Finding a large hidden clique in a random graph *

Noga Alon † Michael Krivelevich ‡ Benny Sudakov §

Abstract

We consider the following probabilistic model of a graph on n labeled vertices. First choose a random graph $G(n, 1/2)$ and then choose randomly a subset Q of vertices of size k and force it to be a clique by joining every pair of vertices of Q by an edge. The problem is to give a polynomial time algorithm for finding this hidden clique almost surely for various values of k. This question was posed independently, in various variants, by Jerrum and by Kučera. In this paper we present an efficient algorithm for all $k > cn^{0.5}$, for any fixed $c > 0$, thus improving the trivial case $k > cn^{0.5}(\log n)^{0.5}$. The algorithm is based on the spectral properties of the graph.

1 Introduction

A *clique* in a graph G is a set of vertices any two of which are connected by an edge. Let $w(G)$ denote the maximum number of vertices in a clique of G.

The problem of determining or estimating $w(G)$ and that of finding a clique of maximum size in G are fundamental problems in Theoretical Computer Science. The problem of computing $w(G)$ is well known to be NP-hard [16]. The best known approximation algorithm for this quantity, designed by Boppana and Halldórsson [8], has a performance guarantee of $O(n/(\log n)^2)$, where n is the number of vertices in the graph. When the graph contains a large clique, there are better algorithms, and the best one, given in [3], shows that if $w(G)$ exceeds $n/k + m$, where k is a fixed integer and $m > 0$, then one can find a clique of size $\tilde{\Omega}(m^{3/(k+1)})$ in polynomial time, where here the notation $g(n) = \tilde{\Omega}(f(n))$ means, as usual, that $g(n) \geq \Omega(f(n)/(\log n)^c)$ for some constant c independent of n.

*A preliminary version of this paper appeared in Proceedings of the Ninth Annual ACM-SIAM SODA, ACM Press (1998), 594-598.

†Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: noga@math.tau.ac.il. Research supported in part by a USA Israeli BSF grant, by a grant from the Israel Science Foundation and by the Hermann Minkowski Minerva Center for Geometry at Tel Aviv University.

‡Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: krivelev@math.tau.ac.il. Research supported in part by a Charles Clore Fellowship.

§Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email: sudakov@math.tau.ac.il.

On the negative side, it is known, by the work of [5] following [9] and [6], that for some $b > 0$ it is impossible to approximate $w(G)$ in polynomial time for a graph on n vertices within a factor of n^b, assuming P $\neq$ NP. The exponent b has since been improved in various papers and recently it has been shown by Håstad [13] that it is in fact larger than $(1 - \delta)$ for every positive δ, assuming NP does not have polynomial time randomized algorithms. Another negative result, proved in [1] following [20], shows that it is impossible to approximate $w(G)$ for an n vertex graph within a factor of $n/\log^7 n$ by a polynomial size *monotone* circuit.

These facts suggest that the problem of finding the largest clique in a general graph is intractable. It is thus natural to study this problem for appropriately randomly generated input graphs. This is of interest theoretically, and is also motivated by the fact that in real applications the input graphs often have certain random properties. The study of the performance of algorithms on random input graphs gained popularity recently, see the survey of Frieze and McDiarmid [10] and its many references.

Let $G(n, 1/2)$ denote the random graph on n labeled vertices obtained by choosing, randomly and independently, every pair ij of vertices to be an edge with probability $1/2$. It is known that almost surely (that is, with probability that approaches 1 as n tends to infinity), the value of $w(G)$ is either $\lfloor r(n) \rfloor$ or $\lceil r(n) \rceil$, for a certain function $r(n) = (2 + o(1)) \log_2 n$ which can be written explicitly (cf., e.g., [4]). Several simple polynomial time algorithms (see e.g., [12]) find, almost surely, a clique of size $(1 + o(1)) \log_2 n$ in $G(n, 1/2)$, that is, a clique of roughly half the size of the largest one. However, there is no known polynomial time algorithm that finds, almost surely, a clique of size at least $(1 + \epsilon) \log_2 n$ for any fixed $\epsilon > 0$. The problem of finding such an algorithm was suggested by Karp [17]. His results, as well as more recent ones of Jerrum [14] implied that several natural algorithms do not achieve this goal and it seems plausible to conjecture (see [14]) that in fact there is no polynomial time algorithm that finds, with probability more than a half, say, a clique of size bigger than $(1 + \epsilon) \log_2 n$. This conjecture has certain interesting cryptographic consequences, as shown in [15].

