Combinatorics, Probability and Computing (2003) 12, 245–267. © 2003 Cambridge University Press
DOI: 10.1017/S0963548303005637 Printed in the United Kingdom

Some Results on Random Unsatisfiable *k*-Sat Instances and Approximation Algorithms Applied to Random Structures

ANDREAS GOERDT[1] and TOMASZ JURDZIŃSKI[2†]

[1] Technische Universität Chemnitz, Fakultät für Informatik,
Straße der Nationen 62, 09107 Chemnitz, Germany
(e-mail: goerdt@informatik.tu-chemnitz.de)

[2] Wroclaw University, Institute of Computer Science,
Przesmyckiego 20, 50-151 Wroclaw, Poland
(e-mail: tju@ii.uni.wroc.pl)

Received 10 June 2002; revised 11 November 2002

It is known that random k-Sat instances with at least $(2^k \cdot \ln 2) \cdot n$ random clauses are unsatisfiable with high probability. This result is simply obtained by showing that the expected number of satisfying assignments tends to 0 when the number of variables n tends to infinity. This proof does not directly provide us with an efficient algorithm certifying the unsatisfiability of a given random formula. Concerning efficient algorithms, it is essentially known that random formulas with $n^{\varepsilon} \cdot n^{k/2}$ clauses with k literals can be efficiently certified as unsatisfiable. The present paper is the result of trying to lower this bound. We obtain better bounds for some specialized satisfiability problems. These results are based on discrepancy investigations for hypergraphs.

Further, we show that random formulas with a linear number of clauses can be efficiently certified as unsatisfiable in the Not-All-Equal-3-Sat sense. A similar result holds for the non-3-colourability of random graphs with a linear number of edges. We obtain these results by direct application of approximation algorithms.

1. Introduction

Part of the interest in random k-Sat instances is due to the interesting threshold behaviour, in that there exist numbers $c_k = c_k(n)$ such that random k-Sat instances with asymptotically less than $c_k \cdot n$ random clauses are satisfiable with high probability, whereas for more than $c_k \cdot n$ random clauses we have unsatisfiability with high probability (*i.e.*, with probability tending to 1 when n, the number of variables, tends to infinity). In particular, we point out that, according to current knowledge, $c_k = c_k(n)$ lies in a bounded interval depending only on k, but it is not proved that it really is a constant independent of n; see [10].

† Supported by DFG, grant GO 493/1-1. This research was done while the second author was at Technische Universität Chemnitz, Fakultät für Informatik.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

We are interested in 'efficient certification' of unsatisfiability of a random k-Sat instance. That is, we look for an efficient, deterministic algorithm which, given a k-Sat instance as input, either states that it cannot determine that the input formula is unsatisfiable, or states that the input formula is unsatisfiable. First of all, we require the algorithm to be *correct* in that the unsatisfiable answer implies that the input formula really is unsatisfiable.

Second, we require the algorithm to be *complete*, a notion which we explain now. We are given a family of probability spaces of random k-Sat instances indexed by n, the number of propositional variables considered. For example, let c be a constant; then a typical nth probability space of such a family might be the set of formulas consisting of $c \cdot n$ clauses endowed with the uniform probability distribution. Now assume that we are given such a family, and the formulas from the probability spaces in question are unsatisfiable with high probability (that is, the probability of the set of unsatisfiable formulas in the nth space tends to 1 when n tends to infinity). *Completeness* of the algorithm simply means that, when picking formulas from the nth space, the answer is 'unsatisfiable' with high probability, too. Note that this does *not* mean that the algorithm answers 'unsatisfiable' on all unsatisfiable inputs from the nth space, but only on a subset whose probability tends to 1. We always use the notion of efficient certification in the sense described, as well as for properties other than unsatisfiability.

A k-clause is a disjunction of k literals; the exact definition we have used can be found at the beginning of Section 4. From [11] and [15], it is essentially known that for (the family of probability spaces of) random k-Sat instances with $n^{\varepsilon} \cdot n^{k/2}$ k-clauses, we can efficiently certify unsatisfiability. Note that, probabilistically, we know much more: random k-Sat instances with at least $(2^k \cdot \ln 2) \cdot n$ k-clauses are unsatisfiable with high probability. It is thus an obvious problem to attempt to decrease the bound of $n^{\varepsilon} \cdot n^{k/2}$. Apparently, two recent papers ([9] and [3]) are motivated by this problem; however, a better bound is not known. Note that it is often the case that we know probabilistically much more than can be efficiently certified.

Here we look at the following specialized satisfiability problems. For l-Out-of-k-Sat we require that at least l literals per clause must be *true* in order for the formula to be satisfied. The usual satisfiability problem has $l = 1$. For $l = 2$ and $k = 5$ we can certify unsatisfiability for $n^{\varepsilon} \cdot n^{3/2}$ random 5-clauses. This is better than the bound of $n^{\varepsilon} \cdot n^{5/2}$ clauses which can be obtained for 1-Out-of-5-Sat (5-Sat) along the lines of [11]. To obtain our result, we make some observations concerning the efficient certification of 'discrepancy properties' (see Definition 1, of random 3-uniform hypergraphs). These observations may be of independent interest.

Moreover, we look at the Not-All-Equal-3-Sat problem. Recall that a formula is satisfied in the Not-All-Equal sense if each clause contains at least one true literal and one false literal. We show that unsatisfiability in this case can in fact be certified for a linear number of 3-clauses. We apply an approximation algorithm from [18] to get our result. To the best of the authors' knowledge, approximation algorithms have so far not been used in this way to certify efficiently properties of random structures. The result is particularly interesting in view of the fact that the following hypothesis is put forward in [9]. For the classical 3-Sat problem, efficient certification of unsatisfiability for a linear number of random

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

clauses is not possible. Further, by direct application of an approximation algorithm for the Max-3-Cut problem, we obtain an algorithm efficiently certifying non-3-colourability of random graphs with a (sufficently large, but) linear number of edges.

Section 2 serves to introduce the techniques of the present paper in a clear and simple context. Section 3 contains our results concerning 3-uniform hypergraphs. In Section 4 we look into the l-Out-of-k-Sat problem. Further, in Sections 5 and 6 we deal with the Not-All-Equal-3-Sat and 3-colourability problems. A preliminary version of this paper is [16].

2. Certifying discrepancy properties of random graphs

This section is mainly based on [8] and serves as a motivation for the next section, in which discrepancy problems for hypergraphs are treated. In particular, we refer to [8, pp. 71 *ff*]. We consider undirected graphs $G = (V, E)$, as usual, where $E \subseteq \{\{x, y\} \mid x, y \in V, x \neq y\}$. For $x \in V$ we denote the degree of x by d_x. For $X, Y \subseteq V$ we write

$$E(X, Y) = \{(x, y) \mid \{x, y\} \in E, x \in X \text{ and } y \in Y\},$$

and $e(X, Y) = |E(X, Y)|$. Note that $E(X, Y)$ consists of *ordered* pairs. Let $E(X) = \{\{x, y\} \mid \{x, y\} \in E, x, y \in X\}$ be the set of *edges* of G inside X and $e(X) = |E(X)|$. For X, Y disjoint, $E(X, Y)$ corresponds to the set of edges of G with one endpoint in X and the other endpoint in Y. In $E(X, X)$, each edge of $E(X)$ occurs twice, and we have $e(X) = (1/2) \cdot e(X, X)$.

Let $e = e(V)$ denote the number of all edges of the graph $G = (V, E)$ and let n be the number of vertices; then the *edge density* of G is $\rho = e/\binom{n}{2} \approx 2e/n^2$. When picking a set $\{x, y\}$ of two vertices $x \neq y$ uniformly at random, ρ is the probability that $\{x, y\} \in E$. Picking a set $X \subseteq V$ with m elements uniformly at random from all such sets, the expectation of $e(X)$ is

$$\rho \cdot \binom{m}{2} \approx e \cdot \frac{m}{n} \cdot \frac{m}{n}. \tag{2.1}$$

For $Y = V \backslash X$ the expectation of $e(X, Y)$ is

$$\rho \cdot m \cdot (n - m) \approx 2 \cdot e \cdot \frac{m}{n} \cdot \left(1 - \frac{m}{n}\right). \tag{2.2}$$

These expectations can be computed as follows. For each edge of G we introduce indicator random variables indicating that the edge considered is inside X or goes from X to Y. The expectations of these indicators are easily calculated, and linearity of expectation yields the rest. Discrepancy refers to the deviation of $e(X)$ and $e(X, Y)$ from the expectation.

Definition 1. (Discrepancy notions)

(a) For $G = (V, E)$ the discrepancy of $X \subseteq V$ is

$$\mathrm{Disc}_G(X) = \left| e(X) - \rho \cdot \binom{|X|}{2} \right|.$$

(b) The discrepancy of X and $Y \subseteq V$, where $Y = V \backslash X$, in the graph G is

$$\mathrm{Disc}_G(X, Y) = \left| e(X, Y) - \rho \cdot |X| \cdot |Y| \right|.$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

(c) For $0 < \alpha < 1$ and $m = \alpha n$, the (relative) α-discrepancy of G is the maximum of all

$$\frac{\mathrm{Disc}_G(X)}{\rho \cdot \binom{m}{2}} \quad \text{and} \quad \frac{\mathrm{Disc}_G(X, Y)}{\rho \cdot m \cdot (n-m)}$$

where $X \subseteq V$, $|X| = \alpha n$ and $Y = V \backslash X$.

