Efficient recognition of random unsatisfiable k-SAT instances by spectral methods

Andreas Goerdt[1] and Michael Krivelevich[2]*

[1] Fakultät für Informatik, TU Chemnitz, 09107 Chemnitz, Germany,
e-mail: goerdt@informatik.tu-chemnitz.de
[2] Department of Mathematics, Faculty of Exact Sciences, Tel Aviv University,
Tel Aviv 69978, Israel,
e-mail: krivelev@math.tau.ac.il

Abstract. It is known that random k-SAT instances with at least cn clauses where $c = c_k$ is a suitable constant are unsatisfiable (with high probability). We consider the problem to certify efficiently the unsatisfiability of such formulas. A result of Beame et al. shows that k-SAT instances with at least $n^{k-1}/\log n$ clauses can be certified unsatisfiable in polynomial time. We employ spectral methods to improve on this: We present a polynomial time algorithm which certifies random k-SAT instances for k even with at least $2^k \cdot (k/2)^7 \cdot (\ln n)^7 \cdot n^{k/2} = n^{(k/2)+o(1)}$ clauses as unsatisfiable (with high probability).

Introduction

We study the complexity of certifying unsatisfiability of random k-SAT instances (or k-CNF formulas) over n propositional variables. (All our discussion refers to k fixed and then letting n be sufficiently large.) The probability space of random k-SAT instances has been widely studied in recent years for several good reasons. The most recent literature is [Ac2000],[Fr99], [Be et al98].

One of the reasons for studying random k-SAT instances is that they have the following sharp threshold behaviour [Fr99]: There exists a constant $c = c_k$ such that for any $\varepsilon > 0$ formulas with at most $(1-\varepsilon) \cdot c \cdot n$ clauses are satisfiable whereas formulas with at least $(1+\varepsilon) \cdot c \cdot n$ are unsatisfiable with high probability (that means with probability tending to 1 when n goes to infinity). In fact, it is by now not proven that c_k is a constant. It might be that $c_k = c_k(n)$ depends on n. However, it is known that c_k is at most $2^k \cdot \ln 2$ and the general conjecture is that c_k converges to a constant. For formulas with at least $2^k \cdot (\ln 2) \cdot n$ clauses the expected number of satisfying assignments of a random formula tends to 0 and the formulas are unsatisfiable with high probability. For 3-SAT instances much effort is spent to approximate the value of c_3. The currently best results are that c_3 is at least 3.125 [Ac2000] and at most 4.601 [KiKrKrSt98]. In [Du et al2000] it is claimed that $c_3 \leq 4.501$. (For $k = 2$ we have $c_2 = 1$ [ChRe92], [Go96].)

* Partially supported by a USA-Israeli BSF grant

The algorithmic interest in this threshold is due to the empirical obeservation that random k-SAT instances at the threshold, i.e. with around $c_k n$ random clauses are hard instances. The following behaviour has been reported consistently in experimental studies with suitably optimized backtracking algorithms searching for a satisfying assignment,see for example [SeMiLe96] [CrAu96]: The average running time is quite low for instances below the threshold. For 3-SAT instances we observe: Formulas with at most $4n$ clauses are satisfiable and it is quite easy to find a satisfying assignment. A precipitous increase in the average running time is observed at the threshold. For 3-SAT: About half of the formulas with $4.2n$ clauses are satisfiable and it is difficult to decide if a formula is satisfiable or not. Finally a speedy decline to lower complexity is observed beyond the threshold. For 3-SAT: All formulas with $4.5n$ clauses are unsatisfiable and the running time decreases again (in spite of the fact that now always the whole backtracking tree must be searched.)

There are no general complexity theoretical results relating the threshold to hardness. The following observation is trivial: If we can efficiently certify almost all instances with dn clauses where d is above the threshold as unsatisfiable, then we can efficiently certify almost all instances with $d'n$ clauses where $d' > d$ as unsatisfiable by simply chopping off the superfluous clauses. The analogous fact holds below the threshold, where we extend a given formula by some random clauses. Analogous observations apply to the number of literals in clauses.

The relationship of hardness and thresholds is rather general and not restricted to satisfiability. It is known for k-colourability of random graphs with a linear number of edges. In [PeWe89] a peak in running time seemingly related to the threshold is reported. The existence of a threshold is proved in [AcFr99] but again the value and convergence to a constant are only known experimentally. For the subset sum problem which is of a quite different nature we have also this relationship between threshold and hardness: The threshold is known and some discussion related to hardness is found in [ImNa96].