The situation may become better in a random model in which the biggest clique is larger. Following [14], let $G(n, 1/2, k)$ denote the probability space whose members are generated by choosing a random graph $G(n, 1/2)$ and then by placing randomly a clique of size k in it. As observed by Kučera [18], if k is bigger than $c\sqrt{n \log n}$ for an appropriate constant c, the vertices of the clique would almost surely be the ones with the largest degrees in G, and hence it is easy to find them efficiently. Can we design an algorithm that finds the biggest clique almost surely if k is $o(\sqrt{n \log n})$? This problem was mentioned in [18]. Here we solve it, by showing that for every $\epsilon > 0$ there is a polynomial time algorithm that finds, almost surely, the unique largest clique of size k in $G(n, 1/2, k)$, provided $k \geq \epsilon n^{1/2}$. Although this beats the trivial algorithm based on the degrees only by a logarithmic factor, the technique applied here, which is based on the spectral properties of the graph, and

resembles the basic approach in [3], is interesting, and may be useful for tackling related problems as well.

2 The main result

In this section we describe our algorithm and analyze its performance on graphs generated according to the distribution $G(n, 1/2, k)$. The results can easily be extended to similar models of random graphs. Since the trivial algorithm based on the degrees solves the clique problem almost surely for $k > c\sqrt{n \log n}$ we assume, from now on, that $k = O(\sqrt{n \log n})$. We also assume, whenever this is needed, that n is sufficiently large. To simplify the presentation, we omit all floor and ceiling signs whenever these are not crucial.

2.1 The basic algorithm

In this subsection we describe the basic algorithm dealing with a hidden clique of size at least $10\sqrt{n}$. The algorithm is based on the spectral properties of the adjacency matrix of the graph. After the analysis of the algorithm in the next subsection we explain, in subsection 2.3, how to modify the basic algorithm to reduce the constant 10 to any positive constant.

Given a graph $G = (V, E)$ denote by A the adjacency matrix of G, that is, the n by n matrix $(a_{uv})_{u,v \in V}$ defined by $a_{uv} = 1$ if $uv \in E$ and $a_{uv} = 0$ otherwise. It is well known that since A is symmetric it has real eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and an orthonormal basis of eigenvectors $v_1, \ldots, v_n$, such that $Av_i = \lambda_i v_i$. The crucial point of the algorithm is that one can almost surely find a big portion of the hidden clique from the second eigenvector of A. Since there are several efficient algorithms to compute the eigenvectors and eigenvalues of symmetric matrices (see, e.g., [19]), we can certainly calculate v_2 in polynomial time. Our first algorithm is very simple and consists of two stages.

Algorithm A

Input: A graph $G = (V, E)$ from the distribution $G(n, 1/2, k)$ with $k \geq 10\sqrt{n}$.

1. Find the second eigenvector v_2 of the adjacency matrix of G

2. Sort the vertices of V by decreasing order of the absolute values of their coordinates in v_2 (where equalities are broken arbitrarily) and let W be the first k vertices in this order. Let $Q \subset V$ be the set of all vertices of G which have at least $3k/4$ neighbors in W.

Output: The subset $Q \subset V$.

This completes the description of the algorithm.

2.2 The properties of the second eigenvector

We claim that almost surely the above algorithm finds the (unique) clique of size k in G. To prove this fact we first need to establish some results about the spectrum of G. For the analysis of the algorithm we assume that the set of vertices V is $\{1, \ldots, n\}$ and the hidden clique Q in G consists of the first k vertices of V.