When considering $G = (V, E)$ with $V = \{1, \ldots, n\}$ and $\varepsilon' > 0$, constant, we say that G has *low discrepancy* with respect to ε' if and only if the α-discrepancy for all $\varepsilon' < \alpha < 1 - \varepsilon'$ of G is $o(1)$ (uniformly in α). That is, low discrepancy means that, for all α considered, and for every subset X with $|X| = \alpha n$, $Y = V \backslash X$, we have

$$e(X) = e \cdot \alpha^2 \cdot (1 + o(1)) \tag{2.3}$$

and

$$e(X, Y) = 2 \cdot e \cdot \alpha \cdot (1 - \alpha) \cdot (1 + o(1)). \tag{2.4}$$

From equations (2.1) and (2.2) we see that equations (2.3) and (2.4) imply that $e(X)$ and $e(X, Y)$ are asymptotically equal to their expected values. In the case of (2.3) and (2.4) and similar situations, we use the terminology that the asymptotic fraction of the edges inside X is α^2 and that of edges between X and Y is $2 \cdot \alpha \cdot (1 - \alpha)$.

The probability space $G_{n,p}$ is the space of random graphs over the set of vertices $V = \{1, \ldots, n\}$ in which each possible edge is thrown in with probability p. Usually our graphs are moderately sparse in that $p = n^{\varepsilon}/n$, with $\varepsilon > 0$ being a constant.

Lemma 2.1. *Let $p = n^{\varepsilon}/n$, where $\varepsilon > 0$ is a constant. The following statements hold with high probability for a random G from $G_{n,p}$.*

(a) *For all vertices x of G, we have that the degree of x is $d_x = n^{\varepsilon} \cdot (1 + o(1))$.*
(b) *Given constant $\varepsilon' > 0$, then G is of low discrepancy with respect to ε'.*

Proof. **(a)** For a fixed vertex x, the degree of x is a random variable defined on the space $G_{n,p}$. This random variable follows the binomial distribution with parameters $n - 1$ and p. The expectation of d_x is $n^{\varepsilon}(1 + o(1))$. Now we apply the usual Chernoff bound. For X binomially distributed and all $1 \geqslant \delta \geqslant 0$, we have

$$\mathrm{Prob}[|X - E[X]| \geqslant \delta \cdot E[X]] \leqslant 2 \cdot e^{-(1/3)\delta^2 E[X]}. \tag{2.5}$$

With high probability we get concentration of d_x at n^{ε} for *all* n vertices x when we pick $\delta = 1/n^{\varepsilon/4}$, for example, and observe that $n \cdot 2 \cdot e^{-(1/3)\delta^2 n^{\varepsilon}}$ is still $o(1)$.

(b) First we determine the edge density of a random G from $G_{n,p}$. Not surprisingly, we have with high probability that $\rho = p(1 + o(1))$ as follows, with inequality (2.5).

Let $\varepsilon' < \alpha < 1 - \varepsilon'$, and let $X \subseteq V$ with $|X| = \alpha n$ be a fixed set of vertices. The random variable $e(X)$ is binomially distributed with parameters $\binom{\alpha n}{2}$ and p. The expectation of $e(X)$ is

$$p \cdot \binom{\alpha n}{2} = \frac{1}{2} \cdot n \cdot n^{\varepsilon} \cdot \alpha^2 \cdot (1 + o(1)).$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

Using the bound (2.5) with $\delta = 1/\log_2 n$, for example, the result follows. Note that we have at most 2^n possible sets X, and multiplying yields an exponent certainly bounded above by $n - (1/3) \cdot \delta^2 \cdot (1/2) \cdot n \cdot n^{\varepsilon} \cdot \varepsilon'^2$, which still tends to $-\infty$. Therefore, with high probability for *all* X with $|X| = \alpha n$, where $\varepsilon' < \alpha < 1 - \varepsilon'$, we have

$$e(X) = p \cdot \binom{\alpha n}{2} \cdot (1 + o(1)) = \rho \cdot \binom{\alpha n}{2} \cdot (1 + o(1)).$$

The second requirement follows similarly, as the expectation of $e(X, Y)$ is $p \cdot \alpha n \cdot (n - \alpha n) = \rho \cdot \alpha n \cdot (n - \alpha n) \cdot (1 + o(1))$. □

We are interested in efficient certification, in the sense of the Introduction, of the properties of Lemma 2.1. As the degrees can simply be counted, picking a suitable function for the $o(1)$-term, for example $1/n^{\varepsilon/4}$, can be derived from the proof; the degree concentration property as stated in Lemma 2.1(a) can trivially be efficiently certified for a random G from $G_{n,p}$. As far as Lemma 2.1(b) is concerned, this is not so obvious, as bounding the discrepancy of a given graph is in general co-$\mathcal{NP}$-complete (see a remark in [8, p. 71], or see [5]). For graphs in which all vertices have the same degree, that is, for regular graphs, the eigenvalues of the adjacency matrix are used to obtain efficient bounds on the discrepancy: see [1, pp. 119 *ff*]. The linear algebra required can be found in [22]. In the regular case, we have that the all-1s-vector is an eigenvector of the adjacency matrix whose eigenvalue is the degree of the vertices of the graph considered. In our 'almost regular' case, this no longer holds, and we have to resort to the Laplacian matrix instead of the adjacency matrix, as in [8]. For $G = (V, E)$ with $V = \{1, \ldots, n\}$, we define the matrix $L_G(x, y)$ for $x, y \in V$ by $L_G(x, y) = 1$ if $x = y$ and $d_x \geqslant 1$ and $L_G(x, y) = 0$ otherwise. The matrix $H_G(x, y)$ for $x, y \in V$ is defined by $H_G(x, y) = 1/\sqrt{d_x \cdot d_y}$ if $\{x, y\} \in E$ and 0 otherwise. The Laplacian matrix of G is $\mathscr{L}_{\mathscr{G}} = L_G - H_G$. Throughout we assume that, for all x, the degree satisfies $d_x \geqslant 1$, and L_G is the identity matrix. For $v = (\sqrt{d_1}, \ldots, \sqrt{d_n})^{\mathrm{T}}$ (T denoting the transpose) it is easy to check that $H_G \cdot v = v$. That is, v is an eigenvector with eigenvalue 1 of H_G. As H_G is symmetric and real-valued, it is known that H_G has n real eigenvalues ρ_i, which can be ordered as $\rho_0 = 1 \geqslant \rho_1 \geqslant \rho_2 \geqslant \cdots \geqslant \rho_{n-1}$. The actual Laplacian $\mathscr{L}_{\mathscr{G}}$ has the same eigenvectors as H_G, and the eigenvalues are $\lambda_i = 1 - \rho_i$. Let $\bar{\lambda}$ be defined by

$$\bar{\lambda} = \max\{|1 - \lambda_i| \mid i \neq 0\} = \max\{|\rho_i| \mid i \neq 0\}.$$

That is, $\bar{\lambda}$ is simply the maximum of the absolute values of all eigenvalues, except the largest eigenvalue ρ_0 of the modified adjacency matrix H_G. The importance of $\bar{\lambda}$ is that it helps to bound the discrepancy. The subsequent Fact 2.2 can be found as Theorem 5.1 and its corollaries in [8, pp. 72 and 73]. For a graph G and X a subset of the vertices of G, the *volume* of X is $\mathrm{Vol}\, X = \sum_{x \in X} d_x$.

Fact 2.2. *Let $G = (V, E)$ and let $X, Y \subseteq V$. Then we have:*

(a)

$$\left| e(X, Y) - \frac{\mathrm{Vol}\, X \cdot \mathrm{Vol}\, Y}{\mathrm{Vol}\, V} \right| \leqslant \bar{\lambda} \cdot \sqrt{\mathrm{Vol}\, X \cdot \mathrm{Vol}\, Y},$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

(b)

$$\left| e(X, X) - \frac{(\mathrm{Vol}\, X)^2}{\mathrm{Vol}\, V} \right| \leqslant \bar{\lambda} \cdot \mathrm{Vol}\, X,$$

(c)

$$\left| e(X) - \frac{1}{2} \cdot \frac{(\mathrm{Vol}\, X)^2}{\mathrm{Vol}\, V} \right| \leqslant \frac{1}{2} \cdot \bar{\lambda} \cdot \mathrm{Vol}\, X.$$

Let $d = d(n)$ be given. For $G = (V, E)$ with $V = \{1, \ldots, n\}$, we say G is *almost d-regular* if and only if, for all vertices x of G, we have $d_x = d \cdot (1 + o(1))$. Note that Lemma 2.1(a) states that, with high probability, a random G from $G_{n,p}$ is almost d-regular where $d = n^{\varepsilon}$. In the case of almost-d-regularity, we have for all $X \subseteq V$ that $\mathrm{Vol}\, X = |X| \cdot d \cdot (1 + o(1))$. Under the premise of almost-d-regularity we have, abbreviating $e = |E|$,

$$\frac{(\mathrm{Vol}\, X)^2}{\mathrm{Vol}\, V} = \frac{(d \cdot |X|)^2}{d \cdot n} \cdot (1 + o(1)) = n \cdot d \cdot \frac{|X|^2}{n^2} \cdot (1 + o(1)) = 2e \cdot \frac{|X|^2}{n^2} \cdot (1 + o(1))$$

and, for $Y = V \backslash X$,

$$\frac{\mathrm{Vol}\, X \cdot \mathrm{Vol}\, Y}{\mathrm{Vol}\, V} = \frac{n \cdot d \cdot |X| \cdot d \cdot |Y|}{d \cdot n^2} \cdot (1 + o(1)) = 2e \cdot \frac{|X|}{n} \cdot \frac{n - |X|}{n} \cdot (1 + o(1)),$$

and get the following fact.

