Certifying Unsatisfiability of Random $2k$-SAT Formulas using Approximation Techniques

Amin Coja-Oghlan[2], Andreas Goerdt[1], André Lanka[1], and Frank Schädlich[1]

[1] Technische Universität Chemnitz, Fakultät für Informatik
Straße der Nationen 62, 09107 Chemnitz, Germany
e-mail: {goerdt, lanka, frs}@informatik.tu-chemnitz.de
[2] Humboldt-Universität zu Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin, Germany
e-mail: coja@informatik.hu-berlin.de

Abstract. It is known that random k-SAT formulas with at least $(2^k \cdot \ln 2) \cdot n$ random clauses are unsatisfiable with high probability. This result is simply obtained by bounding the expected number of satisfying assignments of a random k-SAT instance by an expression tending to 0 when n, the number of variables tends to infinity. This argument does not give us an efficient algorithm certifying the unsatisfiability of a given random instance. For even k it is known that random k-SAT instances with at least $\mathrm{Poly}(\log n) \cdot n^{k/2}$ clauses can be efficiently certified as unsatisfiable. For $k = 3$ we need at least $n^{(3/2)+\varepsilon}$ random clauses.
In case of even k we improve the aforementioned results in two ways. There exists a constant C such that 4-SAT instances with at least $C \cdot n^2$ clauses can be efficiently certified as unsatisfiable. Moreover, we give a satisfiability algorithm which runs in expected polynomial time over all k-SAT instances with $C \cdot n^{k/2}$ clauses. Our proofs are based on the direct application of known approximation algorithms on the one hand, and on a recent estimate of the ϑ-function for random graphs with a linear number of edges, on the other hand.

1 Introduction

The *k-SAT problem* is to find out whether a given k-SAT formula over n propositional variables is satisfiable or not. Since it is well-known that the k-SAT problem for $k \geq 3$ is $\mathcal{NP}$-complete and $\mathcal{NP}$-completeness is a worst case notion, it is natural to ask for algorithms that can handle *random* formulas efficiently. Given a set of n propositional variables and a function $c = c(n)$, a random k-SAT instance is obtained by picking c k-clauses over the set of n variables uniformly

at random and independently of each other. Part of the recent interest in random k-Sat instances is due to the interesting threshold behavior, in that there exist values $c_k = c_k(n)$ such that random k-Sat instances with at most $(1 - \varepsilon) \cdot c_k \cdot n$ random clauses are satisfiable with high probability, whereas for at least $(1 + \varepsilon) \cdot c_k \cdot n$ random clauses we have unsatisfiability with high probability. (Here, "with high probability" means "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 on k only. However, it is not known whether the threshold really is a constant independent of n, cf. [Fr 99]. The consistent experimental observation that the satisfiability problem for those instances close to the threshold seems to be algorithmically much harder than for those further away makes this model particularly interesting. In this paper, we are concerned with values of $c(n)$ well above the threshold, hence our main problem is to *efficiently* find a proof that a random formula is unsatisfiable.

There are two different types of algorithms for deciding whether a random k-SAT formula is satisfiable or not. First, there are algorithms that on *any* input formula have a polynomial running time, and that with high probability with respect to the input give the correct answer, "satisfiable" or "unsatisfiable". However, with probability $o(1)$, the algorithm may give an inconclusive answer. We shall refer to algorithms of this type as *efficient certification algorithms*. Secondly, there are algorithms that *always* answer correctly either "satisfiable" or "unsatisfiable", and applied to a random formula have a polynomial *expected* running time [DyFr 89].

Let us emphasize that although an efficient certification algorithm may give an inconclusive answer in some (rare) cases, such an algorithm is still *complete* in the following sense. Consider random k-SAT instances where the number of clauses is above the satisfiability threshold c_k mentioned above. Completeness of a certification algorithm in this context means that when picking such formulas at random the algorithm gives the right answer, "unsatisfiable" in the present case, with high probability, too. Note that no efficient algorithm can answer "unsatisfiable" on all unsatisfiable inputs from each space; completeness only refers to a subset whose probability tends to 1. Completeness of efficient certification algorithms is always

Then

$$M(F) = |\{i \in \{1,\dots,m\}|\; h(C_i) = h(C_j) \text{ for some } i \neq j\}| < \varepsilon 2^{-k} m.$$

Since $|E_F| \geq P(F) - L(F) - M(F)$, the lemma is an immediate consequence of 1–3 above.

In order to prove that 1. holds with sufficiently high probability, note that the number of all-positive clauses is binomially distributed with expectation $2^{-k}m$. By Chernoff bounds, $\mathrm{Prob}[P(F) < (1-\varepsilon)2^{-k}m] \leq \exp(-\varepsilon^2 2^{-k-1}m)$. Similarly, since the number of clauses as in 2. is binomially distributed with expectation $m(2n)^{-k/2} \leq \varepsilon 2^{-k-1}m$, we conclude that

$$\mathrm{Prob}[L(F) > \mathrm{E}[L(F)] + \varepsilon 2^{-k-1}m] \leq \exp(-\varepsilon^2 2^{-k-2}m).$$

The probability that 3. is violated is most easily bounded using Talagrand's inequality. Indeed, we may consider $\mathrm{Form}_{n,k,m}$ as a product space $\Lambda_1 \times \cdots \times \Lambda_m$, where Λ_i is a random clause, $i = 1,\dots,m$. First, let us estimate the expectation of $M(F)$. Let $F = C_1 \wedge \cdots \wedge C_m$. If $i \neq j$ are fixed, then the probability that $h(C_i) = h(C_j)$ is $2n^{-k}$. Hence, by our assumption $m \leq n^{k-1}$, $\mathrm{E}[M(F)] \leq 2m^2 n^{-k} \leq 2m/n < \varepsilon 2^{-k-2}m$. In order to apply Talagrand's inequality, we observe that if $F_1, F_2 \in \mathrm{Form}_{n,k,m}$ differ only in the jth clause, then $|M(F_1) - M(F_2)| \leq 2$. Furthermore, let $r > 0$. Suppose that $F = C_1 \wedge \cdots \wedge C_m \in \mathrm{Form}_{n,k,m}$ satisfies $M(F) \geq r$, and let

$$J = \{i \in \{1,\dots,m\}|\; \text{there exists } j \neq i \text{ such that } h(C_i) = h(C_j)\}.$$

Then there exists $J_0 \subset J$, $|J_0| \leq r+1$, such that the following condition holds: For any $F' = D_1 \wedge \cdots \wedge D_m \in \mathrm{Form}_{n,k,m}$, such that $D_j = C_j$ for all $j \in J_0$, we have $M(F') \geq r$. Put $\psi(r) = 4(r+1)$. Then Talagrand's inequality (in the version [Ja et al. 2000, p. 40]) yields

$$\mathrm{Prob}[M(F) > 2\mathrm{E}[M(F)] + t] \leq 2\exp\left(-\frac{t^2}{4\psi(2\mathrm{E}[M(F)] + t)}\right).$$

Letting $t = \varepsilon 2^{-k-1}m$ entails $\mathrm{Prob}[M(F) > \varepsilon 2^{-k}m] \leq \exp(-\varepsilon 2^{-k}m/100)$.

understood as described, also for properties different from unsatisfiability.

Any certification algorithm can be turned into a satisfiability algorithm that answers correctly on any input, simply by invoking an enumeration procedure in case that the efficient certification procedure gives an inconclusive answer. But even the completeness of the certification algorithm does not ensure that the corresponding satisfiability algorithm runs in expected polynomial time. The probability of an inconclusive answer may be too large to even out the time needed for the enumeration procedure, even though it is $o(1)$. On the other hand, any satisfiability algorithm with expected polynomial time can be turned into an efficient certification algorithm, because the probability that the running time of the algorithm will be superpolynomial is $o(1)$. Hence, having a polynomial expected running time is a slightly stronger requirement.

From [FrGo 2001] and [GoKr 2001] it is essentially known that for random k-SAT instances with $\mathrm{Poly}(\log n) \cdot n^{k/2}$ clauses we can efficiently certify unsatisfiability, in case of even k. For odd k we need $n^{(k/2)+\varepsilon}$ random clauses. Recall that probabilistically we know much more: Random k-SAT instances with at least $(2^k \cdot \ln 2) \cdot n$ clauses are unsatisfiable with high probability. Hence, it is an obvious problem to design algorithms that can certify unsatisfiability of random formulas efficiently for smaller numbers of clauses than given in [FrGo 2001,GoKr 2001]. To make further progress on this question, new techniques seem to be necessary. Therefore, in this paper, we examine what various algorithmic techniques contribute to the random k-SAT problem. We achieve some improvements for the case of even k.

Based on the direct application of known approximation algorithms in the sense advocated by [Fe 2002] or also [GoJu 2002], we obtain an efficient certification algorithm for the case of at least $C \cdot n^2$ 4-clauses, thereby gaining a polylogarithmic factor. We present two different certification algorithms. One algorithm applies the Max-Cut approximation algorithm of Goemans and Williamson [GoWi 95]. The other one employs the Min-Bisection approximation algorithm of Feige and Krauthgamer [FeKr 2000]. Since the Max-Cut approximation algorithm is based on semidefinite programming, our first algorithm is not a purely combinatorial algorithm. In contrast,

the application of the Min-Bisection algorithm yields a combinatorial algorithm. We state our result only for $k = 4$, but we feel that it is only a technical matter to extend it to any even k and $C \cdot n^{k/2}$ clauses.