Proposition 2.1 *Let $G = G(n, 1/2, k)$, where $k = o(n)$, then almost surely the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ of the adjacency matrix A of G satisfy:*

(i) $\lambda_1 \geq (\frac{1}{2} + o(1))n$

(ii) $\lambda_i \leq (1 + o(1))\sqrt{n}$ *for all* $i \geq 3$

Proof. By the variational definition of the eigenvalues of A (see e.g. [23], pp. 99–101) we have that

$$\lambda_i = \max_{\dim F = i} \min_{x \in F, x \neq 0} \frac{x^t A x}{x^t x} = \min_{\dim F = n-i+1} \max_{x \in F, x \neq 0} \frac{x^t A x}{x^t x},$$

where F ranges over all subspaces of R^n of the appropriate dimension. In particular λ_1 is simply the maximum of $x^t Ax / x^t x$ over all nonzero vectors x. Therefore by taking x to be the all 1 vector we obtain the well known result that λ_1 is at least the average degree of G. By the known estimates for the Binomial distribution, the average degree of G is $(1/2 + o(1))n$ almost surely. This proves (i).

To prove (ii) we need the following result about the spectrum of the random graph, proved by Füredi and Komlós [11].

Lemma 2.2 *Let $\lambda_1 \geq \ldots \geq \lambda_m$ be the eigenvalues of the adjacency matrix of the random graph $G(m, 1/2)$, then almost surely*

$$\max_{i \geq 2} |\lambda_i| \leq \sqrt{m} + O(m^{1/3} \log m).$$

In order to bound the eigenvalues of the matrix A we represent the graph G as an edge disjoint union of two random graphs. Let $G_2 = G(k, 1/2)$ be the random graph on the set of vertices of the clique Q. Denote by A_2 the adjacency matrix of the graph, which is the union of G_2 together with the remaining $n - k$ isolated vertices. Remove all the edges of G_2 from G and denote by A_1 the adjacency matrix of the remaining graph G_1. It is easy to see that G_1 is obtained according to the distribution $G(n, 1/2)$. By definition $A = A_1 + A_2$. Denote by u_i the eigenvector of A_i corresponding to the largest eigenvalue of A_i, for $i = 1, 2$ respectively. Let F be the subspace of all vectors which are orthogonal to both u_1 and u_2. By the definition of F together with Lemma 2.2 we have that

almost surely for any vector $x \in F, x \neq 0$, $x^t A_1 x / x^t x \leq (1+o(1))\sqrt{n}$ and $x^t A_2 x / x^t x \leq (1+o(1))\sqrt{k}$. Therefore

$$\frac{x^t A x}{x^t x} = \frac{x^t A_1 x}{x^t x} + \frac{x^t A_2 x}{x^t x} \leq (1+o(1))\sqrt{n}$$

for all $x \in F, x \neq 0$ where here we used the fact that $k = o(n)$. Since $\dim F \geq n-2$, by the variational definition of the eigenvalues of the matrix A we conclude that $\lambda_i \leq (1+o(1))\sqrt{n}$ for all $i \geq 3$. This completes the proof of (ii). $\Box$

The crucial observation for the analysis of the algorithm is that the eigenvector v_2 has most of its weight on the clique. To show this we exhibit a vector z whose first k coordinates are considerably larger then the rest of the coordinates and prove that it is close to the second eigenvector of A. Let $z = (z_i, 1 \leq i \leq n)$ be the vector defined by $z_i = n-k$ if $i \leq k$ and $z_i = -k$ otherwise. We denote by $\| x \|$ the l_2-norm of a vector x.

Proposition 2.3 *In the above notation almost surely there exists a vector $\delta = (\delta_i, 1 \leq i \leq n)$, satisfying $\| \delta \|^2 \leq \frac{1}{60} \| z \|^2$ so that $z - \delta$ is collinear with the second eigenvector v_2 of A.*

Proof. We use the following lemma.