Fact 2.3. *Let $G = (V, E)$, where $V = \{1, \ldots, n\}$, be almost d-regular; let $\varepsilon' < \alpha < 1 - \varepsilon'$; let $X \subseteq V$ with $|X| = \alpha n$, $Y = V \backslash X$. Finally, let $e = (1/2) \cdot d \cdot n$ be the asymptotic number of edges of G. Then we have, uniformly in α:*

(a)

$$|e(X) - e\alpha^2| \leqslant \bar{\lambda} \cdot e \cdot \alpha \cdot (1 + o(1)) + o(e),$$

(b)

$$|e(X, Y) - 2 \cdot e \cdot \alpha \cdot (1 - \alpha)| \leqslant \bar{\lambda} \cdot 2 \cdot e \cdot \sqrt{\alpha(1 - \alpha)} \cdot (1 + o(1)) + o(e).$$

If $\bar{\lambda} = o(1)$, Fact 2.3 implies low discrepancy: see equations (2.3) and (2.4).

We consider the usual 0–1-adjacency matrix of a random graph G from $G_{n,p}$ where $p = n^{\varepsilon}/n$. This adjacency matrix is symmetric and has n real eigenvalues. It is known and, for example, proved in [11] that the eigenvalue with the second-largest absolute value among all eigenvalues of such an adjacency matrix is $o(n^{\varepsilon})$ with high probability. In the case of the matrix H_G, the nonzero entries are $1/\sqrt{d_x \cdot d_y} = 1/n^{\varepsilon} \cdot (1 + o(1))$, and the eigenvalue bound for the adjacency matrix implies that $\bar{\lambda} = o(1)$ with high probability. We explain this next.

For both matrices the 'trace method' allows us to estimate the eigenvalues. More precisely, we use the fact

$$\mathrm{Trace}\left(H_G^k\right) = \sum_{i=0}^{n-1} \rho_i^k,$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

where, as usual, $\mathrm{Trace}(H_G^k)$ is the sum of the diagonal elements on H_G^k. A simple inductive argument shows that

$$H_G^k(x,x) = \sum (H_G(x,x_1)\cdot H_G(x_1,x_2)\cdot \;\cdots\; \cdot H_G(x_{k-1},x)),$$

where the sum goes over all tuples of vertices $(x_1, x_2, \ldots, x_{k-1})$, with repetitions allowed. The entry $H_G^k(x,x)$ corresponds to the number of closed walks from x to x in G. The analogous fact holds for the usual adjacency matrix, and allows us to estimate the trace of the adjacency matrix of a random graph with high probability. With $k = k(\varepsilon)$ being a sufficiently large constant, we find that the trace of the kth power of the adjacency matrix is bounded above by $n^{k\cdot\varepsilon} + o(n^{k\cdot\varepsilon})$ with high probability. In the case of H_G we get, simply by factoring out k-times $1/n^{\varepsilon}\cdot(1+o(1))$, that $\mathrm{Trace}(H_G^k) = 1 + o(1)$. The factoring out process works without difficulties as k is a constant and $(1+o(1))^k$ is still $1+o(1)$. Picking k as an even number, all terms of $\sum_{i=0}^{n}\rho_i^k$ are nonnegative; moreover, $\rho_0 = 1$. This implies that we have only an $o(1)$-remainder for the sum of all remaining ρ_i^ks. Again, k being a constant ensures that $\bar{\lambda} = o(1)$.

Given $\varepsilon' > 0$, we can now efficiently certify the following property for a random G from $G_{n,p}$ where $p = n^{\varepsilon}/n$: G has low discrepancy with respect to ε'.

The certification algorithm goes as follows. Given a random graph G from $G_{n,p}$, the algorithm first checks almost-d-regularity for $d = n^{\varepsilon}$. If this check fails, the algorithm can only give an inconclusive answer. Second, computing the matrix H_G^k for a suitable constant $k = k(\varepsilon)$, the algorithm bounds $\bar{\lambda}$. It checks that $\bar{\lambda} = o(1)$. If the check is positive the algorithm gives 'low discrepancy' as output. If the check fails, we get an inconclusive answer.

Fact 2.3 and equations (2.3) and (2.4) imply the correctness of the algorithm. Completeness follows from Lemma 2.1(a) and the discussion above on $\bar{\lambda}$.

At this point we know that an algorithm efficiently certifying low discrepancy exists, because there are suitable $o(1)$-terms, as we know from our previous consideration. In order to be of practical use, we would have to specify the $o(1)$-terms. Suitable functions can be derived from our proofs. The algorithm clearly works in polynomial time, with the polynomial depending on ε. The reason for this dependence on ε is that, for the k above, we have $k = k(\varepsilon)$. Analytical approximation methods to estimate the eigenvalues may possibly be used to remove this dependency on ε, but we have not looked into this.

3. Certifying discrepancy properties of random uniform hypergraphs

We are interested in k-uniform hypergraphs $H = (V, E)$, where $V = \{1, \ldots, n\}$ is a standard set of n vertices and $E \subseteq V^k$ is simply a set of *ordered* k-tuples. In this section we focus on $k = 3$. For $X, Y, Z \subseteq V$ we let $E(X,Y,Z) = E \cap (X \times Y \times Z)$, and thus, for example, $E(X,X,X)$ is the set of 3-tuples of H inside X. We denote $e(X,Y,Z) = |E(X,Y,Z)|$ and usually $e = |E|$. We say that a 3-tuple is of *type* (X,Y,Z) if it belongs to $X \times Y \times Z$. Let $\varepsilon' > 0$. We say that H has low discrepancy with respect to ε' if and only if, for all α with $\varepsilon' < \alpha < 1 - \varepsilon'$ and for all $X \subseteq V$ with $|X| = \alpha n$, $Y = V \setminus X$, we have that $e(X,X,X) = \alpha^3 \cdot e \cdot (1 + o(1))$ and $e(X,X,Y) = \alpha^2 \cdot (1-\alpha) \cdot e \cdot (1 + o(1))$, and analogously

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

for the remaining 6 possibilities of placing X and Y into the 3 positions. This is analogous to the case of graphs: compare equations (2.3) and (2.4).

Let $d = d(n)$ be given. The hypergraph H is *almost d-regular* if, for each $x \in V$, the number of occurrences of x at each of the 3 possible positions is $d \cdot (1 + o(1))$. Moreover, almost-d-regularity requires two additional properties:

$$|\{(x,y,z) \in E| \text{ there is an } (x',y',z') \neq (x,y,z) \text{ in } E \text{ such that } |\{x,y,z\} \cap \{x',y',z'\}| \geqslant 2\}| \leqslant o(e)$$

and

$$|\{(x,y,z) \in E| \; |\{x,y,z\}| \leqslant 2\}| \leqslant o(e).$$

Note that d-regularity implies in particular that $e = d \cdot n \cdot (1 + o(1))$. Selecting each possible 3-tuple with probability p, we get the probability space $H_{n,p}$ of random 3-uniform hypergraphs over n vertices. For $m = \alpha n$ and $X \subseteq V$ with $|X| = m$, the random variable $e(X,X,X)$ follows the binomial distribution with parameters m^3 and p. As in Lemma 2.1, we get that $e(X,X,X)$ is with high probability concentrated at $\alpha^3 \cdot n^{1+\varepsilon}$, provided that $p = n^{\varepsilon}/n^2$. The analogous fact applies to all types consisting of X and $Y = V \backslash X$. As in the case of graphs, this applies to *all* sets X with $|X| = \alpha n$ uniformly for $\varepsilon' < \alpha < 1 - \varepsilon'$, and we say that we have low discrepancy with respect to ε', just as in the case of graphs.

As we have no obvious notion of adjacency matrix, let alone Laplacian matrix, for hypergraphs, discrepancy properties of random hypergraphs cannot be as easily certified as in the case of graphs at the end of the preceding section. In spite of the fact that there are some attempts to generalize spectral graph theory to hypergraphs, results relevant to our context are not known, to the best of the authors' knowledge: see [12] and [6]. Results related to our approach are obtained in [7]. Unfortunately, these results are only useful for k-uniform hypergraphs with at least $\Omega(n^{k-1})$ edges. This is too dense to be interesting to us. The related theory of quasi-random hypergraphs has so far only been developed for $\Omega(n^k)$ k-uniform edges, see the references in [7].

In order to deal with sparser hypergraphs, we define some graphs associated with a hypergraph. The product graph defined below looks to the authors like a new concept in the present context. A similar graph is only used in [11].

Definition 2. (Projection graph, product graph) Let $H = (V,E)$ be a 3-uniform hypergraph.

(a) The *projection graph* on the first and second coordinate is the graph $G = (V,F)$, where F contains all edges $\{x,y\}$ with $x \neq y$ such that there is a z with $(x,y,z) \in E$ or $(y,x,z) \in E$. Projection graphs are defined for all 3 pairs of two different coordinates.

