\lim_{i\to\infty} B_i)$. Thus $\psi_k^p(n, rn, sn)$ is satisfiable w.h.p. by Lemma 3.8. □

The weaker lower bound on $\#F_k(n, rn)$ from Lemma 3.5 corresponds to a weaker lower bound on the location of the phase-transition.

Lemma 3.10. *Let $k \geq 3$, $p \in (0, 1/2]$, $s \geq 0$, and $r \geq 0$. If $r < 2^k \ln(2) - \frac{1}{2}(k + 1)\ln(2) + \frac{3}{2}$ and $s < \frac{1}{2}\log_2(\Lambda_b(k, r))$, then $\psi_k^p(n, rn, sn)$ is satisfiable w.h.p..*

Proof. Let $B_i = (\Lambda_b(k,r) - 1/i)^{1/2}$. This is an increasing sequence in i and so $\log_2(B_{i+1}/B_i)$ is positive for all $i \geq 1$. Consider one such $i \geq 1$ and define $N_i = 1/\log_2(B_{i+1}/B_i)$. Then for all $n > N_i$ it follows that $2^{1/n} < B_{i+1}/B_i$ and so $B_i^n < \frac{1}{2}B_{i+1}^n$. By Lemma 3.5, $\frac{1}{2}B_{i+1}^n \leq \#F_k(n, rn)$ w.h.p. and therefore $B_i^n < \frac{1}{2}B_{i+1}^n \leq \#F_k(n, rn)$ w.h.p. as well.

Furthermore, $\lim_{i\to\infty} B_i = \Lambda_b(k,r)^{1/2}$ and so $s < \log_2(\lim_{i\to\infty} B_i)$. It follows that $\psi_k^p(n, rn, sn)$ is satisfiable w.h.p. by Lemma 3.8. □

3.3.2 A Proof of the Upper Bound

We now establish Theorem 3.1.(b) and Theorem 3.2.(b), which follow directly from Lemma 3.14 and Lemma 3.15 respectively.

Similar to Section 3.3.1, the key idea in the proof of these lemmas is to decompose $\psi_k^p(n, rn, sn)$ into independently generated k-CNF and XOR formulas, so that $\psi_k^p(n, rn, sn) = F_k(n, rn) \wedge Q^p(n, sn)$. We can then bound the number of solutions to $F_k(n, rn)$ from above with high probability and bound from below the probability that $F_k(n, rn)$ becomes unsatisfiable after including XOR-clauses on top of $F_k(n, rn)$.

The first of these two tasks is accomplished through Lemma 3.3.(c), which gives a tight upper bound on $\#F_k(n, rn)$ for small k-clause densities, and by Lemma 3.11, which gives a weaker explicit upper bound on $\#F_k(n, rn)$.

Lemma 3.11. *For all $\epsilon > 1$, $k \geq 2$, and $r \geq 0$, w.h.p. $\#F_k(n, rn) < (2\epsilon \cdot (1 - 2^{-k})^r)^n$.*

Proof. Let $X = \#F_k(n, rn)$. For a random assignment on n variables σ, note that $\mathsf{Pr}\left[\sigma \text{ satisfies } F_k(n, 1)\right] = (1 - 2^{-k})$. Since the $\lceil rn \rceil$ k-clauses of $F_k(n, rn)$ were chosen independently, this implies that $\mathsf{E}\left[X\right] = 2^n(1 - 2^{-k})^{\lceil rn \rceil}$.

By Markov's inequality, we get $\mathsf{Pr}\left[X \geq \epsilon^n \mathsf{E}\left[X\right]\right] \leq \mathsf{E}\left[X\right]/(\epsilon^n \mathsf{E}\left[X\right]) = \epsilon^{-n}$. Since $1 - 2^{-k} < 1$ and so $\epsilon^n \mathsf{E}\left[X\right] = 2^n \epsilon^n (1 - 2^{-k})^{\lceil rn \rceil} \leq 2^n \epsilon^n (1 - 2^{-k})^{rn}$, it follows that

$\mathsf{Pr}\left[X \geq \epsilon^n 2^n (1-2^{-k})^{rn}\right] \leq \mathsf{Pr}\left[X \geq \epsilon^n \mathsf{E}\left[X\right]\right] \leq \epsilon^{-n}$. Since $\epsilon > 1$, this implies that $\lim_{n\to\infty} \mathsf{Pr}\left[X < \epsilon^n 2^n (1-2^{-k})^{rn}\right] = 1$ as desired. □

The following lemma bounds from below the probability that a formula H (in Lemma 3.13 we take $H = F_k(n, rn)$) becomes unsatisfiable after including XOR-clauses on top of H. This result and proof is similar to Corollary 1 from [9].

Lemma 3.12. *Let $\alpha \geq 1$, $p \in (0, 1/2]$, $s \geq 0$, $n \geq 0$, and let H be a formula defined over $X = \{X_1, \cdots, X_n\}$. Then $\mathsf{Pr}\left[H \wedge Q^p(n, sn) \text{ is unsatisfiable} \mid \#H \leq 2^{\lceil sn \rceil - \alpha}\right] \geq 1 - 2^{-\alpha}$.*

Proof. Let R be the set of all truth assignments to the variables in X that satisfy H; there are $\#H$ such truth assignments. For every truth assignment $\sigma \in R$, let Y_σ be a 0-1 random variable that is 1 if σ satisfies $H \wedge Q^p(n, sn)$ and 0 otherwise. Note that $\mathsf{Pr}\left[\sigma \text{ satisfies } Q^p(n, 1)\right] = 1/2$ (since for all possible sets of variables $A \subseteq X$, $\mathsf{Pr}\left[\sigma \text{ satisfies } Q^p(n, 1) \mid \text{the variables of } Q^p(n, 1) \text{ are exactly } A\right] = 1/2$). Since the $\lceil sn \rceil$ XOR-clauses of $Q^p(n, sn)$ were chosen independently, this implies that $\mathsf{E}\left[Y_\sigma\right] = \mathsf{Pr}\left[\sigma \text{ satisfies } Q^p(n, sn)\right] = 2^{-\lceil sn \rceil}$.

Let the random variable Y be the number of solutions to $H \wedge Q^p(n, sn)$, so $Y = \#(H \wedge Q^p(n, sn)) = \sum_\sigma Y_\sigma$. Thus $\mathsf{E}\left[Y\right] = \sum_\sigma \mathsf{E}\left[Y_\sigma\right] = \#H \cdot 2^{-\lceil sn \rceil}$.

