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

Jeffrey M. Dudek
Rice University

Kuldeep S. Meel
Rice Univesity

Moshe Y. Vardi
Rice University

Abstract

Recent universal-hashing based approaches to sampling and counting crucially depend on the runtime performance of SAT solvers on formulas expressed as the conjunction of both CNF constraints and variable-width XOR constraints (known as CNF-XOR formulas). In this paper, we present the first study of the runtime behavior of SAT solvers equipped with XOR-reasoning techniques on random CNF-XOR formulas. We empirically demonstrate that a state-of-the-art SAT solver scales exponentially on random CNF-XOR formulas across a wide range of XOR-clause densities, peaking around the empirical phase-transition location. On the theoretical front, we prove that the solution space of a random CNF-XOR formula 'shatters' at *all* nonzero XOR-clause densities into well-separated components, similar to the behavior seen in random CNF formulas known to be difficult for many SAT-solving algorithms.

1 Introduction

The Boolean-Satisfaction Problem (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, biology and the like [Biere *et al.*, 2009]. While SAT is NP-complete, the study of the runtime behavior of SAT techniques is a topic of major interest in AI [Liang *et al.*, 2015] owing to its practical usage. Of specific interest is the behavior of SAT solvers on random problems [Cheeseman *et al.*, 1991], motivated by the connection between the clause density (the ratio of clauses to variables) of a random SAT instance and algorithmic properties of the solution space. Early experiments [Mitchell *et al.*, 1992; Crawford and Auton, 1993; Kirkpatrick and Selman, 1994] on random fixed-width CNF formulas (where each clause contains a fixed number of literals) revealed a surprising phase-transition behavior in the satisfiability of random formulas: the probability of satisfiability undergoes a precipitous drop around a fixed density, the location of which depends only on the clause width (for CNF formulas with clause width 3, this occurs around a clause density of 4.26). Moreover, the runtime of SAT solvers (using DPLL and related algorithms) on random CNF formulas was shown to follow an *easy-hard-easy* pattern [Kirkpatrick and Selman, 1994]: 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 fixed-width 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 [Coarfa *et al.*, 2003]. Theoretical analysis of this phenomenon [Daudé *et al.*, 2008; Mézard *et al.*, 2005; Achlioptas *et al.*, 2011] 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). In this region, several algorithms are known to solve random fixed-width CNF formulas w.h.p. in polynomial time [Achlioptas, 2009]. 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 [Achlioptas and Menchaca-Mendez, 2012; Coja-Oghlan, 2011].

Although this prior work exists on the runtime scaling behavior of SAT solvers on random fixed-width CNF formulas and on certain other classes of random constraints, no prior work considers the runtime scaling behavior of SAT solvers on formulas composed of both CNF-clauses and XOR-clauses, known as CNF-XOR formulas. Recently, successful hashing-based approaches to the fundamental problems of constrained sampling and counting employ SAT solvers to solve CNF-XOR formulas [Gomes *et al.*, 2006; Chakraborty *et al.*, 2013; Zhao *et al.*, 2016; Meel *et al.*, 2016]. The scalability of these hashing-based algorithms crucially depends on the runtime performance of SAT solvers in handling

*The author list has been sorted alphabetically by last name; this should not be used to determine the extent of authors' contributions.

CNF-XOR formulas. Although XOR-formulas can be solved individually in polynomial time (using Gaussian Elimination [Schaefer, 1978]), XOR-formulas are empirically hard [Haanpää *et al.*, 2006] 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 [Soos *et al.*, 2009], that combine SAT-solving techniques with algebraic techniques and so can reason about about both the CNF-clauses and XOR-clauses within a single CNF-XOR formula.

The runtime behavior of these specialized CNF-XOR solvers is an area of active research. Recent work [Dudek *et al.*, 2016] analyzed the satisfiability of random formulas composed of both random k-clauses (i.e. CNF-clauses of fixed-width k) and random variable-width XOR-clauses (where the width of the XOR-clauses used is stochastic), known as random k-CNF-XOR formulas, to begin to demystify the behavior of CNF-XOR solvers. Since the scaling behavior of SAT solvers on random k-clauses has been analyzed to explain the runtime behavior of SAT solvers in practice [Achlioptas, 2009], we believe that analysis of the scaling behavior of CNF-XOR solvers on random k-CNF-XOR formulas is the next step towards explaining the runtime behavior of CNF-XOR solvers in practice and thus explaining the runtime behavior of hashing-based algorithms.

