Spectral Techniques in Graph Algorithms

Noga Alon *

Abstract

The existence of efficient algorithms to compute the eigenvectors and eigenvalues of graphs supplies a useful tool for the design of various graph algorithms.

In this survey we describe several algorithms based on spectral techniques focusing on their performance for randomly generated input graphs.

1 Introduction

Graph bisection, graph coloring and finding the independence number of a graph are three well studied algorithmic problems. All of them are NP-hard, and even the task of solving any of them approximately cannot be done in polynomial time under the common assumptions in Complexity Theory. It is possible, however, to develop efficient algorithms that solve these problems for almost all graphs in appropriately defined classes. Such algorithms are desirable, since all three problems arise often in practice, where one might hope that the input instances are not necessarily worst case examples. Spectral techniques, based on the eigenvalues and the eigenvectors of the adjacency or the Laplace matrices of graphs, appear to be very successful in the design of such algorithms, and can provably solve the above problems for various classes of randomly generated graphs, where all previous techniques failed. The analysis of the performance of algorithms for random graphs has gained popularity recently (see [25] and its many references), and it seems to provide a useful measure for the behaviour of algorithmic techniques. In this paper we describe the relevance of spectral techniques to the above mentioned problems, discuss the algorithms and study their performance. This is mostly a survey paper and hence the focus here is on the underlying ideas and not on the detailed proofs which can be found in the relevant references.

The *adjacency matrix* of a graph $G=(V,E)$ is the matrix $A=(a_{u,v})_{u,v \in V}$, in which $a_{u,v}=1$ if $uv \in E$ and $a_{u,v}=0$ otherwise. The *Laplace matrix* of G is $Q=D-A$, where $D=(d_{u,v})_{u,v \in V}$ is the

*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.

diagonal matrix in which $d_{u,u}$ is the degree $d(u)$ of u in G and $d_{u,v} = 0$ for all $u \neq v$. Both matrices above are symmetric and hence have real eigenvalues and an orthonormal basis of eigenvectors. The Laplace matrix is easily seen to be positive semi-definite and hence its eigenvalues are nonnegative. It is well known that there is a tight relation between the eigenvalues of A and Q and several structural properties of the graph G, and it is natural to expect that this fact, and the fact that all eigenvalues and eigenvectors can be computed efficiently (see, e.g., [43]), may be useful in the design of efficient algorithms. In the following sections we demonstrate this approach in the study of several algorithmic questions.

2 Expansion and bisection

The *vertex expansion* $c_V(G)$ of a graph $G = (V, E)$ on n vertices is

$$c_V(G) = \ min_{X \subset V, |X| \leq n/2} \frac{|N(X) - X|}{|X|},$$

where $N(X) = \{y \in v : \ xy \in E \ \text{ for some } \ x \in X\}$ is the set of all neighbors of X in G. The *edge expansion* $c_E(G)$ of G is defined by

$$c_E(G) = \ min_{X \subset V, |X| \leq n/2} \frac{e(X, V - X)}{|X|},$$

where $e(X, V - X)$ is the number of edges with one end in X and another end in its complement $V - X$. A *bisection* of $G = (V, E)$ is a partition of its set of vertices into two equal parts. The *size* of the bisection is the number of edges with one end in each part.

The problem of computing the vertex expansion or edge expansion of a graph and that of finding the minimum size of a bisection in it are useful for tackling various problems in VLSI design, and thus received a considerable amount of attention. The task of finding an optimal solution to any of these problems is NP-hard, and there is no known polynomial time algorithm that approximates any of these quantities up to a constant factor. In fact, it is known that if, as is widely believed, the complexity classes P and NP differ, then even the problem of approximating quantities related to the above three are NP-hard (see [12]).

