ON THE SPINE OF 3-XORSAT

DAVID SOLYMOSI

Abstract. The spine of a constraint satisfaction formula is the set of variables which are forced to take on only one value by some subformula. Introduced as a tool to study resolution complexity, it has applications in analyzing algorithms on CSPs as well as the structure of random formulas. We show that the spine of 3-XORSAT exhibits a sharp threshold at the point where satisfiability does; the size of the spine is $o(n^{2/3})$ before this threshold, and at least cn for some $c > 0$ after the threshold.

1. Introduction

A k-XORSAT formula $\mathcal{F}$ on n boolean variables $v_1, \dots v_n$ is a collection of XOR constraints on subsets of the variables of size exactly k. An example of such a constraint could be $v_1 \oplus \overline{v_2} \oplus v_3$, where $\oplus$ stands for the XOR operation. Each constraint is called a k-clause, or simply a clause when k is clear from the context, and we say a clause is satisfied by an assignment of the variables if the XORSAT constraint evaluates to true when the assignments are substituted for the variables. We say $\mathcal{F}$ is satisfiable if there exists some assignment of variables such that every clause in $\mathcal{F}$ is satisfied under this assignment.

k-XORSAT is different from many other constraint satisfaction problems in that it can be interpreted as a system of linear equations. If instead of boolean variables we let each variable take values from $\mathbb{F}_2$, then each clause can be converted into a linear equation over $\mathbb{F}_2$ in a straightforward way. Our previous example would change from $v_1 \oplus \overline{v_2} \oplus v_3$ to $v_1 + (1 - v_2) + v_3 = 1$, or equivalently $v_1 + v_2 + v_3 = 0$. This interpretation leads to a polynomial time algorithm for finding a satisfying assignment for an instance of a k-XORSAT formula, or for showing that none exists, through Gaussian elimination.

1.1. Random XORSAT.

Much interest lies in random instances of k-XORSAT. Not only is it an important example in the general family of random constraint satisfaction problems which are extensively studied in computer science, but random XORSAT has interesting applications in physics, where it is known as the zero temperature k-spin model [21, 15].

Random constraint satisfaction problems have been studied extensively in the past 30 years (see Chapter 8 of [5] for a recent survey of the area). A typical question asks for the probability of some property holding for a CSP chosen uniformly at random, on n variables with $f(n)$ clauses, as n tends to infinity. When this probability tends to 1, we say this property holds *with high probability*, or w.h.p. for short. The most natural property to ask is satisfiability—how many clauses can we to pick while still ensuring the formula is satisfiable with high probability? How many do we need to pick to ensure it is unsatisfiable with high probability? Other

properties, such as algorithms succeeding, or the solution space taking on certain structure are also well studied (see e.g. [2, 8] and [3, 16] respectively).

Perhaps the most famous problem in random constraint satisfaction is the satisfiability threshold for k-SAT. It is believed that for each $k \geq 3$ there exists come r_k such that a random k-SAT equation on n vertices with cn clauses is

(i) with high probability satisfiable if $c < r_k$, and
(ii) with high probability unsatisfiable if $c > r_k$.

Recently in [13] this was shown to be true for very large k, but the question remains open for smaller cases.

In the case of k-XORSAT, such a result is known. For $k = 3$, the 3-XORSAT satisfiability threshold, which we denote c_{sat}, was first shown in [14]. The authors examined what is called the *2-core* of the formula, the largest subformula in which every variable is in at least two clauses. The aptly named 2-core contains all of the 'difficulty' in satisfying the formula: If a variable is present in only one clause, then that clause can always be satisfied by a good choice of that variable, without having any effect on other clauses. Thus the subformula obtained by removing that one variable and the clause that contains it will be satisfiable iff the original formula was satisfiable. Iterating this, if we repeatedly remove any variables which are in at most one clause, we end up with the 2-core, which is satisfiable iff the original formula was.

By [22] we know that w.h.p. the 2-core of a random 3-XORSAT instance with cn clauses on n variables contains

$$\left(1 - e^{-x}(1 + x)\right) n + o(n)$$

variables, and

$$\frac{x}{3}\left(1 - e^{-x}\right) n + o(n)$$

clauses, where x is the largest solution to

$$6c = \frac{2x}{(1 - e^{-x})^2}.$$

It was shown in [14] that if the 2-core of a random 3-XORSAT formula contains more clauses than variables, then it is unsatisfiable with high probability, while if the 2-core contains fewer clauses than variables, then it is satisfiable with high probability. Showing satisfiability was much harder, and required maximizing a complicated function to approximate the second moment of the number of satisfying assignments. The 2-core formulas above allow us to compute c_{sat} as the value of c when the number of clauses and variables in the 2-core are equal.

While only the $k = 3$ case of random k-XORSAT satisfiability was tackled in [14], the authors stated that the same method will work for larger k. In [12] a similar second moment calculation showed that for random k-XORSAT with $k > 3$, the satisfiability threshold is the point where the number of clauses in the 2-core exceeds the number of variables.

An alternate proof of the k-XORSAT threshold was more recently used in [23], relying instead on the linear system interpretation to examine the number of sets of equations which may lead to a contradiction. The formula is reduced to the 2-core before analysis in this paper as well.

1.2. **Algorithms and the Spine.** Although finding the k-XORSAT satisfiability threshold seems simpler than finding the k-SAT threshold, the reason for this is not the polynomial time algorithm which can find a satisfying assignment for XORSAT. In [9] the authors examined what makes the proof of the satisfiability threshold for 3-XORSAT in [14] work, and defined an NP-complete CSP for which a similar second moment calculation shows the satisfiability threshold. Like the second moment calculation in [14], the global maximum of a complicated function is determined over many pages of calculations, some of which are delegated to a computer-aided proof.

While Gaussian elimination can solve any instance of XORSAT, no other efficient algorithms are known for random XORSAT which do not rely on this linear system structure in some way.