Moreover, we obtain the first algorithm for deciding satisfiability of random $2k$-SAT formulas in expected polynomial time. The algorithm can handle even *semirandom* formulas, cf. Sec. 4 for details. Our algorithm is based on computing the Lovasz number $\vartheta(G)$ of a graph G, which provides an efficiently computable upper bound on the independence number of G [Gr et al. 88]. Only recently an estimate on the probable value of the ϑ-function of random graphs with a linear number of edges has been obtained [Co 2003]. Based on this estimate we give an (un-)satisfiability algorithm running in expected polynomial time on the space of random k-SAT instances with $C \cdot n^{k/2}$ clauses. Note that the computation of the ϑ-function is based on semidefinite programming again, hence, our algorithm for deciding satisfiability in expected polynomial time is not purely combinatorial.

In Section 2 we give our certification algorithms and in Section 3 we state the theorem crucial for their correctness. In Section 4 we deal with the expected polynomial time algorithm. This section can be read independently from the rest.

Related work. There are two more recent papers motivated by the random k-SAT problem, [Fe 2002] and [BeBi 2002]. Instead of proposing new algorithms for certifying unsatisfiability, Feige [Fe 2002] shows that non-approximability results can be based on a certain assumption on the average-case complexity of random k-SAT. Hence, Feiges results emphasize the intimate relationship between approximation techniques and random k-SAT.

2 Efficient certification of unsatisfiability

Given a set of n propositional variables, $\mathrm{Var} = \mathrm{Var}_n = \{v_1, \ldots, v_n\}$, a literal over Var is a variable v_i or a negated variable $\neg v_i$. A k-clause is an ordered k-tuple $l_1 \vee l_2 \vee \cdots \vee l_k$ of literals such that the variables underlying the literals are distinct. A k-SAT instance is a

[DyFr 89] Martin Dyer, Allan Frieze. *The solution of some NP-hard problems in polynomial expected time.* J. Algorithms **10**, 1989, 451–489

[Fr 99] Ehud Friegut. *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.

[Fe 2002] Uriel Feige. *Relations between average case complexity and approximation complexity.* In Proceedings STOC 2002

[FeKi 2001] Uriel Feige, J. Kilian. *Heuristics for semirandom graph problems.* J. Comput. and System Sci. **63**, 2001, 639–671.

[FeKr 2000] Uriel Feige, Robert Krauthgamer. *A polylogarithmic approximation of the minimum bisection.* In Proceedings FoCS 2000, 105–115

[FrGo 2001] Joel Friedman, Andreas Goerdt. *Recognizing more Unsatisfiable Random 3-Sat Instances efficiently.* In Proceedings ICALP 2001, LNCS 2076, 310–321.

[GoJu 2002] Andreas Goerdt, Tomasz Jurdzinski. *Some Results on Random Unsatisfiable k-Sat Instances and Approximation Algorithms Applied to Random Structures.* In Proc. MFCS 2002, LNCS 2420, 280–291.

[GoKr 2001] Andreas Goerdt, Michael Krivelevich. *Efficient recognition of random unsatisfiable k-SAT instances by spectral methods.* Proc. 18th STACS, 2001, LNCS **2010**, 294–304.

[GoWi 95] M. X. Goemans, D. P. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.* J. ACM 42, 1115–1145.

[Gr et al. 88] Martin Grötschel, M., Laszlo Lovász, A. Schrijver. *Geometric algorithms and combinatorial optimization.* Springer 1988

[Ho 87] Micha Hofri. *Probabilistic Analysis of Algorithms.* Springer 1987.

[Ja et al. 2000] Svante Janson, Tomasz Łuczak, Andrei Ruciński. *Random Graphs.* Wiley 2000

[Kn 94] Donald Knuth. *The sandwich theorem, Electron.* J. Combin. **1**, 1994.

Appendix

Proof of Lemma 5 The first part of the lemma is proven in [GoKr 2001]. We shall prove that with probability $\geq 1 - \exp(-\Omega(m))$ the graph G_F enjoys the following three properties.

1. The number $P(F)$ of all-positive clauses in F is at least $(1 - \varepsilon)2^{-k}m$.
2. The number $L(F)$ of all-positive clauses of type

$$x_1 \vee \cdots \vee x_{k/2} \vee x_1 \vee \cdots \vee x_{k/2}$$

   that occur in F is $\leq \varepsilon 2^{-k}m$.
3. Let $F = C_1 \wedge \cdots \wedge C_m$. Given literals $l_1, \ldots, l_k$, put

$$h(l_1 \vee \cdots \vee l_k) = \{l_1 \vee \cdots \vee l_{k/2}, l_{k/2+1} \vee \cdots \vee l_k\}.$$

1. Compute G_F and G'_F. Let $p = m(2n)^{-k}$. If both `DecideMIS`(G_F, p) and `DecideMIS`(G'_F, p) answer "typical", then terminate with output "unsatisfiable".
2. Enumerate all 2^n assignments and look for a satisfiable one. If none is found output "unsatisfiable", otherwise "satisfiable" and the assignment.

Proof of Thm. 12. We may assume that $m = \lceil C 2^k n^{k/2} \rceil$. The correctness of `DecideSAT` follows from Lemma 4. As for the running time, let F_0 denote the random k-SAT formula from which $F \in \mathrm{Form}^+_{n,k,m}$ has been constructed. Clearly, the graphs G_F and G'_F can be computed efficiently. By the second part of Lemma 5, we may assume that $\min\{|E(G_{F_0})|, |E(G'_{F_0})|\} \geq 2^{-k-1}m$. By the first part of Lemma 5, the graphs G_F and G'_F both can be made up as follows. First, a random graph $G_{\nu,\mu}$ is chosen where $\nu = n^{k/2}$, $\mu = 2^{-k-1}m$. Then, an adversary adds some edges. The graphs make up a subset of $G^+_{n,p}$ whose probability is bounded from below by an inverse polynomial fraction. This is the case because our choice of p implies that $\mathrm{Prob}[|E(G_{\nu,p})| \leq \mu]$ is bounded from below by some inverse polynomial and the semirandom model $G^+_{\nu,p}$ allows us to add the required edges. Therefore, the assertion follows from Prop. 1. $\square$

References

[Au et al. 99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. *Complexity and Approximation – Combinatorial Optimization Problems and their Approximability Properties.* Springer 1999.

[AlSp 92] Noga Alon, Joel Spencer. *The Probabilistic Method.* John Wiley and Sons 1992.

[BeBi 2002] Eli Ben-Sasson, Yonatan Bilu, *A Gap in Average Proof Complexity.* Electronic Colloquium on Computational Complexity (ECCC) 003, 2002.

[Ch 97] Fan R. K. Chung. *S*pectral Graph Theory. American Mathematical Society, 1997.

[Co 2003] Amin Coja-Oghlan. *The Lovasz number of random graphs.* Hamburger Beiträge zur Mathematik 169; available from http://www.informatik.hu-berlin.de/∼coja/

[Co 2003b] Amin Coja-Oghlan, A. Taraz. *Finding large independent sets in polynomial expected time.* In Proceedings STACS 2003, LNCS.

[CoTa 2003] Amin Coja-Oghlan, A. Taraz. *Colouring random graphs in expected polynomial time.* In Proceedings STACS 2003, LNCS.

set of k-clauses. We think of a k-SAT instance as $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where each C_i is a k-clause. Given a truth value assignment a of Var, that is simply a mapping assigning true (=1) or false (=0) to each variable, we can assign true or false to a k-SAT instance as usual. We let T_a be the set of variables x with $a(x) =$ true and F_a the set of variables x with $a(x) =$ false.

The probability space $\mathrm{Form}_{n,k,p}$ is the probability space of k-SAT instances obtained by picking each k-clause with probability p independently. There are slightly different ways to define probability spaces of k-SAT instances. For example with $m \approx p \cdot 2^k \cdot (n)_k$ where $(n)_k = n \cdot (n-1) \cdot (n-2) \cdots (n-k+1)$ we might consider the uniform distribution of all k-SAT instances with m clauses. Note that m is about the expected number of clauses of a random formula from $\mathrm{Form}_{n,k,p}$. One might also define clauses as sets of literals or one might allow tautological clauses ... . In line with common usage we assume that it is only a technical matter to transfer our results to any of these possibilities to define random k-SAT instances, but do not check the details.

A k-uniform hyperedge or simply k-tuple over the vertex set V is a vector $(x_1, x_2, \ldots, x_k)$ where the $x_i \in V$ are *all distinct.* $H = (V, E)$ is a k-uniform hypergraph if E is a set of k-tuples over the vertex set V. In the context of k-uniform hypergraphs we use the notion of *type* in the following sense: Let $X_1, X_2, \ldots, X_k \subseteq V$, a k-tuple $(x_1, x_2, \ldots, x_k)$ is of type $(X_1, X_2, \ldots, X_k)$ if we have for all i that $x_i \in X_i$. A random hypergraph $H \in HG_{n,k,p}$ is obtained by picking each of the possible $(n)_k$ k-tuples with probability p, independently.