Leighton and Rao [37] designed a polynomial time algorithm that approximates $c_E(G)$, for any n-vertex graph G, up to a multiplicative factor of $\log n$. The known results concerning the tight relation between the expansion properties of a graph and its spectral properties provide some (very rough) efficient approximation algorithms for vertex expansion and edge expansion, which are based on eigenvalue bounds. In particular, by the results in [20], [6] and [1], for every graph G with maximum degree d in which λ is the second smallest Laplace eigenvalue,

$$\Omega(\frac{\lambda}{d}) \leq c_V(G) \leq O(\sqrt{\lambda}).$$

Similarly, by the results of [20], [6] and [47], for every graph G as above

$$\frac{\lambda}{2} \leq c_E(G) \leq \sqrt{2d\lambda}.$$

Donath and Hoffman [18] were the first to suggest to use spectral techniques for graph partitioning. Their work, that of Fiedler [20], and substantial experimental work (see, e.g., [41], [46]) demonstrated that this is indeed a very good heuristics. See also [16] for some related results.

The basic idea in the Donath-Hoffman algorithm as well as in its more recent variants is that the eigenvector corresponding to the second smallest eigenvalue of the Laplace matrix of the graph provides some information that can be used to find a good partition of the graph. Indeed, in the extreme case that the graph consists of two connected components of equal size, this eigenvector is a constant on each component, and since we always may assume that the eigenvector of the smallest eigenvalue is the all 1 vector, and that the eigenvectors are orthogonal, this means that in the above extreme case the sign of the coordinates of the second eigenvector provides the desired partition. It is thus natural to expect that even in less trivial cases some information about a good bisection can be deduced from the coordinates of the second eigenvector.

Random graphs were initiated by Erdős and Rényi [19], and their extensive study (see, e.g., the comprehensive book of Bollobás [13] and its many references) motivated the investigation of the performance of heuristic algorithms for input graphs generated randomly. Boppana [15] showed that a variant of the basic spectral technique finds, with high probability, the minimum bisection in a *random* graph with n vertices, m edges and bisection of size b, provided $0 \leq b \leq m/2 - 5\sqrt{mn \log n}$. Therefore, these techniques work provably well on appropriate randomly generated input graphs.

More recently, Spielman and Teng showed in [48] that by partitioning the vertices of a planar, bounded-degree graph on n vertices according to the coordinates of the eigenvector of its second smallest Laplace eigenvalue, one obtains a cut for which the ratio between the number of edges and the number of vertices in the smaller side is $O(1/\sqrt{n})$. They also obtained similar results for other classes of graphs. It is well known that every bounded degree planar graph has such a separator, by the Lipton Tarjan separator Theorem [38], which also provides a linear time algorithm for finding such a cut. Although the results of [48] do not supply a better algorithm for planar graphs, they do provide insight for the behaviour of spectral techniques in partitioning algorithms and show that in some cases these techniques are provably useful.

3 Coloring

The *chromatic number* $\chi(G)$ of a graph G is the minimum number of colors needed to color the vertices of G so that adjacent vertices have distinct colors.

The problem of determining or estimating this parameter has received a considerable amount of attention in Combinatorics and in Theoretical Computer Science, as several scheduling problems are naturally formulated as graph coloring problems. It is well known (see [30, 26]) that the problem of properly coloring a graph of chromatic number k with k colors is NP-hard, even for any fixed $k \geq 3$. Moreover, even the problem of approximating the chromatic number of an n vertex graph up to an n^c multiplicative factor, for an appropriate positive c, is NP-hard, as shown by Lund and Yannakakis [39]. In fact, if NP does not have efficient *randomized* algorithms, then there is no polynomial time algorithm for approximating the chromatic number of an n vertex graph up to a factor of $n^{1-\epsilon}$, for any fixed $\epsilon > 0$, as proved in [23], using the result of [27]. In addition, from the results in [2] it follows that the chromatic number of an n vertex graph cannot be approximated up to a factor of $n/(\log n)^7$ by a *monotone* polynomial size circuit.

Several sophisticated polynomial time algorithms provide some very rough estimates for the chromatic number of a graph. Despite a lot of efforts (see [34] and its references) there is no known polynomial time algorithm that finds a proper coloring of a 3-colorable graph on n vertices by less than, say, $n^{1/5}$ colors.

There are, however, several results that indicate that the sectral properties of a graph provide some information on its chromatic number. If G is a graph on n vertices with (adjacency matrix) eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ then, as proved by Hoffman [28], $\chi(G) \geq 1 - \frac{\lambda_1}{\lambda_n}$. Similarly, as proved by Wilf (cf., e.g., [36]), $\chi(G) \leq 1 + \lambda_1$.