For example, in [17], WalkSAT is tested for a variety of NP-complete CSPs and random 3-XORSAT, near their respective satisfiability thresholds. WalkSAT is an algorithm which tries to satisfy a given formula by first picking a random assignment, and then trying to improve this assignment by picking an unsatisfied clause uniformly at random, and flipping the assignment for a variable in the clause. The flipped variable is usually chosen to minimize the number of unsatisfied clauses after the step.

The mean number of steps taken by WalkSAT was examined numerically in [17], as the number of variables is increased. Out of all the random CSPs tested, random 3-XORSAT took the greatest number of steps by several orders of magnitude.

One reason for this is a special type of symmetry of the clauses: Suppose we have some XORSAT formula, and an assignment for the variables. Flipping the value of one variable changes the status of all clauses it contains—if a clause was satisfied before, it becomes unsatisfied after the flip, and vica versa. This is not true for SAT, for example, where one variable alone can satisfy a clause, and changing other variables will not make it unsatisfiable.

A special case of finding solutions for rare satisfiable instances of random CSPs above the satisfiability threshold is examined in [5], where simple message-parsing algorithms work for random k-SAT, but no local or decimation-based algorithms take less than an exponential number of steps for random k-XORSAT.

Other algorithms are known to be hard for both random XORSAT and random SAT. For example, DPLL algorithms, introduced in [11], assign a value to some variable in each step, after which the formula is reduced to the remaining variables, with the clauses appropriately modified. This step is then repeated. Variants of DPLL pick the variable to be set according to different heuristics, and might allow backtracking to occur. See [1] and [2] for analysis of such algorithms on k-SAT.

Two common DPLL heuristics for random k-SAT are examined for random k-XORSAT in [10] and [8]. In the unit clause heuristic, variables present in 1-clauses are picked first and are set to satisfy those clauses, and if no 1-clauses are present, then variables are picked and set uniformly at random. In the generalized unit clause heuristic, a variable is picked uniformly from a smallest clause in the formula, and set to satisfy the clause if it is of size 1, or set uniformly to either true or false otherwise. Both heuristics are shown to take exponential time to find a satisfying assignment well below the satisfiability threshold.

A DPLL algorithm will fail or will have to backtrack if a variable is set 'wrong', that is in all solutions of the remaining formula, the variable takes on the other

value. It is natural to keep track of the set S of variables which only take on one value in the formula, so that they can be avoided. As we will later see, S only depends on which variables are in each clause, not their values. Some choices of values for these clauses will lead to a satisfiable formula, while others will not. The set S will be constant among all of these satisfiable formulas, and we will show that there is a simple structural reason for these variables being in this set, which is independent of whether the formula is satisfiable or not. Therefore it is useful to define this set in a way which is already independent of satisfiability.

This brings us to the definition of the *spine*, the main topic of interest of this paper. It was introduced in [6] as a tool to study the resolution complexity of 2-SAT, and the definition carries over to XORSAT:

Definition. A variable v in the formula $\mathcal{F}$ is in the spine iff there is a satisfiable subformula $\mathcal{H}$ where the value of v is constant among all satisfying assignments.

This definition for the spine allows us to talk about the spine past the satisfiability threshold. Although we motivated the spine as a set to keep an eye on during the execution of algorithms, it is much more versatile. Since adding clauses can only increase the size of the spine, it proved to be a good set to study very close to the satisfiability threshold of 2-SAT in [6], to gain knowledge of that satisfiability transition. Moreover, it provides knowledge about the structure of the solution space of a formula. A large spine limits the solutions to live in a small subspace of all assignments, while a very small spine can only be achieved through either a large number of solutions, or solutions which differ greatly from each other.

The spine of an XORSAT instance $\mathcal{F}$ is determined only by which variables are in each clause; whether the variables are negated or not does not make a difference. Thus the spine of a formula will remain the same, even if we change each XOR clause to contain no negations, ensuring that the formula is satisfiable (with the all 1 assignment).

To see this suppose v is in the spine of $\mathcal{F}$ with $\mathcal{H}$ being a corresponding subformula as per the definition of the spine, and view $\mathcal{H}$ as a set of linear equations over $\mathbb{F}_2$. Each linear equation is of the form $v_{a_1} + v_{a_2} + \ldots + v_{a_k} = \{0, 1\}$, and since $\mathcal{H}$ has a solution it is consistent. However, adding one of $v = 0$ or $v = 1$ will make the system inconsistent, so there must be some subset of equations which sum to $v = 1$ or $v = 0$. This gives an alternate definition for the spine, as the set of variables v for which there exists a subset of linear equations which sum to $v = 0$ or $v = 1$.

This inspires yet another way of looking at a k-XORSAT formula. For each formula $\mathcal{F}$ we define a corresponding k-uniform hypergraph $G(\mathcal{F})$ on the variables, where we add one edge for each clause, containing the variables the clause contains. We lose the right hand sides of the linear equations when we look at $\mathcal{F}$ this way, but we retain all other structure. We can rephrase the spine using this hypergraph: a variable v is in the spine of the formula $\mathcal{F}$ iff $G(\mathcal{F})$ contains a (not necessarily induced) subgraph in which only v has odd degree. To see why this is equivalent, consider summing the linear equations corresponding to the hyperedges in the subgraph. Either $v = 0$ or $v = 1$ falls out as the result, since all other variables appear an even number of times and cancel out, as we are in $\mathbb{F}_2$. Conversely, if v is in the spine we know $v = 0$ or $v = 1$ can be obtained as a sum of of linear equations corresponding to a subset of clauses. Since all other variables cancel out, each variable must be in an even number of these equations. Thus taking the edges corresponding to these equations will give a hypergraph where only v has odd degree.