Abandoning the general complexity theoretic point of view and looking at concrete algorithms the following results are known for random k-SAT instances: All progress approximating the threshold from below is based on the analysis of rather simple polynomial time heuristics. In fact the most advanced heuristic being analyzed [Ac2000] only finds a satisfying assignment with probability of at least ε where $\varepsilon > 0$ is a small constant for 3-SAT formulas with at most $3.145n$ clauses. The heuristic in [FrSu96] finds a satisfying assignment for 3-SAT almost always for 3-SAT instances with at most $3.003n$ clauses. On the other hand the progress made in approximating the threshold from above does not provide us at all with efficient algorithms certifying the unsatisfiability of the formula at hand. Only the expectation of the number of satisfying assignments is calculated and is shown to tend to 0.

In fact beyond the threshold we have negative results: For arbitrary but fixed $d \geq 2^k \cdot \ln 2$ random k-SAT instances with dn clauses (are unsatisfiable and) have only resolution proofs with an exponential number, that is with at least $(1+\varepsilon)^n = 2^{\Omega(n)}$ clauses [ChSz88]. This has been improved upon by [Fu98], [BePi96], and [Be et al98] all proving (exponential) lower bounds for somewhat larger clause/variable ratios. Note that a lower bound on the size of resolution proofs provides a lower bound on the number of nodes in *any* classical backtracking tree as generated by any variant of the well known Davis-Putnam procedure.

Provably polynomial time results beyond the threshold are rather limited by now: In [Fu98] it is shown that k-SAT formulas with at least n^{k-1} clauses allow for polynomial size resolution proofs and thus can be certified unsatisfiable efficiently. This is strengthened in [Be et al98] to the best result known by now: For at least $n^{k-1}/\log n$ random clauses a backtracking based algorithm proves unsatisfiability in polynomial time with high probability. (In fact the result of Beame et al. is slightly stronger as it applies to formulas with $\Omega(n^{k-1}/\log n)$ random clauses.)

We extend the number of clauses for which a provably polynomial time algorithm exists. We give an algorithm which works when the number of clauses is only n to a constant fraction on k (with high probability). Our algorithm certifies k-SAT instances for k even with at least $2^k \cdot (k/2)^7 \cdot (\ln n)^7 \cdot n^{k/2} = n^{(k/2)+o(1)}$ clauses as unsatisfiable. We thus get the first improvement of existing bounds for $k = 4$. To obtain our result we leave the area of strictly combinatorial algorithms considered by now. Instead we associate a graph with a given formula and show how to certify unsatisfiability of the formula with the help of the eigenvalue spectrum of a certain matrix associated to this graph. Note that the eigenvalue spectrum can be calculated in polynomial time by standard linear algebra methods.

Eigenvalues are used in two ways in the algorithmic theory of random structures: They can be used to find a solution of an NP-hard problem in a random instance generated in such a way that it has a solution (not known to the algorithm). An example for 3-colourability is [AlKa94]. They can also be used to prove the absence of a solution of an NP-problem. However these applications are somewhat rare at the moment. The most prominent example here is the expansion property of random regular graphs [AlSp92]. Note that the expansion property is coNP-complete [Bl et al81] and the eigenvalues certify the absence of a non-expanding subset of vertices (which is the solution in this case). Our result is an example of the second kind.

1 From random formulas to random graphs

We use the following notation throughout: $\mathrm{Form}_{n,k,m}$ is our probabilistic model of k-SAT instances with m clauses over n propositional variables. Most of the time we assume that k is even. $\mathrm{Form}_{n,k,m}$ is defined as follows: The probability space of clauses of size k, $\mathrm{Clause}_{n,k}$, is the set of orderd k-tuples of literals over n propositional variables $v_1, \ldots, v_n$. We write $l_1 \vee \ldots \vee l_k$ with $l_i = x$ or $l_i = \neg x$ where x is one of our variables. Our definition of $\mathrm{Clause}_{n,k}$ allows for clauses containing the same literal twice and clauses which contain a variable and its negation in order to simplify the subsequent presentation. We consider $\mathrm{Clause}_{n,k}$ as endowed with the uniform probability distribution: The probability of a clause is given by $P(l_1 \vee \ldots \vee l_k) = (1/(2n))^k$. $\mathrm{Form}_{n,k,m}$ is the m-fold cartesian product space of $\mathrm{Clause}_{n,k}$. We write $F = C_1 \wedge \ldots \wedge C_m$ and $P(F) = (1/(2n))^{k \cdot m}$. There are several ways of defining k-SAT probability spaces. Our results refer to these spaces, too. We discuss this matter after the presentation of the algorithm.

