Phase Transition Behavior of Cardinality and XOR Constraints

Yash Pote[1] , **Saurabh Joshi**[2] and **Kuldeep S. Meel**[1]
[1]National University of Singapore
[2]IIT Hyderabad, India

Abstract

The runtime performance of modern SAT solvers is deeply connected to the phase transition behavior of CNF formulas. While CNF solving has witnessed significant runtime improvement over the past two decades, the same does not hold for several other classes such as the conjunction of cardinality and XOR constraints, denoted as CARD-XOR formulas. The problem of determining satisfiability of CARD-XOR formulas is a fundamental problem with wide variety of applications ranging from discrete integration in the field of artificial intelligence to maximum likelihood decoding in coding theory. The runtime behavior of random CARD-XOR formulas is unexplored in prior work. In this paper, we present the first rigorous empirical study to characterize the runtime behavior of 1-CARD-XOR formulas. We show empirical evidence of a surprising phase-transition that follows a non-linear tradeoff between CARD and XOR constraints.

1 Introduction

The study of runtime behavior of algorithmic techniques in the context of constraint satisfaction problems (CSP) has been key to several breakthroughs in the design of new solvers [Dechter and Meiri, 1994]. Specifically, a deep connection was discovered between the density (ratio of the number of clauses to the number of variables) of random propositional CNF fixed-width (fixed number of literals per clause) formulas and the runtime behavior of SAT solvers on such formulas. For random k-CNF formulas, where every clause contains exactly k literals, experiments suggest a specific phase-transition density, for example 4.26 for random 3-SAT, but establishing this analytically has been highly challenging [Coja-Oghlan and Panagiotou, 2013], and it has been established only for $k = 2$ [Chvátal and Reed, 1992] and all large enough k [Ding *et al.*, 2015]. A phase-transition phenomenon has also been identified in random XOR formulas (conjunctions of XOR constraints). Creignou and Daudé [1999] proved a phase-transition at density 1 for variable-width random XOR formulas.

Recently, Dudek, Meel, and Vardi [2016; 2017] extended such studies to the conjunction of CNF and XOR constraints, called CNF-XOR formulas. The motivation for their study was the usage of CNF-XOR formulas in the recent hashing-based techniques for the problem of propositional model counting [Stockmeyer, 1983; Chakraborty *et al.*, 2013; 2016; Soos and Meel, 2019].

Satisfiability of conjunction of a cardinality constraint and XOR constraints gives rise to an interesting problem, which we shall refer to as 1-CARD-XOR. Given a set of propositional variables, a cardinality constraint (CARD) puts bounds on how many of these variables can be set to `true`. It is worth noting that 1-CARD-XOR is still NP-complete [Berlekamp *et al.*, 1978], even to approximate [Arora *et al.*, 1997], and as our experimental evaluation demonstrates, the study of 1-CARD-XOR alone is computationally expensive. Furthermore, 1-CARD-XOR formulas are necessary and sufficient for maximum likelihood decoding (MLD), one of the most crucial problems in coding theory, in which one seeks to extract the maximum amount of information from a noisy channel. The problem of maximum likelihood decoding is equivalent to determining satisfiability of a 1-CARD-XOR formula. Consequently, MLD has been subject to theoretical and practical investigations for over 50 years [Chase, 1985; Tal and Vardy, 2015].

Generalization of a cardinality constraint is a Pseudo-Boolean (PB) constraint which enforces bounds on the summation of the weights of the propositional variables that can be set to `true`. A variant and a more generalized version of 1-CARD-XOR problem is the satisfiability of conjunction of CNF constraints, one Pseudo-Boolean constraint and random XOR constraints, denoted as CNF-PB-XOR formulas. Formulas of this kind play a crucial role in solving one of the fundamental problems in artificial intelligence: discrete integration. Given a set of constraints as a Boolean formula F and a weight function W, the problem of discrete integration is to compute the total weight of the set of solutions of input constraints. This has applications in numerous areas, including probabilistic reasoning, machine learning, planning, statistical physics, inexact computing, and constrained-random verification [Jerrum and Sinclair, 1996; Madras and Piccioni, 1999; Bacchus *et al.*, 2003; Sang *et al.*, 2004; Domshlak and Hoffmann, 2007; Gomes *et al.*, 2009; Murphy, 2012; Ermon *et al.*, 2014].

Recently, two hashing-based approaches have been proposed for discrete integration: WISH [Ermon *et al.*, 2013] and WeightMC [Chakraborty *et al.*, 2014]. Both of these approaches provide strong Probably Approximately Correct (PAC)-style guarantees, i.e., (ε, δ) guarantees. The core idea of WeightMC is to partition the problem of discrete integration into linearly many *regions* such that the weight of satisfying assignments in each of the *regions* is *almost equal*; thereby allowing the usage of hashing-based unweighed counting techniques for each of the regions. Each of the regions can be represented by the conjunction of F and one Pseudo-Boolean (PB) constraint. Consequently, the underlying SAT solver invoked during unweighted counting subroutine needs to handle the CNF-PB-XOR formulas. While the elegant formulation of WISH and WeightMC promises scalability and strong theoretical guarantees, WISH and WeightMC have not witnessed scalability similar to that of unweighted counting algorithms such as ApproxMC3. Unlike CryptoMiniSAT, which is optimized for CNF-XOR formulas, to the best of our knowledge, there do not exist specialized solvers that can handle CNF-PB-XOR formulas efficiently. Design of solvers to efficiently handle CARD-XOR formulas alone would push the boundaries of state-of-the-art techniques to handle several problems of interest [Duenas-Osorio *et al.*, 2017].