We generalize the spine using these observations, to define an *implied ℓ-clause* of an XORSAT formula $\mathcal{F}$ as a set of ℓ variables $v_{a_1}, v_{a_2}, \ldots v_{a_\ell}$ such that the following equivalent conditions hold:

(1) There exists a satisfiable subformula where the value of $v_{a_1} \oplus v_{a_2} \oplus \ldots \oplus v_{a_\ell}$ is constant among all satisfying assignments.
(2) When viewing $\mathcal{F}$ as a linear system, there exists a subset of equations which sum to either $v_{a_1} + v_{a_2} + \ldots v_{a_\ell} = 0$ or $v_{a_1} + v_{a_2} + \ldots v_{a_\ell} = 1$.
(3) There exists a subgraph of the corresponding hypergraph $G(\mathcal{F})$ where only the variables $v_{a_1}, v_{a_2}, \ldots v_{a_\ell}$ have odd degree.

Elements of the spine are exactly the variables which lie in implied 1-clauses.

The spine of general constraint satisfaction problems was examined in [20], but it was generalized in a way that only agrees with the spine for constraints in conjunctive normal form (i.e. SAT). For all other CSPs, the set examined is only guaranteed to contain the spine, but the converse is usually false.

2. Results

Our main result states that the size of the spine of 3-XORSAT exhibits a sharp threshold at the point where satisfiability does.

Theorem 1. *(a) For any $c > c_{sat}$ there exists a $\delta > 0$ such that the size of the spine in a random 3-XORSAT instance with cn clauses is w.h.p. at least δn.*
(b) For any $c < c_{sat}$ the size of the spine in a random 3-XORSAT instance with cn clauses is $o(n^{2/3})$ w.h.p.

This statement, with 3-XORSAT replaced with k-XORSAT appears in [20], but without a correct proof. Their argument works for a small class of CSPs, including k-SAT but not k-XORSAT. In k-SAT each clause is an OR of variables, so every variable in a minimal unsatisfiable subformula will be in the spine; we sketch the argument from [20]:
Let $\mathcal{M}$ be a minimal unsatisfiable subformula of a k-SAT instance. If X is a clause in $\mathcal{M}$, then $\overline{X}$ must be true in all satisfying assignments of $\mathcal{M} \setminus X$. Since the negation of X is an AND of variable assignments, these variables must take on only one value in all satisfying assignments of $\mathcal{M} \setminus X$, meaning that they must be in the spine. By the classic result [7], a minimal unsatisfiable subformula is w.h.p. at least of linear size in random k-SAT, which leads to a linear sized spine.

The error in [20] is that while for random k-XORSAT a minimal unsatisfiable formula is still linear sized, it might have a small spine! For example, when k is even, the spine will always be empty.

In fact an XORSAT formula $\mathcal{F}$ with only even size clauses has no implied ℓ-clauses for any odd ℓ. This is easy to see by the hypergraph formulation of the handshaking lemma. For any subformula $\mathcal{H}$ of $\mathcal{F}$ we know

$$\sum_{x \in v(G(\mathcal{H}))} deg(x) = \sum_{e \in G(\mathcal{H})} |e|.$$

Since the right hand sum is even, we cannot have an odd number of odd degree vertices.

However, the statement of Theorem 1 should be true for odd $k > 3$. Moreover, the size of the spine below the satisfiability threshold seems to be $O(1)$, but we

were not able to show this. Instead, we use a simple concentration bound to show Theorem 1(b).

The size of the spine of random 3-XORSAT was computed using numerical experiments in [25]. After ensuring that the all zero assignment is a satisfying assignment, experiments were run to find what fraction of variables are fixed to 0 in every satisfying assignment, as the number of clauses is increased. These variables will clearly be contained in the spine. Moreover, since the all zero assignment is satisfying, there will be no variables that take the value 1 in all satisfying assignments, so the set of these variables will be the spine! The numeric results indicate that before the satisfiability threshold is passed, the spine is very small, and tends to 0 as the number of variables grows, while a constant fraction of the variables are in the spine immediately after the satisfiability threshold. Theorem 1 makes these observations rigorous.

Theorem 1 was partially motivated by a study of a DPLL type algorithm in [24]. Although Theorem 1 is not strong enough to justify their observations which are in a mixed 2- and 3-XORSAT setting, we discuss what is needed and how an extension of Theorem 1 could be shown in Section 6.

We prove Theorem 1(a) in Section 5, relying on a series of results. The first step, which might be of interest in its own, is

Theorem 2. *For each $c > c_{sat}$ there exists a $\delta > 0$ such that the number of implied 3-clauses in a random 3-XORSAT instance with cn clauses is almost surely at least δn^3.*

We prove this in Section 3. Intuitively this says that a constant fraction of triples are bound by some implicit constraint past the satisfiability threshold. The XORSAT formula being a random formula is necessary, as one can easily construct a 3-XORSAT formula with cn clauses and only cn implied 3-clauses, for example by using only $(cn)^{1/3}$ variables among all clauses.

We move down from implied 3-clauses to implied 2-clauses using the following key linear algebra lemma, which is proven in Section 4. Recall that the Hamming weight of a vector in $\mathbb{F}_2^n$ is the number of non-zero entries it has, and the Hamming distance between two vectors is the number of entries they differ in.

Lemma 3. *Let V be a subspace of $\mathbb{F}_2^n$, and let $1 \leq i \leq n$. Suppose V contains d vectors of Hamming weight 3 which are non-zero at the ith entry. Then V contains at least $d - \frac{n}{2}$ vectors of Hamming weight 2.*

We will translate implied 3-clauses to vectors of Hamming weight 3, and apply this lemma with their span as V. The resulting Hamming weight 2 vectors will then correspond to implied 2-clauses, and if we know that a formula has $\Theta(n^2)$ implied 2-clauses, a simple argument will yield Theorem 1(a).

The result of $d - \frac{n}{2}$ vectors of Hamming weight 2 is tight. The existence of single error correcting Hamming codes shows this. A single error correcting Hamming code V is a subspace of $\mathbb{F}_2^n$ with special properties (see any standard coding theory textbook e.g. [26]). Of these properties, we will use the following:

(i) V contains no vectors of Hamming weight 1 or 2.
(ii) Any vector in $\mathbb{F}_2^n$ is at most Hamming distance 1 away from some vector in V.