(b) The *product graph* of H with respect to the first coordinate is the graph $G = (W,F)$, where $W = V \times V$ and F is defined as follows. For $x_1, x_2, y_1, y_2 \in V$ with $(x_1,y_1) \neq (x_2,y_2)$ we have $\{(x_1,y_1),(x_2,y_2)\} \in F$ if and only if there exists an $u \in V$ and two different 3-tuples $(u,x_1,x_2) \in E$ and $(u,y_1,y_2) \in E$. The product graph with respect to the second or third coordinate is defined in the same way.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

When H is almost d-regular, any projection graph has asymptotically e edges. Concerning the product graph, we observe, under the additional assumption that $d = d(n)$ tends to infinity, that the number of edges is given by

$$n \cdot d^2 \cdot (1 + o(1)) = \frac{e^2}{n} \cdot (1 + o(1)).$$

Note that two different 3-tuples from H, (u, x_1, x_2) and (u, y_1, y_2), induce the two edges $\{(x_1, y_1), (x_2, y_2)\}$ and $\{(y_1, x_1), (y_2, x_2)\}$ in the product graph. We are mainly interested in the case when the underlying hypergraph is a random hypergraph. In this case we have the following result.

Corollary 3.1. *Let $H = (V, E)$ be a random 3-uniform hypergraph from $H_{n,p}$ where $p = n^{\varepsilon}/n^{3/2}$ and $1/2 > \varepsilon \geqslant 0$ is a constant.*

(a) *The projection graphs are random graphs from $G_{n,p'}$ with $p' = 2 \cdot n^{(1/2)+\varepsilon}/n \cdot (1 + o(1))$.*
(b) *With $d = 2 \cdot n^{2\varepsilon}$ each product graph is almost d-regular with high probability. Moreover, with high probability, each product graph has $\bar{\lambda} = o(1)$.*

Proof. **(a)** Taking into account that the edges in a hypergraph are ordered, whereas in the projection graphs they are not, we get that a given edge $\{x, y\}$ with $x \neq y$ is present in the projection graph independently, with probability $p' = 1 - (1 - p)^{2n}$, which is the probability that a random H contains at least one of the $2n$ 3-tuples (x, y, z) or (y, x, z) for a $z \in V$. From this we get that $p' \leqslant 2np = 2 \cdot n^{(1/2)+\varepsilon}/n$, as $2np$ is the expected number of 3-tuples (x, y, z) or (y, x, z) in a random G.

A lower bound for p' can be obtained using the binomial theorem and the geometric series

$$\begin{aligned} p' &= 1 - (1 - p)^{2n} \\ &= 1 - 1 + 2np - \binom{2n}{2} p^2 + \cdots - \binom{2n}{2n} p^{2n} \\ &\geqslant 2np - (2np)^2 - (2np)^3 - \cdots - (2np)^{2n} \\ &\geqslant 2np \cdot \left(1 - \frac{2np}{1 - 2np}\right) \\ &= 2np \cdot (1 - o(1)), \end{aligned}$$

where we need $2np = o(1)$, which holds as $\varepsilon < 1/2$.

(b) Concerning the degree of a vertex (x_1, y_1), we proceed as in the proof of [13, Theorem 11]. The elementary proof is based on the following ideas. Let X be the random variable that counts the number of vertices u such that H has an ordered pair of distinct 3-tuples $((u, x_1, x_2), (u, y_1, y_2))$. Let Y count the number of us with a pair of distinct 3-tuples $((u, x_2, x_1), (u, y_2, y_1))$ in H. We can show that the degree of (x_1, y_1) is, with sufficiently large probability, meaning here at least $1 - o(1/n^2)$, asymptotically equal to $X + Y$. Moreover, with similarly high probability we have that $X + Y$ is $2 \cdot n^{2\varepsilon} \cdot (1 + o(1))$. This implies the claim. Note that the expectation of $X + Y$ is asymptotically $2 \cdot n^{2\varepsilon}$.

In [11] it is shown that the eigenvalue of the adjacency matrix of a product graph with the second largest absolute value is $o(n^{2\varepsilon})$. In fact the product graph from [11] is defined

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

slightly differently from the product graph here, but this difference causes no problem. As the trace method with walks of bounded length is used and the product graph G is almost $2n^{2\varepsilon}$-regular, we have no problem in transferring this proof to the matrix H_G, as considered in the preceding section, and the claim follows. □

Note that the assumptions of the preceding lemma imply that the number of the 3-tuples of H is with high probability $e \approx n^{3/2+\varepsilon}$. The number of edges of the product graph is

$$e^2/n \approx n^{2+2\varepsilon} = \frac{2n^{2\varepsilon} \cdot n^2}{2},$$

which reflects that the product graph is almost $2n^{2\varepsilon}$-regular.

Theorem 3.2. *Let $\varepsilon' > 0$ be fixed and let $d = d(n)$ tend to infinity. Let $V = \{1, \ldots, n\}$ and let $H = (V, E)$ be a 3-uniform hypergraph having the following properties:*

- *H is almost d-regular;*
- *the projection graphs of H have low discrepancy with respect to ε';*
- *the product graphs of H have low discrepancy with respect to ε'^2.*

Then H itself has low discrepancy with respect to ε'.

Proof. Let $|E| = e$ and let $X \subseteq V$ with $|X| = \alpha n$. We show that the asymptotic fraction of 3-tuples of type (X, X, X) in H is α^3. Here $\varepsilon' \leqslant \alpha \leqslant 1 - \varepsilon'$.

Low discrepancy of the projection graph on the second and third coordinate implies that the asymptotic fraction of edges inside X is α^2 of all edges of the projection graph. The edges inside X in the projection graph can only be induced by 3-tuples of type (V, X, X) of H. By almost-d-regularity we have that the projection graph has asymptotically e edges. Therefore the fraction of 3-tuples of H of type (V, X, X) must be asymptotically α^2, too. Let $Y = V \backslash X$. We partition the 3-tuples of H of type (V, X, X) into those of type (X, X, X) and those of type (Y, X, X).

Assume for a contradiction that the asymptotic fraction of 3-tuples of type (X, X, X) in H is not α^3. To be totally precise, this means that there exists a constant $\delta > 0$ and $\varepsilon = \varepsilon(n) \geqslant \delta$ such that, for infinitely many n, we have that the fraction of 3-tuples of type (X, X, X) of H is equal to

$$\alpha^3 + \varepsilon \quad \text{or} \quad \alpha^3 - \varepsilon.$$

We assume that the first case applies; the second case can be treated similarly.

We look at the product graph with respect to the first coordinate. We show below that the asymptotic fraction of edges inside $X \times X$ in the product graph is at least

$$\alpha^4 + \frac{\varepsilon^2}{\alpha} + \frac{\varepsilon^2}{1 - \alpha}. \tag{3.1}$$

As $|X \times X| = \alpha^2 n^2$, the number of vertices of the product graph is n^2, $\varepsilon \geqslant \delta > 0$, and n can be chosen arbitrarily large, (3.1) contradicts low discrepancy with respect to ε'^2 of the product graph: recall equation (2.3).

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

For $u \in X$, let a_u be the number of 3-tuples of type $(\{u\}, X, X)$ of H. The number of edges of the product graph induced by 3-tuples of H of type (X, X, X) is asymptotically, using d-regularity,

$$\sum_{u \in X} a_u(a_u - 1) = \sum_{u \in X} \left(a_u^2 - a_u\right) = \left(\sum_{u \in X} a_u^2\right) - (\alpha^3 + \varepsilon) \cdot e, \tag{3.2}$$

as the number of 3-tuples of type (X, X, X) of H is equal to $\sum a_u = (\alpha^3 + \varepsilon) \cdot e$. Moreover, recall that two different 3-tuples (u, x_1, x_2) and (u, y_1, y_2) of H induce two edges in the product graph. By the preceding constraint on the sum of the a_u, we get that the first term, $\sum a_u^2$, in (3.2) is minimized when each a_u is the arithmetic mean $a = (\alpha^3 + \varepsilon) \cdot e/(\alpha n)$ of the a_us. Therefore,

$$\sum_{u \in X} a_u^2 \geqslant \sum_{u \in X} \left(\frac{(\alpha^3 + \varepsilon) \cdot e}{\alpha n}\right)^2 = \frac{(\alpha^3 + \varepsilon)^2}{\alpha n} \cdot e^2 = \left(\alpha^5 + 2\alpha^2\varepsilon + \frac{\varepsilon^2}{\alpha}\right) \cdot \frac{e^2}{n}. \tag{3.3}$$

As d tends to infinity in n, the number of edges of the product graph is asymptotically e^2/n and e is asymptotically negligible with respect to e^2/n. Therefore (3.2) and (3.3) imply that the asymptotic fraction of edges of the product graph which are induced by 3-tuples of type (X, X, X) is

$$\alpha^5 + 2\alpha^2\varepsilon + \frac{\varepsilon^2}{\alpha}. \tag{3.4}$$

Now we look at the fraction of the number of edges of the product graph induced by 3-tuples of H of the type (Y, X, X). We have that the asymptotic fraction of 3-tuples of H of type (Y, X, X) is

$$\alpha^2 - (\alpha^3 + \varepsilon) = (1 - \alpha) \cdot \alpha^2 - \varepsilon.$$

As before, we come to

$$\sum_{u \in Y} a_u^2 \geqslant \frac{((1 - \alpha) \cdot \alpha^2 - \varepsilon)^2}{(1 - \alpha)n} \cdot e^2 = \left(\alpha^4 - \alpha^5 - 2\alpha^2\varepsilon + \frac{\varepsilon^2}{1 - \alpha}\right) \cdot \frac{e^2}{n}, \tag{3.5}$$

and the asymptotic fraction of the number of edges of the product graph induced by 3-tuples of type (Y, X, X) is at least

$$\alpha^4 - \alpha^5 - 2\alpha^2\varepsilon + \frac{\varepsilon^2}{1 - \alpha}. \tag{3.6}$$

Since the sum of (3.4) and (3.6) gives (3.1), we are done.

