Why almost all satisfiable k-CNF formulas are easy

Amin Coja-Oghlan, Michael Krivelevich, Dan Vilenchik

Combinatorial Dominance Guarantees for Heuristic Algorithms

2007 Conference on Analysis of Algorithms, AofA 07 DMTCS proc. **AH**, 2007, 95–108

Why almost all satisfiable k-CNF formulas are easy

Amin Coja-Oghlan[1†] and Michael Krivelevich[2‡] and Dan Vilenchik[3§]

[1] *Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA, USA. Supported by the German Research Foundation (project CO 646).*
[2] *School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel. Supported in part by USA-Israel BSF Grant 2002-133, and by grant 526/05 from the Israel Science Foundation.*
[3] *School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel.*

received 26^{th} *February 2007*, *revised 19th January 2008*, *accepted 19th January 2008*.

Finding a satisfying assignment for a k-CNF formula ($k \geq 3$), assuming such exists, is a notoriously hard problem. In this work we consider the uniform distribution over satisfiable k-CNF formulas with a linear number of clauses (clause-variable ratio greater than some constant). We rigorously analyze the structure of the space of satisfying assignments of a random formula in that distribution, showing that basically all satisfying assignments are clustered in one cluster, and agree on all but a small, though linear, number of variables. This observation enables us to describe a polynomial time algorithm that finds *whp* a satisfying assignment for such formulas, thus asserting that most satisfiable k-CNF formulas are easy (whenever the clause-variable ratio is greater than some constant). This should be contrasted with the setting of very sparse k-CNF formulas (which are satisfiable *whp*), where experimental results show some regime of clause density to be difficult for many SAT heuristics. One explanation for this phenomena, backed up by partially non-rigorous analytical tools from statistical physics, is the complicated clustering of the solution space at that regime, unlike the more "regular" structure that denser formulas possess. Thus in some sense, our result rigorously supports this explanation.

Keywords: computational and structural complexity, algorithms and data structures, message passing algorithms, SAT.

Contents

†Email: amincoja@andrew.cmu.edu
‡Email: krivelev@post.tau.ac.il.
§Email: vilenchi@post.tau.ac.il

1 Introduction

Constraint satisfaction problems play an important role in many areas of computer science.The main challenge is to devise efficient algorithms for finding satisfying assignments (when such exist), or conversely to provide a certificate of unsatisfiability. One of the best known examples of a constraint satisfaction problem is k-SAT, which is the first to be proven as NP-complete. Although satisfactory approximation algorithms are known for several NP-hard problems, the problem of finding a satisfying assignment (if such exists) is not amongst them. In fact, Håstad (15) proved that it is NP-hard to approximate MAX-3SAT (the problem of finding an assignment that satisfies as many clauses as possible) within a ratio better than 7/8.

In trying to understand the inherent hardness of the problem, many researchers analyzed structural properties of formulas drawn from different distributions. One such natural distribution is the following: fix $c, n > 0$ (c may depend on n), choose $m = cn$ clauses uniformly at random out of $8\binom{n}{3}$ possible ones. We denote this distribution by $\mathcal{P}_{n,m}$. Despite its simplicity, many essential properties of this model are yet to be understood. In particular, the hardness of deciding if a random formula is satisfiable, and finding a satisfying assignment for a random formula, are both major open problems (9; 19).

1.1 Our Contribution

Remarkable phenomena occurring in the random model $\mathcal{P}_{n,m}$ are **phase transitions**. With respect to the property of being satisfiable, such a phase transition takes place too. More precisely, there exists a threshold $d_k = d_k(n)$ such that a k-CNF formula with clause-variable ratio greater than d_k is not satisfiable *whp*[(i)], while one with ratio smaller than d_k is (14). In this work we consider satisfiable k-CNF formulas with cn clauses, c greater than some sufficiently large constant. In this regime almost all formulas are not satisfiable, and therefore we consider the following natural extension of $\mathcal{P}_{n,m}$, which we denote by $\mathcal{P}^{\mathrm{sat}}_{n,m}$: fix $c, n > 0$ (c may depend on n), choose $m = cn$ clauses uniformly at random out of $8\binom{n}{3}$ possible ones, *conditioned* on the fact that the received formula is satisfiable. To simplify the presentation we consider the most popular setting – the case $k = 3$, namely random 3SAT, and remark that our results extend to any fixed k.

(i) Writing *whp* we mean with probability tending to 1 as n goes to infinity.