We will show that for every i these codes contain $\lceil \frac{n-1}{2} \rceil$ vectors of Hamming weight 3 which are non-zero at the ith entry, but no vectors of Hamming weight 2.

Let V be a single error correcting Hamming code. Fix some $0 < i \leq n$, and consider all Hamming weight 2 vectors in $\mathbb{F}_2^n$ which are non-zero at the ith entry (these will not be in V). This is a set of $n-1$ vectors, and by property (ii) above each vector in this set will have a vector in V at Hamming distance 1 away from it. By property (i), these vectors in V which are Hamming distance 1 away can only be Hamming weight 3 vectors which are non-zero at the ith entry! Since one weight 3 vector can only 'cover' 2 vectors from this set, there must be at least $\lceil \frac{n-1}{2} \rceil$ Hamming weight 3 vectors which are non-zero in the ith entry.

3. Proof of Theorem 2

Recall that the *2-core* of a 3-XORSAT formula $\mathcal{F}$ is the largest subformula on a subset of the variables in which every variable is in at least two clauses. As mentioned in the introduction, the satisfiability threshold was shown by inspecting the 2-core. As the density of a random formula increases, so does the density of its 2-core. The satisfiability threshold is the point where the density of the 2-core is 1.

A clause x is said to be *dependent* in a k-XORSAT formula $\mathcal{F}$ if x is an implied clause in $\mathcal{F} \setminus x$, otherwise it is called *independent*. Note that a k-XORSAT formula with no dependent k-clauses is always satisfiable, to see this think of the formula as a system of linear equations; since each clause is independent, no sum of any subset of equations can contradict another equation in the system. In other words, the system is of full rank, and always has a solution.

On the other hand, if we are given a satisfiable formula and we add a clause which is dependent, we may or may not make the formula unsatisfiable, depending on the right hand side of the linear equation it corresponds to. If this is chosen uniformly at random, then the formula will become unsatisfiable with probability $1/2$. Note however that we cannot have more than n independent clauses on n variables, by the usual dimension bound from linear algebra.

Lemma 4. *When $0 < \gamma < 0.001$, the 2-core of a random 3-XORSAT formula on n vertices with $(c_{sat} - \gamma)n$ clauses lies on at least $0.6n$ vertices and has density greater than $1 - 2\gamma$.*

Proof. We make use of results on the 2-core from [22] mentioned in the first section. Recall that if we define x as the largest solution to the equation

$$6(c_{sat} - \gamma) = \frac{2x}{(1 - e^{-x})^2}, \tag{1}$$

then the number of variables in the 2-core is w.h.p.

$$\left(1 - e^{-x}(1 + x)\right) n + o(n),$$

and the number of clauses in the 2-core is w.h.p.

$$\frac{x}{3}\left(1 - e^{-x}\right) n + o(n).$$

Note that a smaller γ leads to a larger x, and for $0 < \gamma < 0.001$ the formula for the number of variables increases with x. Thus it suffices to show that when $\gamma = 0.001$, the size of the 2-core is at least $0.6n$, which is a straightforward calculation.

To see that the density is always greater than $1-2\gamma$, denote the ratio of clauses to variables by r,

$$r := \frac{\frac{x}{3}(1-e^{-x})}{1-e^{-x}(1+x)}. \tag{2}$$

We will show that $\frac{\partial r}{\partial \gamma} < -2$ for $0 < \gamma < 0.001$. Recall that c_{sat} was defined as the value of c when $r = 1$, so when $\gamma = 0$ we have $r = 1$. Together these two facts show that for any $0 < \gamma < 0.001$, the value of r is at most $1 - 2\gamma$.

We first determine $\frac{\partial x}{\partial \gamma}$ from (1):

$$-6 = \frac{2e^{2x}(e^x - 2x - 1)}{(e^x-1)^3} \cdot \frac{\partial x}{\partial \gamma},$$

so

$$\frac{\partial x}{\partial \gamma} = \frac{-3(e^x-1)^3}{e^{2x}(e^x-2x-1)}.$$

Then from (2) we have

$$\frac{\partial r}{\partial x} = \frac{e^{2x} - e^{-x}(x^2+2)+1}{3(x-e^x+1)^2}.$$

Thus

$$\frac{\partial r}{\partial \gamma} = \frac{\partial r}{\partial x}\frac{\partial x}{\partial \gamma} = -\frac{(e^x-1)^3(e^{2x}-e^x(x^2+2)+1)}{e^{2x}(e^x-2x-1)(x-e^x+2)^2}.$$

When $0 < \gamma < 0.001$, we have $2.14 < x < 2.15$. It is easy to check that $\frac{\partial r}{\partial \gamma}$ is increasing in this interval, so its value at $x = 2.15$ will yield an upper bound here. The result is less than -2, so $\frac{\partial r}{\partial \gamma} < -2$ when $2.14 < x < 2.15$, and thus when $0 < \gamma < 0.001$.

□

Proof of Theorem 2. Fix some $0 < \gamma < 0.001$. We pick a random 3-XORSAT formula $\mathcal{F}$ with $(c_{sat} - \gamma)n$ clauses. This formula will be satisfiable with high probability. We will add $11\gamma n$ clauses to this formula one by one, uniformly at random, and examine where the clauses fall during this process, which will make the resulting formula unsatisfiable with high probability. Define the following sets of clauses:

Let $\mathcal{C}$ be the 2-core of $\mathcal{F}$, and let S denote the set of vertices this 2-core contains.

Let $\mathcal{Z}$ be the set of all implied 3-clauses in $\mathcal{C}$.