Finally, to show that the 3-tuples of the remaining 7 types occur with the required frequencies among the 3-tuples of H, we can proceed in an analogous way. Note that the proof for the type (Y, X, X) is implicitly contained in the proof above. □

Let $0 \leqslant \varepsilon < 1/2$ be a constant and let $\varepsilon' > 0$. We give an algorithm which efficiently certifies that a random 3-uniform hypergraph H from $H_{n,p}$, where $p = n^{\varepsilon}/n^{3/2}$ has low discrepancy with respect to ε'. First, we check almost-d-regularity with $d = n^{(1/2)+\varepsilon}$. Note that the expectation of the number of occurrences of a given vertex at a given coordinate of a random H is $n^2 p = n^{(1/2)+\varepsilon} = d$. As in Lemma 2.1, the number of occurrences of

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

each vertex is concentrated at the expectation with high probability. As $\varepsilon < 1/2$ and constant, simple expectation calculations together with Markov's inequality show that the remaining two requirements of almost-d-regularity hold. Thus the check of almost-d-regularity succeeds with high probability, and the algorithm continues.

We approximate the $\bar{\lambda}$s of all projection and product graphs of H, and certify that they are $o(1)$. This check is positive with high probability, which follows from Corollary 3.1 and the remarks on $G_{n,p}$ in the preceding section. Note that the application of Corollary 3.1(b) requires $p = n^{\varepsilon}/n^{3/2}$.

The algorithm is correct, as Fact 2.3 and the preceding Theorem 3.2 imply low discrepancy of H with respect to ε'.

4. Applications to random unsatisfiable k-Sat instances

We consider propositional formulas over a standard set of n propositional variables $\mathrm{Var}_n = \{x_1, \ldots, x_n\}$. For a variable x, we call x positive literal and $\neg x$ negative literal. The set of k-clauses over Var_n, $\mathrm{Clause}_{n,k}$, is simply the k-fold Cartesian product of positive and negative literals over Var_n. Thus clauses are allowed to contain the same literal twice, or x and $\neg x$ together. However, this is only a technical concession, and using different common notions of clauses would not change our results. Altogether we have that $|\mathrm{Clause}_{n,k}| = (2n)^k$. Clauses that contain only positive literals (negative literals, respectively) are called *all-positive* clauses (*all-negative* clauses, respectively). The probability space of random k-Sat instances, $\mathrm{Form}_{n,k,p}$, is obtained by selecting each clause independently with probability p.

It is natural to make a k-uniform hypergraph in the sense of Section 3 from a k-Sat instance: the vertex set is always the set of variables, Var_n (note: not the set of literals), and the k-tuples of the hypergraph are obtained by substituting $\neg x$ with x in the clauses. Similarly, each subset of clauses of a formula induces a hypergraph. When we use terminology originally introduced for hypergraphs for subsets of clauses of a formula, we view this subset as a hypergraph, as described.

In [9], the following Lemma 4.1 is proved for random 3-Sat instances with a linear number of clauses. The proof uses approximation algorithms based on semidefinite programming. Our subsequent Lemma 4.1 is weaker than the result from [9], in that we assume that the number of clauses is superlinear. However, our proof method is different from that of Feige. Despite being of independent interest because of its conceptual simplicity, its purpose is to introduce the technique of the subsequent 4-Sat result.

Let a be a truth value assignment for Var_n. Then F_a denotes the set of variables assigned false by a and T_a the set of variables assigned true.

Lemma 4.1. *We consider the probability space* $\mathrm{Form}_{n,3,p}$, *where* $p = n^{\varepsilon}/n^2$ *and* $1 > \varepsilon > 0$ *constant. For a random formula* F, *the following statements can be efficiently certified.*

(a) *All satisfying assignments* a *for* F *have*

$$|F_a| = (n/2)(1 + o(1)) \quad \textit{and} \quad |T_a| = (n/2)(1 + o(1)).$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

(b) *All satisfying assignments a for F satisfy the following statements; recall the hypergraph notion of type introduced at the beginning of Section 3.*

(i) *For the hypergraph from the all-positive clauses of F, the asymptotic fraction of 3-tuples of each of the types*

$$(F_a, F_a, T_a), \quad (F_a, T_a, F_a), \quad (T_a, F_a, F_a), \quad (T_a, T_a, T_a)$$

is 1/4.

(ii) *For the hypergraph from the all-negative clauses, we have the analogous statement with the types*

$$(T_a, T_a, F_a), \quad (T_a, F_a, T_a), \quad (F_a, T_a, T_a), \quad (F_a, F_a, F_a).$$

(iii) *For the hypergraph from the clauses with the first literal negative and the remaining two literals positive, we get the analogous statement for the types*

$$(T_a, F_a, T_a), \quad (T_a, T_a, F_a), \quad (F_a, F_a, F_a), \quad (F_a, T_a, T_a).$$

(iv) *For the hypergraph from the clauses with the first two literals negative and the third literal positive, we get the analogous statement with the types*

$$(T_a, T_a, T_a), \quad (T_a, F_a, F_a), \quad (F_a, T_a, F_a), \quad (F_a, F_a, T_a).$$

(v) *For the remaining possibilities of placing positive and negative literals in clauses, symmetric statements hold.*

Note that (a) means that we can efficiently certify that none of the assignments which assign true or false to at least $(n/2) \cdot (1 + \delta)$ variables, with $\delta > 0$ constant, satisfies a random F. Note that (b) implies that the number of clauses with *exactly* 2 literals becoming true under the satisfying assignment a is negligible.

Proof. **(a)** We run the following algorithm. Let H be the hypergraph made up from the all-positive clauses of F. Certify that all three projection graphs of H have low discrepancy with respect to a given sufficiently small $\varepsilon' > 0$. We use the algorithm from the end of Section 2. We need a d' for almost-d'-regularity of the projection graphs. We take

$$d' = 2np \cdot n = 2n^{\varepsilon}.$$

Certify that H is almost d-regular, where $d = n^{\varepsilon}$ is the expected degree of a given vertex at a given position of the 3-tuples.

First, observe that with high probability the certifications are successful. Let G be a projection graph of H. Then G is a random graph from $G_{n,p'}$, where

$$p' = 2np \cdot (1 + o(1)) = 2n^{\varepsilon}/n \cdot (1 + o(1)).$$

First we have $p' \leqslant 2np$ and

$$p' \geqslant 2np - \binom{2n}{2} \cdot p^2 = 2np \cdot (1 + o(1))$$

as $\varepsilon < 1$. Note that $\{x, y\}$ is an edge of the projection graph on the first and second coordinate if and only if H has one of the $2n$ 3-tuples (x, y, u) or (y, x, u) for $u \in \mathrm{Var}_n$.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

From Lemma 2.1 we know that G is almost d'-regular and has low discrepancy with high probability. The algorithm explained at the end of Section 2 certifies this. As almost-d-regularity of H holds with high probability, we can certify it. Thus the algorithm is successful with high probability.

We still have to prove the correctness of the algorithm. Assume that the algorithm has been successful for F. We show that the certified properties imply that F is not satisfied by any assignment a with $|F_a| \geqslant n \cdot (1/2 + \delta)$ where $\delta > 0$. In order to obtain a contradiction, we assume that a is an assignment with $|F_a| \geqslant n \cdot (1/2 + \delta)$, and $\delta > 0$ is a constant. Then, in each of the three projection graphs of H, we have that the edges inside F_a are asymptotically a fraction of at least $(1/2 + \delta)^2$ of all edges. Together with the certified almost-d-regularity of H, this implies that each of the following types of 3-tuples,

$$(F_a, F_a, -), \quad (F_a, -, F_a) \quad \text{and} \quad (-, F_a, F_a),$$

makes up an asymptotic fraction of $(1/2 + \delta)^2$ of the 3-tuples of H. As a is a satisfying assignment, we have that the vacant position in these 3-tuples must be filled with a variable from T_a. We therefore know that we have a fraction of $(1/2 + \delta)^2$ of 3-tuples of each of the types

$$(F_a, F_a, T_a), \quad (F_a, T_a, F_a) \quad \text{and} \quad (T_a, F_a, F_a)$$

in H.

Looking at the projection graph on the first and second coordinate, we see, using almost-d-regularity of H, that we have a fraction of at least $2 \cdot (1/2 + \delta)^2 = 1/2 + 2\delta + 2\delta^2$ edges between F_a and T_a. By low discrepancy, this fraction should be $2 \cdot (1/2 + \delta) \cdot (1/2 - \delta) = 1/2 - 2\delta^2$. This is a contradiction as long as δ does not approach 0.

Running the same algorithm for the hypergraph made up from the all-negative clauses, we can also conclude that $|T_a| \geqslant 1/2 + \delta$ does not hold for any satisfying assignment a.

(b(i)) We run the algorithm from (a). To prove the correctness of the algorithm, let F be a formula for which the certification has been successful. Let $\alpha = 1/2 \cdot (1 + o(1))$. Assume that a is a satisfying assignment of F and $|T_a| = \alpha n$. As we have certified low discrepancy of the projection graphs of H, we get as in (a) that the asymptotic fraction of the number of 3-tuples from H of each of the following types is $1/4$:

$$(F_a, F_a, -), \quad (F_a, -, F_a), \quad (-, F_a, F_a), \quad (T_a, T_a, -), \quad (T_a, -, T_a), \quad (-, T_a, T_a).$$

As a satisfies F, we know that we get an asymptotic fraction of $1/4$ of the 3-tuples of each of the types

$$(F_a, F_a, T_a), \quad (F_a, T_a, F_a), \quad \text{and} \quad (T_a, F_a, F_a).$$

Now $3/4$ of the 3-tuples from H are used up, because the preceding types correspond to disjoint sets of 3-tuples. We only have a fraction of $1/4$ of the 3-tuples for the remaining partially specified types above. This can only be achieved with $1/4$ of the 3-tuples being of type (T_a, T_a, T_a).