Markov's inequality implies that $\mathsf{Pr}\left[Y \geq 1\right] \leq \mathsf{E}\left[Y\right]$, so $\mathsf{Pr}\left[Y \geq 1\right] \leq \#H \cdot 2^{-\lceil sn \rceil}$. If $\#H \leq 2^{\lceil sn \rceil - \alpha}$, then $\#H \cdot 2^{-\lceil sn \rceil} \leq 2^{-\alpha}$. Thus $\mathsf{Pr}\left[Y \geq 1 \mid \#H \leq 2^{\lceil sn \rceil - \alpha}\right] \leq 2^{-\alpha}$. Since $H \wedge Q^p(n, sn)$ is unsatisfiable exactly when $Y = 0$, we conclude that $\mathsf{Pr}\left[H \wedge Q^p(n, sn) \text{ is unsatisfiable}\right] \geq 1 - 2^{-\alpha}$. □

Using the key behavior of XOR-clauses described in Lemma 3.12, we can transform upper bounds (w.h.p.) on the number of solutions to $F_k(n, rn)$ into upper bounds on the location of a possible k-CNF-XOR phase-transition.

Lemma 3.13. *Let $k \geq 2$, $p \in (0, 1/2]$, $s \geq 0$, and $r \geq 0$. Let B_1, B_2, $\cdots$ be an infinite convergent sequence of positive real numbers such that $\#F_k(n, rn) \leq B_i^n$ occurs w.h.p. for all $i \geq 1$. If $s > \log_2(\lim_{i\to\infty} B_i)$, then w.h.p. $\psi_k^p(n, rn, sn)$ is unsatisfiable.*

Proof. For all integers $n \geq 0$, let the event $\neg E_n$ denote the event when $\psi_k^p(n, rn, sn)$ is unsatisfiable. We would like to show that $\mathsf{Pr}\left[\neg E_n\right]$ converges to 1 as $n \to \infty$.

The general idea of the proof follows. Note that $\psi_k^p(n, rn, sn) = F_k(n, rn) \wedge Q^p(n, sn)$ as in Lemma 3.8. Let the event U_n denote the event when the number of solutions of $F_k(n, rn)$ is bounded from above (by an upper bound to be specified later). We show that U_n occurs w.h.p.. Next, we use Lemma 3.12 to bound from below the probability that $F_k(n, rn) \wedge Q^p(n, sn)$ becomes unsatisfiable given that $F_k(n, rn)$ has few solutions; we use this to show that $\mathsf{Pr}\left[\neg E_n \mid U_n\right]$ converges to 1 as $n \to \infty$. Finally, we combine these results to prove that $\mathsf{Pr}\left[\neg E_n\right]$ converges to 1.

Since $2^s > \lim_{i\to\infty} B_i$, there is some integer $i \geq 1$ such that $2^s > B_i$. Define the event U_n as the event when $\#F_k(n, rn) \leq B_i^n$. Then U_n occurs w.h.p. by hypothesis.

Next, we show that $\mathsf{Pr}\left[\neg E_n \mid U_n\right]$ converges to 1. Choose $\delta > 0$ and $N > 0$ such that $2^{s-\delta-1/N} > B_i$. As in Lemma 3.8 we are concerned only with the behavior of $\mathsf{Pr}\left[\neg E_n \mid U_n\right]$ in the limit so we can restrict our attention only to large enough n. In particular, consider $n > N$. Then we get that $2^{\lceil sn\rceil - \delta n - 1} > 2^{sn - \delta n - n/N} > B_i^n$. Let $\alpha = \delta n + 1$, so that $2^{\lceil sn\rceil - \alpha} \geq B_i^n$. Then Lemma 3.12 says that $\mathsf{Pr}\left[\neg E_n \mid U_n\right] \geq 1 - 2^{-\delta n - 1}$. Since $1 - 2^{-\delta n - 1}$ converges to 1 as $n \to \infty$, $\mathsf{Pr}\left[\neg E_n \mid U_n\right]$ must also converge to 1.

Therefore both $\mathsf{Pr}\left[\neg E_n \mid U_n\right]$ and $\mathsf{Pr}\left[U_n\right]$ converge to 1 as $n \to \infty$. This implies that $\mathsf{Pr}\left[\neg E_n \cap U_n\right] = \mathsf{Pr}\left[\neg E_n \mid U_n\right] \cdot \mathsf{Pr}\left[U_n\right]$ also converges to 1. Since $\mathsf{Pr}\left[\neg E_n \cap U_n\right] \leq \mathsf{Pr}\left[\neg E_n\right] \leq 1$, this implies that $\mathsf{Pr}\left[\neg E_n\right]$ converges to 1 as desired. □

Finally, it remains only to use Lemma 3.13 to obtain bounds on the k-CNF-

XOR phase-transition. The tight upper bound on $\#F_k(n, rn)$ from Lemma 3.3.(c) corresponds to a tight upper bound on the location of the phase-transition.

Lemma 3.14. *Let $k \geq 2$, and let α_k, $\mathcal{C}_k$, and $\phi_k(r)$ be as defined in Lemma 3.3. Then for all $p \in (0, 1/2]$, $r \in [0, \alpha_k) \backslash \mathcal{C}_k$ and $s > \phi_k(r)$, $\psi_k^p(n, rn, sn)$ is unsatisfiable w.h.p..*

Proof. Let $B_i = 2^{\phi_k(r)+1/i}$. By Lemma 3.3.(c), $B_i^n \geq \#F_k(n, rn)$ w.h.p. for all $i \geq 1$. Furthermore, $\lim_{i\to\infty} B_i = 2^{\phi_k(r)}$ and so $s > \log_2(\lim_{i\to\infty} B_i)$. Thus $\psi_k^p(n, rn, sn)$ is unsatisfiable w.h.p. by Lemma 3.13. □

The weaker upper bound on $\#F_k(n, rn)$ from Lemma 3.11 corresponds to a weaker upper bound on the phase-transition.

Lemma 3.15. *Let $k \geq 2$, $p \in (0, 1/2]$, $s \geq 0$, and $r \geq 0$. If $s > 1 + r\log_2(1 - 2^{-k})$, then $\psi_k^p(n, rn, sn)$ is unsatisfiable w.h.p..*

Proof. Let $B_i = ((1+1/i) \cdot 2(1-2^{-k})^r)$. By Lemma 3.11, $B_i^n \geq \#F_k(n, rn)$ w.h.p. for all $i \geq 1$. Furthermore, $\lim_{i\to\infty} B_i = 2(1 - 2^{-k})^r$ and so $s > \log_2(\lim_{i\to\infty} B_i)$. Thus $\psi_k^p(n, rn, sn)$ is unsatisfiable w.h.p. by Lemma 3.13. □

3.4 Extending the Phase-Transition Region

Section 3.3 proved that a phase-transition exists for k-CNF-XOR formulas when the k-clause density is small. Our empirical observations in Section 3.2 suggest that a phase-transition exists for higher k-clause densities as well. In this section, we conjecture two possible extensions to our theoretical results.

The first extension follows from Theorem 3.1, which implies that $s = \phi_k(r)$ gives the location of the phase-transition for small k-clause densities. It is thus natural to conjecture that $\phi_k(r)$ gives the location of the k-CNF-XOR phase-transition for all (except perhaps countably many) $r > 0$. This would follow from a conjecture of [48].