Let $\mathcal{I}$ be the largest set of independent clauses in $\mathcal{C}$. Since our formula is below the satisfiability threshold, with high probability it is satisfiable. Recall that any random dependent clause makes the formula unsatisfiable with probability $1/2$, so we must have $\mathcal{I} = \mathcal{C}$.

As we add clauses to our formula one at a time, we define the following sets as well:

Let $\mathcal{F}_i$ denote the entire formula after the ith edge has been added.

Let $\mathcal{C}_i$ denote the subformula on S, after the ith step.

Let $\mathcal{Z}_i$ be defined the following way: Let $c := 0.0026$. We will always have $|\mathcal{Z}_i| = cn^3$. If the number of implied 3-clauses in $\mathcal{C}_i$ is at least cn^3, then let $\mathcal{Z}_i$ be a uniformly selected cn^3 sized subset of them. If the number of implied 3-clauses in $\mathcal{C}_i$ is less than cn^3, then $\mathcal{Z}_i$ contains these implied 3-clauses, and the rest of the

clauses are selected uniformly at random from all possible 3-clauses that can appear on S.

Let $\mathcal{I}_i$ be defined the following way: $\mathcal{I}_0 := \mathcal{I}$. During the process, on the ith round we define $\mathcal{I}_i := \mathcal{I}_{i-1}$, but we also add the ith edge to $\mathcal{I}_i$ if it is not in $\mathcal{Z}_{i-1}$ but is in S.

With these sets in hand, we can outline what we are interested in during the process. We want to show that the number of implied 3-clauses is larger than cn^3. Note that if this is not true, then no clause added to $\mathcal{I}_i$ lands on an implied 3-clause, by our definition of $\mathcal{Z}_i$. Since an added clause can only be dependent if it was an implied 3-clause before being added, the set $\mathcal{I}_i$ will always consist of independent clauses. However, since $\mathcal{I}$ is already so large, we will show that too many clauses fall in $\mathcal{I}_i$ during the process, more than the total number of variables, forcing some clauses to be dependent.

We examine the probability of the ith added edge to fall in $\mathcal{I}_i$ during the process.

$$P(i\text{th clause is added to } \mathcal{I}_i) \geq \left(1 - \frac{|\mathcal{Z}_{i-1}|}{\binom{|S|}{3}}\right) \cdot P(i\text{th clause lands fully in } S)$$

Note that by Lemma 4, $|S|$ is w.h.p. at least $0.6n$, so

$$\begin{aligned} P(i\text{th clause is added to } \mathcal{I}_i) &\geq \left(1 - \frac{|\mathcal{Z}_{i-1}|}{\frac{1}{6}(0.6n)^3}\right) \frac{\binom{0.6n}{3}}{\binom{n}{3}} \\ &\geq \left(1 - \frac{|\mathcal{Z}_{i-1}|}{0.036n^3}\right) 0.216 \\ &= \left(1 - \frac{c}{0.036}\right) 0.216 \\ &\geq 0.2. \end{aligned}$$

Note that during the ith round, whether this clause is added to $\mathcal{I}_i$ or not does not depend on i in any way. Furthermore, it is independent of all previous rounds. Thus if we let X denote the random variable corresponding to the number of clauses added to $\mathcal{I}_i$ for all i during the run of this process, it will follow a binomial distribution:

$$X \sim BIN(11\gamma n, p),$$

where $p > 0.2$. As we run the process at least $2.2\gamma n$ clauses are expected to land in one of the $\mathcal{I}_i$. By Lemma 4 the size of $\mathcal{C}$ is at least $(1-2\gamma)|S|$ with high probability, and since $\mathcal{I} = \mathcal{C}$ with high probability, adding these $2.2\gamma n$ clauses will make the expected size of $\mathcal{I}_{11\gamma n}$ at least $|\mathcal{I}| + E[X] \geq (1 + 0.2\gamma)|S|$.

We can show concentration for X, using a Chernoff bound (see [18]):

$$P(X \leq 2.1\gamma n) \leq e^{-2.2n\left(1-\frac{2.1}{2.2}\right)^2/3} \leq e^{-0.001n}.$$

Thus with high probability $\mathcal{I}_{11\gamma n}$ is of size at least $(1 + 0.1\gamma)|S|$. However, we cannot have an independent set of size greater than $|S|$, so the only way $\mathcal{I}_{11\gamma n}$ could be so large if there were some implied 3-clauses which were not contained in $\mathcal{Z}_i$, meaning that there were more than cn^3 implied 3-clauses!

Although we have only shown that we have cn^3 implied 3-clauses for densities between c_{sat} and $c_{sat} + 0.001$, this is enough. As we add more clauses, the number of implied 3-clauses cannot decrease, so any random formula with larger densities will have cn^3 implied 3-clauses. □

4. Proof of Lemma 3

Proof. Suppose a vector space $V \subseteq \mathbb{F}_2^n$ is given containing d vectors of Hamming weight 3 which are non-zero at the ith entry. Define a 3-uniform hypergraph H on $[n]$ where an edge is present for each vector with Hamming weight 3. The ith vertex will have d edges on it, that is, it has degree at least d. Define an auxillary graph G_i on $[n] \setminus \{i\}$, where an edge $\{x, y\}$ is present iff $\{i, x, y\}$ is an edge of H. Note that each edge of G_i corresponds to an edge of H which corresponds to a vector in V.

If G_i contains an odd cycle $a_1, a_2, \dots a_{2k+1}$, then H must contain edges

$$\{i, a_1, a_2\}, \{i, a_2, a_3\}, \dots \{i, a_{2k}, a_{2k+1}\}, \{i, a_{2k+1}, a_1\}.$$

Each a_i appears in two edges, and i appears in all $2k+1$ edges. Thus the sum of the d corresponding vectors will be the ith unit vector, as all the a_i cancel out. Therefore the ith unit vector must be in V, and we can form d Hamming weight 2 vectors by summing the ith unit vector with the d Hamming weight 3 vectors.