(b(iii)) We first run the algorithm from (a). We look at the set of clauses of F whose first literal is negative and whose remaining two literals are positive. Let K be the

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

hypergraph from these clauses. We certify low discrepancy of the projection graphs and almost-d-regularity of K, where d is as above. The certification is successful with high probability.

Let F be such that efficient certification has taken place. Let a be a satisfying assignment of F. We have that $|F_a| = \alpha n$ where $\alpha = 1/2 \cdot (1 + o(1))$. From low discrepancy and almost-d-regularity of K, we get that the asymptotic fraction of 3-tuples of K of type $(-, F_a, F_a)$ is $1/4$. As a is a satisfying assignment of F, the first position in the corresponding 3-tuples must be filled with a variable from F_a, and we get $1/4$ of 3-tuples of type (F_a, F_a, F_a). Now, from low discrepancy of the projection graphs and almost-d-regularity of K, we know, further, that only a negligible fraction of the remaining 3-tuples has a type with two F_as. We know also that the 3-tuples of one of the types

$$(T_a, F_a, -) \quad \text{or} \quad (T_a, -, F_a)$$

have a T_a in the vacant position. By looking at the projection graph on the second and third position and almost-d-regularity, we have that the fraction of 3-tuples of these types altogether is $1/2$. Moreover, each type occurs for a fraction of $1/4$, as we see from the first and third and from the first and second position and low discrepancy of the corresponding projection graphs. We now have $1/4$ of the 3-tuples of each of the types

$$(F_a, F_a, F_a), \quad (T_a, T_a, F_a), \quad (T_a, F_a, T_a).$$

Looking at the projection graph on the second and third position, we must still have the type $(-, T_a, T_a)$ with a fraction of $1/4$ among the 3-tuples of K, because such a type has not yet occurred. The vacant position must be filled with F_a, and the claim follows.

We omit the remaining cases, which can be dealt with similarly. □

Now we come to clause size $k = 4$. When considering the space $\text{Form}_{n,4,p}$ with $p = n^{\varepsilon}/n^2$, the number of clauses is, with high probability, asymptotically $n^{2+\varepsilon}$. Splitting each clause into two halves of two literals each, we get a graph with n^2 vertices and $n^{2+\varepsilon}$ random edges. Discrepancy certifications as in [15] certify the unsatisfiability of a random formula. Not much is known for sparser random 4-Sat instances, and the simple trick described no longer works when the degree of most of the n^2 vertices of the graph is 0. However, an analogue of Lemma 4.1 can be proved for sparser random 4-Sat instances.

Theorem 4.2. *For a random formula F from $\text{Form}_{n,4,p}$, where $p = n^{\varepsilon}/n^{5/2}$ and $1/2 > \varepsilon > 0$, we can efficiently certify the following properties.*

(a) *All satisfying assignments a of F have the property that $|T_a|$ and $|F_a|$ is asymptotically $n/2$.*

(b) *All satisfying assignments a of F satisfy the following statement. Among the 4-tuples of the hypergraph made from the all-positive clauses of F, the asymptotic fraction of 4-tuples of each of the following types is $1/8$. The types are all possibilities of having an odd number of variables from T_a:*

$$(F_a, F_a, F_a, T_a), \ldots, (T_a, F_a, F_a, F_a), \quad \text{and} \quad (T_a, T_a, T_a, F_a), \ldots, (F_a, T_a, T_a, T_a).$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

As in Lemma 4.1, we get analogous statements for all kinds of clauses containing negative literals. The types occurring are always such that an odd number of literals per clause is true under a.

Proof. **(a)** We look at the 4-uniform hypergraph made up from the all-positive clauses of a given random F. We consider the projection hypergraph on the first three positions of this hypergraph. That is, we simply forget the last coordinate and consider the remaining 3-uniform hypergraph. Call this hypergraph H. Then H is a random 3-uniform hypergraph with edge probability $p' = 1-(1-p)^n = n^e/n^{3/2} \cdot (1+o(1))$. With the algorithm as stated at the end of Section 3, we certify low discrepancy of H. Moreover, we certify that H has asymptotically the same number of 3-tuples as we have 4-tuples in the 4-uniform hypergraph. We certify the analogous properties for the projection hypergraph on the last three positions, H'.

Let $\delta > 0$ and let a be a satisfying assignment with $|F_a| \geqslant (1/2+\delta) \cdot n$. As H has low discrepancy, we know that H has an asymptotic fraction of $(1/2+\delta)^3$ among its 3-tuples which are of type (F_a, F_a, F_a). Moreover, the type (F_a, F_a, F_a, T_a) among the 4-tuples must be a fraction of $(1/2+\delta)^3$, as we assume a to be satisfying. Looking at H' and the type (F_a, F_a, T_a) in this hypergraph, we can derive a contradiction as in Lemma 4.1(a), if $\delta > 0$ is a constant.

We do the same for the 4-uniform hypergraph obtained from the all-negative clauses to exclude assignments with asymptotically more variables set to true than false.

(b) Our certification algorithm now runs as follows. We first run the algorithm from (a). After this we certify low discrepancy of all four 3-uniform projection hypergraphs obtained from the hypergraph made up from the all-positive clauses. We apply the algorithm from the end of Section 3, which is successful with high probability because the 3-uniform projection hypergraphs are random hypergraphs. After this we certify low discrepancy of all 6 projection graphs obtained from the hypergraph of the all-positive clauses. Note that, for each pair of positions, we get one projection graph. These projection graphs are random graphs. We apply the algorithm from the end of Section 2 to certify low discrepancy of these projection graphs. Again we can assume that the certification is successful.

From (a) we know that any satisfying assignment a has asymptotically $n/2$ variables set to true and false. From low discrepancy of the 3-uniform projection hypergraphs and the assumption that a is a satisfying assignment, we get that the fraction of 4-tuples of each of the types

$$(F_a, F_a, F_a, T_a), \quad (F_a, F_a, T_a, F_a), \quad (F_a, T_a, F_a, F_a), \quad (T_a, F_a, F_a, F_a)$$

is $1/8$. The number of edges inside F_a in each of the projection *graphs* must be asymptotically $1/4$. Therefore only a negligible fraction of the 4-tuples that are not of the 4 types mentioned is of a type with two F_as. This is because, for each pair of positions, we already have a fraction of $1/8+1/8 = 1/4$ with F_a in both positions. This implies that the remaining fraction of $1/2$ of the 4-tuples must be of one of the types

$$(T_a, T_a, T_a, F_a), \quad (T_a, T_a, F_a, T_a), \quad (T_a, F_a, T_a, T_a), \quad (F_a, T_a, T_a, T_a), \quad (T_a, T_a, T_a, T_a).$$

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

In all 6 projection *graphs* we have an asymptotic fraction of $1/2$ of the edges between T_a and F_a. From the types above we already have a fraction of $1/4$ and we need another $1/4$. Thus we need an asymptotic fraction of $1/4$ of the 4-tuples of one of the types

$$(T_a, T_a, T_a, F_a) \quad \text{or} \quad (T_a, T_a, F_a, T_a).$$

The same applies to the first two positions. As the 4-tuples corresponding to these types are disjoint, the type (T_a, T_a, T_a, T_a) can asymptotically not occur. Since, in each projection graph, we get a fraction of $1/4$ of the edges inside T_a, we can conclude that each type with 3 T_as occurs with a fraction of $1/8$. □

Now let us come to the l-Out-of-k-Sat problem. For 2-Out-of-5-Sat we get the following result.

Corollary 4.3. *For F from* $\text{Form}_{n,5,p}$ *where* $p = n^{\varepsilon}/n^{7/2}$, *we can efficiently certify that F is unsatisfiable in the* 2-*Out-of*-5-*Sat sense. Here* $1/2 > \varepsilon > 0$.

Proof. Our certification algorithm works as follows. Let F be a given random 5-Sat instance. Let E be the subset of clauses of F such that the literal in the last position is a positive literal, x. Let D be the random 4-Sat instance obtained from E by simply deleting the variable in the last position. D is a random 4-Sat instance from $\text{Form}_{n,4,p'}$ with $p' = np \cdot (1 + o(1))$. Thus the preceding theorem is applicable and we apply the algorithm to D. Next we look at the 5-uniform hypergraph H of the all-positive clauses of F. We certify low discrepancy of the projection hypergraph on the last 3 positions of H with the algorithm from the end of Section 3. Our probability assumptions imply that certification is successful.