Lemma 2.4 *Almost surely $\| (A - \frac{k}{2} I) z \|^2 \leq (\frac{1}{4} + o(1)) n^3 k$.*

Before proving the lemma, we apply it to deduce the existence of δ as above. Let $z = c_1 v_1 + \ldots + c_n v_n$ be the representation of z as a linear combination of the eigenvectors v_i. We show that the coefficients $c_1, c_3, \ldots, c_n$ are small compared to $\| z \|$. Indeed, $(A - \frac{k}{2} I) z = \sum_{i=1}^n c_i (\lambda_i - \frac{k}{2}) v_i$ and thus

$$\begin{aligned} \| (A - \frac{k}{2} I) z \|^2 &= \sum_{i=1}^n {c_i}^2 (\lambda_i - \frac{k}{2})^2 \\ &\geq (1+o(1)) (\frac{k}{2} - \sqrt{n})^2 \sum_{i \neq 2} {c_i}^2, \end{aligned} \tag{1}$$

where the last inequality follows from Proposition 2.1, whose assertion holds, as $k = o(n)$. Define $\delta = c_1 v_1 + c_3 v_3 + \ldots + c_n v_n$. By (1) and Lemma 2.4 it follows that

$$\| \delta \|^2 = \sum_{i \neq 2} {c_i}^2 \leq (1+o(1)) \frac{n^3 k}{(k - 2\sqrt{n})^2} < \frac{1}{60} k n (n-k) = \frac{1}{60} \| z \|^2,$$

where here we used the fact that $k \geq 10\sqrt{n}$. On the other hand $z - \delta = c_2 v_2$ is collinear with v_2. $\Box$

Note that the above discussion supplies an estimate of the second eigenvalue of A. Indeed, $\| z \|^2 - \| \delta \|^2 = {c_2}^2 \geq (59/60) \| z \|^2$. By substituting this inequality into (1) we obtain, using Lemma 2.4, that

$$(\frac{1}{4} + o(1)) n^3 k \geq {c_2}^2 (\lambda_2 - \frac{k}{2})^2 \geq \frac{59}{60} k n (n-k) (\lambda_2 - \frac{k}{2})^2 \geq \frac{2}{3} k n^2 (\lambda_2 - \frac{k}{2})^2.$$

This implies that $(\lambda_2 - k/2)^2 \leq n/2$, thus proving the following corollary.

Corollary 2.5 *The second eigenvalue of the matrix A almost surely satisfies the following inequality:*

$$\frac{k}{2} - \sqrt{\frac{n}{2}} \leq \lambda_2 \leq \frac{k}{2} + \sqrt{\frac{n}{2}}.$$

In particular when $k \geq 10\sqrt{n}$, λ_2 is much bigger than λ_i for all $i \geq 3$. $\Box$

Proof of Lemma 2.4 Let $(A - \frac{k}{2}I)z = (t_1, t_2, \ldots, t_n)$. Denote by $B(m,p)$ the binomial distribution with parameters m and p. By the definition of the matrix A we have that the random variable t_i is given by

$$t_i = \begin{cases} (\frac{k}{2} - 1)(n-k) - kY_i, & 1 \leq i \leq k \\ \frac{k^2}{2} + (n-k)X_i - kY_i, & k+1 \leq i \leq n \end{cases}$$

where X_i is a binomially distributed random variable $B(k, 1/2)$, and Y_i is a binomially distributed random variable $B(n-k, 1/2)$ for $i \leq k$, and $B(n-k-1, 1/2)$ for $i > k$. Using the standard estimates for Binomial distributions (see, e.g., [4], Appendix A) we get that almost surely $Y_i = (n-k)/2 + O(\sqrt{n \log n})$ for all $1 \leq i \leq k$. Therefore almost surely $t_i = -(n-k) + O(k\sqrt{n \log n})$ for all $i \leq k$. Thus $\sum_{i=1}^{k} {t_i}^2 = O(k(k\sqrt{n \log n})^2) = O(k^3 n \log n) = o(n^3 k)$. In order to bound the remaining $\sum_{i=k+1}^{n} {t_i}^2$ we first modify the expression for t_i in the following way,

$$t_i = (n-k)(X_i - \frac{k}{2}) - k(Y_i - \frac{n-k-1}{2} - \frac{k+1}{2}).$$

Then $\sum_{i=k+1}^{n} {t_i}^2$ can be written as $S_1 + S_2 + S_3$, where

$$\begin{aligned} S_1 &= (n-k)^2 \textstyle\sum_{i=k+1}^{n} (X_i - \frac{k}{2})^2, \\ S_2 &= k^2 \textstyle\sum_{i=k+1}^{n} (Y_i - \frac{n-k-1}{2} - \frac{k+1}{2})^2 \\ S_3 &= -2k(n-k) \textstyle\sum_{i=k+1}^{n} (X_i - \frac{k}{2})(Y_i - \frac{n-k-1}{2} - \frac{k+1}{2}). \end{aligned}$$