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 [Gomes *et al.*, 2007; Ivrii *et al.*, 2016] have focused on designing hashing-based techniques that employ XOR-clauses of smaller width. In this paper, we present empirical evidence that using smaller width XOR-clauses does not necessarily improve the scaling behavior of CNF-XOR solvers.

The primary contribution of this work is the first empirical and theoretical 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:

1. We present (in Section 3) experimental evidence that the runtime of 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.
2. We present (in Section 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.
3. We prove (in Section 4) that the solution space of random variable-width XOR formulas (and therefore of random k-CNF-XOR formulas) shatters. We hypothesize that the exponential scaling behavior of random k-CNF-XOR formulas within the satisfiable region is caused by this solution space shattering.

2 Notations and Preliminaries

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. We say that F is *satisfiable* (or *sat.*) if there exists a satisfying assignment of F and that F is *unsatisfiable* (or *unsat.*) otherwise.

We describe the solution space of F using terminology from Achlioptas and Molloy [2013]. 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 $\mathsf{Pr}[E]$ to denote the probability of event E. We say that an infinite sequence of random events $E_1, E_2, \cdots$ occurs *with high probability* (denoted, w.h.p.) if $\lim_{n\to\infty} \mathsf{Pr}[E_n] = 1$.

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 $F_k(n, rn)$ [Mitchell *et al.*, 1992; Crawford and Auton, 1993; Kirkpatrick and Selman, 1994] 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 has been proven for $k = 2$ and for all sufficiently large k [Ding *et al.*, 2015]. The Conjecture has remained elusive for small values of $k \geq 3$, although values for these r_k can be estimated experimentally (e.g., r_3 seems to be near 4.26) and predicted analytically using techniques from statistical physics [Mertens *et al.*, 2006].

When the k-clause density is small (e.g. below $O(2^k/k)$) there are algorithms that are known to solve $F_k(n, rn)$ with high probability in polynomial time [Cook and Mitchell, 1997]. 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 [Achlioptas *et al.*, 2011]. The solution space geometry of $F_k(n, rn)$ can 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 [Achlioptas *et al.*, 2011]. This 'shattering' of the solution space into linearly separated clusters is known to

be difficult for a variety of SAT-solving algorithms [Achlioptas and Menchaca-Mendez, 2012; Coja-Oghlan, 2011].

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. The solution space geometry of $Q^p(n, sn)$ has not been characterized in prior work. There is a related model of random XOR-formulas where every XOR-clause contains a fixed number of variables. In this case, 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 [Achlioptas and Molloy, 2013; Ibrahimi *et al.*, 2012].

The random variable $Q^{1/2}(n, sn)$ matches the XOR-clauses used in several hashing-based constrained sampling and counting algorithms [Chakraborty *et al.*, 2013]. Recent work [Zhao *et al.*, 2016] has made use of $Q^p(n, sn)$ with $p < 1/2$ for constrained sampling and counting algorithms.

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. There exists a phase-transition in the satisfiability of $\psi_k^{1/2}(n, rn, sn)$ when the k-clause density is small, shown by the following theorem [Dudek *et al.*, 2016]:

Theorem 1. *Let $k \geq 2$. There is a function $\phi_k(r)$ and a constant $\alpha_k \geq 1$ such that for all $s \geq 0$ and all $r \in [0, \alpha_k)$ (except for at most countably many r):*

1. *If $s < \phi_k(r)$, then w.h.p. $\psi_k^{1/2}(n, rn, sn)$ is sat.*
2. *If $s > \phi_k(r)$, then w.h.p. $\psi_k^{1/2}(n, rn, sn)$ is unsat.*

3 Experimental Results

To explore empirically the runtime behavior of solvers on randomly constructed k-CNF-XOR formulas, we built a prototype implementation in Python that employs the CryptoMiniSAT[1] [Soos *et al.*, 2009] 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 [Chakraborty *et al.*, 2014].

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.

3.1 Experimental Setup

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 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.

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 results from these experiments in Section 3.2.
- To study the effect of the XOR variable-probability on the runtime, we ran 1295 experiments with $r = 2$, p ranging from 0.02 to 0.94 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 3.3.