Let S be a set of k-clauses over the set of variables Var, as defined above. The hypergraph $H = (V, E)$ associated to S is defined by $V =$ Var and $(x_1, x_2, x_3, \ldots, x_k) \in E$ if and only if there is a k-clause $l_1 \vee l_2 \vee \cdots \vee l_k \in S$ such that for all i $l_i = x_i$ or $l_i = \neg x_i$. In case of even k the graph $G = (V, E)$ associated to S is defined by $V = \{(x_1, \ldots, x_{k/2}) \,|\, x_i \in \mathrm{Var} \text{ and } x_i \neq x_j \text{ for } i \neq j\}$ and $\{(x_1, x_2, \ldots, x_{k/2}), (x_{(k/2)+1}, \ldots, x_k)\} \in E$ if and only if there is a k-clause $l_1 \vee l_2 \vee \cdots \vee l_k \in S$ such that the variable underlying l_i is x_i.

We use the following standard asymptotic terminology: When $f(n) \to 0$, $O(f(n))$ stands for a term $g(n)$ such that $|g(n)| \leq C \cdot$

$f(n)$ for a constant C and all sufficiently large n. The following abbreviations are more specialized: $f(n) \sim_s g(n)$ iff there is an $\varepsilon > 0$ such that $f(n) = g(n) \cdot (1 + O(1/n^{\varepsilon}))$. Here $\sim_s$ stands for strong asymptotic equality. Similarly we use $f(n) = so(g(n))$ iff $f(n) = O(1/n^{\varepsilon}) \cdot g(n)$. We say $f(n)$ is negligible iff $f(n) = so(1)$.

Parity properties analogous to the next theorem have been proved in [Fe 2002] for 3-Sat instances with a linear number of clauses and in [GoJu 2002] for 4-Sat instances. But in the proof of [GoJu 2002] it is important that the probability of each clause is $p \leq 1/n^{2-\varepsilon}$ where $\varepsilon > 0$ is a constant. In this case the number of clauses is with high probability at most $O(n^{2-\varepsilon})$. This implies that the number of occurrences of two given literals in several clauses of a random formula is small. Therefore we can essentially assume that we have no occurrences of two given literals together in several clauses. This is not any more the case for $p = C/n^2$ and some additional complications arise when proving the following theorem.

Theorem 1 (Parity Theorem). *For a random $F \in Form_{n,4,p}$ where $p = C/n^2$ and C is a sufficiently large constant, we can efficiently certify the following properties.*

(a) Let $S \subseteq F$ be the subset of all clauses of F corresponding to one of the 16 possibilities of placing negated and non-negated variables into the four slots of clauses available. Let $G = (V, E)$ be the graph associated to S. Then $|S| = C \cdot n^2 \cdot (1 + so(1))$ and $|E| = C \cdot n^2 \cdot (1 + so(1))$.

(b) For all satisfying assignments a of F we have that $|T_a| \sim_s (1/2) \cdot n$ and $|F_a| \sim_s (1/2) \cdot n$.

(c) Let S be the set of clauses of F consisting only of non-negated variables. Let H be the hypergraph associated to S. For all satisfying assignments a of F the number of 4-tuples of H of each of the 8 types

$$(T_a, T_a, T_a, F_a), (T_a, T_a, F_a, T_a), (T_a, F_a, T_a, T_a), (F_a, T_a, T_a, T_a),$$

$$(F_a, F_a, F_a, T_a), (F_a, F_a, T_a, F_a), (F_a, T_a, F_a, F_a), (T_a, F_a, F_a, F_a)$$

is $(1/8) \cdot C \cdot n^2 \cdot (1 + so(1))$. The same statement applies when S is one of the remaining seven subsets of clauses of F which have a given even number of negated variables in a given subset of the four slots available.

1. If $\vartheta(G) \leq C'(\ln(\nu p))^{1/2}(\nu/p)^{1/2}$, then terminate with output "typical". Here C' denotes some sufficiently large constant (independent of p, ν).
2. Check whether there exists a subset S of V, $|S| = 25\ln(\nu p)/p$, such that $|V \setminus (S \cup N(S))| > 12(\nu/p)^{1/2}$. Here $N(S)$ is the set of neighbours of S, that is vertices adjacent to a vertex form S. If no such set S exists, then output "typical" and terminate.
3. Check whether in G there is an independent set of size $12(\nu/p)^{1/2}$. If this is not the case, then output "typical". Otherwise, output "not typical".

The analysis of `DecideMIS` is based on two results concerning $\vartheta(G_{\nu,p})$ from [Co 2003] and [Co 2003b], respectively.

Theorem 14. *Suppose that $D \leq \nu p \leq 0.99\nu$ for some large constant $D > 0$. There exist constants $c_1, c_2 > 0$ such that with high probability*

$$c_1(\ln(\nu p))^{-1/2}(\nu/p)^{1/2} \leq \vartheta(G_{\nu,p}) \leq c_2(\ln(\nu p))^{1/2}(\nu/p)^{1/2}.$$

Theorem 15. *Suppose that $p \leq 0.99$. Let M be a median of $\vartheta(G_{\nu,p})$. Let $\xi \geq \max\{10, M^{1/2}\}$. Then $Prob[\vartheta(G_{\nu,p}) \geq M + \xi] \leq 30\exp\left(-\frac{\xi^2}{5M+10\xi}\right)$.*

Proposition 1. *For any G, if `DecideMIS`(G,p) outputs "typical", then $\alpha(G) \leq C'(\ln(\nu p))^{1/2}(\nu/p)^{1/2}$. Moreover, with respect to $G^{+}_{\nu,p}$ the probability that `DecideMIS`(G,p) outputs "not typical" is $< \exp(-\nu)$. Applied to $G^{+}_{\nu,p}$, `DecideMIS` has a polynomial expected running time, provided $\nu p \geq C''$, for some large constant $C'' > 0$.*

Sketch of proof. The proof goes along the lines of [CoTa 2003,Co 2003]. In addition to Thm. 14 and 15, to handle the semirandom graph $G^{+}_{\nu,p}$, we make use of the monotonicity of ϑ: If G_1 is a (weak) subgraph of G_2, then $\vartheta(G_1) \geq \vartheta(G_2)$, cf. [Kn 94]. □

Finally, our expected polynomial time algorithm for deciding whether a k-SAT instance over Var_n is satisfiable is as follows.

Algorithm 16 `DecideSAT`(F)
Input: A k-SAT formula F over Var_n.
Output: Either a satisfying assignment of F or "unsatisfiable".

The following lemma is proved in [GoKr 2001] without the exponentially low probability bounds. It shows that the above reduction from certifying unsatisfiability to bounding the independence number from above, maps random formulas to random graphs. Let $G_{\nu,\mu}$ denote a graph with ν vertices and $\mu \leq \binom{\nu}{2}$ edges, chosen uniformly at random.

Lemma 5. *Let $F \in \mathrm{Form}_{n,k,m}$ be a random formula.*

1. *Conditioned on $|E(G_F)| = \mu$, the graph G_F is uniformly distributed; i.e. $G_F = G_{\nu,\mu}$. A similar statement holds for G'_F.*
2. *Let $\varepsilon > 0$. Suppose that $2^k \cdot n^{k/2} \leq m \leq n^{k-1}$. Then with probability at least $1 - \exp(-\Omega(m))$ we have $\min\{|E(G_F)|, |E(G'_F)|\} \geq (1-\varepsilon) \cdot 2^{-k} \cdot m$.*

Thus, our next aim is to estimate efficiently the independence number of a semirandom graph. Let $0 < p = p(\nu) < 1$. The semirandom graph $G^+_{\nu,p}$ is produced in two steps as follows (similarly to $\mathrm{Form}^+_{n,k,m}$).

1. Choose a random graph $G_0 = G_{\nu,p}$ (remember that $G_{\nu,p}$ is obtained by including each of the $\binom{\nu}{2}$ possible edges with probability p independently).
2. An adversary adds to G_0 arbitrary edges, thereby completing the instance $G = G^+_{\nu,p}$.

(We will later set $p = m(2n)^{-k}$.) We employ the *Lovász number* ϑ (cf. [Gr et al. 88,Kn 94]), which can be seen as a semidefinite programming relaxation of the independence number. Indeed, $\vartheta(G) \geq \alpha(G)$ for any graph G. In contrast to the independence number, the Lovász number $\vartheta(G)$ can be computed in polynomial time using the ellipsoid algorithm [Gr et al. 88] (rounding issues can be ignored for our purposes). We shall now adapt the algorithms given in [CoTa 2003,Co 2003] for approximating the independence number of classical random graphs to our purpose: Our algorithm `DecideMIS` will output "typical", if the independence number of the input graph is "small", and "atypical" otherwise.

Algorithm 13 `DecideMIS`(G, p)
Input: A graph G of order ν, and a number p.
Output: Either "typical" or "not typical".