The phase-transition behavior of CNF constraints has been analyzed to explain runtime behavior of SAT solvers [Achlioptas and Coja-Oghlan, 2008]. Furthermore, the study of Dudek et al [2016; 2017] contributed to the development of a new architecture for handling CNF-XOR constraints [Soos and Meel, 2019]. We believe that analysis of the phase-transition phenomenon for CARD-XOR formulas would be pivotal towards demystifying the runtime behavior of the current state of the art solvers. Therefore, a deeper understanding of the runtime behavior of CSP/SAT solvers for CARD-XOR constraints can have far-reaching consequences.

The primary contribution of this work is the first rigorous empirical study to characterize the runtime behavior of 1-CARD-XOR formulas. In particular:

1. We prove (in Section 4) upper and lower bounds on the location of the 1-CARD-XOR phase-transition region.

2. We present (in Section 5) experimental evidence for phase transition behavior of 1-CARD-XOR formulas, henceforth known as 1-CARD-XOR phase-transition that follows a non-linear trade-off between k-CNF clauses and XOR clauses.

3. We demonstrate that the runtime behavior of SAT solver around phase transition is reminiscent of random CNF formulas but is surprisingly different from CNF-XOR formulas. This observation underscores the need for further exploration in this direction for deeper understanding.

The surprising nature of our observations opens up future directions of research and we hope that a deeper understanding would lead to the design of efficient CARD-XOR solvers in the future. The rest of the paper is organized as follows. We discuss notations and preliminaries in Section 2. We survey related work in Section 3. We present a theoretical analysis to obtain lower and upper bounds on the location of phase transition in Section 4. We then present the empirical behavior of 1-CARD-XOR constraints in Section 5. We finally conclude in Section 6.

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 a *witness* of F is an assignment of truth values to the variables in X such that F evaluates to true. Let $\#F$ denote the number of satisfying assignments of F. We say that F is satisfiable (or SAT) if $\#F > 0$ and unsatisfiable (or UNSAT) if $\#F = 0$.

A single XOR constraint (also called a *XOR clause*) over X is specified as $a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n = b_0$, where all $a_i, b_j \in \{0, 1\}$. Satisfiability of a system of m XOR constraints (XORSAT) over n variables can be thought of as a matrix $A \in \{0,1\}^{m\times n}$, a vector $b \in \{0,1\}^m$, and a variable vector $x \in \{0,1\}^n$ which satisfy $Ax = b$. The density s of a system of m XOR constraints is the ratio $s = m/n$.

To generate a random XORSAT instance, we create such a matrix A and the vector b with each element either 0 or 1 with probability $\frac{1}{2}$. Let the random variable $Q^{\frac{1}{2}}(n, sn)$ denote such a randomly generate XORSAT instance over n with $\lceil sn \rceil$ XOR clauses, where $s = \frac{m}{n}$ is called the XOR density. On expectation a XOR clause in such an instance would have $\frac{n}{2}$ variables.

An at-most-k cardinality constraint is satisfiable by an assignment if and only if at most k of the n literals are set to $\top$ by that assignment. The set of satisfying assignments for this constraint forms a Hamming ball of radius k, which has volume $\sum_{w=0}^{k} \binom{n}{w}$. Let $F(n, k)$ represent the CNF encoding of the at-most-k cardinality constraint over n variables. We will use $\#F(n, k)$ to represent the number of solutions to $F(n, k)$, $\#F(n, k) = \sum_{w=0}^{k} \binom{n}{w}$.

A 1-CARD-XOR formula is the conjunction of some number of XOR clauses and a cardinality constraint. For fixed positive integers k and n, and a fixed positive real number s, let the random variable $\psi^{\frac{1}{2}}(n, k, sn)$ denote the formula $Q^{\frac{1}{2}}(n, sn) \wedge F(n, k)$.

We use $\Pr[E]$ to denote the probability of an 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 $\lim_{n\to\infty} \Pr[E_n] = 1$. Let $H(\mu) = -\mu \log_2(\mu) - (1-\mu)\log_2(1-\mu)$ denote the binary entropy function.

3 Related Work

Phase Transitions

Motivated from statistical physics, the study of satisfiability of random constraint satisfaction problems led to the observation of a phase transition behavior [Kirkpatrick and Selman, 1994]. In particular, the probability of satisfiability was observed to undergo a sharp transition from one to zero at the *critical* density, defined as the ratio of number of clauses to number of variables. For example, for random 3-SAT, the critical point was observed at density 4.26. Theoretical investigations into the location of random k-SAT have led to the precise identification of density for $k = 2$ and existence of a sharp transition for large k [Ding *et al.*, 2015].

Furthermore, a phase-transition phenomenon has also been identified in random l-XOR formulas (conjunctions of XOR constraints of length l), for $l \geq 1$, without specifying an exact location for the phase-transition [Creignou and Daudé, 2003].