Our algorithm uses the following graphs:

Definition 1. Let $F \in \mathrm{Form}_{n,k,m}$ be given. The graph $G = G_F$ depends only on the sequence of all-positive clauses of F:

- The set of vertices of G is $V = V_F = \{x_1 \vee \ldots \vee x_{k/2} \mid x_i \text{ a variable}\}$. We have $|V| = n^{k/2}$ and V is independent of F.
- The set of edges of G, $E = E_F$ is given by: For two different vertices $x_1 \vee \ldots \vee x_{k/2}$ and $y_1 \vee \ldots \vee y_{k/2}$ we have that $\{x_1 \vee \ldots \vee x_{k/2},\ y_1 \vee \ldots \vee y_{k/2}\} \in E$ iff $x_1 \vee \ldots \vee x_{k/2} \vee y_1 \vee \ldots \vee y_{k/2}$ (or $y_1 \vee \ldots \vee y_{k/2} \vee x_1 \vee \ldots \vee x_{k/2}$) is a clause of F. Note that it is possible that $|E| < m$ as a clause might induce no edge or two clauses induce the same edge. Our definition does not allow for loops or multiple edges.

The graph H_F is defined in a totally analogous way for the all-negative clauses of F. □

Recall that an *independent set* of a graph G is a subset of vertices W of G such that we have no edge $\{v, w\}$ in G where both $v, w \in W$. The independence number of the graph G denoted by $\alpha(G)$ is the number of vertices in a largest independent set. It is NP-hard to determine $\alpha(G)$.

Lemma 2. *If $F \in \mathit{Form}_{n,k,m}$ is satisfiable then*

$$\alpha(G_F) \geq (n/2)^{k/2} = (1/2)^{k/2} \cdot |V| \text{ or } \alpha(H_F) \geq (1/2)^{k/2} \cdot |V|.$$

As k remains constant when n gets large this means that we have independent sets consisting of a constant fraction of all vertices of G_F of H_F.

Proof. Let $\mathcal{A}$ be an assignment of the n underlying variables with the truth values $0, 1$ (where $0 =$ false and $1 =$ true) which makes F true. We assume that $\mathcal{A}$ assigns 1 to at least $n/2$ variables. Let S be this set of variables then F has no all-negative clause consisting only of literals over S. Therefore H_F has an independent set with at least $|S|^{k/2} \geq (1/2)^{k/2} \cdot n^{k/2}$ vertices. If the assignment assigns more than half of the variables a 0 the analogous statement applies to G_F. □

In the subsequent discussion we refer mainly to G_F. Of course everything applies also to H_F. We need to show that the distribution of G_F is just the distribution of a usual random graph. To this end let be $G_{n,m}$ be the probability space of random graphs with n labelled vertices and m different edges. Each graph is equally likely, that is the probability of G is $P(G) = 1/\binom{\binom{n}{2}}{m}$.

Lemma 3. *(1) Conditional on the event in* $Form_{n,k,m}$ *that* $|E_F| = r$ *the graph* G_F *is a random member of the space* $G_{\nu,r}$ *where* $\nu = n^{k/2}$ *is the number of vertices of* G_F.
(2) Let $\varepsilon > 0$. *For* $F \in Form_{n,k,m}$ *the number of edges of* G_F *is between* $m \cdot (1/2)^k \cdot (1-\varepsilon)$ *and* $m \cdot (1/2)^k \cdot (1+\varepsilon)$ *with high probability.*

Proof. (1) Let $V = \{x_1 \vee \ldots \vee x_{k/2} \,|\, x_i \text{ a variable}\}$ be the set of vertices. Let $G = (V, E)$ be a graph with $|E| = r$. We show further below that the probability of the event that $F \in \mathrm{Form}_{n,k,m}$ induces the edges set E, denoted by $P(F; E_F = E)$, depends only on r, but is independent of the actual edge set E. This implies the claim because

$$P(F; E_F = E \,|\, |E_F| = r) = \frac{P(F; E_F = E)}{P(|E_F| = r)} = 1/\binom{\binom{\nu}{2}}{r},$$

where the last equation holds because $P(F; E_F = E)$ is independent of E and therefore must be the same for all E with r edges.