Applying again the standard estimates for Binomial distributions we get that almost surely $X_i = k/2 + O(\sqrt{k \log k})$ and $Y_i = (n-k-1)/2 + O(\sqrt{n \log n})$ for $i \geq k+1$. This implies that $S_2 = O(k^2(n-k)n \log n) = o(n^3 k)$ and $S_3 = O(k(n-k)(n-k)\sqrt{k \log k}\sqrt{n \log n}) = o(n^3 k)$.

It remains to bound S_1. By the definition of X_i, $X_i - k/2$ for $i > k$ can be viewed as the sum of k independent random variables each taking values $1/2$ and $-1/2$ with equal probability. This implies that the expected value of $(X_i - k/2)^2$ is $k/4$ and the expected value of $(X_i - k/2)^4$ is $O(k^2)$. Note that X_i and X_j are independent random variables for $i \neq j$, since they correspond to the edges going from two different vertices of G to the clique. Thus the expected value μ of $\sum_{i=k+1}^{n} (X_i - k/2)^2$ is $\mu = (n-k)k/4$ and its variance is equal to the sum of the variances, which is $O(k^2(n-k)) = o(\mu^2)$. Therefore by Chebyshev's Inequality we obtain that almost surely $S_1 = (1 + o(1))(n-k)^2 \mu = (1/4 + o(1))n^3 k$. This completes the proof of Lemma 2.4. $\Box$

Let the normalized second eigenvector of A be $v_2 = (a_i, 1 \leq i \leq n)$. Note that by Corollary 2.5 it is unique almost surely. Recall that in the algorithm, W is the set of indices which correspond to the k largest values of $|a_i|, 1 \leq i \leq n$. We use Proposition 2.3 to show that almost surely $|W \cap \{1, \ldots, k\}| \geq 5k/6$, thus proving that at least $5/6$ of the k largest (in absolute value) coordinates of the second eigenvector correspond to the vertices of the clique. Note that one gets the same set of indices W for every $\alpha v_2, \alpha \neq 0$. Consider $c_2 v_2 = z - \delta$ from Proposition 2.3, where $\| \delta \|^2 \leq (1/60)kn^2$. The number of coordinates of δ which are greater in absolute value than $n/3$ is at most $k/6$. Since the coordinates of $c_2 v_2$ are $z_i - \delta_i$ and $z_1 = \ldots = z_k = n - k$, $z_{k+1} = \ldots = z_n = -k$ we conclude that at least $k - k_1$ of the first k coordinates of $c_2 v_2$ are greater than $n - k - n/3 > n/2$ and at least $n - k - k_2$ of the last $n - k$ coordinates are at most $k + n/3 < n/2$, where $k_1 + k_2 \leq k/6$. This implies that $|W \cap \{1, \ldots, k\}| \geq 5k/6$.

To finish the proof of the correctness of the algorithm we show that every vertex outside the clique is almost surely adjacent to less than $3k/4$ vertices of W. Indeed, every edge outside the clique appears in $G(n, 1/2, k)$ randomly and independently with probability $1/2$. Thus all vertices outside the clique are adjacent, almost surely, to at most $(1 + o(1))k/2$ vertices of the clique. Since W has at most $k - 5k/6 = k/6$ vertices not in the clique, it follows that, almost surely, each vertex not in the clique has at most $(1 + o(1))k/2 + k/6 < 3k/4$ neighbors in W. This guarantees that in stage two of the algorithm we choose only vertices of the clique, and choose all of them because every vertex of the planted clique is adjacent to at least $5k/6$ vertices of W, as shown above.