Pittel and Sorkin [2016] identified the location of the phase transition for l-XOR formulas for $l > 3$. Dudek *et al.* [2016] first studied the satisfiability threshold for the conjunction of random k-CNF and random variable-width XOR formulas. As is the case with random k-CNFs, experiments confirm that the hardest instances are at the critical threshold for k-CNF-XOR formulas [Dudek *et al.*, 2017]. To the best of our knowledge, no prior work exists regarding the study of phase transition for a formula with one cardinality constraint in conjunction with a set of random variable-width XOR constraints (1-CARD-XOR formulas).

Cardinality Encodings

Cardinality (CARD) constraints naturally arise in many different contexts, such as computer tomography [Gardner *et al.*, 1999], MaxSAT algorithms [Fu and Malik, 2006], radio frequency assignment [Yang and Dong, 2013], product configuration [Yang and Dong, 2013], program repair [Joshi and Kroening, 2015] and weighted counting problems [Duenas-Osorio *et al.*, 2017]. Due to their ubiquity in several application domains, several encodings have been developed which translate them into the Boolean CNF form such as the Totalizer encoding [Bailleux and Boufkhad, 2003], the Sequential counter [Sinz, 2005], Adder [Eén and Sörensson, 2006], BDD based encoding [Bailleux *et al.*, 2006], Cardinality Networks [Achá *et al.*, 2009], and the like. These encodings exhibit different characteristics in terms of their size (e.g., number of clauses and number of variables) and whether they preserve arc-consistency, i.e., the solver is able to detect inconsistencies by unit propagation alone [Zhang and Yap, 2000]. Therefore, we focus on observing the repeatability of behavior across different encodings before drawing a conclusion in our study.

4 Establishing a Phase-Transition

In this section we will first define and show the existence of a phase transition phenomenon in 1-CARD-XOR formulas. A phase transition boundary is defined by a function ϕ such that, w.h.p., a random formula with $s < \phi$ is satisfiable, and becomes unsatisfiable as soon as $s > \phi$. Note that the random variable denoting a 1-CARD-XOR formula is $\psi^{\frac{1}{2}}(n, k, sn) = F(n, k) \wedge Q^{\frac{1}{2}}(n, sn)$.

Theorem 1. *([Dudek* et al.*, 2016], Theorem 1)*
If $\phi(k/n) = \frac{1}{n}\log_2(\#F(n, k))$, then for all $k \geq 1$ and $s \geq 0$:

(a). If $s < \phi(k/n)$, then w.h.p. $\psi^{\frac{1}{2}}(n, k, sn)$ is satisfiable.

(b). If $s > \phi(k/n)$, then w.h.p. $\psi^{\frac{1}{2}}(n, k, sn)$ is unsatisfiable.

This is a special case of Theorem 1 given in [Dudek *et al.*, 2016], which establishes the existence of a phase transition for random CNF-XOR formulas. The region of satisfiability is sharply separated from the region of unsatisfiability by the function $\phi(k/n)$. Since we give an explicit function ϕ in Lemma 5 and 7, we are able to show sharp numerical bounds on the phase transition boundary. We plot the transition function, $\phi(k/n)$, with a red line in all figures in this paper.

Next, we use a result from [Dudek *et al.*, 2016], which gives us the probability of a CNF being satisfiable when conjuncted with some number of XOR constraints, in terms of the count of solutions of that CNF. Using Theorem 2 we relate the satisfiability threshold with the model count of any formula when conjuncted with *random* XORs.

Theorem 2. *([Dudek* et al.*, 2016] , Lemma 7 and 12)*
Let $\alpha \geq 1$, $s \geq 0$, $n \geq 0$, and let F be a formula defined over $\{X_1, \ldots, X_n\}$. Then

(a). $\Pr\left[F \wedge Q^{\frac{1}{2}}(n, sn) \text{ is satisfiable} \;\middle|\; \#F \geq 2^{\lceil sn \rceil + \alpha}\right] \geq 1 - 2^{-\alpha}$.

(b). $\Pr\left[F \wedge Q^{\frac{1}{2}}(n, sn) \text{ is unsatisfiable} \;\middle|\; \#F \leq 2^{\lceil sn \rceil - \alpha}\right] \geq 1 - 2^{-\alpha}$.

Since the bounds given in Theorem 1 are not in closed form, we provide analytic bounds which are weaker. The separation in the lower and upper bound is exactly 1 for $k > \lfloor n/2 \rfloor$ while for $k \leq \lfloor n/2 \rfloor$ the separation is $\mathcal{O}(\log(n)/n)$

Theorem 3. *$\psi^{\frac{1}{2}}(n, k, sn)$ is satisfiable w.h.p. if:*

(a). $s < H(k/n) - \log(8k(1 - k/n))/n$, and $0 < k < n/2$

(b). $s < 1 - 1/n$, and $n/2 \leq k \leq n$

$\psi^{\frac{1}{2}}(n, k, sn)$ is unsatisfiable w.h.p. if:

(c). If $s > H(k/n)$, and $0 < k < n/2$

(d). If $s > 1$, and $n/2 \leq k \leq n$

Proof. Part(a) and (b) follow from Lemma 6 and Part(c) and (d) follow from Lemma 8 presented in Sections 4.1 and 4.2 respectively. □

We will use the facts that $\#F(n, k) = \sum_{w=0}^{k}\binom{n}{w}$ and $\psi^{\frac{1}{2}}(n, k, sn) = F(n, k) \wedge Q^{\frac{1}{2}}(n, sn)$.