It remains to show that $P(F; E_F = E)$ is independent of E. To this end we show that

$$P(F; E_F = E \,|\, F \text{ has exactly } s \text{ all-positive clauses})$$

is independent of E. This implies the claim because by conditioning

$$P(F; E_F = E) = \sum_{s \geq 0} P(F; F \text{ has } s \text{ positive clauses }) \cdot P(F; E_F = E \,|\, F \text{ has } s \text{ positive clauses}).$$

The distribution of $\mathrm{Form}_{n,k,m}$ conditional on the set of formulas with exactly s all-positive clauses is the uniform one. We therefore just need to count the number of formulas F with exactly s positive clauses such that $E_F = E$. Each such formula F with $E_F = E$ is obtained exactly once by the following choosing process: 1. Pick a sequence of s positive clauses with k literals $(C_1, \ldots, C_s) \in (\mathrm{Clause}_{n,k})^s$ which induce the edge set E. 2. Pick s positions from the m positions available and put the clauses $(C_1, \ldots, C_s)$ from left to right into the corresponding slots. 3. Fill the remaining $m - s$ positions of F with clauses containing at least one negative literal.

For 2 edge sets E, E' with r edges there is a natural (but technically not easy to describe) bijective correspondence between the $(C_1, \ldots, C_s)$ for E and the $(C'_1, \ldots, C'_s)$ for E' picked in step 1. Therefore the number of choosing possibilities is independent of the actual set E and we are done.

(2) The claim follows from the following statements which we prove further below:

- Let $\varepsilon > 0$ be fixed. The number of all-positive clauses of $F \in \mathrm{Form}$ is between $(1-\varepsilon) \cdot (1/2)^k \cdot m$ and $(1+\varepsilon) \cdot (1/2)^k \cdot m$ with high probability.

- The number of all-positive clauses like $x_1 \vee \ldots \vee x_{k/2} \vee x_1 \vee \ldots \vee x_{k/2}$, that is with the same first and second half, is $o(m)$.
- The number of unordered pairs of positions of F on which we have positive clauses which induce only one edge, that is pairs of clauses $\{x_1 \vee \ldots \vee x_k,\, y_1 \vee \ldots \vee y_k\}$ where $\{x_1 \vee \ldots \vee x_{k/2},\, x_{k/2+1} \vee \ldots \vee x_k\} = \{y_1 \vee \ldots \vee y_{k/2},\, y_{k/2+1} \vee \ldots \vee y_k\}$ is also $o(m)$ with high probability.

This implies the claim of the lemma with ε slightly lower than the ε from the first statement above because we have only $o(m)$ clauses inducing no additional edge.

The first statement: This statement follows with Chernoff bounds because the probability that a clause at a fixed position is all-positive is $(1/2)^k$ and clauses at different positions are independent. The second statement: The probability that the clause at position i has the same first and second half is $(1/n)^{k/2}$. The expected number of such clauses in a random F is therefore $m \cdot (1/n)^{k/2} = o(m)$. The third statement: We fix 2 positions $i \neq j$ of F. The probability that the clauses at these positions have the same set of first and second halves is $2 \cdot (1/n)^k$ and the expected number of such unordered pairs is at most $m^2 \cdot 2 \cdot (1/n)^k = O(m/n)$ provided $m = O(n^{k-1})$ which we can assume. Let X be the random variable counting the number of unordered pairs of positions with clauses with the same first and second half and let $\varepsilon > 0$. Markov's inequality gives us

$$P(X > n^{\varepsilon} \cdot EX) \leq EX/(n^{\varepsilon} \cdot EX) = 1/n^{\varepsilon}.$$

Therefore we get that with high probability $X \leq n^{\varepsilon} \cdot (m/n) = o(m)$. □

2 Spectral considerations

Eigenvalues of matrices associated with general graphs are somewhat less common at least in Computer Science applications than those of regular graphs. The monograph [Ch97] is a standard reference for the general case. The easier regular case is dealt with in [AlSp92]. The necessary Linear Algebra details cannot all be given here. They are very well presented in the textbook [St88].