Our contribution is composed of three parts. The **first** part *rigorously* establishes the following fact:

Theorem 1 *There exists a polynomial time algorithm that* whp *finds a satisfying assignment for 3CNF instances from $\mathcal{P}^{\rm sat}_{n,m}$, $m \geq C_0 n$, C_0 a sufficiently large constant.*

The algorithm is described in Section 3. Thus we partially answer the open problem asking to decide the hardness of $\mathcal{P}_{n,m}$ (9; 19). Specifically, we assert that for all but a vanishing fraction of satisfiable 3CNF formulas over n variables with m clauses, one can efficiently find a satisfying assignment (whenever m/n is greater than some constant). Our proof of Theorem 1 is constructive – that is, we present an algorithm that meets the requirements of Theorem 1.

The **second** part of our result concerns another exciting area. One of the most surprising recent developments in satisfiability problems comes from statistical physics. More specifically, in their well-known work, Mezard, Parisi and Zecchina (6) designed a new algorithm, known as Survey Propagation, for solving k-SAT instances. A particularly dramatic feature of this method is that it appears to remain efficient in solving very large instances of random k-SAT even with densities very close to the conjectured satisfiability threshold, a regime where other algorithms (e.g., the WalkSAT method (22)) typically fail. Nonetheless, despite the considerable progress to date the reasons underlying the remarkable performance of Survey Propagation are not yet fully understood, let alone rigorously analyzed.

The difficulty that Survey Propagation apparently overcomes lies in the complicated structure of the solution space of such formulas. That is, the *conjectured* picture, some supporting evidence of which was proved rigorously for $k \geq 8$ (20; 1; 21), is that typically random k-CNF formulas in the near-threshold regime have an exponential number of **clusters** of satisfying assignments. While any two assignments in distinct clusters disagree on at least εn variables, any two assignments within one cluster coincide on $(1-\varepsilon)n$ variables. Furthermore, each cluster has a linear number of **frozen** variables (a variable is said to be *frozen* in some cluster if *all* satisfying assignments within that cluster assign it in the same way). The algorithmic difficulty with such a clustered solution space seems to be that most known algorithms do not "steer" into one cluster but try to find a "compromise" between the satisfying assignments in distinct clusters, which actually is impossible.

Complementing this picture *rigorously*, we show that typically for satisfiable 3CNF formulas in the above-threshold regime the solution space contains only one cluster, though its size may be exponential in n. Formally,

Theorem 2 *Let $\mathcal{F}$ be random 3CNF from $\mathcal{P}^{\rm sat}_{n,m}$, $m \geq C_0 n$, C_0 a sufficiently large constant. Then* whp *$\mathcal{F}$ enjoys the following properties:*

1. *All but $e^{-\Theta(m/n)}n$ variables are frozen.*
2. *The formula induced by the non-frozen variables decomposes into connected components of at most logarithmic size.*
3. *Letting $\beta(\mathcal{F})$ be the number of satisfying assignments of $\mathcal{F}$, we have $\frac{1}{n}\log\beta(G) = e^{-\Theta(m/n)}$.*

Combining Theorems 1 and 2 supports the following common thesis: the main key to understanding the hardness (even experimental one) of a certain distribution over satisfiable formulas lies in the structure of the solution space of a typical formula in that distribution. Specifically, our results show (at least in our setting) that typically when a formula has a single cluster of satisfying assignments, though its volume

might be exponential, then the problem is "easy". On the other hand, when the clustering is complicated, for example in the near threshold regime, experimental results predict that many "simple" heuristics fail, while "heavy machinery" such as Survey Propagation works. Heightening this last point, consider the recent work in (11), where the naïve Warning Propagation algorithm is rigorously shown to work *whp* for 3CNF formulas taken from a somewhat different distribution than the one we consider, nonetheless (as we shall prove) sharing with $\mathcal{P}^{\rm sat}_{n,m}$ the same simple cluster structure. Fitting the result in (11) to our perspective – when the clustering is simple, then a simple message passing algorithm works (Warning Propagation), when the clustering is complicated, then only a much more complicated message passing algorithm is known (and even this only experimentally) to work (Survey Propagation).

The **third** part of our result is more "philosophical" in nature. As we already mentioned, the event of a random formula in $\mathcal{P}_{n,m}$ being satisfiable, when m/n is some constant above the satisfiability threshold, is very unlikely. Therefore, the distribution $\mathcal{P}^{\rm sat}_{n,m}$ differs from the $\mathcal{P}_{n,m}$ distribution significantly. In effect, many techniques that have become standard in the study of random instances (3CNF formulas and grahps) just do not carry over to $\mathcal{P}^{\rm sat}_{n,m}$ – at least not directly. In particular, the contriving event of being satisfiable causes the clauses in $\mathcal{P}^{\rm sat}_{n,m}$ to be dependent.