(d) Let H be the hypergraph associated to those clauses of F whose first slot contains a negated variable and whose remaining three slots contain non-negated variables. The number of 4-tuples of H of each of the 8 types

$$(T_a, T_a, T_a, T_a), (T_a, T_a, F_a, F_a), (T_a, F_a, T_a, F_a), (T_a, F_a, F_a, T_a),$$

$$(F_a, F_a, F_a, F_a), (F_a, F_a, T_a, T_a), (F_a, T_a, F_a, T_a), (F_a, T_a, T_a, F_a)$$

is $(1/8) \cdot C \cdot n^2 \cdot (1 + so(1))$. The same statement applies when S is one of the remaining seven subsets of clauses of F which have a given odd number of negated variables in a given subset of the four slots available. □

The technical notion type of a 4-tuple of a hypergraph is defined above. On probabilistic grounds we know that satisfying assignments do generally not exist under the assumptions of the theorem. However, we cannot efficiently certify this. Instead, we can only certify the weaker statements of the theorem. Statement (b) means that we have an $\varepsilon > 0$ such that we can certify that all assignments a with $|T_a| \geq (1/2) \cdot n \cdot (1 + 1/n^{\varepsilon})$ or $|F_a| \geq (1/2) \cdot n \cdot (1 + 1/n^{\varepsilon})$ do not satisfy a random F. In the same way (c) means that we can efficiently certify that all assignments a for which the number of all positive clauses of one of the prescribed types is not between $(1/8) \cdot C \cdot n^2 \cdot (1 - 1/n^{\varepsilon})$ and $(1/8) \cdot C \cdot n^2 \cdot (1 + 1/n^{\varepsilon})$ do not satisfy F. In particular the number of clauses with an even number of literals true under a is asymptotically negligible. We present the proof of the Parity Theorem in the next section. First we state our unsatisfiability certification algorithms.

Given a graph $G = (V, E)$, a cut is a partition of V into two subsets V_1 and V_2. The Max-Cut problem is the problem to maximize the number of crossing edges, that is the number of edges with one endpoint in V_1 and the other endpoint in V_2. In this context $\mathrm{Opt}(G)$ is the maximal number of crossing edges of a cut. There is a polynomial time approximation algorithm which, given G, finds a cut such that the number of crossing edges is guaranteed to be at least $0.87 \cdot \mathrm{Opt}(G)$, see [Au et al. 99], page 399. Note that the algorithm is a deterministic algorithm.

Algorithm 2 *The input is a 4-Sat instance F.*

1. Certify the properties as stated in Theorem 1. If this certification is not successful the algorithm stops with an inconclusive answer.

2. Let S be the subset of all clauses of F containing only non-negated variables. We construct the graph $G = (V, E)$, as defined above, associated to this subset S.

3. Apply the Max-Cut approximation algorithm to G.

4. If the cut found in 3. contains at most $0.86 \cdot |E|$ edges the output is "unsatisfiable", otherwise the algorithm gives an inconclusive answer as it cannot determine whether F is satisfiable or unsatisfiable. □

Theorem 3. *Algorithm 2 efficiently certifies the unsatisfiability for $Form_{n,4,p}$ where $p = C/n^2$ and C is sufficiently large.*

Proof. To show that the algorithm is correct, let F be any satisfiable 4-SAT instance. Let a be a satisfying truth value assignment of F. Let H be the hypergraph associated to S. From Step 1 we know that only an asymptotically negligible fraction of the 4-tuples of H has a type not among the eight types with an odd number of T_a's. Now consider the partition of the vertices of the graph G where $V_1 = \{(x, y) \mid x \neq y \text{ and } x, y \in T_a \text{ or } x, y \in F_a\}$ and $V_2 = \{(x, y) \mid x \in T_a, y \in F_a \text{ or } x \in F_a, y \in T_a\}$. From Step 1 we know that the number of edges between V_1 and V_2 is equal to $|E| \cdot (1 + so(1))$. The Max-Cut approximation algorithm finds a cut which with at least $0.87 \cdot |E|$ crossing edges. Therefore the algorithm does not answer "unsatisfiable" and is correct.

To show the completeness of the algorithm let $F \in \mathrm{Form}_{n,4,p}$ where $p = C/n^2$ be a random formula. Step 1 is successful with high probability as we know from Theorem 1. Let V_1, V_2 be an arbitrary cut of G with $|V_1| = m$ and $l = |V_2| = n(n-1) - m$. The number of crossing edges possible at all is bounded above by $m \cdot l$. Each edge possible is present in G with probability $p' = 1 - (1-p)^2 = 2p \cdot (1 + o(1))$, independently. The probability of the event that the number of crossing edges of the given cut V_1, V_2 is at least b is bounded above by the probability that the binomial distribution with parameters $m \cdot l$ and p' is at least b. Moreover, $m \cdot l$ is maximized when $m = l = n(n-1)/2$. In this case the expectation of the number of

1. A formula $F_0 = C_1 \wedge \cdots \wedge C_m \in \mathrm{Form}_{n,k,m}$ is chosen from the uniform distribution.
2. An adversary picks any formula $F = \mathrm{Form}^+_{n,k,m}$ over the standard set of variables Var_n in which at least one copy of each C_i, $i = 1, \ldots, m$, occurs.

Note that in general we cannot reconstruct F_0 from F. For each $F_0 \in \mathrm{Form}_{n,k,m}$, let $\mathcal{I}(F_0)$ denote the set of all formulas that can be obtained according to 2. above. For any k-SAT instance F, let $|F|$ signify the number of clauses of F. We say that an algorithm $\mathcal{A}$ *has a polynomial expected running time applied to* $\mathrm{Form}^+_{n,k,m}$ if there exists a constant $l > 0$ such that the following condition holds. For any map I that assigns to each $F_0 \in \mathrm{Form}_{n,k,m}$ an instance $I(F_0) \in \mathcal{I}(F_0)$ we have $\sum_{F_0 \in \mathrm{Form}_{n,k,m}} 1/(2n)^{k \cdot m} \cdot R_{\mathcal{A}}(I(F_0)) = O((n + |I(F_0)|)^l)$, where $R_{\mathcal{A}}(F)$ denotes the running time of $\mathcal{A}$ on input F.

Theorem 12. *Let $k \geq 4$ be an even integer. Suppose that $m \geq C \cdot 2^k \cdot n^{k/2}$, for some sufficiently large constant $C > 0$. There exists an algorithm* ***DecideSAT*** *that satisfies the following conditions.*

1. *Let F be any k-SAT instance over Var_n.*
   - *If F is satisfiable, then* ***DecideSAT***(F) *finds a satisfying assignment.*
   - *If F is unsatisfiable, then* ***DecideSAT***(F) *outputs "unsatisfiable"*
2. *Applied to* $\mathrm{Form}^+_{n,k,m}$, ***DecideSAT*** *runs in polynomial expected time.*

Our algorithm exploits the following connection between k-SAT and the maximum independent set problem. Let $V = \{1, \ldots, n\}^{k/2}$, and $\nu = n^{k/2}$ for the rest of this paper. Given any k-SAT instance F over Var_n we define the two graphs, $G_F = (V, E_F)$, $G'_F = (V, E'_F)$ as follows. We let $\{(v_1, \ldots, v_{k/2}), (w_1, \ldots, w_{k/2})\} \in E_F$ iff the k-clause $x_{v_1} \vee \cdots \vee x_{v_{k/2}} \vee x_{w_1} \vee \cdots \vee x_{w_{k/2}}$ occurs in F, and $\{(v_1, \ldots, v_{k/2}), (w_1, \ldots, w_{k/2})\} \in E'_F$ iff the k-clause $\neg x_{v_1} \vee \cdots \vee \neg x_{v_{k/2}} \vee \neg x_{w_1} \vee \cdots \vee \neg x_{w_{k/2}}$ occurs in F. For a graph G let $\alpha(G)$ be the independence number of G. The next lemma is an observation from [GoKr 2001].

Lemma 4. *If F is satisfiable, then* $\max\{\alpha(G_F), \alpha(G'_F)\} \geq 2^{-k/2} n^{k/2}$.

5. Announce successful certification of the property of Theorem 1 (c) if all preceeding algorithms are successful. Give an inconclusive answer otherwise.

The correctness of the algorithm follows because for a satisfying assignment a we get a fraction of $(1/8) \cdot (1 + so(1))$ of the 4-tuples of H of each of the following types,

$$(F_a, F_a, F_a, -), (F_a, F_a, , -, F_a), (F_a, -, F_a, F_a), (-, F_a, F_a, F_a).$$

This follows from Step 1 and Step 3. The unspecified position must be filled with T_a. Similarly we get a fraction of $(1/8) \cdot (1 + so(1))$ of the 4-tuples of H of each of the types

$$(T_a, T_a, T_a, -), (T_a, T_a, -, T_a), (T_a, -, T_a, T_a), (-, T_a, T_a, T_a).$$

Negligible discrepancy of the labelled projection graphs implies that the vacant slot must be filled with an F_a and the algorithm is correct. Completeness follows easily from our previous considerations.

Those cases of Theorem 1 which are left open by now can be treated similarlyand the Parity Theorem is proved.

4 Deciding Satisfiability in Expected Polynomial Time

Considerations of this section are based on the probabilistic model $\mathrm{Form}_{n,k,m}$ of random k-SAT instances. Given a standard set $\mathrm{Var} = \mathrm{Var}_n = \{x_1, \ldots, x_n\}$ of n propositional variables a k-clause is an ordered k-tuple of literals, that is negated or non-negated variables. Note that here any kind of double occurrence is allowed. A formula from $\mathrm{Form}_{n,k,m}$ is an ordered tuple $C_1 \wedge \cdots \wedge C_m$ of k-clauses and each formula occurs with the same probability $1/(2n)^{k \cdot m}$. Recall our remark from the beginning of Section 2 concerning different models of random k-SAT instances.