The second extension follows from the experimental results in Section 3.2, which suggest that the location of the phase-transition follows a linear trade-off between k-clauses and XOR-clauses. This leads to the following conjecture:

Conjecture 2 (k-CNF-XOR Linear Phase-Transition Conjecture)**.** *Let $k \geq 2$. Then there exists a slope $L_k < 0$ and a constant $\alpha_k^* > 0$ such that for all $p \in (0, 1/2]$, $r \in [0, \alpha_k^*)$, and $s \geq 0$:*

(a). If $s < rL_k + 1$, then w.h.p. $\psi_k^p(n, rn, sn)$ is satisfiable.

(b). If $s > rL_k + 1$, then w.h.p. $\psi_k^p(n, rn, sn)$ is unsatisfiable.

Theorem 3.2 bounds the possible values for L_k. Moreover, if the Linear k-CNF-XOR Phase-Transition Conjecture holds, then Theorem 3.1 implies that $\phi_k(r)$ is linear for all $r < \alpha_k$ and $r < \alpha_k^*$. Explicit computations of $\phi_k(r)$ (or sufficiently tight bounds) would resolve this conjecture.

Note that this conjecture does not necessarily describe the entire k-CNF-XOR phase-transition; a phase-transition may exist when $r > \alpha_k^*$ as well. The experimental results in Section 3.2 for $k = 2$ and $k = 3$ suggest that the location of the phase-transition may "kink" and become non-linear for large enough k-clause densities. We leave the full characterization of the k-CNF-XOR phase-transition for future work, noting that a full characterization would resolve the Satisfiability Phase-Transition Conjecture.

Chapter 4

Runtime Scaling Behavior

In this chapter, we explicitly study the runtime of CryptoMiniSAT, a specialized CNF-XOR solver, on random k-CNF-XOR formulas.

As introduced in Chapter 1, the runtime of SAT solvers (using DPLL and related algorithms) on random fixed-width CNF formulas (where each clause contains a fixed number of literals) was shown to follow an *easy-hard-easy* pattern [25]: the runtime is low when the clause density is very low or very high and peaks near the phase-transition point. Further analysis of the relationship between the clause density and SAT solver runtime revealed a more nuanced picture of the scaling behavior of SAT solvers on random k-CNF instances: a secondary phase-transition was observed within the satisfiable region, where the median runtime transitions from polynomial to exponential in the number of variables [30].

Theoretical analysis of this phenomenon [31–33] has shown that the solution space of a random fixed-width CNF formula undergoes a dramatic 'shattering'. When the clause density is small, almost all solutions are contained in a single connected-component (where solutions are adjacent if their Hamming distance is 1). Above a specific clause density the solution space 'shatters' into exponentially many connected-components. Moreover, these clusters are with high probability all linearly separated i.e. the Hamming distance between all pairs of connected-components is bounded from below by some function linear in the number of variables. This 'shattering' of the solution space into linearly separated solutions is known to be difficult for a variety of SAT-solving algorithms [34, 35].

In Chapter 3, we identified the phase-transition location for random k-CNF-XOR formulas. The next step towards explaining the runtime behavior of CNF-XOR solvers in practice (and thus explaining the runtime behavior of hashing-based algorithms), then, is the analysis of the scaling behavior of CNF-XOR solvers on random k-CNF-XOR formulas.

For example, it is widely believed that the performance of CNF-XOR solvers on CNF-XOR formulas depends on the width of the XOR-clauses. Consequently, recent efforts [50, 51] have focused on designing hashing-based techniques that employ XOR-clauses of smaller width. In this chapter, we use our framework of random k-CNF-XOR formulas to present empirical evidence that using smaller width XOR-clauses does not necessarily improve the scaling behavior of CNF-XOR solvers.

In the remainder of this chapter, we present the first study of the runtime behavior of CNF-XOR solvers on random k-CNF-XOR formulas and on the solution space of random k-CNF-XOR formulas. In particular:

(a). We present (in Section 4.2) experimental evidence that the runtime of the CNF-XOR solver CryptoMiniSAT scales exponentially in the number of variables at many k-clause and XOR-clause densities well within the satisfiable region, even when both the CNF and XOR subformulas are separately solvable in polynomial time by CryptoMiniSAT.

(b). We present (in Section 4.3) experimental evidence that this exponential scaling peaks around the empirical phase-transition location for random k-CNF-XOR formulas, and further that the scaling behavior does *not* monotonically improve as the XOR-clauses get shorter.

(c). We hypothesize (in Section 4.4) that this exponential scaling behavior within the satisfiable region is caused by the shattering of the solution space of random k-CNF-XOR formulas. We use recent theoretical results from the field of constrained counting [12] to prove that the solution space of random variable-width XOR formulas (and therefore of random k-CNF-XOR formulas) shatters.

4.1 Experimental Setup

We used the experimental setup described in Section 2.5 to explore empirically the runtime behavior of CryptoMiniSAT on random k-CNF-XOR formulas. In particular, the objective of the experimental setup was to empirically determine the scaling behavior, as a function of n, in the median runtime of checking satisfiability of $\psi_k^p(n, rn, sn)$ with respect to r (the k-clause density), s (the XOR-clause density), and p (the XOR variable-probability) for fixed k.

In all experiments we fix the clause length $k = 3$. The 3-clause density r, the XOR-clause density s, and the XOR variable-probability p varied in each experiment, as follows:

- To study the effect of the 3-clause and XOR-clause densities on the runtime, we ran 124 experiments with $r \in \{1, 2, 3, 4\}$, $p = 1/2$, and s ranging from 0.3 to 0.9 in increments of 0.02. We present selected results from these experiments in Section 4.2.

- To study the effect of the XOR variable-probability on the runtime, we ran 679 experiments with $r = 2$, p ranging from 0.02 to 0.5 in increments of 0.005, and s ranging from 0.3 to 0.9 in increments of 0.1. We chose these clause-densities so that approximately half of the clause-densities were in the satisfiable region.

We present selected results from these experiments in Section 4.3.

To determine the scaling behavior of CryptoMiniSAT on random k-CNF-XOR formulas with parameters k, r, s, and p, we determined a number of variables N so that the median runtime of CryptoMiniSAT on $\psi_k^p(N, rN, sN)$ was as large as possible while remaining below the set formula timeout. We then allowed n to range from 10 to N in increments of 1, and estimated the median solve time of CryptoMiniSAT on $\psi_k^p(n, rn, sn)$ using the experimental setup described in Section 2.5. Finally, we used the curve_fit function in the Python scipy.optimize* library to determine the relationship between the number of variables n and the medium runtime of CryptoMiniSAT on $\psi_k^p(n, rn, sn)$. We attempted to fit linear $(an + b)$, quadratic $(an^2 + bn + c)$, cubic $(an^3 + bn^2 + cn + d)$, and exponential $(\beta 2^{\alpha n})$ curves; the best-fit curve was the curve with the smallest mean squared error.

All formula were given a timeout of 5 seconds. We were not able to run informative experiments for formulas with higher timeouts; as the runtime of CryptoMiniSAT increases past 5 seconds, the variance in runtime significantly increases as well and so experiments require a number of trials at each data point far beyond our computational abilities.