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. For each n, we used CryptoMiniSAT to check the satisfiability of 100 formulas sampled from $\psi_k^p(n, rn, sn)$ by constructing the conjunction of $\lceil rn \rceil$ k-clauses and $\lceil sn \rceil$ XOR-clauses, with each clause chosen independently as described above. The solving of each formula was individually timed. The median runtime is an estimate for the median runtime of CryptoMiniSAT on $\psi_k^p(n, rn, sn)$.

Finally, we used the curve_fit function in the Python scipy.optimize[2] 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),

[1] http://www.msoos.org/cryptominisat4/

[2] https://www.scipy.org/

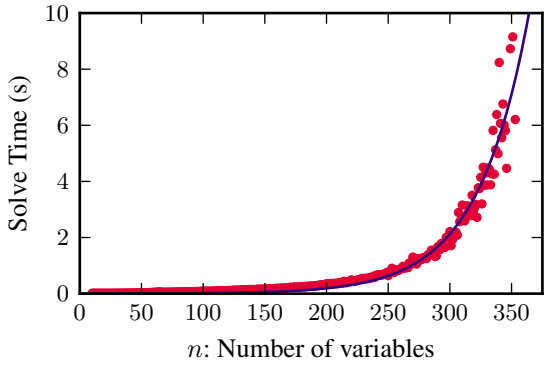

Figure 1: Runtime for 3-CNF-XOR formulas at 3-clause density $r = 2$, XOR-clause density $s = 0.3$, and XOR variable-probability $p = 1/2$, together with the best-fit curve $0.00152 \cdot 2^{0.0348n}$.

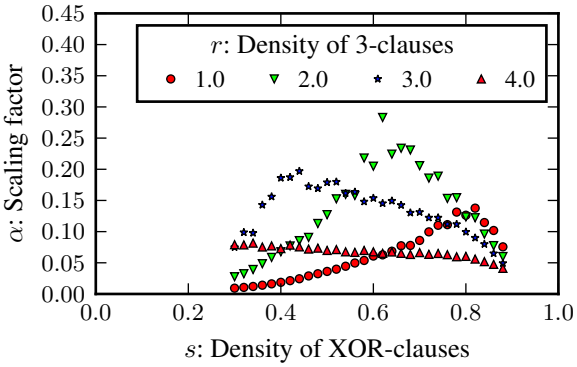

Figure 2: Exponential scaling factor for 3-CNF-XOR formulas with 3-clause density $r \in \{1, 2, 3, 4\}$ and XOR variable-probability $p = 1/2$. The scaling factor α is the exponent of the best-fit line for the runtime of $\psi_3^{1/2}(n, rn, sn)$.

and exponential ($\beta 2^{\alpha n}$) curves; the best-fit curve was the curve with the smallest mean squared error.

Each experiment was 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. Each formula was given a timeout of 10 seconds. We were not able to run informative experiments for formulas with higher timeouts; as the runtime of CryptoMiniSAT increases past 10 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.

3.2 Results on the Impact of 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.

Figure 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 CryptoMiniSATon $\psi_3^{1/2}(n, 4n, 0.3n)$.

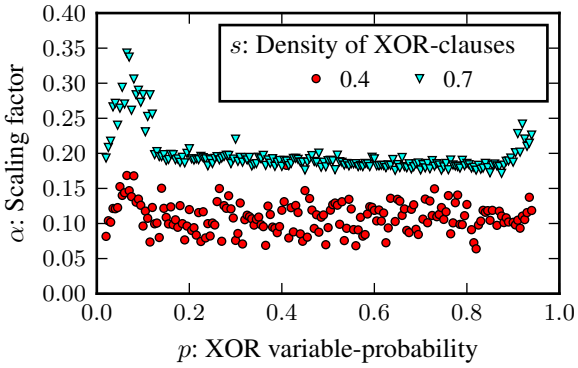

Figure 3: Exponential scaling factor for 3-CNF-XOR formulas with 3-clause density $r = 2$ and XOR-clause densities $s = 0.4$ and 0.7.

We observe that the median runtime increases exponentially in the number of variables. In this case, the best-fit curve is the exponential function $0.00152 \cdot 2^{0.0348n}$.