Our algorithm can even handle *semirandom formulas*. In the semirandom case, input instances are made up from a random share and a worst case part added by an adversary (cf. [FeKi 2001] for several semirandom models of combinatorial optimization problems and a motivating discussion). We shall study the following simple semirandom model $\mathrm{Form}^+_{n,k,m}$.

crossing edges is bounded above by $C \cdot n \cdot (n-1)/2 \cdot (1 + so(1))$. Tail bounds for the binomial distribution imply that we have at least $(1 + \varepsilon) \cdot C \cdot n \cdot (n-1)/2 \cdot (1 + so(1))$ crossing edges for a given constant $\varepsilon > 0$ with probability bounded above by $e^{-\Omega(C \cdot n(n-1))}$. As there are $2^{n(n-1)}$ possible cuts altogether, the probability that we have a cut with at least $(1 + \varepsilon) \cdot C \cdot n \cdot (n-1)/2 \cdot (1 + so(1))$ crossing edges is bounded above by $2^{n(n-1)} \cdot e^{-\Omega(C \cdot n(n-1))} = o(1)$ if we pick C sufficiently large. From Step 1 we know that $|E| = Cn^2(1 + so(1))$, therefore with high probability the algorithm cannot find a cut with at least $0.87 \cdot |E|$ edges and the output is "unsatisfiable" with high probability. □

As asymptotic terms in algorithms tend to cause confusion the following remark is in order. At this point we know that an algorithm efficiently certifying unsatisfiability exists, because there exist suitable $so(1)$-terms as we know from our theorems and considerations. In order to be of practical use we would have to specify the $so(1)$-terms. Suitable functions can be derived from our proofs.

Given a graph $G = (V, E)$, where $|V|$ is even. A bisection of G is a partition of V into two subsets V_1 and V_2 with $|V_1| = |V_2| = |V|/2$. The Min-Bisection problem is the problem to minimize the number of crossing edges, that is the number of edges with one endpoint in V_1 and the other endpoint in V_2. In this context $\mathrm{Opt}(G)$ is the minimal number of crossing edges of a bisection. There is a polynomial time approximation algorithm which, given G, finds a bisection such that the number of crossing edges is guaranteed to be at most $O((\log n)^2) \cdot \mathrm{Opt}(G)$, $|V| = n$, see [FeKr 2000].

Algorithm 4 *The input is a 4-Sat instance F.*

1. Certify the properties as stated in Theorem 1. If this certification is not successful the algorithm stops with an inconclusive answer.

2. Let S be the subset of all clauses of F whose first literal is a negated variable and whose remaining literals are non-negated variables. We construct the graph $G = (V, E)$ associated to this set S. Check if the maximal degree of G is at most $3 \cdot \ln n$. If this check is not successful the algorithm gives an inconclusive answer.

3. Apply the Min-Bisection approximation algorithm to G.

4. If the bisection found contains at least $(1/3)\cdot|E|$ edges, then the output is "unsatisfiable". Otherwise the algorithm cannot determine if F is satisfiable or not. □

Theorem 5. *Algorithm 4 efficiently certifies the unsatisfiability for $\mathrm{Form}_{n,4,p}$ where $p = C/n^2$ and C is sufficiently large.*

Proof. To show the correctness of the algorithm let F be a satisfiable formula and let a be a satisfying assignment of F. Let H be the hypergraph associated to S. From Step 1 we know that all up to $C \cdot n^2 \cdot (1 + so(1))$ 4-tuples from S are of one of the 8 types with an even number of T_a's. Now consider the partition of the vertices of the graph G where $V_1 = \{(x, y) \mid x \neq y \text{ and } x, y \in T_a \text{ or } x, y \in F_a\}$ and $V_2 = \{(x, y) \mid x \in T_a, y \in F_a \text{ or } x \in F_a, y \in T_a\}$. From Step 1 we know that $|F_a| = 1/2\cdot n\cdot(1+so(1))$ and $|T_a| = 1/2\cdot n\cdot(1+so(1))$. Therefore we have that $|V_1| = 1/2 \cdot n(n-1) \cdot (1 + so(1))$ and $|V_2| = 1/2 \cdot n(n-1) \cdot (1 + so(1))$. Thus V_1, V_2 is a cut which is almost a bisection. From Step 1 we know that the number of edges of G between V_1 and V_2 is $so(n^2)$. The cut V_1, V_2 can be made into a bisection by moving at most $so(n^2)$ vertices from V_1 to V_2 or vice versa. This may increase the number of crossing edges by $so(n^2)$, as we check in Step 2 that the maximal degree of G is at most logarithmic. Therefore we have that $\mathrm{Opt}(G)$ is at most $so(n^2)$. Then the Min-Bisection approximation algorithm will find a bisection with at most $O((\log n^2)^2) \cdot so(n^2)$ many crossing edges which is less than $(1/3) \cdot |E|$ and the algorithm does not give "unsatisfiable" as output.

To show the completeness of the algorithm let $F \in \mathrm{Form}_{n,4,p}$ where $p = C/n^2$ be a random formula. Step 1 is successful with high probability as we know from Theorem 1. As to Step 2, we know that the number of neighbors of a given vertex (x, y) follows the binomial distribution with parameters $(n-2)(n-3)$ and p', as defined above. Using the bound from [AlSp 92], Theorem A.12, page 237, we see that the certification in Step 2 successful with high probability. Let V_1, V_2 be an arbitrary bisection of G. As in the completeness proof of Algorithm 2 we have that with high probability the number of crossing edges of any bisection is at least $(1-\varepsilon) \cdot C \cdot n \cdot (n-1)/2$. Therefore Min-Bisection approximation algorithm can only find a bisection with at least this number of crossing edges. As we know

1. Use Algorithm 9 to check that the labelled projection of H onto coordinates $1, 2, 3$ has negligible discrepancy.

2. Use Algorithm 9 to check that the labelled projection of H onto coordinates $2, 3, 4$ has negligible discrepancy.

3. Do the same as 1. and 2. for the hypergraph associated to the clauses consisting only of negated variables.

4. If all checks have been successful, then certify that $|T_a| \sim_s (1/2)n$ and
$|F_a| \sim_s (1/2)n$ for any satisfying assignment a, where n is the number of variables of F.

Let F be any 4-Sat instance such that the algorithm is successful. Let a be an assignment with $|F_a| \geq (1/2) \cdot n \cdot (1 + \delta)$ where $\delta = \delta(n) > 0$ is not negligible in the sense of Section 2 (for example $\delta = 1/\log n$). From Step 1 we know that the fraction of 4-tuples of H of type $(F_a, F_a, F_a, -)$ is $((1/2)\cdot(1+\delta))^3\cdot(1+so(1))$. Under the assumption that a satisfies F, the empty slot is filled with a variable from T_a. From Step 2 we know that the fraction of 4-tuples of H of type $(-, F_a, F_a, T_a)$ is $((1/2) \cdot (1 + \delta))^2 \cdot (1/2)(1 - \delta)$. As δ is not negligible this contradicts negligible discrepancy of the labelled projection onto coordinates 2, 3, and 4 of H. In the same way we can exclude assignments with more variables set to true than false because Step 3 is successful. Therefore the algorithm is correct.

For random F the constructed hypergraphs are random hypergraphs and the completeness of Algorithms 9 implies the completeness of the algorithm.

Concerning Theorem 1 (c) we consider the following algorithm.

Algorithm 11 *The input is a 4-Sat instance F.*

1. Invoke Algorithm 10.

2. Let H be the hypergraph associated to the clauses of F consisting only of non-negated variables.

3. Certify that all 4 labelled projections onto any 3 different coordinates of H have negligible discrepancy (wrt. a suitable $\beta > 0$). Use Algorithm 9 .

4. Certify that all 6 labelled projections onto any two coordinates of H have negligible discrepancy. Use Algorithm 7 .

$x_7 = x_1$, $y_7 = y_1$. In case all the h_i, k_i are distinct we can bound the expectation by $n^4 \cdot (2c^2)^6 + so(n^4)$. In case that the h_i, k_i are not all distinct we get a bound of $so(n^4)$. Then we have that $E[\mathrm{Trace}](A_P^6)] = (2c^2n)^6 + so(n^6)$. Using (a) we can argue as in Lemma 2. □

The following algorithm certifies negligible discrepancy of labelled projections onto 3 coordinates of 4-uniform hypergraphs.

Algorithm 9 *The input is a* 4*-uniform hypergraph* $H = (V, E)$*. Let* $G = (V, E)$ *be the projection of* H *onto the coordinates* 1, 2*, and* 3*.*

1. Check if there is a suitable d *such that* G *is almost* d*-regular. That is check if* $d_{x,i} = d(1 + so(1))$ *for all vertices* x *and all* $i = 1, 2, 3$*.*

2. Check if the labelled projections onto any two of the coordinates 1, 2, 3 *of* H *have negligible discrepancy. Apply Algorithm 7.*

3. Check if the products of G *are almost* d*-regular with* $d = 2c^2n$*.*

4. For each of the 3 *labelled products* P *of* G *check if* $\mathrm{Trace}\,(A_P^6) = (2 \cdot c^2 \cdot n)^6 \cdot (1 + so(1))$ *where* A_P *is the adjacency matrix of* P*.*

5. If all checks are positive then certify negligible discrepancy of the labelled projection G*. Otherwise the algorithm fails.*

The correctness of the algorithm follows similarly to the previous considerations proving the correctness of Algorithm 7. For random hypergraphs $H \in HG_{n,4,p}$ the algorithm is complete.