Let $G = (V, E)$ be an undirected graph (loopless and without multiple edges) with $V = \{1, \ldots, n\}$ being a standard set of n vertices. For $0 < p < 1$ we consider the matrix $A = A_{G,p}$ as in [KrVu2000] and [Ju82] which is defined as follows:

The $(n \times n)$-matrix $A = A_{G,p} = (a_{i,j})_{1 \leq i,j \leq n}$ has $a_{i,j} = 1$ iff $\{i, j\} \notin E$ and $a_{i,j} = -(1-p)/p = 1 - 1/p$ iff $\{i, j\} \in E$. In particular $a_{i,i} = 1$. As A is real and symmetric A has n real eigenvalues when counting them with their multiplicities. We denote these eigenvalues by $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$.

Now we have an efficiently computable upper bound for $\alpha(G)$:

Lemma 4. *(Lemma 4 of [KrVu2000]) For any possible p $\lambda_1(A_{G,p}) \geq \alpha(G)$.*

Proof. Proof: Let $l = \alpha(G)$. Then the matrix $A_{G,p}$ has an $l \times l$-block which contains only 1's. This block of course is indexed with the vertices from a largest independent set. It follows from interlacing with a suitable $l \times n$-matrix N (cf. Lemma 31.5, page 396 of [vLWi]) that $\lambda_1(A_{G,p})$ is at least as large as l. This is the claim. $\square$

In order to bound the size of the eigenvalues of $A_{G,p}$ when G is a random graph we rely on a suitably modified version of the following theorem:

Theorem 5. *(Theorem 2 of [FuKo81]) Let for $1 \le i, j \le n$ and $i \le j$ $a_{i,j}$ be independent , real valued random variables (not necessarily identically distributed) satisfying the following conditions:*

- $|a_{i,j}| \le K$ *for all* $i \le j$,
- *the expectation* $Ea_{i,i} = \nu$ *for all* i,
- *the expectation* $Ea_{i,j} = 0$ *for all* $i < j$,
- *the variance* $Va_{i,j} = E[a_{i,j}^2] - (Ea_{i,j})^2 = \sigma^2$ *for all* $i < j$,

where the values K, ν, σ are constants independent of n.

For $j \ge i$ let $a_{j,i} = a_{i,j}$ and let $A = (a_{i,j})_{1 \le i,j \le n}$ be the random $(n \times n)$-matrix defined by the $a_{i,j}$. Let the eigenvalues of A be $\lambda_1(A) \ge \lambda_2(A) \ge \cdots \ge \lambda_n(A)$. With probability at least $1 - (1/n)^{10}$ the matrix A is such that

$$\mathrm{Max}\{|\lambda_i(A)| \,|\, 1 \le i \le n\} = 2 \cdot \sigma \cdot \sqrt{n} + O(n^{1/3} \cdot \log n) = 2 \cdot \sigma \cdot \sqrt{n} \cdot (1 + o(1)).$$

$\square$

We intend to apply this theorem to a random matrix $A = A_{G,p}$ where G is a random graph from the probability space $G_{n,m}$. However, in this case the entries of A are not strictly independent and Theorem 5 cannot be directly applied. We first consider random graphs from the space $G_{n,p}$ and proceed to $G_{n,m}$ later on. Recall that a random graph G from $G_{n,p}$ is obtained by inserting each possible edge with probability p independently of other edges.

For p constant and G a random member from $G_{n,p}$ the assumptions of Theorem 5 can easily be checked to apply to $A_{G,p}$. However, for sparser random graphs that is $p = p(n) = o(1)$ the situation changes. We have that $a_{i,j}$ can assume the value $-1/o(1) + 1$ and thus is not any more bounded above by a constant. The same applies to the variance: $\sigma^2 = (1-p)/p = 1/o(1) - 1$.