We use a commonly known bound for summation of binomial coefficients,

Lemma 4. *([MacWilliams and Sloane, 1978], Lemma 10.8).*
$2^{nH(k/n)}/(8k(1 - k/n)) \leq \sum_{w=0}^{k}\binom{n}{w} \leq 2^{nH(k/n)}$,
for $0 < k \leq n/2$ and for all $n \geq 1$.

4.1 Lower bound

Lemma 5. *Let $k \geq 2$ and $s \geq 0$. If $s < \frac{1}{n}log_2\#F(n, k)$ as $\lim n \to \infty$, then w.h.p. $\psi^{\frac{1}{2}}(n, k, sn)$ is satisfiable.*

Proof. Since $2^s < \#F(n, k)^{1/n}$ we can choose $\delta > 0$ and $N > 0$ such that $2^{s+\delta+1/N} < \#F(n, k)^{1/n}$. We can always find a small enough δ and a sufficiently large N such that this is true. Since we are concerned with behavior asymptotic in n, we consider only $n > 2N$. Then we have $2^{sn+\delta n+2} < \#F(n, k)$ and so $2^{\lceil sn \rceil + \delta n + 1} < \#F(n, k)$. Let $\alpha = \delta n + 1$, so we get $2^{\lceil sn \rceil + \alpha} \leq \#F(n, k)$. Using Theorem 2a we see that $\Pr\left[\psi^{\frac{1}{2}}(n, k, sn) \text{ is SAT} \;\middle|\; \#F(n, k) \geq 2^{\lceil sn \rceil + \alpha}\right] \geq 1 - 2^{-\alpha}$. Since $\lim\limits_{n\to\infty} 1 - 2^{-\delta n - 1}$ converges to 1, $\psi^{\frac{1}{2}}(n, k, sn)$ is satisfiable w.h.p. □

Lemma 6. *For $s \geq 0$ and $0 < k < n$ and $\lim n \to \infty$, $\psi^{\frac{1}{2}}(n, k, sn)$ is satisfiable w.h.p. if:*

(a). $k \leq n/2$ *and* $s < H(k/n) - \log(8k(1-k/n))/n$

(b). $k > n/2$ *and* $s < 1 - 1/n$

Proof. For $n/2 < k \leq n$, $\#F(n,k) > 2^{n-1}$. Observing that $s < 1 - 1/n < \frac{1}{n}\log_2 \#F(n,k)$ we see that Part(b) is an immediate consequence of Lemma 5.

Using the bound shown in Lemma , for $0 < k < n/2$ we get that . Since $2^s < 2^{H(k/n)-\log(8k(1-k/n))/n}$ we can choose $\delta > 0$ and $N > 0$ such that $2^{s+\delta+1/N} < 2^{H(k/n)-\log(8k(1-k/n))/n}$. We can always find a small enough δ and a sufficiently large N such that this is true. Since we are concerned with behavior asymptotic in n, we consider only $n > 2N$. Then we have $2^{sn+\delta n+1} < 2^{nH(k/n)-\log(8k(1-k/n))}$ and so $2^{\lceil sn \rceil+\delta n+1} < 2^{nH(k/n)-\log(8k(1-k/n))}$. Let $\alpha = \delta n + 1$, so we get $2^{\lceil sn \rceil+\alpha} < 2^{nH(k/n)-\log(8k(1-k/n))}$. Using Theorem 2a we see that $\Pr\left[\psi^{\frac{1}{2}}(n,k,sn) \text{ is SAT} \mid 2^{nH(k/n)-\log(8k(1-k/n))} \geq 2^{\lceil sn \rceil+\alpha}\right] \geq 1 - 2^{-\alpha}$. Since $\lim\limits_{n\to\infty} 1 - 2^{-\delta n-1}$ converges to 1, $\psi^{\frac{1}{2}}(n,k,sn)$ is satisfiable w.h.p. □

4.2 Upper bound

Lemma 7. *Let* $k \geq 2, s \geq 0$, *and* $r \geq 0$. *If* $s > \lim\limits_{n\to\infty} \frac{1}{n}log_2\#F(n,k)$, *then w.h.p.* $\psi^{\frac{1}{2}}(n,k,sn)$ *is unsatisfiable.*

Proof. Since $2^s > \#F(n,k)^{1/n}$ we can choose $\delta > 0$ and $N > 0$ such that $2^{s-\delta-1/N} > \#F(n,k)^{1/n}$. We can always find a small enough δ and a sufficiently large N such that this is true. Since we are concerned with behavior asymptotic in n, we consider only $n > N$. Then we have $2^{sn-\delta n-1} > \#F(n,k)$ and so $2^{\lceil sn \rceil-\delta n-1} > \#F(n,k)$. Let $\alpha = \delta n + 1$, so we get $2^{\lceil sn \rceil-\alpha} > \#F(n,k)$. Using Theorem 2b we see $\Pr\left[\psi^{\frac{1}{2}}(n,k,sn) \text{ is UNSAT} \mid \#F(n,k) \geq 2^{\lceil sn \rceil-\alpha}\right] \geq 1 - 2^{-\alpha}$. Since $\lim\limits_{n\to\infty} 1 - 2^{-\delta n-1}$ converges to 1, $\psi^{\frac{1}{2}}(n,k,sn)$ is satisfiable w.h.p. □

Lemma 8. *For* $s \geq 0$, $0 \leq k \leq n$ *and* $\lim n \to \infty$, $\psi^{\frac{1}{2}}(n,k,sn)$ *is unsatisfiable w.h.p. if:*

(a). $k < n/2$ *and* $s \geq H(k/n)$

(b). $k \geq n/2$ *and* $s > 1$

Proof. For $n/2 \leq k < n$ observing that $s > 1 > \frac{1}{n}\log_2 \#F(n,k)$ we see that Part(b) is an immediate consequence of Lemma 7.