2.3 Reducing the constant

The main idea in improving the performance of Algorithm A is to consider the subgraph of G induced on the set $V_1 \subset V$ of all common neighbors of some fixed number of vertices in the clique Q. Doing this we achieve two goals simultaneously. First, $G[V_1]$ still contains a clique of almost the same size k, second, since our graph is random, V_1 is much smaller than V. Thus we improve the ratio between the clique and the size of the graph and can now use the algorithm A. For any subset $S \subset V$ we define $N^*(S) = \{v \in V \setminus S : vu \in E(G) \text{ for all } u \in S\}$.

Algorithm B

Input: A graph $G = (V, E)$ from the distribution $G(n, 1/2, k)$ with $k = c\sqrt{n}$.

1. Define $s = 2[\log_2(10/c)] + 2$.

2. For all subsets $S \subset V, |S| = s$ do

begin

3. Run the Algorithm A on the induced subgraph $G[N^*(S)]$ and denote by Q_S the resulting set.

4. If $Q_S \cup S$ is a clique of size k, then $Q = Q_S \cup S$ and go to 6.

end

5. Take Q to be an arbitrary k-subset of V.

6. Output: The subset $Q \subset V$.

We claim that for any fixed c Algorithm B almost surely produces the hidden clique. To prove this let us first observe that for any fixed subset $S \subset V$ of size $|S| = s$ the cardinality of $N^*(S)$ in the random graph $G(n, 1/2)$ is a binomially distributed random variable with parameters $n - s$ and $1/2^s$. Thus almost surely $|N^*(S)| = (1 + o(1))n/2^s$ for all subsets of vertices of size s in $G(n, 1/2)$. The addition of a clique of size k can increase $|N^*(S)|$ only by at most $k - s$. Therefore $|N^*(S)| = (1 + o(1))n/2^s$ almost surely also in $G(n, 1/2, k)$.

Since Algorithm B checks all the subsets of V of size s, in some step it will reach a subset $S, |S| = s$, which belongs to the clique Q. At this iteration we almost surely get the hidden clique. Indeed, for a fixed subset S of the clique, $|S| = s$, and a fixed $N^*(S)$, the induced subgraph $G[N^*(S)]$ can be treated as a truly random graph $G(|N^*(S)|, 1/2, k - s)$. This is because one can generate $G[N^*(S)]$ as follows: first choose a clique Q and fix a subset S of size s in it, then expose the edges from S to $V \setminus S$ thus fixing $N^*(S)$, and then expose all the edges inside $N^*(S)$. We have $|N^*(S)| = (1 + o(1))n/2^s$ and $G[N^*(S)]$ contains a clique of size $k - s = (1 + o(1))k$. By our choice of s, the size of the hidden clique satisfies $k - s \geq 10\sqrt{|N^*(S)|}$. This guarantees that at this iteration the algorithm A will find the clique $Q \setminus S$ and proves the correctness of Algorithm B.

3 Concluding remarks

We described a polynomial time algorithm that finds, almost surely, the unique clique of size k in $G(n, 1/2, k)$ for $k \geq \Omega(\sqrt{n})$. The obvious challenge that remains open is to design efficient algorithms that work, almost surely, for smaller values of k. If $k = n^{1/2-\epsilon}$ for some fixed $\epsilon > 0$, even the problem of finding a clique of size at least $(1 + \epsilon) \log_2 n$ in $G(n, 1/2, k)$, suggested in [14], is open and seems to require new ideas.

Another interesting version of this problem was suggested by Saks [21]. Suppose G is a graph on n vertices which has been generated either according to the distribution $G(n, 1/2)$ or according to the distribution $G(n, 1/2, k)$ for, say, $k = n^{0.49}$. It is then obvious that an all powerful prover can convince a polynomial time verifier deterministically that, almost surely, G has been generated according to the distribution $G(n, 1/2, k)$ (if indeed that was the case). To do so, he simply presents the clique to the verifier. However, suppose G has been generated according to the distribution $G(n, 1/2)$. Can the prover convince the verifier (without using randomness, of course) that this is the case, almost surely ? At the moment we cannot design such a protocol if $k = o(\sqrt{n})$ (while for

$k \geq \Omega(\sqrt{n})$ the verifier can clearly convince himself, using Algorithm B.)