It can however be checked that the proof of Theorem 5 as given in [FuKo81] goes through as long as we consider matrices $A_{G,p}$ where $p = (\ln n)^7/n$. In this case we have that $K = n/(\ln n)^7 - 1$ and $\sigma = n/(\ln n)^7 - 1$. With this modification and the other assumptions just as before the proof of [FuKo81] leads to:

Corollary 6. *With probability at least $1 - (1/n)^{10}$ the random matrix A satisfies*

$$\mathrm{Max}\ \{|\lambda_i(A)| \,|\, 1 \le i \le n\} = 2 \cdot \sigma \cdot \sqrt{n} + O(n/(\ln n)^{22/6}) = 2 \cdot (1/(\ln n)^{7/2}) \cdot n \cdot (1 + o(1)).$$

Proof. We sketch the changes which need to be applied to the proof of Theorem 2 in [FuKo81]. These changes refer to the final estimations of the proof on page 237. We set

$$k := (\sigma/K)^{1/3} \cdot n^{1/6} = (\ln n)^{7/6}(1 + o(1)),$$

in fact k should be the closest even number. We set the error term

$$v := 50 \cdot n/(\ln n)^{22/6}.$$

We have

$$2 \cdot \sigma \cdot \sqrt{n} = 2 \cdot n/(\ln n)^{7/2} = 2 \cdot n/(\ln n)^{21/6}$$

which implies that $v = o(2 \cdot \sigma \cdot \sqrt{n})$. Concerning the error estimate we get

$$\frac{v \cdot k}{2 \cdot \sigma \cdot \sqrt{n} + v} = \frac{50 \cdot (\ln n)^{7/6}}{(\ln n)^{1/6}} \cdot (1 + o(1)) = 50 \cdot \ln n \cdot (1 + o(1)).$$

This implies the claim. □

Together with Lemma 4 we now get an efficiently computable certificate bounding the size of independent sets in random graphs from $G_{n,m}$.

Corollary 7. *Let G be a random member from $G_{n,m}$ where $m = ((\ln n)^7/2) \cdot n$. and let $p = m/\binom{n}{2} = (\ln n)^7/(n-1)$. We have with high probability that*

$$\lambda_1(A_{G,p}) \leq 2 \cdot (1/(\ln n)^{7/2}) \cdot n \cdot (1 + o(1)).$$

Proof. The proof is a standard transfer from the random graph model $G_{n,p}$ to $G_{n,m}$. For G random from $G_{n,p}$ the induced random matrix $A_{G,p}$ satisfies the assumptions of the last corollary. We have that with probability at least $1 - (1/n)^{10}$ the eigenvalues of $A_{G,p}$ are bounded by $2 \cdot (1/(\ln n)^{7/2}) \cdot n \cdot (1 + o(1))$.

By the Local Limit Theorem for the binomial distribution the probability that a random graph from $G_{n,p}$ has *exactly* m edges is of $\Omega(1/(n \cdot p)^{1/2}) = \Omega(1/(\ln n)^{7/2})$. This implies the claim as the probability in $G_{n,p}$ that the eigenvalue is not bounded as claimed is $O((1/n)^{10}) = o(1/(\ln n)^{7/2})$. (We omit the formal conditioning argument.) □

3 The algorithm

We consider the probability space of formulas $\text{Form} = \text{Form}_{n,k,m}$ where k is even and the number of clauses is

$$m = 2^k \cdot (\ln n^{k/2})^7 \cdot n^{k/2} = 2^k \cdot (k/2)^7 \cdot (\ln n)^7 \cdot n^{k/2}.$$

Given a random formula F from Form the algorithm first considers the all-positive clauses from F and constructs the graph G_F. From Lemma 3 we know that $G = G_F$ is a random member of $G_{\nu,\mu}$ where $\nu = n^{k/2}$ and $\mu \geq m \cdot (1/2)^k \cdot (1-\varepsilon) = (\ln \nu)^7 \cdot \nu \cdot (1-\varepsilon)$, where we fix $\varepsilon > 0$ sufficiently small, in fact $\varepsilon = 1/2$ will do. In case the number of edges is smaller than this bound the algorithm fails.

The algorithm determines the matrix $A = A_{G,p}$ where $p = \mu/\binom{\nu}{2} \geq (\ln \nu)^7/(\nu - 1)$. ¿From Corollary 7 we get that with high probability

$$\lambda_1(A) \leq 2 \cdot (1/(\ln \nu)^{7/2}) \cdot \nu \cdot (1 + o(1)) < (1/2)^{k/2} \cdot \nu$$

for n sufficiently large. In case the second inequality does not hold the algorithm fails. By Lemma 4 G_F has no independent set with $(1/2)^{k/2} \cdot \nu$ vertices.

The algorithm proceeds in the same way for the all negative clauses and the graph H_F. In case it succeeds (which happens with high probability) we have that F is unsatisfiable by Lemma 2.