4.2 Experimental Results on XOR-clause density

We analyzed the median runtime of CryptoMiniSAT on $\psi_3^{1/2}(n, rn, sn)$ for a fixed r and s as a function of the number of variables n. We present here results only for the experiments with $r \in \{2, 3\}$ †.

*https://www.scipy.org/

†The data from all experiments is available at http://www.cs.rice.edu/CS/Verification/Projects/CUSP/

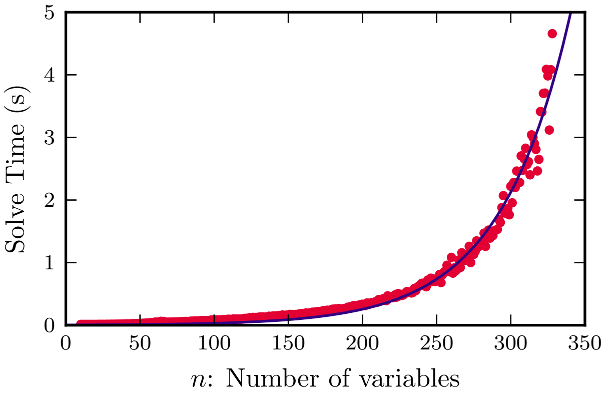


Figure 4.1 : Runtime for 3-CNF-XOR formulas at 3-clause density $r = 2$, XOR-clause density $s = 0.3$, and XOR variable-probability $p = 1/2$, together with the best-fit curve $0.00370 \cdot 2^{0.0305n}$.

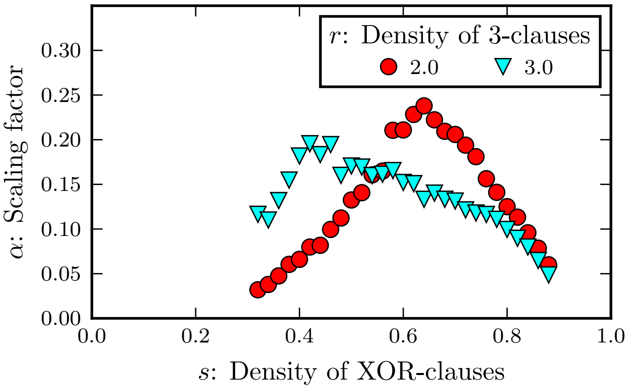


Figure 4.2 : Exponential scaling factor for 3-CNF-XOR formulas with 3-clause density $r = 2$ and 3 and XOR variable-probability $p = 1/2$. The scaling factor α is the exponent of the best-fit line for the runtime of $\psi_3^{1/2}(n, rn, sn)$.

Figure 4.1 plots the median runtime at $k = 3$, $r = 2$, and $s = 0.3$ as a function of n, together with the best-fit curve. The x-axis indicates the number of variables n. The y-axis indicates the median runtime of CryptoMiniSAT on $\psi_3^{1/2}(n, 2n, 0.3n)$. We observe that the median runtime increases exponentially in the number of variables. In this case, the best-fit curve is the exponential function $0.0370 \cdot 2^{0.0305n}$.

In fact, for all experiments with $r = 2, 3$ and $0.3 \leq s \leq 0.9$ the best-fit curve to the median runtime as a function of n is proportional to an exponential function of the form $2^{\alpha n}$ for some $\alpha > 0$. Figure 4.2 plots the scaling behavior with respect to n of the median runtime of CryptoMiniSAT on both $\psi_3^{1/2}(n, 2n, sn)$ and $\psi_3^{1/2}(n, 3n, sn)$. The x-axis indicates the density of XOR-clauses s. The legend indicates the density of 3-clauses r. The value α, known as the *scaling factor*, shown on the y-axis indicates that the best-fit curve to the median runtime of $\psi_3^p(n, rn, sn)$ as a function of n was proportional to $2^{\alpha n}$. We observe that the scaling factor is closely related to the 3-clause density and the XOR-clause density: when the XOR-clause density is low or high the scaling factor is low, and the scaling factor peaks at some intermediate value. When $r = 2$, we observe that this peak in the scaling factor occurs when $s \in (0.6, 0.7)$. When $r = 3$, we observe that this peak in the scaling factor is near $s = 0.4$. As seen in Chapter 3, there is a phase-transition in the satisfiability of random 3-CNF-XOR formulas near $r = 2$, $s = 0.65$ and near $r = 3$, $s = 0.4$. Thus we observe a peak in the runtime scaling factor around the 3-CNF-XOR phase-transition, similar to the peak observed in the runtime factor for $F_k(n, rn)$ around the k-CNF phase-transition [30].

Our experimental results do not describe extremely low 3-clause densities and XOR-clause densities (for example, when the XOR-clause density is below 0.3). At such low densities, conclusive evidence of polynomial or exponential behavior requires computational power beyond our capabilities.

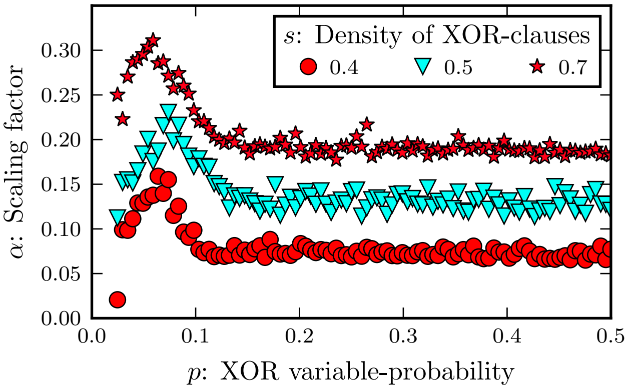


Figure 4.3 : Exponential scaling factor for 3-CNF-XOR formulas with 3-clause density $r = 2$ and XOR-clause density $s = 0.4, 0.5,$ and 0.7.

4.3 Experimental Results on XOR-clause width

We next analyzed the median runtime of CryptoMiniSAT on $\psi_3^p(n, 2n, sn)$ for a fixed p and s as a function of the number of variables n. We present here results only for the experiments with $s \in \{0.4, 0.5, 0.7\}$ [‡].

Figure 4.3 plots the scaling behavior with respect to n of the medium runtime of CryptoMiniSAT on $\psi_3^p(n, 2n, sn)$. The x-axis indicates the XOR variable-probability p. The legend indicates the density of XOR-clauses s. The value α shown on the y-axis indicates that the best-fit curve to the median runtime of $\psi_3^p(n, 2n, sn)$ as a function of n was proportional to $2^{\alpha n}$. In all cases, we observe that the behavior of the scaling factor is independent of the XOR variable-probability, p, when $p \in (0.15, 0.5)$. As the XOR variable-probability decreases further, the scaling factor increases to a peak when $p \in (0.05, 0.1)$, then decreases.