Now we can state the algorithms proving Theorem 1. For Theorem 1 (a) just count and observe that for any possibility of putting negations into a subset of the 4 slots available we have $(n)_4$ clauses altogether. Each clause is picked with $p = C/n^2$ independently. Therefore the hypergraph associated with these clauses is a random hypergraph from $HG_{n,4,p}$ and the number of 4-tuples follows the binomial distribution with parameters $(n)_4$ and p. Tail bounds for the binomial distribution imply Theorem 1 (a).

Concerning Theorem 1 (b) we consider the following algorithm.

Algorithm 10 *The input is a* 4*-Sat instance* F*. Let* $H = (V, E)$ *be the hypergraph associated to the subset of clauses which consist of unnegated variables only.*

that $|E| = C \cdot n^2 \cdot (1 + so(1))$ the algorithm gives "unsatisfiable" as output with high probability. □

3 Proof of the Parity Theorem

We present the algorithms to prove Theorem 1. To deal with the problem of multiple occurrences of pairs of variables in several clauses we need to work with labelled (multi-)graphs and labelled (multi-)hypergraphs, instead of graphs and hypergraphs as in [GoJu 2002]. In a labelled graph we allow for multiple edges between two vertices. The edges between two vertices are distinguished by labels. The same applies to labelled hypergraphs.

Let $H = (V, E)$ be a standard 4-uniform hypergraph. When speaking of the projection of H onto coordinates 1 and 2 we think of H as a labelled multigraph in which the labelled edge $\{x_1, x_2\}_{(x_1,x_2,x_3,x_4)}$ is present if and only $(x_1, x_2, x_3, x_4) \in E$. The choice of the 4-tuples of H as labels ensures that each labelled edge corresponds to exactly one 4-tuple of H. We denote this projection by $G = (V, E)$. We extend the results and the notation from page 71 ff of [Ch 97] to G. Of course, any other set of two coordinates can be treated in the same way. The subsequent considerations refer to projections of 4-uniform hypergraphs only, but they actually apply to general labelled multigraphs.

Let $e = |E|$, $V = \{1, \ldots, n\}$, $X \subseteq V$, and $Y = V \setminus X$. We denote the number of labelled edges of G with one endpoint in X and the other endpoint in Y by $e(X, Y)$. That is

$$e(X,Y) = \begin{array}{l} |\{(x, y, -, -) \in E \mid x \in X, y \in Y\}| \\ \quad + |\{(y, x, -, -) \in E \mid x \in X, y \in Y\}| . \end{array}$$

Similarly $e(X)$ is the number of labelled edges with both endpoints from X, that is

$$e(X) = |\{(x_1, x_2, -, -) \in E \mid x_1, x_2 \in X\}| .$$

The edge density of G is $\rho = e/\binom{n}{2}$. Picking a set of vertices $\{x, y\}$ with $x \neq y$ uniformly at random from all such sets, ρ is the expected number of 4-tuples $(x_1, x_2, -, -) \in E$ with $x_1 = y$, $x_2 = x$ or vice

versa. Picking $X \subseteq V$ with $|X| = m$ uniformly at random from all sets with m elements $e(X)$ is a random variable with expectation

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

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

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

G has discrepancy δ with respect to β iff δ is minimal with the property that for all $X \subseteq V$ with $\beta \cdot n \leq |X| \leq (1 - \beta) \cdot n$,

$$(1 - \delta) \cdot \rho \cdot \binom{|X|}{2} \leq e(X) \leq (1 + \delta) \cdot \rho \cdot \binom{|X|}{2}$$

and for $Y = V \backslash X$

$$(1 - \delta) \cdot \rho \cdot |X| \cdot |Y| \leq e(X, Y) \leq (1 + \delta) \cdot \rho \cdot |X| \cdot |Y| \ .$$

In an asymptotic setting we use our terminology from Section 2 and say that G has negligible discrepancy (with respect to β) iff G has discrepancy δ where $\delta = \delta(n)$ is negligible. Negligible discrepancy (with respect to β) means the same, as for all $X \subseteq V$ with $|X| = \alpha \cdot n$ where $\beta \leq \alpha \leq 1 - \beta$,

$$e(X) \sim_s e\alpha^2 \tag{1}$$

and for $Y = V \backslash X$

$$e(X, Y) \sim_s 2e\alpha(1 - \alpha) \ . \tag{2}$$

Note that $\rho\binom{\alpha n}{2} = e\alpha^2(1 + O(1/n))$ and $\rho\alpha n(1 - \alpha)n = 2e\alpha(1 - \alpha)(1 + O(1/n))$.

The $n \times n$-matrix $A = A_G$ is the adjacency matrix of G where edges are counted with their multiplicity,

$$A(x, y) = |\{(x,\, y,\, -,\, -) \in E\}| + |\{(y,\, x,\, -,\, -) \in E\}| \, .$$

As A is real valued and symmetric, A has n different eigenvectors and corresponding real eigenvalues which we consider ordered as

$$\lambda_{1,A} \geq \lambda_{2,A} \geq \cdots \geq \lambda_{n,A}.$$

$X_1, \ldots, X_n$, independent, with $a \leq X_i \leq b$ for all i and $\mu = \mathrm{E}\left[\sum_i X_i\right]$ we have

$$\mathrm{Prob}\left[\left|\sum_{i=1}^{n} X_i - \mu\right| \geq \delta \cdot n\right] \leq \exp\left(-\frac{2n^2 \cdot \delta}{n(b - a)^2}\right) \ .$$

In our case we only have that $0 \leq X_z \leq n^2$ and the direct application of this bound makes no sense. However using Theorem A.12 from [AlSp 92], page 237, we get

$$\mathrm{Prob}\left[X_z \geq \log n\right] \leq \left(\frac{e}{\log n}\right)^{\log n} = n^{-\Omega(\log\log n)} \ .$$

Conditioning on the event that $X_z, Y_z, X'_z, Y'_z \leq \log n$ for all z, we get with constant $0 < \epsilon < 1/2$ that

$$\mathrm{Prob}\left[\left|\sum_z X_z \cdot Y_z - c^2 \cdot n\right| \geq \frac{1}{n^\epsilon} \cdot n\right] \leq e^{-\Omega\left(n^{\epsilon'}\right)}$$

for an $\epsilon' > 0$. The same argument applies to $\sum_z X'_z \cdot Y'_z$ and we get that $D = 2c^2 n \cdot (1 + so(1))$ with probability $1 - n^{-\Omega(\log\log n)}$. As we have $n(n-1)$ possible vertices altogether the claim follows all vertices (x_1, y_1) with $x_1 \neq y_1$.

Now assume that $x_1 = y_1$. Then we get that $D = \sum_z X_z \cdot (X_z - 1) + \sum_z X'_z \cdot (X'_z - 1)$. With $m = (n-2)(n-3)$, X_z follows the binomial distribution with parameters m and p. As $mp(1-p) = \mathrm{Var}[X_z] = \mathrm{E}[X_z^2] - (\mathrm{E}[X_z])^2$ we get that $\mathrm{E}[X_z^2] = c^2 + c + O(1/n)$. Therefore

$$\mathrm{E}[X_z(X_z - 1)] = c^2 + O(1/n)$$

and we can argue as in the first case.

(b) Let A_P be the adjacency matrix of P. We apply the same technique as in the proof of Lemma 2 but with closed walks of length 6 instead of 4. Given x_1, y_1 we need to bound the expected number of ordered sequences of pairs of 4-tuples like

$$(h_1, k_1),\ (h_2, k_2),\ (h_3, k_3),\ (h_4, k_4),\ (h_5, k_5),\ (h_6,\, k_6) \subseteq E$$

where for each i we have that $h_i = (z_i, x_i, x_{i+1}, -)$ and $k_i = (z_i, y_i, y_{i+1}, -)$ or $h_i = (z_i, x_{i+1}, x_i, -)$ and $k_i = (z_i, y_{i+1}, y_i, -)$ and

Therefore we must have

$$\alpha^4 nd^2(1+so(1)) \geq \left(1+\frac{\gamma^2}{\alpha}+\frac{\gamma^2}{1-\alpha}\right)\alpha^4 nd^2$$

and as $\beta \leq \alpha \leq 1-\beta$, β constant, we have $\gamma^2 = so(1)$ and thus $\gamma = so(1)$. The remaining cases to be considered in order to get negligible discrepancy of G can be treated in the same way. □

Lemma 3. *Let $H = (V,E)$ be a random hypergraph from $HG_{n,4,p}$ where $p = c/n^2$ and c is sufficiently large. Let G be the labelled projection of H onto the coordinates 1, 2, and 3. Let $P = (W,F)$ be the labelled product with respect to the first coordinate of G. Then we have*

(a) P is almost d-regular were $d = 2\cdot c^2\cdot n$ with probability $1-n^{-\Omega(\log\log n)}$.

(b) The adjacency matrix A_P has strong eigenvalue separation with respect to $k=6$.

The analogous claim applies to any other suitable set of coordinates.