In case we want to apply this algorithm when the number of literals per clause k is odd we first extend each clause by a random literal. The algorithm succeeds when the number of clauses is $2^{k+1} \cdot ((k+1)/2)^7 \cdot (\ln n)^7 \cdot n^{(k+1)/2}$.

Some technical matters come up when this algorithm is applied to other $k-$SAT probability spaces used in the literature. The first problem arises when the formulas are defined such that clauses are not allowed to contain the same literal several times. This implies that certain edges are excluded from the graph G_F and we cannot any more speak of a random graph.

The probability that a random clause from our space $\mathrm{Clause}_{n,k}$ has the same literal several times is bounded from above by $O(k^2 \cdot (1/n)) = O(1/n)$ and bounded from below by $1/n$. Thus the expected number of clauses with the same literal several times in a formula from the space $\mathrm{Form}_{n,k,m}$ is $O(m/n)$. Recall that $m > n^{k/2}$ for our algorithm to work so there are quite a few clauses with double occurrences. By the Local Limit Theorem for the binomial distribution with parameters m and $\Theta(1/n)$ the probability that a formula from $\mathrm{Form}_{n,k,m}$ has exactly the expected number of clauses with double occurrences is $\Omega(1/(m/n)^{1/2})$. Let $\nu = n^{k/2}$ and $m = \Theta((\ln \nu)^7 \cdot \nu)$ then still we have that $O(1/\nu)^{10}) = o(1/(\ln \nu)^{7/2} \cdot 1/(m/n)^{1/2})$ cf. the proof of Corollary 7.

Now, given a random sequence of clauses F' without double occurrences of literals we add randomly exactly the expected number of clauses with double occurrences of literals to get the formula F. Then we apply our algorithm to the resulting formula. With high probability (by the above local limit consideration) the algorithm certifies that G_F has only independent sets with $o(\nu)$ vertices given the number of clauses of F is $m = 2^k \cdot (\ln \nu)^7 \cdot \nu$. After deleting the edges of G_F which are induced by the $O(m/n)$ double occurrence clauses any independent set can only increase by $O(2m/n)$ vertices. This implies that we still have no linear size independent set of vertices in G'_F. This and the same consideration for the graph $H_{F'}$ certifies the unsatisfiability of F'.

The ramaining variants of probability spaces (clauses as sets, formulas as sets, picking each clause with a probability p) can more easily be dealt with.

Conclusion

By now a large part of the algorithmic theory of random structures is concerned with efficient algorithms *finding* solutions to an NP-problem. Often the probability spaces used are designed in such a way that we know a solution is present and the algorithm then must find it (or any other solution).

The present paper is concerned with the complementary aspect. We certify efficiently the *absence* of a solution to an NP-problem which we know not to be present by non-efficient means. It seems that this aspect is by now somewhat neglected in the algorithmic theory of random structures. We think it deserves more attention as it may lead to natural questions about natural probability spaces. Spectral methods are one way to deal with these problems. A paper in the same spirit is the recent [KrVu2000] where the *non-existence* of a colouring with a given number of colours is certified by spectral methods.

One problem which can be directly treated based on the ideas developed here is the 3-colouring problem of sparse random graphs: For random graphs with $c \cdot n$ edges the following facts are known: For $c \leq 1.932$ graphs are 3-colourable with high probability [AcMo97]. For $c \geq 2.522$ graphs are not 3-colourable [DuZi98] [AcMo]. There is a sharp threshold [AcFr99] with experimental hardness.

The results of the present paper imply that for $c = c(n) \geq 1/2 \cdot (\ln n)^7$ we can efficiently certify that we do not have any more an independent set with $n/3$ vertices (with high probability). Therefore we have no 3-colouring.

The following two problems however seem to require new ideas: First, the efficient certification of unsatisfiability of formulas with less than $n^{k/2}$ clauses. The problem here is that the average degree in the graph G_F now is $o(1)$ and the bounds on the eigenvalues make no sense. Second, to improve the bound of $n^2/\log n$ known for 3-SAT.

References

[Ac2000] Dimitris Achlioptas. Setting 2 variables at a time yields a new lower bound for random 3-SAT. In Proceedings SToC 2000, ACM.

[AcFr99] Dimitris Achlioptas, Ehud Friedgut. A threshold for random k-colourability. Random Structures and Algorithms 1999.