Now we assume G_i is bipartite. Note that if x and y are connected by an odd length path, then $\{x, y\}$ must be an edge in G_i. To see this sum the corresponding vectors in V as before, the only odd degree vertices will be x, y, and i. Thus V contains the vector which is non-zero at exactly those entries.

This means that G_i is a disjoint union of complete bipartite subgraphs $A_1 \cup B_1, \dots A_k \cup B_k$. Note that if x and y are connected by a path of length two, then V contains the vector which is non-zero exactly at x and y. Thus V contains $\sum_{j=1}^k \binom{|A_j|}{2} + \sum_{j=1}^k \binom{|B_j|}{2}$ vectors that are non-zero at exactly two entries. We want to show that this sum is close to $\sum_{j=1}^k |A_j||B_j| = d$. We compare each term of the two sums,

$$|A_j||B_j| - \binom{|A_j|}{2} - \binom{|B_j|}{2} = \frac{|A_j| + |B_j|}{2} - \frac{1}{2}(|A_j| - |B_j|)^2 \leq \frac{|A_j| + |B_j|}{2}.$$

Thus the difference in the sums is $\frac{1}{2}\sum_j |A_j| + |B_j| \leq \frac{n}{2}$, so V contains at least $d - \frac{n}{2}$ vectors that are non-zero at exactly two entries. □

Note that one can make this lemma more precise by replacing the $-\frac{n}{2}$ term with $-\frac{|N_i|}{2}$, where N_i is the set of vertices that lie in a triple where the ith entry is non-zero. The proof remains identical, but the simpler statement of the lemma suffices for our purposes.

5. Proof of Theorem 1

Suppose some 3-XORSAT formula has δn^3 implied 3-clauses for some $\delta > 0$. We can translate each 3-clause into a vector in $\mathbb{F}_2^n$, which is non-zero in 3 positions corresponding to the 3 variables it encompassed. Now we can use the following corollary of Lemma 3 with V as the span of these vectors.

We adapt Lemma 3 to talk about implied clauses instead of vector spaces.

Corollary 5. *Let $\mathcal{F}$ be a 3-XORSAT formula which contains cn^3 implied 3-clauses for some $c > 0$. Then $\mathcal{F}$ contains at least $3cn^2 - n/2$ implied 2-clauses.*

Proof. Suppose $\mathcal{F}$ contains cn^3 implied 3-clauses. In order to use the lemma, we need to define a vector space V. Let each implied 3-clause correspond to a vector

in $\mathbb{F}_2^n$, which is non-zero in 3 the positions corresponding to the 3 variables it encompassed. Let V be the span of these vectors.

Claim. V contains $3cn^2 - n/2$ vectors of Hamming weight 2.

Proof of claim. To show this we just need to find an i which is non-zero in $3cn^2$ vectors of Hamming weight 3 in V, and apply Lemma 3. As before, define the corresponding 3-uniform hypergraph H where an edge is added for each vector of Hamming weight 3, on the entries which are non-zero. H will have cn^3 edges, so the total degree in H is $3cn^3$. Since there are n vertices, one vertex must have degree at least $3cn^2$. We can let i be the entry corresponding to this vertex, since each edge it lies in corresponds to a vector of Hamming weight 3 which is non-zero at the ith entry. □

All that remains is to show that each Hamming weight 2 vector in V corresponds to an implied 2-clause. We will rely on the linear system definition for implied clauses. Suppose x is a Hamming weight 2 vector in V. Recall that for each implied 3-clause there is a set of linear equations whose sum cancels out all but the three variables. Thus if x is the sum of some vectors in V corresponding to implied 3-clauses, then the sum of all of the linear equations will yield a set of linear equations which cancels out all but the variables which are non-zero in x. In other words, every vector in V corresponds to an implied clause. □

Now we have all the tools for our main result.

Proof of Theorem 1(a). Pick any $\gamma > 0$, and let $c = c_{sat} + \gamma$. By Theorem 2 and Corollary 5 above, a random 3-XORSAT instance with cn clauses will w.h.p. have at least δn^2 implied 2-clauses for some $\delta > 0$. Suppose we add γn more 3-clauses to our formula, uniformly at random. Each clause will have a constant probability $C > 0$ to land in a way such that two of the variables it covers are contained in an implied 2-clause. To see this, draw the random 3-clause one variable at a time. There are $n(n-1)/2$ ways to draw the first two variables, so with probability $1/2\delta$ the first two variables already form an implied 2-clause. If this happens, the third vertex becomes (or remains) a member of the spine.

For convenience we will allow adding the same clause multiple times. Since there are $\binom{n}{3}$ choices for which three variables a clause binds, and at each point at most $(c+2\gamma)n$ are occupied, the probability of a clause landing on an occupied triple is

$$1 - \frac{(c+2\gamma)n}{\binom{n}{3}} \approx 1 - \frac{6(c+2\gamma)}{n^2}.$$

Over the course of adding γn clauses, the probability that at least one clause lands on an occupies triple is at most

$$1 - \left(1 - \frac{6(c+2\gamma)}{n^2}\right)^{\gamma n} = o(1).$$

Thus we can safely ignore these cases—if a property holds with probability p when allowing the same clause to be added multiple times, it will hold with probability at least $(1-o(1))p$ without it.

With this assumption, the random variable X corresponding to the number of 3-clauses which fall on these δn^2 known implied 2-clauses when adding γn clauses

follows a Binomial distribution:

$$X \sim BIN(\gamma n, C).$$

A simple Chernoff bound gives

$$P(X \leq \frac{1}{2}C\gamma n) \leq e^{-C\gamma n/12},$$

where the right hand side tends to 0 very quickly as n grows. Thus w.h.p. we will have at least $\frac{C\gamma}{2}n$ 3-clauses fall on these implied 2-clauses.