Despite the difficulties in coloring graphs efficiently in the worst case, various researchers noticed that *random* k-colorable graphs are usually easy to color optimally. Polynomial time algorithms that optimally color random k-colorable graphs for every fixed k with high probability, have been developed by Kučera [32], by Turner [49] and by Dyer and Frieze [17], where the last paper provides an algorithm whose average running time over all k-colorable graphs on n vertices is polynomial. Note, however, that most k-colorable graphs are quite dense, and hence easy to color. In fact, in a typical k-colorable graph, the number of common neighbors of any pair of vertices with the same color exceeds considerably that of any pair of vertices of distinct colors, and hence a simple coloring algorithm based on this fact already works with high probability. It is more difficult to color sparser random k-colorable graphs. A precise model for generating sparse random k-colorable graphs is described next, and the sparsity in it is governed by a parameter p that specifies the edge probability. There are, in fact, several possible models, but since most of them are equivalent for our purpose here we focus on one. Here is its description.

Let V be a fixed set of kn labelled vertices. For a real $p = p(n)$, let $G_{kn,p,k}$ be the random graph on the set of vertices V obtained as follows; first, split the vertices of V arbitrarily into k color classes $W_1, \ldots, W_k$, each of cardinality n. Next, for each u and v that lie in distinct color classes, choose uv

to be an edge, randomly and independently, with probability p. The input to the coloring algorithm is now a graph $G_{kn,p,k}$ obtained as above, and the algorithm succeeds to color it if it finds a proper k coloring. The interesting case is a fixed value of $k \geq 3$ and large n, and since the case $k = 3$ is similar to the more general one of arbitrarily fixed k we discuss mainly this special case. We say that an algorithm colors $G_{kn,p,k}$ *almost surely* if the probability that a randomly chosen graph as above is properly colored by the algorithm tends to one as n tends to infinity.

Petford and Welsh [42] suggested a randomized heuristic for 3-coloring random 3-colorable graphs and supplied experimental evidence that it works for most edge probabilities. Blum and Spencer [14] (see also [11] for some related results) designed a polynomial algorithm and proved that it colors optimally, with high probability, random 3-colorable graphs on n vertices with edge probability p provided $p \geq n^{\epsilon}/n$, for some arbitrarily small but fixed $\epsilon > 0$. Their algorithm is based on a path counting technique, and can be viewed as a natural generalization of the simple algorithm based on counting common neighbors (that counts paths of length 2), mentioned above.

In [3] the authors designed a polynomial time algorithm that works for sparser random 3-colorable graphs, which is based on spectral techniques. If the edge probability p satisfies $p \geq c/n$, where c is a sufficiently large absolute constant, the algorithm colors optimally the corresponding random 3-colorable graph with high probability. This settled a problem of Blum and Spencer [14], who asked if one can design an algorithm that works almost surely for $p \geq \text{polylog}(n)/n$. (Here, and in what follows, *almost surely* always means: with probability that approaches 1 as n tends to infinity). The algorithm is based on the fact that almost surely a rather accurate approximation of the color classes can be read from the eigenvectors corresponding to the smallest two eigenvalues of the adjacency matrix of a large subgraph. This approximation can then be improved to yield a proper coloring.

As is the case with the bisection heuristics, the intuition here is rather simple. If, in the perfectly symmetric case, there is a constant d such that every vertex has precisely d neighbors in each of the three color classes but its own, then $-d$ is an eigenvector of the adjacency matrix of the graph and the corresponding two dimensional eigenspace consists of all vectors whose sum of coordinates is 0 that attain the same value on each color class. If, in addition, the other eigenvalues of the graph behave like those of random regular graphs (see [24]), then $-d$ is the smallest eigenvalue. Therefore, even if the situation is not that symmetric, enough information about the color classes can still be deduced from the two eigenvectors of the smallest eigenvalue, and with some efforts this information can be used to obtain a full proper 3-coloring.

The algorithm can be easily extended to the case of k-colorable graphs, for any fixed k, and to various models of random *regular* 3-colorable graphs.