Since $2^s > 2^{H(k/n)}$ we can choose $\delta > 0$ and $N > 0$ such that $2^{s-\delta-1/N} > 2^{H(k/n)}$. We can always find a small enough δ and a sufficiently large N such that this is true. Since we are concerned with behavior asymptotic in n, we consider only $n > N$. Then we have $2^{sn-\delta n-1} > 2^{nH(k/n)}$ and so $2^{\lceil sn \rceil-\delta n-1} > 2^{nH(k/n)}$. Let $\alpha = \delta n + 1$, so we get $2^{\lceil sn \rceil-\alpha} > 2^{nH(k/n)}$. Using Theorem 2b we see that $\Pr\left[\psi^{\frac{1}{2}}(n,k,sn) \text{ is UNSAT} \mid 2^{nH(k/n)} \geq 2^{\lceil sn \rceil-\alpha}\right] \geq 1 - 2^{-\alpha}$. Since $\lim\limits_{n\to\infty} 1 - 2^{-\delta n-1}$ converges to 1, $\psi^{\frac{1}{2}}(n,k,sn)$ is satisfiable w.h.p. □

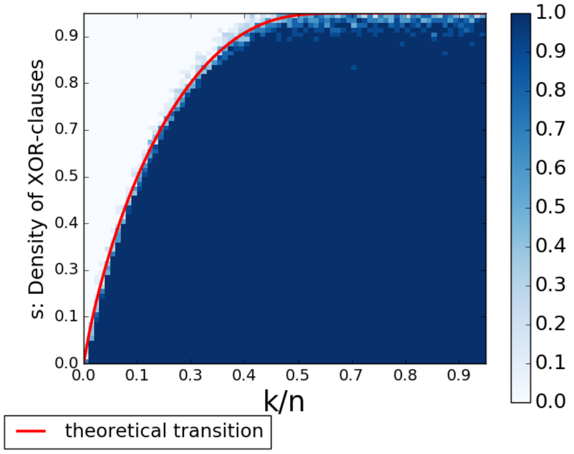


Figure 1: This plot shows the satisfiability behavior for $n = 75$. The darker shade of blue indicates the instances which were satisfiable with higher probability and the red line shows the theoretically derived phase transition. (Best viewed in color)

5 Experimental Results

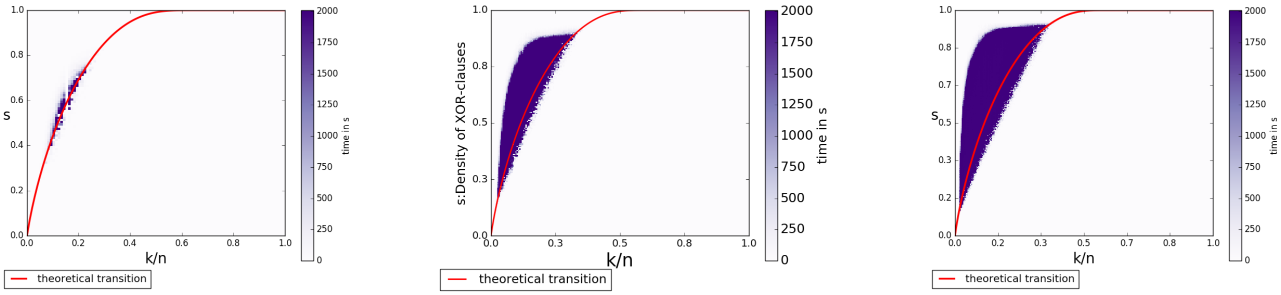
Experimental Setup

For every value of n, k and s we generate $\psi^{\frac{1}{2}}(n,k,sn)$ in the following manner. To uniformly choose an XOR-clause, we include each variable of $\{x_1, \cdots, x_n\}$ with probability $\frac{1}{2}$ in the XOR clause. We then choose exactly one of $\{0,1\}$ with probability $\frac{1}{2}$ in the XOR clause. We repeat such uniform sampling of XOR clauses $\lceil sn \rceil$ many times and conjunct all such XOR-clauses to build $Q^{\frac{1}{2}}(n,sn)$. Finally, we conjuct $Q^{\frac{1}{2}}(n,sn)$ with $F(n,k)$ to generate $\psi^{\frac{1}{2}}(n,k,sn)$.

We performed experiments with 9 values of n. For each $n \in \{50, 75, 100, 125, 150, 175, 200, 250, 300\}$ we generated an instance of 1-CARD-XOR for all values of $k \in [0,n]$ and $\lceil sn \rceil \in [1,n]$. We repeated the experiments 10 times for each data point with $n = \{75, 100\}$ to get a finer estimate on the satisfiability at the transition threshold. We were not able to run experiments for values of n significantly larger than those listed above due to computational constraints.