In fact, for all experiments with $r \in \{1, 2, 3, 4\}$ 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 2 plots the scaling behavior with respect to n of the median runtime of CryptoMiniSAT on $\psi_3^{1/2}(n, rn, sn)$. 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 $\psi_3^{1/2}(n, rn, sn)$ 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. We observe peaks in the scaling factor near ($r = 1$, $s = 0.8$), ($r = 2$, $s = 0.6$) and ($r = 3$, $s = 0.4$). Empirically, there is a phase-transition in the satisfiability of random 3-CNF-XOR formulas exactly at these locations [Dudek *et al.*, 2016]. 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 [Coarfa *et al.*, 2003].

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.

3.3 Results on the Impact of 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. For lack of space, we present results only for the experiments with $s \in \{0.4, 0.7\}$ [3].

Figure 3 plots the scaling behavior with respect to n of the medium runtime of CryptoMiniSAT on $\psi_3^p(n, 2n, sn)$. The

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

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 $2^{\alpha n}$. Note that $(r = 2, s = 0.4)$ is in the satisfiable region and $(r = 2, s = 0.7)$ is in the unsatisfiable region when $p = 1/2$. We observe that the behavior of the scaling factor is independent of the XOR variable-probability, p, when $p \in (0.15, 0.9)$. As p decreases below 0.15, the scaling factor increases to a peak when $p \in (0.05, 0.1)$, then decreases. We also observe a peak in the scaling factor when $p > 0.9$.

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 between 0.15 and 0.9 and the scaling behavior peaks when the XOR variable-probability is between 0.05 and 0.1, independent of the XOR-clause density.

4 The Separation of the XOR 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. [Achlioptas and Menchaca-Mendez, 2012; Coja-Oghlan, 2011]. Prior work has shown that the solution space of fixed-width XOR-clauses has similar behavior; unfortunately, the proof techniques used in this prior work 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 [Achlioptas and Molloy, 2013] or random factor graphs with factors of constant degree ℓ [Ibrahimi *et al.*, 2012]. If the width of each XOR-clause is stochastic, as in $Q^p(n, sn)$, rather than fixed, the corresponding hypergraphs are not uniform and the corresponding factor graphs do not have factors of constant degree.

Nevertheless, we show in Theorem 2 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 2. *Let $s \in (0, 1)$, $\rho > 2$, and $f(n)$ be a nonnegative 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 7. The proof of this lemma appears in Section 4.1. □

Notice that Theorem 2 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 2 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 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 3.

Corollary 3. *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 2 with $f(n) = p$. □

Corollary 3 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. Since the separation of the k-CNF solution space is closely connected to the exponential scaling of SAT solvers, we hypothesize that the exponential scaling of CryptoMiniSAT we observed in Section 3 at many XOR-clause densities and XOR variable-probabilities is closely connected to the $\Omega(n)$-separation of k-CNF-XOR formulas shown in Corollary 3 at all nonzero XOR-clause densities and XOR variable-probabilities (below $1/2$).

4.1 Proofs

In this section we establish Theorem 2, which follows directly from Lemma 7. 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 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 A containing no more than λn variables for some constant λ, we bound 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 6 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. *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 $\Pr\left[\sum_{i=1}^N B_i \text{ is even}\right] = 1/2 + 1/2(1-2p)^N$.*

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 $a_N = 1/2 + 1/2(1-2p)^N$. □

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 5. *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. This proof is given as **Lemma 7** of [Zhao *et al.*, 2016]. □

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 7 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 6. *Let* $f(n)$ *and* $g(n)$ *be nonnegative functions with* $f(n) \leq 1$ *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 $\lim_{n\to\infty} \mathsf{Pr}[D = 1] = 0$.

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 $\mathsf{Pr}[D = 1] \leq \sum_{A\in\mathcal{B}} \mathsf{Pr}[D(A) = 1]$.

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 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 it follows that the probability that Q_1 contains an even number of variables from A is $1/2 + 1/2(1 - 2f(n))^{|A|}$.