Now, let F be such that it has been successfully certified by the algorithm. We show that F is unsatisfiable in the 2-Out-of-5-Sat sense. Assume that a is an assignment which satisfies F in the 2-Out-of-5-Sat sense. Then a satisfies D in particular. From our certification we know that, in the 4-uniform hypergraph from the all-positive clauses of D, we have asymptotically $1/8$ of the 4-tuples of each of the types

$$(F_a, F_a, F_a, T_a), \quad (F_a, F_a, T_a, F_a), \quad (F_a, T_a, F_a, F_a), \quad (T_a, F_a, F_a, F_a).$$

Thus, in the 5-uniform hypergraph H we have $1/8$ of each of the types

$$(F_a, F_a, F_a, T_a, T_a), \quad (F_a, F_a, T_a, F_a, T_a), \quad (F_a, T_a, F_a, F_a, T_a), \quad (T_a, F_a, F_a, F_a, T_a),$$

because otherwise a would not satisfy F in the 2-Out-of-5-Sat sense. From the certified low discrepancy of the projection hypergraph on the last 3 positions of H, we get that we can only have a fraction of $1/8$ of the 3-tuples of type (F_a, F_a, T_a). As above, we have derived a fraction of $1/4$ and hence we have a contradiction. □

For $l = 2$ and $k = 4$ the 3-Sat result [11] implies efficient certification of unsatisfiability for $n^{3/2+\varepsilon}$ many 4-clauses in the 2-Out-of-4-Sat sense. But, looking at the 6 projection graphs, one can even get down to $n^{1+\varepsilon}$ random 4-clauses. The interested reader can fill in the required details easily. Given that we can efficiently certify unsatisfiability

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

of random k-Sat instances with $n^{(k/2)+\varepsilon}$ clauses, the following natural generalization of current knowledge comes to mind. In order to certify unsatisfiability in the l-Out-of-k-Sat sense, $n^{(k/2)+\varepsilon-l}$ random clauses are enough. We leave this as a likely conjecture.

5. Certifying Not-All-Equal-3-Sat unsatisfiability

A 3-Sat instance is satisfied by the assignment a in the Not-All-Equal-3-Sat sense if and only if each clause contains at least 1 literal that is true under a and at least 1 literal that is false. In the Not-All-Equal context we modify our set of clauses, $\text{Clause}_{n,k}$, in that we do not allow double occurrences of literals inside clauses, or x and $\neg x$ both in one clause. The Not-all-Equal-3-Sat problem is well known to be $\mathcal{NP}$-complete. The probability space $\text{Form}_{n,3,m}$ consists of all formulas with exactly m different clauses. Each formula is equally likely. It is well known that, for $\text{Form}_{n,3,p}$ and $\text{Form}_{n,3,m}$, most properties holding in one of these spaces with high probability hold in the other one too, provided p is such that the expected number of clauses is m. We are in line with common usage if we assume, without checking the actual details, that this applies to the properties of interest to us.

We are interested in random inputs to the Not-All-Equal-3-Sat problem. The problem for random instances according to the space $\text{Form}_{n,3,m}$ is investigated in [2]. The authors show that there exist values $c = c(n)$ such that, for any $\varepsilon > 0$, formulas with at most $(1-\varepsilon)\cdot c \cdot n$ clauses are satisfiable, whereas formulas with at least $(1+\varepsilon)\cdot c \cdot n$ clauses are unsatisfiable with high probability. They show more precisely that $1.514 \leqslant c \leqslant 2.215$.

Our result is the efficient certification of unsatisfiability in the Not-All-Equal sense for a linear number of random 3-clauses. It is based on the following probabilistic observation.

Lemma 5.1. *Let $\varepsilon > 0$. Let F be a random formula from* $\text{Form}_{n,3,p}$*, where $p = c/n^2$ and c is a sufficiently large constant. With high probability we have that, for all assignments a, a fraction of at most $3/4 + \varepsilon$ of the all-positive clauses is satisfied by a in the Not-All-Equal sense.*

Proof. Let a be a fixed assignment and assume that a assigns true to αn variables and false to $(1-\alpha)n$ variables. The number of all-positive clauses of a random formula consisting only of literals set to true follows the binomial distribution with parameters $(\alpha \cdot n(1 + o(1)))^3$ and p. The number of all-positive clauses consisting of literals set to false follows the binomial distribution with parameters $((1-\alpha)n(1+o(1)))^3$ and p. The expectation of the number of clauses consisting only of literals set to true or to false is $(\alpha^3 + (1-\alpha)^3)\cdot c \cdot n$. This expectation is at least $(1/4)\cdot c \cdot n$. As in the proof of Lemma 2.1(b), we use equation (2.5) (the Chernoff bound) to show that the number of clauses of interest is concentrated at the expectation. Picking c sufficiently large but constant, we can show that this is even the case for all 2^n assignments. □

We need the following fact from [18].

Fact 5.2. *There exists a polynomial time deterministic algorithm which, for* 3*-Sat instance F, finds a truth value assignment that satisfies at least* $0.878 \cdot \text{Opt}(F)$ *clauses of F*

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

in the Not-All-Equal sense. Here Opt(F) *is the maximum over all truth assignments of the number of clauses of F which can be satisfied in the Not-All-Equal sense.*

Based on this fact we obtain the following algorithm.

Algorithm 5.3. *The input to the algorithm is an instance F of the Not-All-Equal-3-Sat problem. Let P be the set of all-positive clauses of F. Run the algorithm from Fact 5.2 on the set of clauses P. Let S be the set of clauses satisfied in the Not-All-Equal sense by the assignment found by the algorithm. If $|S| < 0.877 \cdot |P|$ then the algorithm answers that F is unsatisfiable in the Not-All-Equal sense. Otherwise the algorithm fails, in that it gives an inconclusive answer.*

The algorithm is correct in that it never finds that a formula is unsatisfiable which is satisfiable in the Not-All-Equal sense. This is because, from Fact 5.2, we know for Opt(P) the maximum number of all clauses from P satisfiable in the Not-All-Equal sense, that $\text{Opt}(P) \leqslant 1/0.878 \cdot |S|$. As $0.877/0.878 < 1$, the correctness of the algorithm follows. Completeness in the sense that, for random F from $\text{Form}_{n,3,p}$ with $p = c/n^2$ and sufficiently large c, the algorithm determines unsatisfiability with high probability, follows from Lemma 5.1. From this lemma, we know with high probability that, for all assignments, the fraction of all-positive clauses satisfied in the Not-All-Equal sense is at most $3/4 + \varepsilon$ with $\varepsilon > 0$ a small constant, provided we pick c sufficiently large. Therefore the algorithm must answer unsatisfiable with high probability.

Now, we move to the Half-k-Sat problem, as introduced in [3]. In this problem we consider clauses with an even number of literals k which, as in the Not-All-Equal case, have no double occurrences of literals, or x and $\neg x$ inside a clause. A formula is satisfied in the Half-k-Sat sense if, in each clause, exactly half of the literals are set to true and the other half are set to false. The problem is also $\mathcal{NP}$-complete: see [3]. In [3] it is shown that unsatisfiability can be efficiently certified for clause size $k = 4$ and $p = \log_2 n/n^3$. Using our Not-All-Equal-3-Sat algorithm above, it is trivial to get an algorithm certifying unsatisfiability in the Half-4-Sat sense of random formulas from $\text{Form}_{n,4,p}$ where $p = c/n^3$. Deleting the last literal from each clause gives a random 3-Sat instance from $\text{Form}_{n,3,p'}$ with $p' = c'/n^2$. For sufficiently large c', we can certify that this instance is unsatisfiable in the Not-All-Equal-3-Sat sense, and the result follows.

6. Certifying non-3-colourability

Now we concentrate on a graph-theoretic $\mathcal{NP}$-complete problem, k-colourability. A graph $G = (V, E)$ is called k-colourable if there exists a colouring $f : V \to \{1, \ldots, k\}$ such that no edge of G is monochromatic under f. The decision version of the problem is to decide whether a particular graph is k-colourable for a natural number k. Karp proved in 1972 [19] that it is $\mathcal{NP}$-complete for any fixed $k \geqslant 3$. As for satisfiability problems, there exist threshold values $c_k = c_k(n)$ such that random graphs from $G_{n,p}$ become not k-colourable with high probability when the edge probability p passes c_k/n.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

Krivelevich [20] proposed an algorithm for deciding k-colourability whose expected running time is polynomial in the probability space $G_{n,p}$ if $p \geqslant c/n$, where $c = c(k)$ is a sufficiently large constant. Like the satisfiability problems considered here, p is chosen such that the graph is not k-colourable, with high probability. Therefore the algorithm efficiently certifies *non-k-colourability* in our sense. Here, we propose a direct application of a known approximation algorithm which gives a polynomial time algorithm that certifies non-3-colourability with high probability for a random G from $G_{n,p}$, if $p = c/n$ and c is sufficiently large.

Let a k-partition of the graph G be a partition of the set of vertices of G into k subsets. Max-k-Cut is the problem: find a k-partition with a maximal number of crossing edges, *i.e.*, edges whose ends belong to different subsets of the partition. We need the following result from [14, 21].

Fact 6.1. *There exists a polynomial time deterministic algorithm which, for each graph $G = (V, E)$, finds a k-partition of G such that the number of crossing edges for this partition is at least $d_k \cdot \mathrm{Opt}(G)$ for $d_k > 1 - \frac{1}{k}$ and d_k tending to $1 - \frac{1}{k} + 2\frac{\ln k}{k^2}$ when k tends to infinity. In particular, we have that $d_3 \geqslant 0.8002$. Here $\mathrm{Opt}(G)$ is the maximum number of crossing edges over all k-partitions.*

Our result is based on the following observation.

Lemma 6.2. *Let $\varepsilon > 0$. Let G be a random graph chosen according to $G_{n,p}$ where $p = c/n$ and c is sufficiently large. With high probability we have, for all 3-partitions of G, that the number of crossing edges is at most $(2/3) \cdot |E| \cdot (1 + \varepsilon)$.*

Proof. The number of edges in G is a random variable following the binomial distribution with parameters $n(n-1)/2 = n^2/2 \cdot (1 + o(1))$ and p. Using equation (2.5) (the Chernoff bound), we see that the number of all edges of G is $cn/2 \cdot (1 + o(1))$ with high probability.