Since CryptoMiniSAT [Soos *et al.*, 2009] is a specialized solver which handles XOR clauses in combination with CNF clauses quite efficiently, we use it as the SAT solver in our experiments. We used a high performance computing cluster of 25 nodes for our experiments. Each node consisted of an Intel® Xeon® E5-2690 v3 CPU with 24 cores and 96GB of RAM divided evenly among the cores, with each core having access to 4GB of RAM. Each experiment was conducted on a single core. Our experimental evaluation used over 80,000 CPU hours.

The data and scripts associated with this project are available at `https://github.com/meelgroup/1-CARD-XOR`



$n = 100$ $n = 200$ $n = 300$

Figure 2: Three plots for cardnet encoding. Each plot shows the runtime behavior for a different number of variables in the following order: (a) $n = 100$, (b) $n = 200$, and (c) $n = 300$. The purple region is where the blowup in runtime was observed. The red line indicates the phase transition. (Best viewed in color)

It is known that encoding cardinality constraint using different encodings may result not only in different sizes for $F(n, k)$ but also may have an impact on the performance of the solver. To investigate the impact of various encodings of cardinality constraints for 1-CARD-XOR formulas, for each value of n, s and k we experimented with three cardinality encodings.

- Adder: $\mathcal{O}(n)$ clauses, no arc consistency [Eén and Sörensson, 2006]
- BDD: $\mathcal{O}(n \cdot k)$ clauses, preserves arc-consistency, is equivalent to $LT_{SEQ}^{n,k}$ encoding [Sinz, 2005]
- Cardinality Network: $\mathcal{O}(n \cdot log^2 k)$ clauses, preserves arc consistency [Achá *et al.*, 2009]

In our experiments, we have used the PBLib [Philipp and Steinke, 2015] tool to encode our constraints. A timeout of 2000 seconds was used for all experiments.

Polarity Caching

The SAT solver goes through an iterative process of search and inference. During search, it selects an unassigned variable and then decides on the truth value (polarity) to be given to this variable. During inference phase it explores the implications of these choices until we either get a contradiction or a satisfying assignment or nothing further can be inferred and the solver has to make another variable selection. Polarity selection heuristics decide which truth value, either `true` or `false` to assign to the selected variable. When the solver backtracks due to a contradiction, it must explore the other choice for the truth value. A heuristic which works very well in practice is polarity caching [Pipatsrisawat and Darwiche, 2007], which involves remembering the previous successful choice made on a particular variable. Another simpler heuristic is setting the polarity to `false`, which instructs the solver to always explore the `false` branch before the `true`. As discussed later, setting the polarity to `false` always significantly improves the runtime performance of the solver. Therefore, the experiments concerning runtime performance of SAT solver with respect to other parameters were performed with setting polarity flag to `false` in CryptoMiniSAT.

Results The objective of our experimental evaluation is to answer these four research questions:

RQ1. How does the satisfiability of $\psi^{\frac{1}{2}}(n, k, sn)$ vary, as parameters k and XOR clause density s vary ?

RQ2. How does the runtime performance of SAT solver for $\psi^{\frac{1}{2}}(n, k, sn)$ vary with respect to n, k, s?

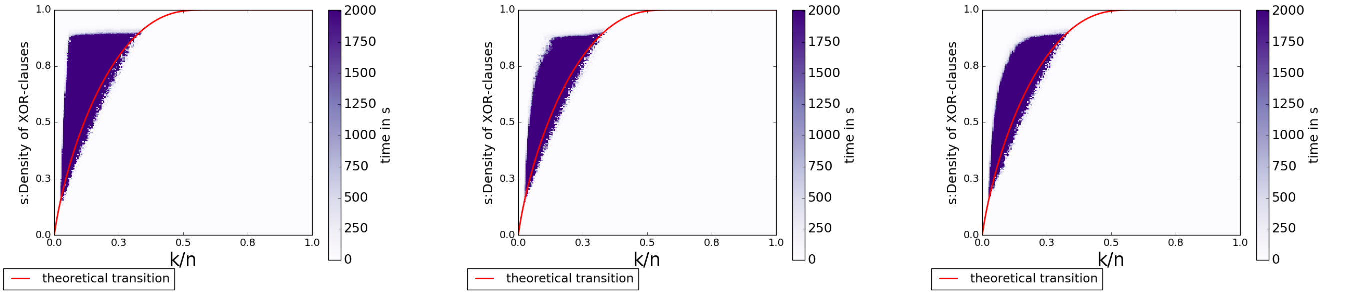
RQ3. How do the different encodings affect the runtime performance of the SAT solver for $\psi^{\frac{1}{2}}(n, k, sn)$?

RQ4. How do the different branching heuristics affect the runtime performance of a SAT solver on $\psi^{\frac{1}{2}}(n, k, sn)$?

We now first present detailed analysis below and then summarize our main conclusions.

RQ1. Figure 1 shows the satisfiability of random instances of $\psi^{\frac{1}{2}}(n, k, sn)$ as the density of XOR-clauses s and the upper bound on at-most-k constraint k varies. The y-axis indicates the density s and the x-axis indicates k/n. Each point on the plot is color-coded to represent satisfiability of the corresponding $\psi^{\frac{1}{2}}(n, k, sn)$. The dark blue color indicates that the corresponding formula is satisfiable while light color indicates unsatisfiability. The red curve represents the theoretical phase transition curve, obtained in Section 4, i.e., a point in the region under the curve is likely to be satisfiable while a point in the region over the curve is likely to be unsatisfiable. We observe that the empirically observed behavior agrees with the analysis, thus demonstrating the tightness of our analysis.