[‡] The data from all experiments is available at `http://www.cs.rice.edu/CS/Verification/Projects/CUSP/`

In summary, we observe that the runtime of CryptoMiniSAT scales exponentially in the number of variables on random 3-CNF-XOR formulas across a wide range of densities and XOR variable-probabilities. The exponential scaling behavior peaks near the empirical location of the 3-CNF-XOR phase-transition. The exponential scaling behavior is constant when the XOR variable-probability is above $p = 0.15$ and the scaling behavior peaks when the XOR variable-probability is between 0.05 and 0.1, independent of the XOR-clause density.

4.4 The Separation of the XOR Formula Solution Space

In the case of k-CNF formulas, the exponential runtime scaling of DPLL-solvers (in the satisfiable region) is closely connected to the 'shattering' of the solution space into exponentially many $\Omega(n)$-separated clusters w.h.p. [34,35]. Prior work has shown that the solution space of fixed-width XOR-clauses has similar behavior. Unfortunately, the proof techniques used for fixed-width XOR-clauses do not easily extend to the solution space of $Q^p(n, sn)$. In particular, the proof techniques for XOR-clauses of fixed-width ℓ heavily involve properties of either random ℓ-uniform hypergraphs [36] or random factor graphs with factors of constant degree ℓ [43]. If the width of each XOR-clause is stochastic, as in $Q^p(n, sn)$, rather than fixed, the corresponding hypergraphs are no longer uniform and the corresponding factor graphs no longer have factors of constant degree.

Nevertheless, we show in Theorem 4.1 that all solutions of a random XOR-formula are w.h.p. $\Omega(n)$-separated (as long as the variable-probability decreases slowly enough as a function of n). This is a stronger separation than the separation seen in the case of k-CNF formulas and fixed-width XOR-formulas, where there may be clusters of nearby solutions.

Theorem 4.1 (XOR Shattering Theorem)**.** *Let $s \in (0, 1)$, $\rho > 2$, and $f(n)$ be a non-negative function. If $\rho \frac{\log(sn)}{sn} \leq f(n) \leq 1/2$ for all large enough n, then $Q^{f(n)}(n, sn)$ is w.h.p. $\Omega(n)$-separated.*

Proof. This follows directly from Lemma 4.6. The proof of this lemma appears in Section 4.4.1. □

Notice that Theorem 4.1 allows the XOR variable-probability to depend on the number of variables. In particular, the XOR variable-probability can decrease as a function of n. Theorem 4.1 does not characterize the solution space of XOR-formulas when the variable-probability decreases faster than $2\frac{\log(sn)}{sn}$ as a function of n. It is possible that the solution space is still $\Omega(n)$-separated in this case, or that clusters of solutions can be found. We leave this for future work.

In Section 4.2 and Section 4.3, we focused on an XOR variable-probability model that is independent of n; this XOR variable-probability is an important special case of the above general theorem. In particular, if the XOR variable-probability is some constant $p \in (0, 1/2]$ then the solution space of a random XOR-formula with variable-probability p is $\Omega(n)$-separated. We highlight this fact as Corollary 4.2.

Corollary 4.2. *For all $s \in (0, 1)$ and $p \in (0, 1/2]$, $Q^p(n, sn)$ is w.h.p. $\Omega(n)$-separated.*

Proof. This follows from Theorem 4.1 with $f(n) = p$. □

Notice that Corollary 4.2 also implies that $\psi_k^p(n, rn, sn) = F_k(n, rn) \wedge Q^p(n, sn)$ is w.h.p. $\Omega(n)$-separated.

The experimental results presented in Section 4.2 suggest that SAT solvers scale exponentially in the number of variables on certain densities of random k-CNF-XOR formulas. Moreover, the experimental results presented in Section 4.2 suggest that this exponential scaling occurs at all observed XOR variable-probabilities. Corollary 4.2 implies that the solution space of $\psi_k^p(n, rn, sn)$ is $\Omega(n)$-separated at all XOR-clause densities and XOR variable-probabilities. Since the separation of the k-CNF solution space is closely connected to the exponential scaling of DPLL-solvers, we hypothesize that the exponential scaling of CryptoMiniSAT we observed at many XOR-clause densities and XOR variable-probabilities is closely connected to the $\Omega(n)$-separation of k-CNF-XOR formulas at all nonzero XOR-clause densities and XOR variable-probabilities.

4.4.1 A Proof of the Separation

In this section we establish Theorem 4.1, which follows directly from Lemma 4.6. To do this, notice that if two solutions of $Q^p(n, sn)$ differ exactly on a set of variables A then every XOR-clause in $Q^p(n, sn)$ must contain an even number of variables from A. We bound from above the probability, for a fixed set of variables A, that a random XOR-clause chosen with variable-probability p contains an even number of variables from A. By summing this bound across all sets containing no more than λn variables for some constant λ, we obtain a bound on the probability that two solutions to $Q^p(n, sn)$ differ in no more than λn variables.

The following lemma presents an elementary result in probability theory. We use this result in Lemma 4.5 to bound the probability that a random XOR-clause chosen with variable-probability p has an even number of variables from a set A.

Lemma 4.3. *Let N be a positive integer and let p be a real number with $0 \leq p \leq 1$. If $B_1, B_2, \cdots, B_N$ are independent Bernoulli random variables with parameter p, then* $\mathsf{Pr}\left[\sum_{1 \leq i \leq n} B_i \text{ is even}\right] = \frac{1}{2} + \frac{1}{2}(1-2p)^N$.

Proof. Fix $p \in [0, 1]$. For all $N \geq 0$, let a_N be the probability that the sum of n

independent Bernoulli random variables with parameter p is even. Then $a_0 = 1$ and $a_N = (1-p)a_{N-1} + p(1-a_{N-1}) = p + a_{N-1} - 2pa_{N-1}$ for all $N \geq 1$. It follows that $a_N = \frac{1}{2} + \frac{1}{2}(1-2p)^N$. □

We will ultimately require that the sum of these probabilities across many sets of variables A is sufficiently small. In particular, the following lemma shows that the sum of these probabilities across all sets whose size is smaller than λn goes to 0 in the limit as $n \to \infty$ when the XOR variable-probability is proportional to $\log(sn)/(sn)$.

Lemma 4.4. *Let $\alpha, \delta \in (0,1)$, $m = \alpha n$, $\kappa > -\frac{\log(2/(1+\delta)-1)}{\log(1+\delta)}$ and $\lambda^* < 1/2$ such that $-\lambda^* \log(\lambda^*) - (1-\lambda^*)\log(1-\lambda^*) = \alpha \log(1+\delta)$. Then for all $\lambda < \lambda^*$:*

$$\lim_{n\to\infty} \sum_{w=1}^{\lambda n} \binom{n}{w} \left(\frac{1}{2} + \frac{1}{2}\left(1 - 2\kappa\frac{\log m}{m}\right)^w\right)^m = 0$$

Proof. The proof of this is given as **Lemma 7** of [12]. □

The following lemma allows us to show that the XOR solution-space is $g(n)$-separated if the XOR variable-probability is $f(n)$ for some functions f and g provided that the sum of probability of all sets of variables whose size is below $g(n)$ goes to 0. In particular, in Lemma 4.6 we use this lemma with $f(n) \propto \log(sn)/(sn)$ and $g(n) \in \Omega(n)$ to show that the solution-space of $Q^{f(n)}(n, sn)$ is $\Omega(n)$-separated.