Each time a 3-clause landed on one of these δn^2 2-clauses, the third variable was uniformly chosen among all possible variables. Since each edge was added independently, this gives a uniform distribution for the variables this process guarantees to be in the spine. This is the same as if we picked these variables uniformly at random with replacement, $\frac{1}{2}C\gamma n$ times. With high probability we will not have much overlap: Picking the variables one by one, each time we will pick an unoccupied variable with probability at least $1 - \frac{1}{2}C\gamma$. Thus the probability of picking at least $\frac{1}{4}C\gamma n$ unique variables is at least as much as the probability that a random binomial variable $Y \sim BIN\left(\frac{1}{2}C\gamma n, 1 - \frac{1}{2}C\gamma\right)$ will be at least $\frac{1}{4}C\gamma n$. This, by the usual Chernoff bound gives

$$P(Y \leq \frac{1}{4}C\gamma n) \leq e^{-C'n},$$

where

$$C' = \frac{\frac{1}{2}C\gamma(2(1 - \frac{1}{2}C\gamma) - 1)^2}{12(1 - \frac{1}{2}C\gamma)}$$

is a constant greater than 0 as long as $\frac{1}{2}C\gamma < \frac{1}{2}$, which is true as $C < 1$ and $\gamma < 1$.

This shows that for any $\gamma > 0$, w.h.p. a random 3-XORSAT instance with $(c_{sat} + 2\gamma)n$ clauses has a spine of size cn for some $c = c(\gamma) > 0$. □

Proof of Theorem 1(b). We need to show that below the satisfiability threshold, we almost surely have a small spine.

Suppose that for a $\gamma > 0$ with some constant probability $p > 0$ a random 3-XORSAT formula with $(c_{sat} - 2\gamma)n$ clauses has a spine of size at least $cn^{2/3}$ for some $c > 0$. Choose $\mathcal{F}$ uniformly among all 3-XORSAT formulas with $(c_{sat} - 2\gamma)n$ clauses which has a spine of size at least $cn^{2/3}$. We will add γn more 3-clauses uniformly at random to $\mathcal{F}$, such that the formula is still below the satisfiability threshold. If we can show that with constant probability q the formula becomes unsatisfiable, that implies that a random formula with $(c_{sat} - \gamma)n$ clauses is unsatisfiable with probability $pq > 0$, a constant which does not depend on n. This leads to a contradiction, as below the satisfiability threshold a random formula is satisfiable with probability tending to 1 as n grows.

Now we examine the process. Each edge has probability

$$\frac{\binom{cn^{2/3}}{3}}{\binom{n}{3}} = \frac{c'}{n}$$

of landing entirely in the spine, where $c' > 0$ depends only on c. Thus the probability of making the formula $\mathcal{F}$ unsatisfiable is at least

$$1 - \left(1 - \frac{c'}{2n}\right)^{\gamma n}.$$

This comes from the fact that if a clause lands entirely in the spine, then it has probability 1/2 of making the entire formula unsatisfiable. Since the value of this tends to $1 - e^{-c'\gamma/2}$, for large enough n the probability of making $\mathcal{F}$ unsatisfiable is at least $(1 - e^{-c'\gamma/2})/2$, a constant which does not depend on n. This is a contradiction as our formula was chosen uniformly at random at a density smaller than the satisfiability threshold, so it should be satisfiable with probability that tends to 1. □

6. Mixed Formulas

Although the spine was introduced as a set of possible 'bad choices' during the run of a DPLL type algorithm, it is rare that an intermittent formula consists purely of clauses of size 3. Instead it will contain clauses of size 1, 2, and 3.

In [24], an algorithm similar to DPLL is inspected on random instances of 3-XORSAT. This algorithm uniformly chooses a variable at random, then sets it to satisfy any simple logical implications that are found, or uniformly at random if none are found. Then the formula is reduced, and this step is iterated.

These variables with 'simple logical implications' are the variables which would be set a certain way if we satisfied all 1-clauses repeatedly in the intermittent formula. For convenience we will call the formula resulting from repeatedly satisfying 1-clauses while they exist the reduced intermittent formula. This will only contain clauses of size 2 and 3.

A detailed analysis of the algorithm is present in [24], under some assumptions and with some rigour missing. The statement which motivated this study of the spine is restated here in a more convenient way, with some details omitted:

Hypothesis 6. *There is a linear sized discontinuity in the size of the spine*[1] *of the intermittent formula at the point where the 2-core of the reduced intermittent formula contains more clauses than variables.*

It is also stated that while the 2-core of the reduced intermittent formula contains less clauses than variables, the only variables in the spine are ones in the intermittent formula but not in the reduced intermittent formula. These variables will always be in the spine; this is because 1-clauses are implied 1-clauses, and variables in implied 1-clauses are exactly the variables in the spine.

When reduced to the reduced intermittent formula, Hypothesis 6 states there will be an empty spine before the point where the 2-clause passes density 1, and a linear number of them after. This looks very similar to Theorem 1, but with satisfiability in terms of the 2-core. To prove the hypothesis using the techniques in this paper, a statement relating the density of the 2-core and satisfiability is needed.

Hypothesis 7. *Suppose a random XORSAT formula $\mathcal{F}$ with $c_2 n$ clauses of size 2 and $c_3 n$ clauses of size 3 has a 2-core which w.h.p. contains less clauses than variables. Then w.h.p. $\mathcal{F}$ is satisfiable.*

Such a result is not known. A second moment calculation like the one used in [14] and [9] seems more difficult here than in those cases, as the function to maximize loses some nice properties when mixed size clauses are present.

[1]The set examined in [24] is actually called the backbone, but it is equal to the spine in this case.