Proof. (a) We consider the vertex $(x_1,y_1) \in W$. First, assume that $x_1 \neq y_1$. We introduce the random variables,

$$X_z = |\{(z,\, x_1,\, -,\, -) \in E\}|,\ Y_z = |\{(z,\, y_1,\, -,\, -) \in E\}|$$

and

$$X'_z = |\{(z,\, -,\, x_1,\, -) \in E\}|,\ Y'_z = |\{(z,\, -,\, y_1,\, -) \in E\}|$$

and finally $D = \sum_z X_z\cdot Y_z + \sum_z X'_z\cdot Y'_z$. Then D is the degree of the vertex (x_1,y_1) in the labelled product P. We get that $\mathrm{E}[X_z] = c - O(1/n)$ and, as X_z and Y_z are independent, that $\mathrm{E}[X_z\cdot Y_z] = c^2 - O(1/n)$. This gives

$$\mathrm{E}[D] = 2c^2 n(1+O(1/n)).$$

As the random variables $X_z\cdot Y_z$ are independent for different z's, we can apply Hoeffding's bound, see [Ho 87], page 104, Theorem 7: For

We let $\lambda = \lambda_A = \max_{2\leq i\leq n}|\lambda_{i,A}| = \max\{|\lambda_{2,A}|,|\lambda_{n,A}|\}$. In an asymptotic context we speak of strong eigenvalue separation with respect to a constant k. By this we mean that $\sum_{i=2}^n \lambda_i^k = so(\lambda_1^k)$. When k is even, strong eigenvalue separation implies in particular that $\lambda^k = so(\lambda_1^k)$ and as k is constant, that $\lambda = so(\lambda_1)$. From Linear Algebra it is known that for any $k \geq 0$

$$\mathrm{Trace}(A^k) = \sum_{x=1}^n A^k(x,x) = \sum_{i=1}^n \lambda_{i,A}^k.$$

Moreover, we have that $\mathrm{Trace}(A^k)$ is equal to the number of closed walks of length k in G.

The degree of the vertex x in G $d_x = d_{x,G}$ is

$$d_x = |\{(x,\, -,\, -,\, -) \in E\}| + |\{(-\,, x\, -,\, -) \in E\}|\,.$$

The $n\times n$-matrix $L = L_G$ is a normalized adjacency matrix, it is related to the Laplacian matrix, we have

$$L(x,y) = \frac{A(x,y)}{\sqrt{d_x d_y}}\ .$$

As $L = L_G$ is real valued and symmetric, too, we use all the eigenvalue notation introduced for A analogously for L. Here $\lambda_{1,L}$ is precisely known, $\lambda_{1,L} = 1$. Let $d = d(n)$ be given. In an asymptotic context we say that G is almost d-regular, if for any vertex x of G $d_{x,G} = d(n)\cdot(1+so(1))$. Theorem 5.1 and its corollaries on page 72/73 of [Ch 97] imply the following fact.

Fact 6 *Let $G = (V,\, E)$ where $V = \{1,\ldots,n\}$ be a projection onto two coordinates of a 4-uniform hypergraph $H = (V,\, E)$ with $e = |E|$. Let G be almost d-regular, let $\beta \leq \alpha \leq 1-\beta$ where $\beta > 0$ is a constant, and let $X \subseteq V$ with $|X| = \alpha n$. Then we have,*

(a) $|e(X) - e\alpha^2| \leq \lambda_L\cdot e\cdot\alpha\cdot(1+so(1))$,

(b) $|e(X,Y) - 2e\alpha(1-\alpha)| \leq \lambda_L\cdot 2\cdot e\cdot\sqrt{\alpha\cdot(1-\alpha)}\cdot(1+so(1))$ *for* $Y = V\setminus X$,

where $\lambda_L = \max_{2\leq i\leq n}|\lambda_{i,L}|$ and $L = L_G$ is defined above.

Note that (a) and (b) of Fact 6 imply negligible discrepancy, cf. (1), (2), provided $\lambda_L = so(1)$. Therefore we need methods to estimate λ_L. As eigenvalue properties are more easily to show for $A = A_G$ than for L_G the following lemma is important and may even be of some independent interest.

Lemma 1. *Let G be the projection onto two given coordinates of the 4-uniform hypergraph $H = (V, E)$ where $V = \{1, \ldots, n\}$. If G is almost d-regular and A_G has strong eigenvalue separation with respect to a given constant k, then L_G has strong eigenvalue separation with respect to k.*

Proof. Let W be the number of closed walks of length k in G. Then $W = \text{Trace}(A^k)$ and an inductive argument shows that

$$\text{Trace}\left(L_G^k\right) = \sum_{x=1}^{n} L_G(x,x) \leq W \cdot \left(\frac{1}{d}\right)^k \cdot (1 + so(1)) \ .$$

Then we get,

$$\begin{aligned}\sum_{i=1}^{n} \lambda_{i,L_G}^k &= \text{Trace}\left(L_G^k\right)\\ &\leq W \cdot \left(\frac{1}{d}\right)^k \cdot (1 + so(1))\\ &= \text{Trace}\left(A_G^k\right) \cdot \left(\frac{1}{d}\right)^k \cdot (1 + so(1))\\ &= \left(\sum_{i=1}^{n} \lambda_{i,A_G}^k\right) \cdot \left(\frac{1}{d}\right)^k \cdot (1 + so(1)) \ .\end{aligned}$$

As $\lambda_{1,L_G} = 1$, whereas $\lambda_{1,A_G}^k = d^k \cdot (1 + so(1))$ we get that $\sum_{i=2}^{n} \lambda_{i,L_G}^k = so(1)$. Note that $\lambda_{1,A}$ is always at most the maximal degree of G and at least the minimal degree. This can be seen from the well known characterization $\lambda_{1,A} = \max_{x \neq 0} x^{tr} Ax / x^{tr} x$ where x^{tr} is the transpose of the n-dimensional column vector x. □

We collect some probabilistic properties of labelled projections when H is a random hypergraph.

Lemma 2. *Let $p = c/n^2$ where c is a sufficiently large constant and let $H = (V, E)$ be a random hypergraph from $HG_{n,4,p}$. Let $G = (V, E)$ be a labelled projection of H onto two coordinates.*

Each of the two remaining terms is minimized when each a_z is the arithmetic mean. In this case we get for the first term

$$\frac{e_G(X,X,X)}{\alpha n}$$

and for the second term

$$\frac{e_G(Y,X,X)}{(1-\alpha)n} \ .$$

From negligible discrepancy of P we get that

$$e_{G_2}(X \times X) \sim_s \alpha^4 n d^2$$

as $|F| \sim_s nd^2$. Altogether we get

$$\begin{aligned}\alpha^4 \cdot n \cdot d^2 &\sim_s e_{G_2}(X \times X)\\ &\geq \frac{e_H(X,X,X)^2}{\alpha n} + \frac{e_H(Y,X,X)}{(1-\alpha)n} - \alpha^2 \cdot e \cdot (1 + so(1)) \ .\end{aligned}$$

As $e \sim_s nd$ and $d \geq n^\epsilon$ we get

$$\alpha^4 n d^2 (1 + so(1)) \geq \frac{e_G(X,X,X)^2}{\alpha n} + \frac{e_G(Y,X,X)^2}{(1-\alpha)n} \ .$$

As $e_G(X,X,X) + e_G(Y,X,X) = e_{G_1}(X) \sim_s \alpha^2 \cdot e$ by negligible discrepancy of G_1, the preceding sum is minimized when

$$e_G(X,X,X) \sim_s \alpha \cdot \alpha^2 \cdot e, \quad e_G(Y,X,X) \sim_s (1-\alpha) \cdot \alpha^2 \cdot e \ .$$

This can be seen as follows, if

$$e_G(X,X,X) = (\alpha + \gamma)\alpha^2 e \quad \text{and} \quad e_G(Y,X,X) = (1 - \alpha - \gamma)\alpha^2 e$$

then

$$\begin{aligned}&\frac{e_G(X,X,X)^2}{\alpha n} + \frac{e_G(Y,X,X)^2}{(1-\alpha)n}\\ &= \left(\frac{(\alpha+\gamma)^2}{\alpha} + \frac{(1-\alpha-\gamma)^2}{(1-\alpha)}\right) \alpha^4 n d^2\\ &= \left(\frac{\alpha^2 + 2\alpha\gamma + \gamma^2}{\alpha} + \frac{(1-\alpha)^2 - 2(1-\alpha)\gamma + \gamma^2}{1-\alpha}\right) \alpha^4 n d^2\\ &= \left(1 + \frac{\gamma^2}{\alpha} + \frac{\gamma^2}{1-\alpha}\right) \alpha^4 n d^2\end{aligned}$$

provided $d \geq n^\epsilon$ for an $\epsilon > 0$. Discrepancy notions for labelled products are totally analogous to those for labelled projection graphs defined above. We can omit the formal definitions at this point. Theorem 8 is an adaption of Theorem 2.3 in [GoJu 2002].

Theorem 8. *Let $\epsilon > 0$ and $d = d(n) \geq n^\epsilon$. Let $G = (V, E)$ with $|V| = n$ be the labelled projection hypergraph onto coordinates 1, 2 and 3 of the 4-uniform hypergraph $H = (V, E)$. Assume that G and H have the following properties.*

1. *G is almost d-regular.*
2. *The labelled projection graphs of H onto any two of the coordinates 1, 2, and 3 have negligible discrepancy with respect to $\beta > 0$.*
3. *The labelled products of G have negligible discrepancy with respect to β^2.*

Then the labelled projection G has negligible discrepancy with respect to β.