The inherent difficulty of $\mathcal{P}^{\rm sat}_{n,m}$ has led many researchers to consider the more approachable, but considerably less natural, **planted distribution**, pioneered by Kučera (18) in the context of graph coloring. In the planted distribution, which we denote by $\mathcal{P}^{\rm plant}_{n,m}$, one first fixes some satisfying assignment, and then includes m clauses uniformly at random out of $7\binom{n}{3}$ clauses that are satisfied by it. This of course guarantees that the formula is satisfiable. Planted solution distributions are favored by many researchers in the context of SAT (13; 4), but also for other graph optimization problems such as max clique, min bisection, and coloring (2; 3; 5; 16; 10), to mention just a few.

Of course the $\mathcal{P}^{\rm plant}_{n,m}$ model is somewhat artificial and therefore provides a less natural model of random instances than $\mathcal{P}^{\rm sat}_{n,m}$. Nevertheless, devising new ideas for analyzing $\mathcal{P}^{\rm sat}_{n,m}$ we show that $\mathcal{P}^{\rm sat}_{n,m}$ and $\mathcal{P}^{\rm plant}_{n,m}$ actually share many structural properties such as the existence of a single cluster of solutions. As a consequence, we can prove that a certain algorithm, designed with $\mathcal{P}^{\rm plant}_{n,m}$ in mind, works for $\mathcal{P}^{\rm sat}_{n,m}$ as well (this algorithm is used to prove Theorem 1). In other words, by presenting new methods for analyzing heuristics on random instances, we can show that algorithmic techniques invented for the somewhat artificial planted model extend to the canonical uniform setting.

1.2 Related Work and Techniques

Almost all exact polynomial-time heuristics suggested so far for random instances (either SAT or graph optimization problems) were analyzed when the input is sampled according to a planted-solution distribution, or various semi-random variants thereof. Alon and Kahale (2) suggest a polynomial time algorithm based on spectral techniques that *whp* properly k-colors a random graph from the planted k-coloring distribution (the distribution of graphs generated by partitioning the n vertices into k equally-sized color classes, and including every edge connecting two different color classes with probability $p = p(n)$), for graphs with average degree greater than some constant. In the SAT context, Flaxman's algorithm, drawing on ideas from (2), solves *whp* planted 3SAT instances where the clause-variable ratio is greater than some constant. Also (12; 11; 17) address the planted 3SAT distribution.

On the other hand, very little work was done on non-planted distributions, such as $\mathcal{P}^{\rm sat}_{n,m}$. In this context one can mention the work of Chen (7) which provides an *exponential* time algorithm for $\mathcal{P}^{\rm sat}_{n,m}$ with m/n

greater than some constant. Ben-Sasson et al. (4) also study $\mathcal{P}^{\rm sat}_{n,m}$ but with $m/n = \Omega(\log n)$, a regime where $\mathcal{P}^{\rm sat}_{n,m}$ and $\mathcal{P}^{\rm plant}_{n,m}$ coincide (since typically there is only one satisfying assignment). They leave as an open problem whether one can characterize $\mathcal{P}^{\rm sat}_{n,m}$ for $m/n = o(\log n)$, and in particular whether there exists a *polynomial* time algorithm that finds *whp* a satisfying assignment in this regime. In this work we answer their question positively.

One should also mention the recent work of (8), where the uniform distribution over k-colorable graphs with average degree greater than some constant is analyzed. Specifically, (8) shows that a similar clustering phenomenon to the one described in Theorem 2 also occurs for k-colorable graphs with constant average degree. Furthermore, (8) shows that the algorithm by Alon and Kahale (2) works *whp* for such graphs as well. The techniques that we use are similar in flavor to the ones introduced in (8), though k-SAT is fundamentally different from k-colorability.

To obtain our results, we use two main techniques. As we mentioned, $\mathcal{P}^{\rm plant}_{n,m}$ is already very well understood, and the probability of some structural properties that we discuss can be easily estimated for $\mathcal{P}^{\rm plant}_{n,m}$ using standard probabilistic calculations. It then remains to find a reasonable "exchange rate" between $\mathcal{P}^{\rm plant}_{n,m}$ and $\mathcal{P}^{\rm sat}_{n,m}$. We use this approach to estimate the probability of "complicated" properties, which hold with extremely high probability in $\mathcal{P}^{\rm plant}_{n,m}$. The other method is directly analyzing $\mathcal{P}^{\rm sat}_{n,m}$, crucially overcoming the clause-dependency issue. This method tends to be more involved than the first one, and necessitates intricate counting arguments.

1.3 Paper's Structure

The rest of the paper is structured as follows. In Section 2 we discuss relevant structural properties that a typical formula in $\mathcal{P}^{\rm sat}_{n,m}$ possesses, along with a proof of Theorem 2. We then prove Theorem 1 in Section 3 by describing an algorithm and showing that it meets the requirements of Theorem 1. Concluding remarks are given in Section 5. For lack of space most propositions are given without a proof.