The spectral properties of a graph encode some detailed structural information on it. The ability to compute the eigenvectors and eigenvalues of a graph in polynomial time provides a powerful algorithmic tool, which has already found several applications (see, e.g., [7], [2], [22]). The spectral approach, and the techniques developed here, may well have additional algorithmic applications in the future too.

References

[1] N. Alon and R. B. Boppana, *The monotone circuit complexity of Boolean functions*, Combinatorica 7 (1987), 1–22.

[2] N. Alon and N. Kahale, *A spectral technique for coloring random 3-colorable graphs*, Proc. of the 26^{th} ACM STOC, ACM Press (1994), 346–355. Also: SIAM J. Comput. 26 (1997), 1733–1748.

[3] N. Alon and N. Kahale, *Approximating the independence number via the θ-function*, Math. Programming 80 (1998), 253-264.

[4] N. Alon and J. Spencer, **The Probabilistic Method**, Wiley, New York, 1992.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and intractability of approximation problems*, Proc. of the 33^{rd} IEEE FOCS, IEEE (1992), 14–23.

[6] S. Arora and S. Safra, *Probabilistic checking of proofs; a new characterization of NP*, Proc. of the 33^{rd} IEEE FOCS, IEEE (1992), 2–13.

[7] R. Boppana, *Eigenvalues and graph bisection: An average case analysis*, Proc. of the 28^{th} IEEE FOCS, IEEE (1987), 280–285.

[8] R. Boppana and M. M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, BIT, 32 (1992),180–196.

[9] U. Feige, S. Goldwasser, L. Lovász, S. Safra and M. Szegedy, *Approximating Clique is almost NP-complete*, Proc. of the 32^{nd} IEEE FOCS, IEEE (1991), 2–12.

[10] A. Frieze and C. McDiarmid, *Algorithmic theory of random graphs*, Random Structures and Algorithms 10 (1997), 5–42.

[11] Z. Füredi and J. Komlós, *The eigenvalues of random symmetric matrices*, Combinatorica 1 (1981), 233–241.

[12] G. Grimmett and C. McDiarmid, *On colouring random graphs*, Math. Proc. Cam. Phil. Soc. 77 (1975), 313–324.

[13] J. Håstad, *Clique is hard to approximate within* $n^{1-\epsilon}$, Proc. of the 37^{th} IEEE FOCS, IEEE (1996), 627–636.

[14] M. Jerrum, *Large cliques elude the metropolis process*, Random Structures and Algorithms 3 (1992), 347–359.

[15] A. Juels and M. Peinado, Hiding cliques for cryptographic security, *Proc. of the Ninth Annual ACM-SIAM SODA*, ACM Press (1998), 678–684.

[16] R. M. Karp, Reducibility among combinatorial problems, In: *Complexity of computer computations*, R. E. Miller and J. W. Thatcher (eds.), Plenum Press, New York, 1972, pp. 85–103.

[17] R. M. Karp, Probabilistic analysis of some combinatorial search problems, In: *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, ed., Academic Press, New York, 1976, pp. 1–19.

[18] L. Kučera, *Expected complexity of graph partitioning problems*, Discrete Applied Math. 57 (1995), 193–212.

[19] A. Ralston, **A First Course in Numerical Analysis**, McGraw-Hill, 1985, Section 10.4.

[20] A. A. Razborov, *Lower bounds for the monotone complexity of some Boolean functions*, Dokl. Ak. Nauk. SSSR 281 (1985), 798–801 (in Russian). English translation in: Sov. Math. Dokl. 31 (1985), 354–357.

[21] M. Saks, Private communication.

[22] D. A. Spielman and S.-H. Teng, *Spectral partitioning works: planar graphs and finite element meshes*, Proc. 37^{th} IEEE FOCS, IEEE (1996), 96–105.

[23] J. H. Wilkinson, **The Algebraic Eigenvalue Problem**, Clarendon Press, 1965.