Lemma 4.5. *Let $f(n)$ and $g(n)$ be nonnegative functions with $f(n) \leq 1/2$ for all sufficiently large n. If*

$$\lim_{n\to\infty} \sum_{w=1}^{g(n)} \binom{n}{w} \left(\frac{1}{2} + \frac{1}{2}\left(1 - 2f(n)\right)^w\right)^{sn} = 0$$

then w.h.p. all solutions of $Q^{f(n)}(n, sn)$ are $g(n)$-separated.

Proof. Let the random variable D be 1 if $Q^{f(n)}(n, sn)$ has two solutions with a Hamming distance less than or equal to $g(n)$ and 0 otherwise. We would like to prove that $\lim_{n\to\infty} \mathsf{Pr}\left[D = 1\right] = 0$.

For all nonempty subsets of variables $A \subseteq X$, let the random variable $D(A)$ be 1 if $Q^p(n, sn)$ has a pair of solutions that differ exactly on the variables of A and 0 otherwise. Then $D(A) = 1$ if and only if each XOR-clause in Q contains an even number of variables from A. Moreover, let $\mathcal{B}$ be the set of all subsets of variables $A \subseteq X$ s.t. $0 < |A| \leq g(n)$ and notice that $D \leq \sum_{A\in\mathcal{B}} D(A)$. Thus $\mathsf{Pr}\left[D = 1\right] \leq \sum_{A\in\mathcal{B}} \mathsf{Pr}\left[D(A) = 1\right]$.

Fix $A \subseteq X$ and let Q_1 be a random XOR-clause chosen with variable-probability $f(n)$. Enumerate the $|A|$ variables in A as $Y_1, Y_2, \cdots, Y_{|A|}$. Then for all $1 \leq i \leq |A|$ we can define a random variable B_i that is 1 if the variable Y_i appears in Q_1 and is 0 otherwise. Notice that each B_i is an independent Bernoulli random variable with parameter $f(n)$, and further that the number of variables from A contained in Q_1 is exactly $\sum_{i=1}^{|A|} B_i$. By Lemma 4.3 it follows that the probability that Q_1 contains an even number of variables from A is $\frac{1}{2} + \frac{1}{2}(1 - 2f(n))^{|A|}$.

Since all $\lceil sn \rceil$ XOR-clauses of $Q^p(n, sn)$ are chosen independently with variable-probability $f(n)$, it follows that $\mathsf{Pr}\left[D(A)\right] = (1/2 + (1 - 2f(n))^{|A|}/2)^{\lceil sn \rceil}$. For all sufficiently large n, $f(n) \leq 1/2$ and so $1/2 + (1 - 2f(n))^{|A|}/2 \leq 1$. Thus $\mathsf{Pr}\left[D(A)\right] \leq (1/2 + (1 - 2f(n))^{|A|}/2)^{sn}$ for all sufficiently large n.

Finally, notice that there are exactly $\binom{n}{w}$ sets in $\mathcal{B}$ of size $w \leq g(n)$ and so $\mathsf{Pr}\left[D = 1\right] \leq \sum_{A\in\mathcal{B}}(1/2 + (1 - 2f(n))^{|A|}/2)^{sn} = \sum_{w=1}^{g(n)} \binom{n}{w}(1/2 + (1 - 2f(n))^{w}/2)^{sn}$. By hypothesis, this implies that $\lim_{n\to\infty} \mathsf{Pr}\left[D = 1\right] = 0$. □

The following lemma combines Lemma 4.4 and Lemma 4.5 to show that a variable-probability above $2\log(sn)/(sn)$ implies $\Omega(n)$-separation. This finishes the proof of

Theorem 4.1.

Lemma 4.6. *Let s and ρ be real numbers such that $0 < s \leq 1$ and $\rho > 2$. If $f(n)$ is a nonnegative function such that $\rho\frac{\log(sn)}{sn} \leq f(n) \leq 1/2$ for all sufficiently large n, then $Q^{f(n)}(n, sn)$ is w.h.p. $\Omega(n)$-separated.*

Proof. Let $a(x) = -\log(2/(1+x)-1)/\log(1+x)$. Notice that $\lim_{x\to 0} a(x) = 2$, $\lim_{x\to 1} a(x) = \infty$, and $a(x)$ is continuous on $(0,1)$. Since $2 < \rho < \infty$, it follows that there is some $\delta \in (0,1)$ with $a(\delta) < \rho$.

Let $H(x) = -x\log(x) - (1-x)\log(1-x)$. Notice that $H(0) = 0$, $H(1/2) = 1$, and H is monotonically increasing on $[0, 1/2]$. Since $0 < s\log(1+\delta) < 1$, it follows that there is some $\lambda^* \in (0, 1/2)$ with $H(\lambda^*) = s\log(1+\delta)$. Define $\widehat{f}(n) = \rho\frac{\log(sn)}{sn}$ and $g(n) = n\lambda^*/2$.

Then by Lemma 4.4 with $\alpha = s$, $\kappa = \rho$, and $\lambda = \lambda^*/2$ (and with δ and λ^* as defined above) we have that $\lim_{n\to\infty}\sum_{w=1}^{g(n)}\binom{n}{w}(1/2 + (1-2\widehat{f}(n))^w/2)^{sn} = 0$.

Notice that $\widehat{f}(n) \leq f(n)$ for all sufficiently large n. It follows that $1 - 2\widehat{f}(n) \geq 1 - 2f(n)$ and so $(1-2\widehat{f}(n))^w \geq (1-2f(n))^w$ for all $w \geq 1$ and for all sufficiently large n. Therefore $\binom{n}{w}(1/2+(1-2\widehat{f}(n))^w/2)^{sn} \geq \binom{n}{w}(1/2+(1-2f(n))^w/2)^{sn}$ for all $w \geq 1$ and for all sufficiently large n. Thus $\lim_{n\to\infty}\sum_{w=1}^{g(n)}\binom{n}{w}(1/2 + (1-2f(n))^w/2)^{sn} = 0$ and so by Lemma 4.5 we conclude that $Q^{f(n)}(n, sn)$ is $\Omega(g(n)) = \Omega(n\lambda^*/2) = \Omega(n)$-separated w.h.p. as desired. □

Chapter 5

Conclusion

In this thesis, we presented the first study of random CNF-XOR formulas. In this chapter, we summarize the main contributions of this thesis, discuss the implications of this thesis on hashing-based approaches to constrained counting and sampling, and finally outline directions for future research.

5.1 Summary of Contributions

In the first part of this thesis, we presented the first study of phase-transition phenomenon in the satisfiability of random k-CNF-XOR formulas. We showed in the k-CNF-XOR Phase-Transition Theorem that the free-entropy density $\phi_k(r)$ of k-CNF formulas gives the location of the phase-transition for k-CNF-XOR formulas when the density of the k-CNF clauses is small. We conjectured in the k-CNF-XOR Linear Phase-Transition Conjecture that this phase-transition is linear.