2 Properties of a Random Instance from $\mathcal{P}^{\rm sat}_{n,m}$

In this section we analyze the structure of a typical formula in $\mathcal{P}^{\rm sat}_{n,m}$. A direct consequence of the discussion in this section is a proof of Theorem 2.

Here and throughout we think of m as $O(n \log n)$. Otherwise, typically a formula in $\mathcal{P}^{\rm sat}_{n,m}$ with $m \geq C_0 n \log n$, C_0 some sufficiently large constant, has only one satisfying assignment (see (4) for example), and therefore by the definition of $\mathcal{P}^{\rm plant}_{n,m}$ it holds that $\mathcal{P}^{\rm sat}_{n,m}$ and $\mathcal{P}^{\rm plant}_{n,m}$ are statistically close. A simple second moment calculation shows that the Majority Vote, discussed ahead, will reconstruct *whp* the satisfying assignment for $\mathcal{P}^{\rm plant}_{n,m}$ in that regime. The interesting case remains $m = O(n \log n)$.

2.1 Setting the Exchange Rate

Let $\mathcal{A}$ be some property of CNF formulas (it would be convenient for the reader to think of $\mathcal{A}$ as a "bad" property). We start by determining the exchange rate for $Pr[\mathcal{A}]$ when moving from the planted distribution to the uniform one. For a property $\mathcal{A}$ we use $Pr^{\rm uniform,m}[\mathcal{A}]$ to denote the probability of $\mathcal{A}$ occurring under $\mathcal{P}^{\rm sat}_{n,m}$, and $Pr^{\rm planted,m}[\mathcal{A}]$ for $\mathcal{P}^{\rm plant}_{n,m}$. The following lemma asserts the exchange rate $\mathcal{P}^{\rm plant}_{n,m} \rightarrow \mathcal{P}^{\rm sat}_{n,m}$.

Lemma 3 *($\mathcal{P}^{\rm plant}_{n,m} \rightarrow \mathcal{P}^{\rm sat}_{n,m}$) Let $\mathcal{A}$ be some property of 3CNF formulas, then*

$$Pr^{\rm uniform,m}[\mathcal{A}] \leq e^{ne^{-m/(3n)}} \cdot Pr^{\rm planted,m}[\mathcal{A}].$$

Remark 4 *The exchange rate between the planted distribution and the uniform is exponential in n. Thus Lemma 3 is useful whenever the "bad" event $\mathcal{A}$ happens with exponentially small probability in $\mathcal{P}^{\rm plant}_{n,m}$.*

2.2 The Majority Vote

For a 3CNF formula $\mathcal{F}$ and a variable x we let $N^+(x)$ be the set of clauses in $\mathcal{F}$ in which x appears positively (namely, as the literal x), and $N^-(x)$ be the set of clauses in which x appears negatively (that is, as $\bar{x}$). The Majority Vote assignment over $\mathcal{F}$, which we denote by MAJ, assigns every x according to the sign of $|N^+(x)| - |N^-(x)|$ (TRUE if the difference is positive and FALSE otherwise).

To show the usefulness of the Majority Vote in $\mathcal{P}^{\rm sat}_{n,m}$ we work our way through $\mathcal{P}^{\rm plant}_{n,m}$ and use the exchange-rate technique. Consider $\mathcal{F}$ in $\mathcal{P}^{\rm plant}_{n,m}$, and let φ be its planted assignment. Consider a variable x whose assignment is w.l.o.g. $\varphi(x) = TRUE$. In every clause of $\mathcal{F}$ that contains x, x appears positively with probability $4/7$ and negatively with probability $3/7$. Therefore in expectation the sign of $|N^+(x)| - |N^-(x)|$ agrees with $\varphi(x)$. More formally, one can prove the following fact (see (17) for the complete proof):

Lemma 5 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\rm plant}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Let F_{MAJ} be a random variable counting the number of variables in $\mathcal{F}$ on which MAJ disagrees with the planted assignment. There exists a constant $a_0 > 0$ (independent of m, n) and a positive monotonically increasing function f s.t. for every $a \geq a_0$ it holds that*

$$Pr[F_{MAJ} \geq e^{-m/(an)}n] \leq e^{-ne^{-m/(f(a)n)}}.$$

Proposition 6 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\rm sat}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Then* whp *there exists a satisfying assignment φ of $\mathcal{F}$ that differs from MAJ on at most $e^{-\Theta(m/n)}n$ variables.*