[AcMo97] Dimitris Achlioptas, Mike Molloy. Analysis of a list colouring algorithm on a random graph. In Proceedings FoCS 1997, IEEE.

[AcMo] Dimitris Achlioptas, Mike Molloy. Almost all graphs with 2.522n edges are not 3-colourable. Undated manuscript.

[AlKa94] Noga Alon, Nabil Kahale. A spectral technique for colouring random 3-colourable graphs (preliminary version). In Proceedings 26th SToC, 1994, ACM, 346-355.

[AlSp92] Noga Alon, Joel H. Spencer. The Probabilistic Method. Wiley & Sons Inc., 1992.

[Be et al98] Paul Beame, Richard Karp, Toniann Pitassi, Michael Saks. On the complexity of unsatisfiability proofs for random k-CNF formulas. In Proceedings 30th SToC, 1998, ACM, 561-571.

[BePi96] Paul Beame, Toniann Pitassi. Simplified and improved resolution lower bounds. In Proceedings 37th FoCS, 1996, IEEE, 274-282.

[Bo85] Bela Bollobas. Random Graphs. Academic Press, 1985.

[Bl et al81] Manuel Blum, Richard Karp, Oliver Vornberger, Christos H. Papadimitriou, Mihalis Yannakakis. The complexity of testing whether a graph is a superconcentrator. Information Processing Letters 13, 1981, 164-167.

[Ch97] Fan R. K. Chung. Spectral Graph Theory. American Mathematical Society, 1997.

[ChRe92] Vasek Chvatal, Bruce Reed. Mick gets some (the odds are on his side). In Proceedings 33nd FoCS, 1992, IEEE, 620-627.

[ChSz88] Vasek Chvatal, Endre Szemeredi. Many hard examples for resolution. Journal of the ACM 35(4), 1988, 759-768.

[CrAu96] J. M. Crawford, L. D. Auton. Experimental results on the crossover point in random 3-SAT. Artificial Intelligence 81, 1996.

[Du et al2000] Olivier Dubois, Yacine Boufkhad, Jacques Mandler. Typical random 3-SAT formulae and the satisfiability threshold. In Proceedings SoDA 2000, SIAM.

[DuZi98] Paul E. Dunne, Michele Zito. An improved upper bound for the non-3-colourability threshold. Information Processing Letters 1998.

[Fu98] Xudong Fu. The complexity of the resolution proofs for the random set of clauses. Computational Complexity 1998.

[Fr99] Ehud Friedgut. Necessary and sufficient conditions for sharp thresholds of graph properties and the k-SAT problem. Journal of the American Mathematical Society 12, 1999, 1017-1054.

[FrSu96] Alan M. Frieze, Stephen Suen. Analysis of two simple heuristics on a random instance of k-SAT. Journal of Algorithms 20(2), 1996, 312-355.

[FuKo81] Z. Furedi, J. Komlos. The eigenvalues of random symmetric matrices. Combinatorica 1(3), 1981, 233-241.

[Go96] Andreas Goerdt. A threshold for unsatisfiability. Journal of Computer and System Sciences 53, 1996, 469-486.

[ImNa96] Russel Impagliazzo, Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. Journal of Cryptology 9, 1996, 199-216.

[Ju82] Ferenc Juhasz. The asymptotic behaviour of Lovasz theta function for random graphs. Combinatorica 2(2), 1982, 153-155.

[KrVu2000] Michael Krivelevich, Van H. Vu. Approximating the independence number and the chromatic number in expected polynomial time. In Proceedings ICALP 2000, LNCS 1853, 13-24.

[KiKrKrSt98] Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, Yiannis Stamatiou. Approximating the unsatisfiability threshold of random formulas. Random Structures and Algorithms 12(3), 1998, 253-269.

[PeWe89] A. D. Petford, Dominic Welsh. A Randomised 3-colouring algorithm. Discrete Mathematics 74, 1989, 253-261.

[SeMiLe96] Bart Selman, David G. Mitchell, Hector J. Levesque. Generating hard satisfiability problems. Artificial Intelligence 81(1-2), 1996, 17-29.

[St88] Gilbert Strang. Linear Algebra and its Applications. Harcourt Brace Jovanovich, Publishers, San Diego, 1988.

[vLWi] J. H. van Lint, R. M. Wilson. A Course in Combinatorics. Cambridge University Press, 1992.