4 Cliques and independent sets

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 [30]. The best known approximation algorithm for this quantity, designed by Boppana and Halldórsson [10], 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 [4], 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.

On the negative side, it is known, by the work of [8] following [21] and [9], that for some $\epsilon > 0$ it is impossible to approximate $w(G)$ in polynomial time for a graph on n vertices within a factor of n^ϵ, assuming P $\neq$ NP. The exponent ϵ has since been improved in various papers and recently it has been shown by Håstad [27] 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 [2] following [44], 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 appropriate classes of randomly generated input graphs.

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 $1/2$. By the results in ([40]) improved by several researchers, 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., [7]). Several simple polynomial time algorithms 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 [31]. His results, as well as more recent ones of Jerrum [29] implied that several natural algorithms do not achieve this goal and it seems plausible to conjecture (see [29]) that in fact there is no polynomial time algorithm that finds, with probability more than a half, say, a clique of size bigger than $(1+\epsilon) \log_2 n$.

The situation may become better in a random model in which the biggest clique is larger. Following [29], 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 [33], 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 one design an algorithm that finds the biggest clique almost surely if k is $o(\sqrt{n \log n})$? This problem was mentioned in [33], and has recently been solved in [5] by showing that for every $\epsilon > 0$ there is a polynomial time algorithm, based on spectral techniques, 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, it seems that the spectral technique is crucial for the algorithm.

The relevance of graph eigenvalues to cliques or independent sets in the graphs is well known and can be traced back to the old result that the independence number of any regular graph is at most $-n\lambda_n/(\lambda_1 - \lambda_n)$ and the related results on the connection between the Shannon capacity of a graph and its eigenvalues (see [35]).

The spectral algorithm of [5] is based on the fact that in the random model considered above one can almost surely extract a big portion of the hidden clique using the eigenvector of the second largest eigenvalue of the adjacency matrix of the graph, provided k is at least, say, $10\sqrt{n}$. Using this portion, it is not too difficult to recover the whole clique. Using some extra tricks this can be extended to yield an algorithm for $k = \epsilon\sqrt{n}$ as well.

The basic idea behind the algorithm is that the largest (adjacency matrix) eigenvalue of a random graph in the model $G(n, 1/2, k)$ is likely to be $(\frac{1}{2} + o(1))n$, all its eigenvalues besides the largest **two** are likely to be at most $(1 + o(1))\sqrt{n}$, by a result of Füredi and Komlós [22], and the second largest eigenvalue is likely to be close to $k/2$, where the corresponding eigenvector is nearly a constant on all vertices of the largest clique and nearly another one on all other vertices.

With some efforts this can be formalized and proved (for $k > 10\sqrt{n}$), supplying the desired algorithm. More details appear in [5].

5 Open problems

It would be interesting to extend the spectral algorithms for coloring and for finding the biggest clique to a wider class of randomly generated graphs. In particular, it would be interesting to design a coloring algorithm that finds, almost surely, a proper three-coloring of $G_{3n,p,3}$ for all possible values of p. (Note that the spectral algorithm works for all $p \geq C/n$ where C is a large absolute constant, whereas a trivial algorithm based on omitting repeatedly vertices of low degree works for $p \leq c/n$ if c is an absolute (small) positive constant. Hence, only the case $c/n \leq p \leq C/n$ remains.)

Similarly, the obvious challenge concerning the problem of finding a large clique in input graphs generated according to the distribution $G(n, 1/2, k)$ 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 [29], is open and seems to require new ideas.

Another interesting version of this problem was suggested by Saks [45]. 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 the algorithm in [5].)

The spectral properties of a graph encode some detailed structural information about 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 and may well have additional algorithmic applications in the future too.

References

[1] N. Alon, Eigenvalues and expanders, *Combinatorica* 6 (1986), 83-96.

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

[3] 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., in press.

[4] N. Alon and N. Kahale, Approximating the independence number via the θ-function, Math. Programming, in press.

[5] N. Alon, M. Krivelevich and B. Sudakov, Finding a large hidden clique in a random graph, Proc. of the Ninth Annual ACM-SIAM SODA, ACM Press (1998), to appear.