Proof: Set $a_0 = f^{-1}(3)$ (f is the function promised in Lemma 5, $f^{-1}(3)$ is taken according to the denominator in $e^{ne^{-m/(3n)}}$ from Lemma 3), and $a_1 = 2a_0$. Let $\mathcal{A}$ be the following property: "there exists no satisfying assignment s.t. MAJ is at distance at most $e^{-m/(a_1 n)}n$ from it" (by distance we mean the Hamming distance). Using the exchange-rate technique we obtain:

$$Pr^{uniform,m}[\mathcal{A}] \underbrace{\leq}_{\text{Lemma 3}} e^{ne^{-m/(3n)}} Pr^{\rm planted,m}[\mathcal{A}] \underbrace{\leq}_{\text{Lemma 5}} e^{ne^{-m/(3n)}} e^{-ne^{-m/(f(a_1)n)}} = e^{n(e^{-m/(3n)} - e^{-m/(f(a_1)n)})} = o(1).$$

The last equality is by the choice of a_1 and the fact that f is increasing, that is $f(a_1) = f(2a_0) > f(a_0) = 3$ and therefore $e^{-m/(3n)} - e^{-m/(f(a_1)n)} < 0$. □

2.3 The Discrepancy Property

A well known result in the theory of random graphs is that a random graph *whp* will not contain a small yet unexpectedly dense subgraph. This is also the case for $\mathcal{P}_{n,m}$ (when considering the graph induced by the formula). This property holds only with probability $1 - 1/poly(n)$ (for example, with probability $1/poly(n)$ a fixed clique on a constant number of vertices will appear). Thus the exchange-rate technique is of no use in this case (as the exchange rate factor is exponential in n). Overcoming the clause-dependency issue, using an intricate counting argument, we directly analyze $\mathcal{P}^{\mathrm{sat}}_{n,m}$ to prove:

Proposition 7 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\mathrm{sat}}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Then* whp *there exists* no *subset of variables U s.t.*

- $|U| \leq n/2000$,
- *There are $|U| \cdot \frac{m}{50n}$ clauses in $\mathcal{F}$ that contain two variables from U.*

The full proof is given in Section 4.

Remark 8 *To see how Proposition 7 corresponds to the random graph context, consider the graph induced by the formula $\mathcal{F}$ (the vertices are the variables, and two variables share an edge if there exists some clause containing them both) and observe that every clause that contains at least two variables from U contributes an edge to the subgraph induced by U. Thus if we have many such clauses, this subgraph will be prohibitively dense. Since $\mathcal{F}$ is random so is its induced graph, and therefore the latter will typically not occur. In our case $\mathcal{F}$ is random but the clauses are dependent – making the analysis more complicated.*

2.4 The Core Variables

We describe a subset of the variables, referred to as the *core variables*, which plays a crucial role in the understanding of $\mathcal{P}^{\mathrm{sat}}_{n,m}$. Recall that a variable is said to be frozen in $\mathcal{F}$ if in every satisfying assignment it takes the same assignment. The notion of core captures this phenomenon. In addition, a core typically contains all but a small (though constant) fraction of the variables. This implies that a large fraction of the variables is frozen, a fact which must leave imprints on various structural properties of the formula. These imprints allow efficient heuristics to recover a satisfying assignment of the core. A second implication of this is an upper bound on the number of possible satisfying assignments, and on the distance between every such two. Thus the notion of core gives a catheterization of the cluster structure of the solution space (matching the properties described in Theorem 2).

Definition 9 *(support) Given a 3CNF formula $\mathcal{F}$ and some assignment ψ to the variables, we say that a literal x* supports *a clause C (in which it appears) w.r.t. ψ if x is the only literal that evaluates to true in C under ψ.*

Definition 10 *(core) A set of variables $\mathcal{H}$ is called a* ***core*** *of $\mathcal{F}$ w.r.t. a satisfying assignment ψ if the following three properties hold:*

- *Every variable $x \in \mathcal{H}$ supports at least $m/(5n)$ clauses in $\mathcal{F}[\mathcal{H}]$ w.r.t. ψ ($\mathcal{F}[\mathcal{H}]$ being the subformula containing the clauses where all three variables belong to $\mathcal{H}$).*
- *x appears in at most $m/(10n)$ clauses in $\mathcal{F} \setminus \mathcal{F}[\mathcal{H}]$.*

Remark 11 *The proof of Theorem 2 (structure of the solution space) uses only the first property in Definition 10. However, since the core is also used for the algorithmic perspective, the second property is needed for the analysis of the algorithm.*

Remark 12 *The choice of $m/(5n)$ corresponds to slightly less than the expected support of a variable w.r.t. the planted assignment (which is roughly $3m/(14n)$), had the underlying probability space been $\mathcal{P}^{\text{plant}}_{n,m}$.*

We proceed by asserting some relevant properties that such a core typically possesses.

Proposition 13 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\text{sat}}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Then* whp *there exits a satisfying assignment φ of $\mathcal{F}$ w.r.t. which there exists a core $\mathcal{H}$ and $|\mathcal{H}| \geq (1 - e^{-\Theta(m/n)})n$.*

The proof of Proposition 13 uses the exchange-rate technique, similar to the proof of Proposition 6 (a proof of Proposition 13 in the planted setting is given in (17), similar to Lemma 5). Details omitted.