Since all $\lceil sn \rceil$ XOR-clauses of $Q^p(n, sn)$ are chosen independently with variable-probability $f(n)$, it follows that $\mathsf{Pr}[D(A)] = (1/2 + (1 - 2f(n))^{|A|}/2)^{\lceil sn \rceil}$. For all sufficiently large n, we have $0 \leq f(n) \leq 1$ and thus $0 \leq 1/2 + (1 - 2f(n))^{|A|}/2 \leq 1$. Thus $\mathsf{Pr}[D(A)] \leq (1/2 + (1 - 2f(n))^{|A|}/2)^{sn}$ 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 $\mathsf{Pr}[D = 1] \leq \sum_{A\in\mathcal{B}}(1/2 + (1 - 2f(n))^{|A|}/2)^{sn} = \sum_{w=1}^{g(n)} \binom{n}{w}(1/2 + (1 - 2f(n))^w/2)^{sn}$. By hypothesis, this implies that $\lim_{n\to\infty} \mathsf{Pr}[D = 1] = 0$. □

The following lemma combines Lemma 5 and Lemma 6 to show that a variable-probability above $2\log(sn)/(sn)$ implies $\Omega(n)$-separation. This finishes the proof of Theorem 2.

Lemma 7. *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 $H(x) = -x\log(x) - (1 - x)\log(1 - x)$. Since H is continuous on $[0, 1/2]$ and $H(0) < s\log(1+\delta) < H(1/2)$, it follows 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 5 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, for all sufficiently large n, $\widehat{f}(n) \leq f(n) \leq 1/2$ and so $(1 - 2\widehat{f}(n))^w \geq (1 - 2f(n))^w \geq 0$ for all $w \geq 1$. 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 6 we conclude that $Q^{f(n)}(n, sn)$ is $\Omega(g(n)) = \Omega(n\lambda^*/2) = \Omega(n)$-separated w.h.p. as desired. □

5 Conclusion

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..

Recent hashing-based algorithms for sampling and counting allow some freedom in the exact parameters (for example, in the XOR-clause density [Chakraborty *et al.*, 2016] or the XOR variable-probability [Zhao *et al.*, 2016]) used to generate CNF-XOR formulas. This paper suggests combinations of clause-densities and XOR variable-probabilities that are likely to be difficult for CNF-XOR solvers and thus should be avoided. Using this information to develop better heuristics for hashing-based algorithms is an exciting direction for future work that may lead to significant runtime improvements. For a more detailed discussion, see [Dudek, 2017].

Acknowledgments

Work supported in part by NSF grants CCF-1319459 and IIS-1527668, by NSF Expeditions in Computing project "ExCAPE: Expeditions in Computer Augmented Program Engineering", by BSF grant 9800096, by the Ken Kennedy Institute Computer Science & Engineering Enhancement Fellowship funded by the Rice Oil & Gas HPC Conference, and Data Analysis and Visualization Cyberinfrastructure funded by NSF under grant OCI-0959097. Kuldeep S. Meel is supported by the IBM PhD Fellowship and the Lodieska Stockbridge Vaughn Fellowship.

References

[Achlioptas and Menchaca-Mendez, 2012] Dimitris Achlioptas and Ricardo Menchaca-Mendez. Exponential lower bounds for DPLL algorithms on satisfiable random 3-CNF formulas. In *Proc. of SAT*, pages 327–340, 2012.

[Achlioptas and Molloy, 2013] Dimitris Achlioptas and Michael Molloy. The solution space geometry of random linear equations. *Random Structures & Algorithms*, 2013.

[Achlioptas *et al.*, 2011] Dimitris Achlioptas, Amin Coja-Oghlan, and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. *Random Structures & Algorithms*, 38(3):251–268, 2011.

[Achlioptas, 2009] Dimitris Achlioptas. Random satisfiability. In *Handbook of Satisfiability* [2009], pages 245–270.

[Biere *et al.*, 2009] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh. *Handbook of Satisfiability*. IOS Press, 2009.

[Chakraborty *et al.*, 2013] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. A scalable and nearly uniform generator of SAT witnesses. In *Proc. of CAV*, pages 608–623, 2013.

[Chakraborty *et al.*, 2014] Supratik Chakraborty, Daniel J. Fremont, Kuldeep S. Meel, Sanjit A. Seshia, and Moshe Y. Vardi. Distribution-aware sampling and weighted model counting for SAT. In *Proc. of AAAI*, pages 1722–1730, 2014.

[Chakraborty *et al.*, 2016] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. Algorithmic improvements in approximate counting for probabilistic inference: From linear to logarithmic SAT calls. In *Proc. of IJCAI*, pages 3569 – 3576, 2016.

[Cheeseman *et al.*, 1991] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *Proc. of IJCAI*, pages 331–340, 1991.