In the second part of this thesis, we presented the first study of the runtime behavior of SAT solvers on random k-CNF-XOR formulas. We presented experimental evidence that CryptoMiniSAT scales exponentially on random k-CNF-XOR formulas across a wide range of k-clause densities, XOR-clause densities, and XOR variable-probabilities. To begin to explain this phenomenon in the satisfiable region, we proved that the solution space of XOR-formulas is linearly separated w.h.p..

5.2 Implications for Sampling and Counting Algorithms

In Chapter 1, we motivated the study of CNF-XOR formulas through their usage in recent hashing-based algorithms for constrained sampling and counting. Given the greater understanding of random CNF-XOR formulas presented in this thesis, in this section we discuss several new insights into hashing-based algorithms.

The results of this thesis have immediate practical implications for the implementation of hashing-based algorithms. Recent hashing-based algorithms allow some freedom in the exact parameters (for example, in the XOR-clause density [52] or the XOR variable-probability [12]) used to generate CNF-XOR formulas. This thesis suggests combinations of clause-densities and XOR variable-probabilities that are likely to be difficult for CNF-XOR solvers and thus should be avoided. Applying these results to develop better heuristics for hashing-based algorithms is an exciting direction for future work that may lead to significant runtime improvements.

The results of this thesis offer theoretical insights into approximate counting and sampling as well. In particular, the properties of variable-width XOR-clauses that allow these XOR-clauses to produce good approximations when used in hashing-based algorithms [12] are exactly the properties used in this thesis to show the solution-space of XOR-formulas 'shatters'. That is, the quality of approximation provided by hashing-based algorithms appears to be fundamentally linked to the hardness of such algorithms. Although allowing approximate solutions made the problems of constrained counting and sampling *easier* than computing an exact solution, computing these approximations are still not *easy* even with state-of-the-art SAT-solving techniques. Further analysis of this apparent trade-off between approximation quality and algorithmic hardness may lead to complexity-theoretic insights into the difficulty of approximate counting and sampling.

5.3 Other Directions for Future Work

In addition to the implications described above for hashing-based algorithms for sampling and counting, there are many other interesting directions for future research. We outline several of them below.

Proving the Complete Phase-Transition In Chapter 3, we proved that that a phase-transition exists for k-CNF-XOR formulas when the k-clause density is small. In Section 3.4, we conjectured several possible extensions to these theoretical results, inspired by our empirical observations. In particular, we conjectured in the k-CNF-XOR Phase Transition Conjecture that this phase-transition is linear. We leave further analysis and proof of this conjecture for future work.

Additional Constraints for Sampling and Counting Pittel and Sorkin [42] recently identified the location of the phase-transition for random ℓ-XOR formulas, where each clause contains exactly ℓ literals. This suggests that a phase-transition may also exist in formulas that mix k-CNF clauses together with ℓ-XOR clauses. All proofs of the upper bound presented in Section 3.3.2 also hold for random ℓ-XOR formulas; however, the proofs of the lower bound presented in Section 3.3.1 may not hold for random ℓ-XOR formulas.

Is Shattering Hard for CNF-XOR Solvers? In the k-CNF case, the shattering of the solution-space into linearly separated components is closely connected to the exponential scaling of SAT algorithms within the satisfiable region [34]. Our experimental results presented in Chapter 4 suggest that such shattering is also difficult for CNF-XOR solvers; proving this analytically is an exciting direction for future work that may lead to practical improvements for SAT solvers used in hashing-based sampling and counting algorithms.

The hardness of unsatisfiable CNF-XOR formulas The shattering of the CNF-XOR solution-space described in Section 4.4 does not explain the exponential scaling of CryptoMiniSAT when the formula is unsatisfiable. Similar exponential scaling was observed for random k-CNF formulas in the unsatisfiable region [30]. In the case of random k-CNF formulas, this is explained by the exponential resolution complexity (w.h.p.) of random k-CNF formulas (when the clause-density is large) [53], which implies that all DPLL-type algorithms require exponential time w.h.p. to prove unsatisfiability within the unsatisfiable region. It may be possible to extend this analysis to show that the reasoning techniques used by specialized CNF-XOR solvers require exponential time w.h.p. to prove unsatisfiability within the unsatisfiable region as well.

Bibliography

[1] A. Biere, M. Heule, H. van Maaren, and T. Walsh, *Handbook of Satisfiability.* IOS Press, 2009.

[2] J. H. Liang, V. Ganesh, E. Zulkoski, A. Zaman, and K. Czarnecki, "Understanding VSIDS branching heuristics in conflict-driven clause-learning SAT solvers," in *Proc. of HVC*, pp. 225–241, 2015.

[3] C. Domshlak and J. Hoffmann, "Probabilistic planning via heuristic forward search and weighted model counting," *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 565–620, 2007.

[4] F. Bacchus, S. Dalmao, and T. Pitassi, "Algorithms and complexity results for# sat and bayesian inference," in *Proc. of FoCS*, pp. 340–351, IEEE, 2003.

[5] Y. Naveh, M. Rimon, I. Jaeger, Y. Katz, M. Vinov, E. Marcus, and G. Shurek, "Constraint-based random stimuli generation for hardware verification," in *Proc of IAAI*, pp. 1720–1727, 2006.

[6] T. Sang, F. Bacchus, P. Beame, H. Kautz, and T. Pitassi, "Combining component caching and clause learning for effective model counting," in *Proc. of SAT*, 2004.

[7] L. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[8] M. Jerrum, L. Valiant, and V. Vazirani, "Random generation of combinatorial structures from a uniform distribution," *Theoretical Computer Science*, vol. 43, no. 2-3, pp. 169–188, 1986.

[9] C. P. Gomes, A. Sabharwal, and B. Selman, "Model counting: A new strategy for obtaining good bounds," in *Proc. of AAAI*, vol. 21, pp. 54–61, 2006.

[10] S. Chakraborty, K. S. Meel, and M. Y. Vardi, "A scalable and nearly uniform generator of SAT witnesses," in *Proc. of CAV*, pp. 608–623, 2013.

[11] S. Chakraborty, K. S. Meel, and M. Y. Vardi, "A scalable approximate model counter," in *Proc. of CP*, pp. 200–216, 2013.

[12] S. Zhao, S. Chaturapruek, A. Sabharwal, and S. Ermon, "Closing the gap between short and long XORs for model counting," in *Proc. of AAAI*, 2016.

[13] K. S. Meel, M. Y. Vardi, S. Chakraborty, D. J. Fremont, S. A. Seshia, D. Fried, A. Ivrii, and S. Malik, "Constrained sampling and counting: Universal hashing meets SAT solving," in *Proc. of Beyond NP Workshop*, 2016.

[14] T. J. Schaefer, "The complexity of satisfiability problems," in *Proc. of SToC*, pp. 216–226, 1978.