[6] N. Alon and V. D. Milman, Eigenvalues, expanders and superconcentrators, *Proc.* 25^{th} *Annual Symp. on Foundations of Computer Science*, Singer Island, Florida, IEEE (1984), 320-322. (Also: λ_1, isoperimetric inequalities for graphs and superconcentrators, *J. Combinatorial Theory, Ser. B* 38 (1985), 73-88.)

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

[8] 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.

[9] 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.

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

[11] A. Blum, *Some tools for approximate 3-coloring*, Proc. 31^{st} IEEE FOCS, IEEE (1990), 554–562.

[12] T.N. Bui and C. Jones, *Finding good approximate vertex and edge partitions is NP-hard*, Infor. Proc. Letters 42 (1992), 153–159.

[13] B. Bollobás, *Random Graphs*, Academic Press, London, 1985.

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

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

[16] F. Chung and S. T. Yau, *Eigenvalues, flows and separators of graphs*, to appear.

[17] M. E. Dyer and A. M. Frieze, *The solution of some random NP-Hard problems in polynomial expected time*, Journal of Algorithms 10 (1989), 451–489.

[18] W. E. Donath and A. J. Hoffman, Lower bounds for the partitioning of graphs, *J. Res. Develop.* 17 (1973), 420-425.

[19] P. Erdős and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960), 17-61.

[20] M. Fiedler, Algebraic connectivity of graphs, *Czechoslovak Math. J.* 23(98) (1973), 298-305.

[21] 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.

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

[23] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, Proc. 11^{th} Annual IEEE Conf. on Computational Complexity, 1996.

[24] J. Friedman, J. Kahn and E. Szemerédi, *On the second eigenvalue in random regular graphs*, Proc. 21^{st} ACM STOC, ACM Press (1989), 587–598.

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

[26] M. R. Garey and D. S. Johnson, **Computers and intractability: a guide to the theory of NP–completeness**, Freeman and Company, 1979.

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

[28] A. J. Hoffman, On eigenvalues and colorings of graphs, in: B. Harris Ed., *Graph Theory and its Applications*, Academic, New York and London, 1970, 79-91.

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

[30] 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.

[31] 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.

[32] L. Kučera, *Expected behavior of graph colouring algorithms*, In *Lecture Notes in Computer Science No. 56*, Springer-Verlag, 1977, pp. 447–451.

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

[34] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semi-definite programming, In 35*th Symposium on Foundations of Computer Science*, pages 2–13. IEEE Computer Society Press, 1994.

[35] L. Lovász, *On the Shannon capacity of a graph*, IEEE Transactions on Information Theory IT-25, (1979), 1-7.

[36] L. Lovász, **Combinatorial Problems and Exercises**, North Holland, Amsterdam, 1979, Chapter 11.

[37] F. T. Leighton and S. Rao, An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms, Proc 29^{th} annual FOCS (1988), 422-431.

[38] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math. 36(1979), 177–189.

[39] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems.* Proc. 25^{th} ACM STOC (1993), 286–293.

[40] D. W. Matula, On the complete subgraph of a random graph, *Combinatory Mathematics and its Applications*, Chapel Hill, North Carolina (1970), 356-369.

[41] A. Pothem, H. D. Simon and K.-P. Liou, Partitioning sparse matrices with eigenvectors of graphs, *SIAM J. Matrix Anal. Appl.* 11 (1990), 430-452.

[42] A. D. Petford and D. J. A. Welsh, *A randomised 3-colouring algorithm*, Discrete Mathematics, 74 (1989), 253–261.

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

[44] 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.

[45] M. Saks, Private communication.

[46] H. D. Simon, Partitioning of unstructured problems for parallel processing, *Computing Systmes in Engineering* 2(2/3) (1991), 135-148.

[47] A. Sinclair and M. R. Jerrum, Approximate counting, uniform generation and rapidly mixing Markov chains, *Information and Computation* 82 (1989), 93-133.

[48] D. A. Spielman and S.-H. Teng, Spectral partitioning works: planar graphs and finite element meshes, to appear.

[49] J. S. Turner, *Almost all k-colorable graphs are easy to color*, Journal of Algorithms 9 (1988), 63–82.