The next proposition ties between the core variables and the property of the Majority Vote, and is crucial to the analysis of the algorithm. The proposition follows by noticing that φ in Lemma 5 and in its core-size counterpart is the same – the planted assignment. Thus, one can apply the exchange-rate technique on the combined property.

Proposition 14 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\text{sat}}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Then* whp *there exists a satisfying assignment φ s.t. the following two properties hold:*

- *MAJ differs from φ on at most $e^{-\Theta(m/n)}n$ variables*
- *There exists a core $\mathcal{H}$ w.r.t. φ as promised in Proposition 13.*

The next proposition characterizes the structure of the formula induced by the non-core variables. The connected components of a formula $\mathcal{F}$ are the sub-formulas $\mathcal{F}[C_1], \ldots, \mathcal{F}[C_k]$, where $C_1, C_2, \ldots, C_k$ are the the connected components in the graph induced by $\mathcal{F}$. Given a core $\mathcal{H}$ of $\mathcal{F}$ w.r.t. a satisfying assignment φ, we denote by $\mathcal{F}^{\varphi}_{out}(\mathcal{H})$ the subformula of $\mathcal{F}$ which is the outcome of the following procedure: set the variables $\mathcal{H}$ in $\mathcal{F}$ according to φ and simplify $\mathcal{F}$.

Proposition 15 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\text{sat}}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Let $\mathcal{H}$ be the core promised in Proposition 13. Then* whp *the largest connected component in $\mathcal{F}^{\varphi}_{out}(\mathcal{H})$ is of size $O(\log n)$.*

Proposition 15 also holds only with probability $1 - 1/poly(n)$, had $\mathcal{F}$ been distributed according to $\mathcal{P}^{\text{plant}}_{n,m}$. Thus, similar to Proposition 7 the analysis is an involved counting argument (in this case even more complicated).

Lastly, we establish the "frozenness" property of the core variables. The proof uses Proposition 7 to show that there are no "close" satisfying assignments, and the exchange-rate technique to prove that there are no "far" ones.

Proposition 16 *Let $\mathcal{F}$ be distributed according to $\mathcal{P}^{\text{sat}}_{n,m}$ with $m \geq C_0 n$, C_0 a sufficiently large constant. Let $\mathcal{H}$ be the core promised in Proposition 13. Then* whp *the assignment of $\mathcal{H}$ in all satisfying assignments of $\mathcal{F}$ is the same.*

3 Proof of Theorems 1 and 2

Theorem 2 is an immediate corollary of Propositions 13, 15 and 16 . Propositions 13 and 16 imply that all but a $e^{-\Theta(m/n)}n$ of the variables are frozen. Therefore, there are at most $2^{e^{-\Theta(m/n)}n}$ possible ways to set the assignment of the remaining variables. Furthermore, every two satisfying assignments of $\mathcal{F}$ can differ on the assignment of at most $e^{-\Theta(m/n)}n$ variables (that of the non-core variables). Proposition 15 completes the proof with the characterization of the formula induced by the non-frozen variables (which are a subset of the non-core ones).

Before proving **Theorem 1** we present an algorithm that meets the requirements of Theorem 1. The algorithm is basically the one given in (13).

Remark 17 *The reader versed in the area will notice some differences from the original algorithm in (13). However, since we consider a different distribution than the one in (13), one can describe a simplified version of that algorithm (e.g., replace the spectral step with a Majority Vote).*

SAT($\mathcal{F}$)
Step 1: Majority Vote
1. $\pi_1 \leftarrow$ Majority Vote over $\mathcal{F}$.
Step 2: Reassignment
2. **for** $i = 1$ to $\log n$
3. **for all** $x \in V$
4. **if** x appears in more than $m/(5n)$ unsatisfied clauses w.r.t. π_i **then** $\pi_{i+1} \leftarrow \pi_i$ with x flipped.
5. **end for.**
6. **end for.**
Step 3: Unassignment
7. set $\psi_1 = \pi_{\log n}$, $i = 1$.
8. **while** $\exists x$ s.t. x supports less than $m/(10n)$ clauses w.r.t. ψ_i
9. set $\psi_{i+1} \leftarrow \psi_i$ with x unassigned.
10. $i \leftarrow i + 1$.
11. **end while.**
Step 4: Exhaustive Search
12. Let ξ be the final partial assignment.
13. let A be the set of assigned variables in ξ.
14. exhaustively search $\mathcal{F}^{\xi}_{out}(A)$, component by component.