For all 3-partitions of G, the maximal number of pairs of vertices i, j belonging to different blocks of the 3-partition is obtained when the sizes of the blocks are equal. Then the number of such pairs is at most

$$\binom{n}{2} - 3\binom{\lfloor n/3 \rfloor}{2} = \frac{1}{3} \cdot n^2 \cdot (1 + o(1)).$$

Thus, for any fixed 3-partition, the number of crossing edges is a random variable following the binomial distribution with parameters $m \leqslant (1/3) \cdot n^2 \cdot (1 + o(1))$ and $p = c/n$. Therefore equation (2.5) (the Chernoff bound) shows that with high probability the number of crossing edges is at most

$$\frac{1}{3} \cdot n^2 \cdot c/n \cdot (1 + \varepsilon') = \frac{2}{3} \cdot \frac{c \cdot n}{2} \cdot (1 + \varepsilon')$$

for any $\varepsilon' > 0$ and sufficiently large c. Moreover, for even larger c, we have from equation (2.5) that, with high probability, for *all* 3-partitions of G, the number of crossing edges is

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

at most

$$\frac{2}{3} \cdot \frac{c \cdot n}{2} \cdot (1 + \varepsilon') < \frac{2}{3} \cdot |E| \cdot (1 + \varepsilon). \qquad \square$$

Based on this lemma, we consider the following algorithm.

Algorithm 6.3. *The input to the algorithm is a graph $G = (V, E)$ chosen according to $G_{n,p}$. Run the algorithm from Fact 6.1 on G. Let q be the number of crossing edges in the partition found by the algorithm. If $q < 0.8 \cdot |E|$ then the algorithm answers that G is not 3-colourable. Otherwise the algorithm fails, i.e., it gives an inconclusive answer.*

Note that the 3-colourability of the graph $G = (V, E)$ induces a 3-partition of G such that all edges in E are crossing edges. Thus, if the approximate solution for the Max-3-Cut problem obtained by the algorithm from [14] contains at most $0.8 \cdot |E|$ crossing edges, then each 3-partition of G contains fewer than $|E|$ edges and G is not 3-colourable. Therefore the algorithm is correct. Completeness in the sense that, for a random graph G from $G_{n,p}$ with $p = c/n$ and sufficiently large c, the algorithm certifies non-3-colourability with high probability, follows from Lemma 6.2.

7. Conclusions

For the Not-All-Equal-3-Sat problem, we have shown how the direct application of an approximation algorithm yields an algorithm certifying the unsatisfiability of a random instance. This application is possible because the approximation ratio of the algorithm for the naturally associated maximization problem is larger than the fraction of the number of clauses a maximal solution has with high probability in a random instance. It may be worthwhile to look for other problems on random instances where approximation algorithms can be applied in this sense. However, we often have bounds on the approximation ratio, making this impossible. For example, for random instances of 3-Sat with high probability, each assignment satisfies a fraction of $7/8$ of all clauses. However, the best-known approximation algorithm of the corresponding maximization problem has an approximation ratio $\leqslant 7/8$. This implies that we know nothing concerning unsatisfiability if we apply such an algorithm to a random formula. The algorithm can only find that a fraction of $7/8$ of all clauses can be satisfied, and it might still be that the formula is satisfiable. In [17] bounds on approximation ratios are derived which show that the 3-Sat situation seems to be rather typical, and that the Not-All-Equal-3-Sat situation is an exception. Already for Not-All-Equal-k-Sat where $k \geqslant 4$, bounds from [17] imply that our method no longer works.

Recently, Feige [9] reversed the direction of what we do for Not-All-Equal-3-Sat. Assuming that the certification of unsatisfiability is hard for random 3-Sat instances with a sufficiently large linear number of clauses, interesting bounds on approximation ratios for several problems are obtained. The investigation uses the observation stated in Lemma 4.1 as a starting point. It is an obvious question whether one can obtain bounds

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

on approximation ratios from intractability assumptions about random k-Sat instances with larger k. Theorem 4.2 might be a starting point here. Lemma 4.1 and Theorem 4.2 can be seen as an efficient reduction of unsatisfiability to solving equations over the 2-element field (up to an asymptotically but not algorithmically negligible remainder). In [4], approximation algorithms for the maximum number of satisfiable equations over a finite field are considered. Can we apply these algorithms in the present context? Or, conversely, can we derive new bounds on the approximation ratio from intractability assumptions for random k-Sat instances? Note that intractability assumptions for random instances are much stronger than complexity-theoretic assumptions, which usually refer only to the worst case.

As the Laplacian matrix allows easy certification of low discrepancy properties of random graphs, it seems worthwhile to think about calculating the $\bar{\lambda}$ of random graphs with a linear number of edges. The case of edge probability $p = n^{\varepsilon}/n$ considered here is relatively simple because of almost-regularity. For sparser random graphs, almost-regularity does not hold. To prove Lemma 4.1 for a linear number of clauses, Feige resorts to the approximation algorithm for the maximum cut by Goemans and Williamson. Using a so far unproven conjecture, such as $\bar{\lambda}$ being small for random graphs with a linear number of edges, should give a different (simpler?) proof of this result.

Finally we ask if the theory of quasi-random graphs, and in particular quasi-random hypergraphs, which is also concerned with the efficient certification of low discrepancy properties, can contribute to the context of the present paper.

References

[1] Alon, N. and Spencer, J. (1992) *The Probabilistic Method*, Wiley.
[2] Achlioptas, D., Chtcherba, A., Istrate, G. and Moore, C. (2001) The phase transition in NAESAT and 1-in-k SAT. In *Proc. SODA 2001*, SIAM, pp. 721–722.
[3] Ben-Sasson, E. and Bilu, Y. (2002) A gap in average proof complexity. *Electron. Colloquium Comput. Complexity* (ECCC) 003.
[4] Berman, P. and Karpinski, M. (2001) Approximating minimum unsatisfiability of linear equations. *Electron. Colloquium Comput. Complexity* (ECCC) 025.
[5] Blum, M., Karp, R. M., Vornberger, O., Papadimitriou, C. H. and Yannakakis, M. (1981) The complexity of testing whether a graph is a superconcentrator. *Inform. Process. Lett.* **13** 164–167.
[6] Bolla, M. (1993) Spectra, Euclidean representations and clusterings of hypergraphs. *Discrete Math.* **117** 19–39.
[7] Chung, F. R. K. (1993) The Laplacian of a hypergraph. In *Expanding Graphs*, Vol. 10 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, AMS, pp. 21–36.
[8] Chung, F. R. K. (1997) Spectral graph theory. In *Conference Board of the Mathematical Sciences*, Vol. 92 of *Regional Conference Series in Mathematics*, AMS.
[9] Feige, U. (2002) Relations between average case complexity and approximation complexity. In *Proc. STOC 2002*, ACM, pp. 534–543.
[10] Friegut, E. (1999) Necessary and sufficient conditions for sharp thresholds of graph properties and the k-Sat problem. *J. Amer. Math. Soc.* **12** 1017–1054.
[11] Friedman, J. and Goerdt, A. (2001) Recognizing more unsatisfiable random 3-Sat instances efficiently. In *Proc. International Colloquium on Automata, Languages and Programming, 2001*, Vol. 2076 of *Lecture Notes in Computer Science*, pp. 310–321.
[12] Friedman, J. and Wigderson, A. (1995) On the second eigenvalue of hypergraphs. *Combinatorica* **15** 43–65.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press

[13] Friedman, J., Goerdt, A. and Krivelevich, M. Efficient recognition of random unsatisfiable k-Sat instances. Submitted.
[14] Frieze, A. M. and Jerrum, M. (1997) Improved approximation algorithms for MAX k-cut and MAX BISECTION. *Algorithmica* **18** 67–81.
[15] Goerdt, A. and Krivelevich, M. (2001) Efficient recognition of random unsatisfiable k-Sat instances by spectral methods. In *Proc. Symposium on Theoretical Aspects of Computer Science, 2001*, Vol. 2010 of *Lecture Notes in Computer Science*, pp. 294–304.
[16] Goerdt, A. and Jurdzinski, T. (2002) Some results on random unsatisfiable k-Sat instances and approximation algorithms applied to random structures. In *Proc. MFCS 2002*, Vol. 2420 of *Lecture Notes in Computer Science*, pp. 280–291.
[17] Håstad, J. (2001) Some optimal inapproximability results. *J. Assoc. Comput. Mach.* **48** 798–859.
[18] Kann, V., Lagergren, J. and Panconesi, A. (1996) Approximability of maximum splitting of k-Sets and some other APX-complete problems. *Inform. Process. Lett.* **58** 105–110.
[19] Karp, R. (1972) Reducibility among combinatorial problems. In *Complexity of Computer Computations* (E. Miller and J. W. Thatcher, eds), Plenum Press, New York, pp. 85–103.
[20] Krivelevich, M. (2002) Deciding k-colorability in expected polynomial time. *Inform. Process. Lett.* **81** 1–6.
[21] Mahajan, S. and Ramesh, H. (1999) Derandomizing Approximation Algorithms Based on Semidefinite Programming. *SIAM J. Comput.* **28** 1641–1663.
[22] Strang, G. (1988) *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, San Diego.

https://doi.org/10.1017/S0963548303005637 Published online by Cambridge University Press