[15] H. Haanpää, M. Järvisalo, P. Kaski, and I. Niemelä, "Hard satisfiable clause sets for benchmarking equivalence reasoning techniques," *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 2, pp. 27–46, 2006.

[16] M. Soos, K. Nohl, and C. Castelluccia, "Extending SAT Solvers to Cryptographic Problems," in *Proc. of SAT*, 2009.

[17] P. Cheeseman, B. Kanefsky, and W. M. Taylor, “Where the really hard problems are.,” in *Proc. of IJCAI*, pp. 331–340, 1991.

[18] D. E. Knuth, *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability.* Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2015.

[19] D. Achlioptas, “Random satisfiability,” in *Handbook of Satisfiability* [1], pp. 245–270.

[20] D. Achlioptas and A. Coja-Oghlan, “Algorithmic barriers from phase transitions,” in *Proc. of FoCS*, pp. 793–802, 2008.

[21] C. P. Gomes, B. Selman, N. Crato, and H. Kautz, “Heavy-tailed phenomena in satisfiability and constraint satisfaction problems,” *Journal of automated reasoning*, vol. 24, no. 1, pp. 67–100, 2000.

[22] J. Franco and M. Paull, “Probabilistic analysis of the davis putnam procedure for solving the satisfiability problem,” *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 77–87, 1983.

[23] D. Mitchell, B. Selman, and H. Levesque, “Hard and easy distributions of SAT problems,” in *Proc. of AAAI*, pp. 459–465, 1992.

[24] J. M. Crawford and L. D. Auton, “Experimental results on the crossover point in satisfiability problems,” in *Proc. of AAAI*, pp. 21–27, 1993.

[25] S. Kirkpatrick and B. Selman, “Critical behavior in the satisfiability of random boolean expressions,” *Science*, vol. 264, no. 5163, pp. 1297–1301, 1994.

[26] A. Coja-Oghlan and K. Panagiotou, "Going after the k-sat threshold," in *Proc. of SToC*, pp. 705–714, 2013.

[27] V. Chvátal and B. Reed, "Mick gets some (the odds are on his side)," in *Proc. of FoCS*, pp. 620–627, 1992.

[28] A. Goerdt, "A threshold for unsatisfiability," *Journal of Computer and System Sciences*, vol. 53, no. 3, pp. 469 – 486, 1996.

[29] J. Ding, A. Sly, and N. Sun, "Proof of the satisfiability conjecture for large k," in *Proc. of SToC*, pp. 59–68, 2015.

[30] C. Coarfa, D. D. Demopoulos, A. S. M. Aguirre, D. Subramanian, and M. Y. Vardi, "Random 3-SAT: The plot thickens," *Constraints*, vol. 8, no. 3, pp. 243–261, 2003.

[31] H. Daudé, M. Mézard, T. Mora, and R. Zecchina, "Pairs of SAT-assignments in random boolean formulae," *Theoretical Computer Science*, vol. 393, no. 1, pp. 260 – 279, 2008.

[32] M. Mézard, T. Mora, and R. Zecchina, "Clustering of solutions in the random satisfiability problem," *Phys. Rev. Lett.*, vol. 94, p. 197205, May 2005.

[33] D. Achlioptas, A. Coja-Oghlan, and F. Ricci-Tersenghi, "On the solution-space geometry of random constraint satisfaction problems," *Random Structures & Algorithms*, vol. 38, no. 3, pp. 251–268, 2011.

[34] D. Achlioptas and R. Menchaca-Mendez, "Exponential lower bounds for DPLL algorithms on satisfiable random 3-CNF formulas," in *Proc. of SAT*, pp. 327–340, 2012.

[35] A. Coja-Oghlan, "On belief propagation guided decimation for random k-SAT," in *Proc. of SIAM*, pp. 957–966, 2011.

[36] D. Achlioptas and M. Molloy, "The solution space geometry of random linear equations," *Random Structures & Algorithms*, 2013.

[37] S. Mertens, M. Mézard, and R. Zecchina, "Threshold values of random k-SAT from the cavity method," *Random Structures & Algorithms*, vol. 28, no. 3, pp. 340–373, 2006.

[38] S. A. Cook and D. G. Mitchell, "Finding hard instances of the satisfiability problem," in *Satisfiability Problem: Theory and Applications: DIMACS Workshop*, vol. 35, pp. 1–17, AMS, 1997.

[39] N. Creignou and H. Daudé, "Satisfiability threshold for random xor-cnf formulas," *Discrete Applied Mathematics*, vol. 9697, pp. 41 – 53, 1999.

[40] N. Creignou and H. Daudé, "Smooth and sharp thresholds for random k-xor-cnf satisfiability," *RAIRO:ITA*, vol. 37, no. 2, pp. 127–147, 2003.

[41] O. Dubois and J. Mandler, "The 3-xorsat threshold," *Comptes Rendus Mathematique*, vol. 335, no. 11, pp. 963 – 966, 2002.

[42] B. Pittel and G. B. Sorkin, "The satisfiability threshold for k-xorsat," *Combinatorics, Probability and Computing*, vol. FirstView, pp. 1–33, 10 2015.

[43] M. Ibrahimi, Y. Kanoria, M. Kraning, and A. Montanari, "The set of solutions of random XORSAT formulae," in *Proc. of SIAM*, pp. 760–779, 2012.

[44] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi, "Distribution-aware sampling and weighted model counting for SAT," in *Proc.*

of AAAI, pp. 1722–1730, 2014.

[45] A. Belov, D. Diepold, M. J. Heule, and M. Järvisalo, "Sat competition 2014," 2014.

[46] D. B. Wilson, "Random 2-sat data." http://dbwilson.com/2sat-data/, 2000.

[47] S. Gogioso, "Aspects of statistical physics in computational complexity," *arXiv:1405.3558*, 2014.

[48] E. Abbe and A. Montanari, "On the concentration of the number of solutions of random satisfiability formulas," *Random Structures & Algorithms*, vol. 45, no. 3, pp. 362–382, 2014.

[49] A. Coja-Oghlan and D. Reichman, "Sharp thresholds and the partition function," in *Journal of Physics: Conference Series*, vol. 473, p. 012015, 2013.

[50] C. P. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman, "Short XORs for model counting: from theory to practice," in *Proc. of SAT*, pp. 100–106, 2007.

[51] A. Ivrii, S. Malik, K. S. Meel, and M. Y. Vardi, "On computing minimal independent support and its applications to sampling and counting," *Constraints*, vol. 21, no. 1, pp. 41–58, 2016.

[52] S. Chakraborty, K. S. Meel, and M. Y. Vardi, "Algorithmic improvements in approximate counting for probabilistic inference: From linear to logarithmic SAT calls," in *Proc. of IJCAI*, pp. 3569 – 3576, 2016.

[53] V. Chvátal and E. Szemerédi, "Many hard examples for resolution," *Journal of the ACM (JACM)*, vol. 35, no. 4, pp. 759–768, 1988.