We now prove that the algorithm SAT meets the requirements of Theorem 1. We say that $\mathcal{F}$ is *typical* in $\mathcal{P}^{\mathrm{sat}}_{n,m}$ if Propositions 7, 14 and 15 hold for it. The discussion in Section 2 guarantees that *whp* $\mathcal{F}$ is typical. Therefore, to prove Theorem 1 it suffices to consider a typical $\mathcal{F}$ and prove that SAT (always) finds a satisfying assignment for $\mathcal{F}$.

We let $\mathcal{H}$ be the core promised in Proposition 13, and φ – the satisfying assignment w.r.t. which $\mathcal{H}$ is defined. In all the following propositions we assume $\mathcal{F}$ is typical (we don't explicitly state it every time for the sake of brevity). Similar propositions to Propositions 18–20 were proven in (13) for example when relying only on the fact that the instance is typical.

Proposition 18 *Let ψ_1 be the assignment defined in line 7 of SAT. Then ψ_1 agrees with φ on the assignment of all variables in $\mathcal{H}$.*

Proposition 19 *Let ξ be the partial assignment defined in line 12 of SAT. Then all assigned variables in ξ are assigned according to φ, and all the variables in $\mathcal{H}$ are assigned.*

Proposition 20 *The exhaustive search, Step 4 of SAT, completes in polynomial time with a satisfying assignment of $\mathcal{F}$.*

Theorem 1 then follows.

4 Proof of Proposition 7

Let V be the set of n variables, and let U be some fixed subset of V, $|U| = u$. Let H be a fixed formula over V with exactly $\frac{um}{50n}$ clauses s.t. each clause contains at least two variables from U. A formula $\mathcal{F}$ is said to be *H-poor* if it contains H as a sub-formula.

Furthermore, let $\mathcal{P}_H$ signify the set of all H-poor satisfiable formulas with exactly m clauses, and $\mathcal{A}$ the set of all satisfiable formulas with exactly m edges. Our first objective is to establish the following.

Lemma 21 $|\mathcal{P}_H| \leq (em/n^3)^{um/(50n)}|\mathcal{A}|$.

This immediately implies that the probability of an H-poor formula in $\mathcal{P}^{\rm sat}_{n,m}$ is at most $(em/n^3)^{um/(50n)}$. Next take the union bound over all possible sub-formulas H (s.t. $|U| \leq n/2000$ – as required by Proposition 7) to show that *whp* none is contained in a random $\mathcal{P}^{\rm sat}_{n,m}$ formula.

To prove Lemma 21 we shall set up an auxiliary bipartite graph $\mathcal{G}$ with vertex set $V(\mathcal{G}) = \mathcal{P}_H \cup \mathcal{A}$. This graph will have the property that the average degree of a vertex in $\mathcal{P}_H$ is Δ, while that of a vertex in $\mathcal{A}$ is Δ', where in addition $\Delta'/\Delta \leq (em/n^3)^{um/(50n)}$. Since $\Delta\#\mathcal{P}_H = \Delta'\#\mathcal{A}$, by double counting, we thus obtain Lemma 21. We describe a nondeterministic procedure **P** that receives a formula $F \in \mathcal{P}_H$ and produces a new formula $F' \in \mathcal{A}$. In our auxiliary graph $\mathcal{G}$, we connect a right-side node F with a left-side one F', if F' can be obtained from F by applying **P** to F. **P** is the following procedure:

given a H-poor formula F do:

- Choose a set $\mathcal{C}$ of $um/(50n)$ fresh clauses (that are not yet in F)
- Obtain F' from F by removing all $um/(50n)$ clauses of H and adding $\mathcal{C}$.
- Output F' if it is satisfiable.

Therefore,

$$\Delta \geq \binom{n^3}{um/(50n)}.$$

This is because we have to choose $um/(50n)$ clauses out of at least $7\binom{n}{3} - m \geq n^3$ possible ones (since F was satisfiable to begin with, there are at least $7\binom{n}{3}$ clauses that are satisfied by the assignment that satisfies F, and we can assume that $m = O(n \log n)$). Conversely, consider the following nondeterministic procedure to recover a formula F from F'. Out of m possible clauses in F', choose $um/(50n)$. Take them out, and reinstall the original clauses of H. Therefore,

$$\Delta' \leq \binom{m}{um/(50n)}.$$

Using standard bounds on the binomial coefficients, the required bound on Δ'/Δ is obtained and Lemma 21 follows. We are now ready to bound the probability that a random formula $\mathcal{F}$ in $\mathcal{P}^{\rm sat}_{n,m}$ violates the condition of Proposition 7. Using the union bound this probability is at most