[Coarfa *et al.*, 2003] Cristian Coarfa, Demetrios D. Demopoulos, Alfonso San Miguel Aguirre, Devika Subramanian, and Moshe Y. Vardi. Random 3-SAT: The plot thickens. *Constraints*, 8(3):243–261, 2003.

[Coja-Oghlan, 2011] Amin Coja-Oghlan. On belief propagation guided decimation for random k-SAT. In *Proc. of SIAM*, pages 957–966, 2011.

[Cook and Mitchell, 1997] Stephen A. Cook and David G. Mitchell. Finding hard instances of the satisfiability problem. In *Satisfiability Problem: Theory and Applications: DIMACS Workshop*, volume 35, pages 1–17. AMS, 1997.

[Crawford and Auton, 1993] James M. Crawford and Larry D. Auton. Experimental results on the crossover point in satisfiability problems. In *Proc. of AAAI*, pages 21–27, 1993.

[Daudé *et al.*, 2008] Hervé Daudé, Marc Mézard, Thierry Mora, and Riccardo Zecchina. Pairs of SAT-assignments in random boolean formulae. *Theoretical Computer Science*, 393(1):260 – 279, 2008.

[Ding *et al.*, 2015] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In *Proc. of SToC*, pages 59–68, 2015.

[Dudek *et al.*, 2016] Jeffrey M. Dudek, Kuldeep S. Meel, and Moshe Y. Vardi. Combining the k-CNF and XOR phase-transitions. In *Proc. of IJCAI*, pages 727 – 734, 2016.

[Dudek, 2017] Jeffrey Dudek. Random CNF-XOR formulas. M.Sc. Thesis, Rice University, 2017.

[Gomes *et al.*, 2006] Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Model counting: A new strategy for obtaining good bounds. In *Proc. of AAAI*, volume 21, pages 54–61, 2006.

[Gomes *et al.*, 2007] Carla P. Gomes, Joerg Hoffmann, Ashish Sabharwal, and Bart Selman. Short XORs for model counting: from theory to practice. In *Proc. of SAT*, pages 100–106, 2007.

[Haanpää *et al.*, 2006] Harri Haanpää, Matti Järvisalo, Petteri Kaski, and Ilkka Niemelä. Hard satisfiable clause sets for benchmarking equivalence reasoning techniques. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:27–46, 2006.

[Ibrahimi *et al.*, 2012] Morteza Ibrahimi, Yash Kanoria, Matt Kraning, and Andrea Montanari. The set of solutions of random XOR-SAT formulae. In *Proc. of SIAM*, pages 760–779, 2012.

[Ivrii *et al.*, 2016] Alexander Ivrii, Sharad Malik, Kuldeep S. Meel, and Moshe Y. Vardi. On computing minimal independent support and its applications to sampling and counting. *Constraints*, 21(1):41–58, 2016.

[Kirkpatrick and Selman, 1994] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 264(5163):1297–1301, 1994.

[Liang *et al.*, 2015] Jia Hui Liang, Vijay Ganesh, Ed Zulkoski, Atulan Zaman, and Krzysztof Czarnecki. Understanding VSIDS branching heuristics in conflict-driven clause-learning SAT solvers. In *Proc. of HVC*, pages 225–241, 2015.

[Meel *et al.*, 2016] Kuldeep S. Meel, Moshe Y. Vardi, Supratik Chakraborty, Daniel J. Fremont, Sanjit A. Seshia, Dror Fried, Alexander Ivrii, and Sharad Malik. Constrained sampling and counting: Universal hashing meets SAT solving. In *Proc. of Beyond NP Workshop*, 2016.

[Mertens *et al.*, 2006] Stephan Mertens, Marc Mézard, and Riccardo Zecchina. Threshold values of random k-SAT from the cavity method. *Random Structures & Algorithms*, 28(3):340–373, 2006.

[Mézard *et al.*, 2005] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Phys. Rev. Lett.*, 94:197205, May 2005.

[Mitchell *et al.*, 1992] David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. In *Proc. of AAAI*, pages 459–465, 1992.

[Schaefer, 1978] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. of SToC*, pages 216–226, 1978.

[Soos *et al.*, 2009] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT Solvers to Cryptographic Problems. In *Proc. of SAT*, 2009.

[Zhao *et al.*, 2016] Shengjia Zhao, Sorathan Chaturapruek, Ashish Sabharwal, and Stefano Ermon. Closing the gap between short and long XORs for model counting. In *Proc. of AAAI*, 2016.