RQ2. We now turn our attention to a study of scaling behavior of runtime performance of SAT solver. Figure 2 shows the runtime performance for increasing values of n, i.e., $n \in \{100, 200, 300\}$. Similar to Figure 1, the x-axis indicates k/n while the y-axis indicates the density s. Note that most of the instances were solvable in just 2 seconds but the instances within the purple region timed out for timeout of 2000 seconds. Furthermore, we observe that the area of *hard* instances increases with n. The sudden increase in runtime around the phase transition is reminiscent of similar behavior for random CNF instances but is unlike the behavior observed in case of CNF-XOR formulas [Dudek *et al.*, 2017]. This behavior necessitates further research for a deeper understanding, and we hope this deeper understanding would be useful in the design of efficient CARD-XOR solvers.

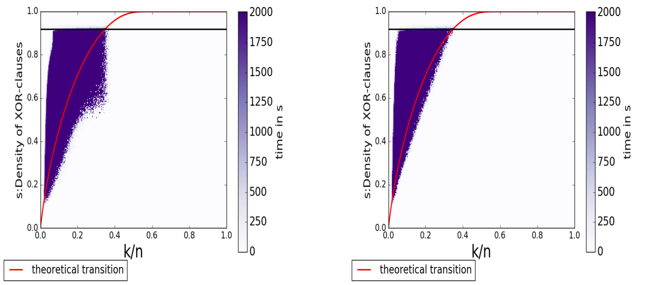


adder

bdd

cardnet

Figure 3: Three plots for $n = 200$. Each plot shows the runtime behavior for a different encoding in the following order: (a) adder, (b) bdd, and (c) cardnet. The purple region is where the blowup in runtime was observed. The red line indicates the phase transition.



(a)

(b)

Figure 4: $n = 250$, adder. The purple region shows the hard instances for the solver when (a) polarity caching was used and (b)when the solver was made to explore the $\perp$ branch first.

RQ3. Given the widespread interest in the design of different encodings for CARD constraints, it is natural to ask whether the runtime behavior of $\psi^{\frac{1}{2}}(n, k, sn)$ is sensitive to encodings. To this end, Figure 3 shows the runtime performance of SAT solvers for the above mentioned three different encodings: adder, bdd, and cardnet. Similar to Figure 1, the x-axis indicates k/n while the y-axis indicates the density s. We observe that while the area of the purple region deviates slightly across for different encodings, the qualitative behavior around phase transition regions very similar.

RQ4. We now turn to the question, what effect do the branching heuristics have on the runtime behavior of SAT solver for $\psi^{\frac{1}{2}}(n, k, sn)$. Figure 4(a) shows the behavior when polarity caching was used while Figure 4(b) shows the behavior when the solver always set the `polarity` flag to `false`. The value of n is set to 200 for both the cases. Interestingly, we observe a significant reduction in the area of the purple region by always setting `polarity` flag to `false`, which is surprising given polarity caching has shown to achieve significant gains for SAT solving. A detailed study of different heuristics is beyond the scope of this work and is left for future work.

6 Conclusion

In this paper, we study 1-CARD-XOR formulas, which are expressed as a conjunction of cardinality constraints and XOR constraints. The CARD-XOR formulas are ubiquitous in several domains of interest such as their close relationship to maximum likelihood decoding and their importance in hashing-based techniques for discrete integration. Our study revealed the empirical existence of phase transition region of satisfiability of random 1-CARD-XOR formulas for which we were able to establish tight theoretical bounds.

The investigation into runtime behavior led to the surprising discovery of behavior reminiscent of random CNF formulas but significantly different from recent studies on CNF-XOR formulas. Furthermore, we observed that despite significant interest in CP/SAT communities devoted to design of encodings, the qualitative nature of runtime behavior remains consistent across different encodings. Finally, we discover a significant impact of branching heuristics on the runtime behavior. Similar to other CSP problems where the study of phase transition have led to development of algorithmic insights, our study opens future directions into the development of algorithmic techniques for efficient CARD-XOR solvers in practice.

Acknowledgements

TThis research is supported in part by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: [AISG-RP-2018-005]), the NUS ODPRT Grant [R-252-000- 685-133] and SERB, DST, India through [ECR 2017/001126]. The computational work for this article was performed on resources of the National Supercomputing Centre, Singapore. `https://www.nscc.sg/`.

References

[Achá *et al.*, 2009] Roberto Javier Asín Achá, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. Cardinality networks and their applications. In *SAT*, 2009.

[Achlioptas and Coja-Oghlan, 2008] Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *FOCS*, 2008.

[Arora *et al.*, 1997] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 1997.

[Bacchus *et al.*, 2003] Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Algorithms and complexity results for #SAT and Bayesian inference. In *FOCS*, 2003.

[Bailleux and Boufkhad, 2003] Olivier Bailleux and Yacine Boufkhad. Efficient CNF encoding of boolean cardinality constraints. In *CP*, 2003.

[Bailleux *et al.*, 2006] Olivier Bailleux, Yacine Boufkhad, and Olivier Roussel. A translation of pseudo boolean constraints to SAT. *JSAT*, 2006.