Proof. Let $|E| = e$ and let $X \subseteq V$ with $|X| = \alpha \cdot n$ where $\beta \leq \alpha \leq (1-\beta)$, and $Y = V \backslash X$. We need to show that $e_G(X, X, X) \sim_s \alpha^3 \cdot e$. Let $G_1 = (V, E)$ be the labelled projection of H onto the coordinates 2 and 3 and let $P = (W, F)$ with $W = V \times V$ be the labelled product of G with respect to the first coordinate.

For $z \in V$ let

$$a_z = |\{(z, x_1, x_2, -) \in E \mid x_1, x_2 \in X\}| \ ,$$

then

$$\begin{aligned} e_P(X \times X) &= \sum_{z \in V} a_z(a_z - 1) \\ &= \sum_{z \in X} a_z(a_z - 1) + \sum_{z \in Y} a_z(a_z - 1) \\ &= \sum_{z \in X} a_z^2 + \sum_{z \in Y} a_z^2 - \sum_{z \in V} a_z \ . \end{aligned}$$

From negligible discrepancy of G_1 we have for the third term of the sum, that

$$\sum_{z \in V} a_z = e_{G_1}(X) \sim_s \alpha^2 \cdot e.$$

(a) Let $d = d(n) = 2 \cdot c \cdot n$. Then G is almost d-regular with probability at least $1 - e^{-\Omega(n^\varepsilon)}$ for a constant $\varepsilon > 0$.
(b) The adjacency matrix $A = A_G$ has strong Eigenvalue separation with respect to $k = 4$.

Proof. (a) Altogether there are $2(n-1)_3$ 4-tuples which might induce a labelled edge of G incident with x. Each of these labelled edges is present with probability $p = c/n^2$ independently. Therefore for a given vertex x the degree of x, d_x, follows the binomial distribution with parameters $2(n-1)_3$ and p. As the expectation is $2cn(1 + O(1/n))$, standard tail bounds for the binomial distribution imply the result.

(b) For definiteness we consider the projection G onto coordinates 1 and 2 of H. The number of closed walks of length 4 in G is equal to

$$\text{Trace}(A^4) = \sum_{x=1}^{n} A^4(x, x) = \sum_{i=1}^{n} \lambda_{i,A}^4 \ .$$

We calculate $E[\text{Trace}(A^4)]$ using linearity of expectation. Given $x_1 \in V$, each closed walk of length 4 starting with x_1 is uniquely represented by an ordered sequence of 4-tuples (h_1, h_2, h_3, h_4) which, given $x_2, x_3, x_4, x_5 = x_1$, can be decomposed as $h_i = (x_i, x_{i+1}, -, -)$ or $h_i = (x_{i+1}, x_i, -, -)$. This allows us to bound the expected number of closed walks starting with x_1 by the following case distinction. For walks such that the h_i are all distinct we get an upper bound of

$$n^3 \cdot (n^2)^4 \cdot \left(\frac{2c}{n^2}\right)^4 \cdot (1 + O(1/n)) = (2 \cdot c)^4 \cdot n^3 \cdot (1 + so(1)) \ .$$

When $h_1 = h_2$ and h_3, h_4 are distinct we get an expectation of

$$n^2 \cdot (n^2)^3 \cdot \left(\frac{2c}{n^2}\right)^3 \cdot (1 + O(1/n)) = so\left((2cn)^3\right) \ .$$

Similarly we get an expectation of $so((2cn)^3)$ for the remaining cases when the h_i are not all distinct. The expected number of closed walks altogether therefore is $n \cdot n^3(2c)^4 + so(n^4)$. As G is asymptotically d-regular, where $d = 2cn$, with probability $1 - e^{-\Omega(n^\varepsilon)}$ for an $\varepsilon > 0$, we get that $\text{E}\left[\lambda_{1,A_G}^4\right] \sim_s (2cn)^4$. Therefore $\text{E}\left[\sum_{i=2}^n \lambda_{i,A_G}^4\right] = so\left(n^4\right)$. Markov's inequality shows for all $f(n)$ not in $so(n^4)$ that

$$\text{Prob}\left[\sum_{i=2}^{n} \lambda_{i,A_G}^4 \geq f(n)\right] \leq \frac{so(n^4)}{f(n)} = o(1)$$

and the claim holds. □

Now we can efficiently certify negligible discrepancy with respect to a given constant β of projection graphs.

Algorithm 7 *Input is a 4-uniform hypergraph $H = (V, E)$. Let $G = (V, E)$ be the projection onto two given coordinates of H.*

1. Check almost regularity of G. If the check fails, the algorithm stops with an inconclusive answer. Determine a d such that for all vertices $x \in V$ $d_x = d \cdot (1 + so(1))$.

2. Let A be the adjacency matrix of G. Compute $\mathrm{Trace}\,(A^4)$.

3. If $\mathrm{Trace}\,(A^4) = d^4 \cdot (1 + so(1))$ *then the algorithm stops with a successful certification. Otherwise it fails with an inconclusive answer.*

To show the correctness of the algorithm, we assume that we have a projection G whose discrepancy is not negligible. The interested reader can easily supply the asymptotic detail to make this assumption precise. If G is not almost regular the algorithm will detect this and answer inconclusively. If G is almost d-regular we can apply Fact 6 and get that λ_{L_G} is not negligible. Therefore $\lambda^4_{L_G}$ is not negligible, too. As all $\lambda^4_{i,L_G} \geq 0$ we have that $\sum_{i=2}^{n} \lambda^4_{i,L_G} \geq \lambda^4_{L_G}$ is not negligible. As $\lambda_{1,L_G} = 1$, we have that the matrix L_G does not have strong eigenvalue separation with respect to $k = 4$. By Lemma 1 we have that A_G does not have strong eigenvalue separation with respect to $k = 4$. By almost d-regularity using the characterization of λ_{1,A_G} from the proof of Lemma 1 we have that $\lambda^4_{1,A_G} = d^4 \cdot (1 + so(1))$. Therefore we cannot have that $\mathrm{Trace}(A^4) = d^4 \cdot (1 + so(1))$ and the algorithm can only answer inconclusively.

To show the completeness of the algorithm let H be a random hypergraph from $HG_{n,4,p}$. Lemma 2 (a) implies almost d-regularity with high probability. Therefore $\lambda_{1,A_G} = d \cdot (1 + so(1))$. With Lemma 2 (b) we have that $\mathrm{Trace}(A^4_G) = d^4 \cdot (1 + so(1))$ with high probability and the algorithm certifies this.

We need to certify discrepancy properties of projections onto 3 given coordinates of a random 4-uniform hypergraph from $HG_{n,4,p}$ where $p = c/n^2$. Let $H = (V, E)$ be a standard 4-uniform hypergraph. When speaking of the projection of H onto coordinates 1, 2, and 3, we think of H as a labelled 3-uniform hypergraph $G = (V, E)$ in which the labelled 3-tuple $(x_1, x_2, x_3)_{(x_1,x_2,x_3,x_4)}$ is present if $(x_1, x_2, x_3, x_4) \in E$. We restrict attention to the projection onto coordinates 1, 2 and 3 in the following. But of course everything can be done in the same way for any other set of 3 coordinates. For $X, Y, Z \subseteq V$ we define

$$e_G(X, Y, Z) = |\{(x, y, z, -) \in E \mid (x, y, z) \text{ is of type } (X, Y, Z)\}| \ .$$

For the notion of type we refer to the beginning of Section 2. With $n = |V|$ and $e = |E|$ we say that the projection G has negligible discrepancy with respect to β if for all X with $|X| = \alpha n$, $\beta \leq \alpha \leq 1 - \beta$, and $Y = V \backslash X$ we have that

$$e_G(X, X, X) \sim_s \alpha^3 \cdot e, \quad e_G(X, Y, X) \sim_s \alpha^2 (1 - \alpha) \cdot e$$

and analogously for the remaining 6 possibilities of placing X and Y into the 3 slots available, compare (1) and (2) for the case of labelled projection graphs. For $1 \leq i \leq 3$ and $x \in V$ we let $d_{x,i}$ be the number of 4-tuples in E which have x in the i'th slot. Given $d = d(n)$, we say that G is almost d-regular if and only if $d_{x,i} = d \cdot (1 + so(1))$ for all $x \in V$ and all $i = 1, 2, 3$. The notion of an adjacency matrix is not known for hypergraphs like G and we cannot directly certify discrepancy properties. Therefore we assign labelled product graphs to G.

Definition 1 (Labelled product). *Let $G = (V, E)$ be the projection onto coordinates 1, 2, and 3 of the 4-uniform hypergraph $H = (V, E)$.*

The labelled product of G with respect to the first coordinate is the labelled graph $P = (W, F)$, where $W = V \times V$ and F is defined as: For $x_1, x_2, y_1, y_2 \in V$ with $(x_1, y_1) \neq (x_2, y_2)$ we have $\{(x_1, y_1), (x_2, y_2)\}_{(h,k)} \in F$ iff $h = (z, x_1, x_2, -) \in E$ and $k = (z, y_1, y_2, -) \in E$ and (!) $h \neq k$.

In the labelled product we can think of each ordered pair of different 4-tuples from E, $((z, x_1, x_2, -), (z, y_1, y_2, -))$ as one edge. Of course labelled products are defined with respect to each of the 3 coordinates of G in the same way. If the projection G is almost d-regular the number of labelled edges of the product is

$$n \cdot d \cdot (d - 1) \cdot (1 + so(1)) = n \cdot d^2 \cdot (1 + so(1))$$