$$\begin{aligned}\sum_{u=1}^{n/2000}\binom{n}{u}\binom{8n\binom{u}{2}}{um/(50n)}\cdot\left(\frac{em}{n^3}\right)^{um/(50n)} &\leq \sum_{u=1}^{n/2000}\left(\frac{en}{u}\right)^u\left(\frac{600un^2}{m}\right)^{um/(50n)}\left(\frac{em}{n^3}\right)^{um/(50n)}\\ &\leq \sum_{u=1}^{n/2000}\left(\frac{en}{u}\right)^u\left(\frac{1800u}{n}\right)^{uC_0/50} \leq \sum_{u=1}^{n/2000}\left(\frac{en}{u}\cdot\frac{1800u}{n}\cdot\left(\frac{1800u}{n}\right)^{C_0/50-1}\right)^u\\ &\leq \sum_{u=1}^{n/2000}\left(5400\cdot\left(\frac{1800u}{n}\right)^{C_0/100}\right)^u = o(1)\end{aligned}$$

The last equality is due to $(u/n)\leq 1/2000$, so the last sum decreases faster than a geometric series with quotient and first element equal $1/poly(n)$, and therefore the whole sum is $o(1)$.

5 Discussion

Though $\mathcal{P}_{n,m}$ has a very simple description (fix $c,n>0$ and choose $m=cn$ clauses uniformly at random out of $8\binom{n}{3}$ possible ones), and is very fundamental to understanding the hardness of 3SAT, it still baffles many researchers and altogether remains very poorly understood. In particular, the hardness of deciding if a random formula is satisfiable, and finding a satisfying assignment for a random formula, are both major open problems (9; 19).

Trying to shed some light on this problem we consider the uniform distribution over satisfiable 3CNF formulas, $\mathcal{P}^{\rm sat}_{n,m}$, with clause-variable ratio greater than some sufficiently large constant. We characterize the typical structure of the solution space of such formulas and show that a relatively simple efficient algorithm recovers *whp* a satisfying assignment of such formulas, thus asserting that almost all satisfiable 3CNF formulas (when the clause-variable ratio is sufficiently large, yet possibly constant) are easy. To obtain our result we had to come up with new analytical tools that apply to a number of further NP-hard problems, including k-colorability. Our result also implies that the algorithmic techniques developed for random formulas from the planted distribution, e.g. (13; 11; 12; 17), can be extended to the significantly more natural uniform distribution.

References

[1] D. Achlioptas and F. Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In *STOC*, pages 130–139, 2006.

[2] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM J. on Comput.*, 26(6):1733–1748, 1997.

[3] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998.

[4] E. Ben-Sasson, Y. Bilu, and D. Gutfreund. Finding a randomly planted assignment in a random $3CNF$. *manuscript*, 2002.

[5] A. Blum and J. Spencer. Coloring random and semi-random k-colorable graphs. *J. of Algorithms*, 19(2):204–234, 1995.

[6] A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27:201–226, 2005.

[7] H. Chen. An algorithm for sat above the threshold. In *6th International Conference on Theory and Applications of Satisfiability Testing*, pages 14–24, 2003.

[8] A. Coja-Oghlan, M. Krivelevich, and D. Vilenchik. Why almost all k-colorable graphs are easy to color. In *STACS*, volume 4393 of *Lect. Notes in Comp. Sci.*, pages 121-132. Springer, Berlin, 2007.

[9] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 534–543, 2002.

[10] U. Feige and R. Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures and Algorithms*, 16(2):195–208, 2000.

[11] U. Feige, E. Mossel, and D. Vilenchik. Complete convergence of message passing algorithms for some satisfiability problems. In *Random*, pages 339–350, 2006.

[12] U. Feige and D. Vilenchik. A local search algorithm for 3SAT. Technical report, The Weizmann Institute of Science, 2004.

[13] A. Flaxman. A spectral technique for random satisfiable 3CNF formulas. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, pages 357–363, 2003.

[14] E. Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *J. Amer. Math. Soc.*, 12(4):1017–1054, 1999.

[15] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[16] C. Hui and A. M. Frieze. Coloring bipartite hypergraphs. In *Proceedings of the 5th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 345–358, 1996.

[17] M. Krivelevich and D. Vilenchik. Solving random satisfiable $3CNF$ formulas in expected polynomial time. In *Proc. 17th ACM-SIAM Symp. on Discrete Algorithms*, pages 454–463, 2006.

[18] L. Kučera. Expected behavior of graph coloring algorithms. In *Proc. Fundamentals of Computation Theory*, volume 56 of *Lecture Notes in Comput. Sci.*, pages 447–451. Springer, Berlin, 1977.

[19] L. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

[20] M. Mezard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94:197–205, 2005.

[21] T. Mora, M. Mezard, and R. Zecchina. Pairs of sat assignments and clustering in random boolean formulae, 2005.

[22] B. Selman, H. A. Kautz, and B. Cohen. Local search strategies for satisfiability testing. In *Proceedings of the Second DIMACS Challange on Cliques, Coloring, and Satisfiability*, 1993.