[Berlekamp *et al.*, 1978] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE TIT*, 1978.

[Chakraborty *et al.*, 2013] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. A scalable approximate model counter. In *CP*, 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 *AAAI*, 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 *IJCAI*, 2016.

[Chase, 1985] D. Chase. Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Transactions on Communications*, 1985.

[Chvátal and Reed, 1992] Vašek Chvátal and Bruce Reed. Mick gets some (the odds are on his side). In *FOCS*, 1992.

[Coja-Oghlan and Panagiotou, 2013] Amin Coja-Oghlan and Konstantinos Panagiotou. Going after the k-sat threshold. In *Proc. of STOC*, 2013.

[Creignou and Daude, 1999] Nadia Creignou and Hervé Daude. Satisfiability threshold for random XOR-CNF formulas. *Discrete Applied Mathematics*, 1999.

[Creignou and Daudé, 2003] Nadia Creignou and Hervé Daudé. Smooth and sharp thresholds for random k-XOR-CNF satisfiability. *RAIRO*, 2003.

[Dechter and Meiri, 1994] Rina Dechter and Itay Meiri. Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. *Artificial Intelligence*, 1994.

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

[Domshlak and Hoffmann, 2007] Carmel Domshlak and Jörg Hoffmann. Probabilistic planning via heuristic forward search and weighted model counting. *JAIR*, 2007.

[Dudek *et al.*, 2016] Jeffrey Dudek, Kuldeep S. Meel, and Moshe Y. Vardi. Combining the k-CNF and XOR phase-transitions. In *IJCAI*, 2016.

[Dudek *et al.*, 2017] Jeffrey Dudek, Kuldeep S. Meel, and Moshe Y. Vardi. The hard problems are almost everywhere for random cnf-xor formulas. In *IJCAI*, 2017.

[Duenas-Osorio *et al.*, 2017] Leonardo Duenas-Osorio, Kuldeep S. Meel, Roger Paredes, and Moshe Y. Vardi. Counting-based reliability estimation for power-transmission grids. In *AAAI*, 2017.

[Ermon *et al.*, 2013] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *ICML*, 2013.

[Ermon *et al.*, 2014] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Low-density parity constraints for hashing-based discrete integration. In *ICML*, 2014.

[Eén and Sörensson, 2006] Niklas Eén and Niklas Sörensson. Translating pseudo-boolean constraints into sat. *Journal on Satisfiability, Boolean Modeling and Computation*, 2006.

[Fu and Malik, 2006] Zhaohui Fu and Sharad Malik. On solving the partial max-sat problem. In *SAT*, 2006.

[Gardner *et al.*, 1999] R.J. Gardner, Peter Gritzmann, and D Prangenberg. On the computational complexity of reconstructing lattice sets from their x-rays. *Discrete Mathematics*, 1999.

[Gomes *et al.*, 2009] C. P. Gomes, A. Sabharwal, and B. Selman. Model counting. In *Handbook of Satisfiability*. 2009.

[Jerrum and Sinclair, 1996] Mark R. Jerrum and Alistair Sinclair. The Markov Chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*, 1996.

[Joshi and Kroening, 2015] Saurabh Joshi and Daniel Kroening. Property-driven fence insertion using reorder bounded model checking. In *FM*, 2015.

[Kirkpatrick and Selman, 1994] Scott Kirkpatrick and Bart Selman. Critical behavior in the satisfiability of random boolean expressions. *Science*, 1994.

[MacWilliams and Sloane, 1978] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. 1978.

[Madras and Piccioni, 1999] N. Madras and M. Piccioni. Importance sampling for families of distributions. *Annals of applied probability*, 1999.

[Murphy, 2012] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[Philipp and Steinke, 2015] Tobias Philipp and Peter Steinke. Pblib – a library for encoding pseudo-boolean constraints into CNF, 2015.

[Pipatsrisawat and Darwiche, 2007] Knot Pipatsrisawat and Adnan Darwiche. A lightweight component caching scheme for satisfiability solvers. In *SAT*, 2007.

[Pittel and Sorkin, 2016] Boris Pittel and Gregory B. Sorkin. The satisfiability threshold for k-xorsat. *Combinatorics, Probability & Computing*, 2016.

[Sang *et al.*, 2004] Tian Sang, Fahiem Bacchus, Paul Beame, Henry A Kautz, and Toniann Pitassi. Combining component caching and clause learning for effective model counting. In *SAT*, 2004.

[Sinz, 2005] Carsten Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *CP*, 2005.

[Soos and Meel, 2019] Mate Soos and Kuldeep S. Meel. Bird: Engineering an efficient CNF-XOR sat solver and its applications to approximate model counting. In *AAAI*, 2019.

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

[Stockmeyer, 1983] Larry Stockmeyer. The complexity of approximate counting. In *STOC*, 1983.

[Tal and Vardy, 2015] I. Tal and A. Vardy. List decoding of polar codes. *IEEE TIT*, 2015.

[Yang and Dong, 2013] Dong Yang and Ming Dong. Applying constraint satisfaction approach to solve product configuration problems with cardinality-based configuration rules. *JIM*, 2013.

[Zhang and Yap, 2000] Yuanlin Zhang and Roland H. C. Yap. Arc consistency on n-ary monotonic and linear constraints. In *CP*, 2000.