The methods in this paper cannot show that the spine will be empty w.h.p. before the satisfiability threshold. However, if Hypothesis 7 were known, our proof techniques could be adapted to show Hypothesis 6, through the following statement:

Hypothesis 8. *Fix $c_2 > 0$, and let c_3^* denote the satisfiability threshold for a random XORSAT formula with $c_2 n$ 2-clauses and $c_3^* n$ 3-clauses*[2]*. Then*
(a) for any $c_3 > c_3^*$ there exists a $\delta > 0$ such that the spine of a random XORSAT instance with $c_2 n$ 2-clauses and $c_3 n$ 3-clauses is w.h.p. of size at least δn.
(b) for any $c_3 < c_3^*$ the spine of a random XORSAT instance with $c_2 n$ 2-clauses and $c_3 n$ 3-clauses is $o(n^{2/3})$ w.h.p.

The only significant difference between the proof of Theorem 1 and Hypothesis 8 would be Lemma 4. An analysis of the 2-core of a random mixed XORSAT formula with $c_2 n$ 2-clauses and $c_3 n$ 3-clauses would have to be carried out, analogously to the random 3-XORSAT case in [22]. Once expressions for the number of vertices in the 2-core and the number of clauses in the 2-core are obtained, what needs to be shown is that close to the point where the number of vertices is equal to the number of clauses, the ratio of clauses to variables changes at a bounded rate as the number of 3-clauses increases.

The 2-core of the reduced intermittent formula in [24] is analyzed in the appendix of that paper. However, we would need a more general result on 2-cores of arbitrary random mixed 2- and 3-XORSAT formulas in our proof. However, since this would only prove Hypothesis 8 if we had a proof for Hypothesis 7, which seems out of reach, no further work in this direction is considered here.

7. Acknowledgements

I would like to thank my advisor Mike Molloy for his guidance, helpful discussions, and encouragement. I also thank Allan Borodin for his careful reading and many helpful remarks.

References

[1] D. Achioptas and G.B. Sorkin. Optimal myopic algorithms for random 3-sat. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 590–600, 2000.
[2] Dimitris Achlioptas. Lower bounds for random 3-sat via differential equations. *Theoretical Computer Science*, 265(12):159 – 185, 2001. Phase Transitions in Combinatorial Problems.
[3] Dimitris Achlioptas, Arthur Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k sat and nae 3-sat. In *SODA*, volume 1, pages 721–722, 2001.
[4] Dimitris Achlioptas and Michael Molloy. The solution space geometry of random linear equations. *Random Structures & Algorithms*, 46(2):197–231, 2015.
[5] A. Biere, M. Heule, and H. van Maaren. *Handbook of Satisfiability*. Frontiers in Artificial Intelligence and Applications. IOS Press, 2009.
[6] Béla Bollobás, Christian Borgs, Jennifer T. Chayes, Jeong Han Kim, and David B. Wilson. The scaling window of the 2-sat transition. *Random Struct. Algorithms*, 18(3):201–256, May 2001.
[7] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, October 1988.
[8] Harold Connamacher. Exact thresholds for dpll on random xor-sat and np-complete extensions of xor-sat. *Theor. Comput. Sci.*, 421:25–55, March 2012.

[2]What we mean by this is that a random XORSAT formula with $c_2 n$ 2-clauses and $c_3 n$ 3-clauses is w.h.p. unsatisfiable when $c_3 > c_3^*$, and satisfiable with probability bounded away from 0 when $c_3 < c_3^*$.

[9] Harold Connamacher and Michael Molloy. The satisfiability threshold for a seemingly intractable random constraint satisfaction problem. *SIAM Journal on Discrete Mathematics*, 26(2):768–800, 2012.

[10] Harold S Connamacher. A random constraint satisfaction problem that seems hard for dpll. In *SAT*, 2004.

[11] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962.

[12] Martin Dietzfelbinger, Andreas Goerdt, Michael Mitzenmacher, Andrea Montanari, Rasmus Pagh, and Michael Rink. Tight thresholds for cuckoo hashing via xorsat. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and PaulG. Spirakis, editors, *Automata, Languages and Programming*, volume 6198 of *Lecture Notes in Computer Science*, pages 213–225. Springer Berlin Heidelberg, 2010.

[13] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. *arXiv preprint arXiv:1411.0650*, 2014.

[14] O. Dubois and J. Mandler. The 3-xorsat threshold. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 769–778, 2002.

[15] Silvio Franz, Michele Leone, Federico Ricci-Tersenghi, and Riccardo Zecchina. Exact solutions for diluted spin glasses and optimization problems. *Phys. Rev. Lett.*, 87:127209, Aug 2001.

[16] Ian P Gent and Toby Walsh. The sat phase transition. In *ECAI*, volume 94, pages 105–109. PITMAN, 1994.

[17] Marco Guidetti and A. P. Young. Complexity of several constraint-satisfaction problems using the heuristic classical algorithm walksat. *Phys. Rev. E*, 84:011102, Jul 2011.

[18] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[19] Morteza Ibrahimi, Yashodhan Kanoria, Matt Kraning, and Andrea Montanari. The set of solutions of random xorsat formulae. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 760–779. SIAM, 2012.

[20] Gabriel Istrate, Stefan Boettcher, and Allon G. Percus. Spines of random constraint satisfaction problems: definition and connection with computational complexity. *Annals of Mathematics and Artificial Inteligence*, 44(4):353–372, 2005.

[21] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina. Two solutions to diluted p-spin models and xorsat problems. *Journal of Statistical Physics*, 111(3-4):505–533, 2003.

[22] Michael Molloy. Cores in random hypergraphs and boolean formulas. *Random Structures & Algorithms*, 27(1):124–135, 2005.

[23] Boris Pittel and Gregory B. Sorkin. The satisfiability threshold for k-xorsat. *Combinatorics, Probability and Computing*, FirstView:1–33, 8 2015.

[24] Federico Ricci-Tersenghi and Guilhem Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):P09001, 2009.

[25] Federico Ricci-Tersenghi, Martin Weigt, and Riccardo Zecchina. Simplest random k-satisfiability problem. *Phys. Rev. E*, 63:026702, Jan 2001.

[26] J. H. van Lint. *Introduction to coding theory*. Springer-Verlag, Berlin, second edition, 1992.