Approximating the independence number and the chromatic number in expected polynomial time

Michael Krivelevich[1] and Van H. Vu[2]

[1] Department of Mathematics, Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@math.tau.ac.il
[2] Microsoft Research, 1 Microsoft Way, Redmond, WA 98052, USA. E-mail: vanhavu@microsoft.com.

Abstract. The independence number of a graph and its chromatic number are hard to approximate. It is known that, unless $coRP = NP$, there is no polynomial time algorithm which approximates any of these quantities within a factor of $n^{1-\epsilon}$ for graphs on n vertices.
We show that the situation is significantly better for the average case. For every edge probability $p = p(n)$ in the range $n^{-1/2+\epsilon} \leq p \leq 3/4$, we present an approximation algorithm for the independence number of graphs on n vertices, whose approximation ratio is $O((np)^{1/2}/\log n)$ and whose expected running time over the probability space $G(n,p)$ is polynomial. An algorithm with similar features is described also for the chromatic number.
A key ingredient in the analysis of both algorithms is a new large deviation inequality for eigenvalues of random matrices, obtained through an application of Talagrand's inequality.

1 Introduction

An *independent set* in a graph $G = (V, E)$ is a subset of vertices spanning no edges. The *independence number* of G, denoted by $\alpha(G)$, is the size of a largest independent set in G. A *coloring* of G is a partition $V = C_1 \dots C_k$ of its vertex set V, in which every part (color class) C_i forms an independent set. The *chromatic number* $\chi(G)$ of G is the minimal possible number of colors in a coloring of G.

Independence number and chromatic number are essential notions in combinatorics and the problem of estimating these parameters is central in both graph theory and theoretical computer science. Unfortunately, it turns out that both of these problems are notoriously difficult. Computing the exact value of $\alpha(G)$ or $\chi(G)$ is known to be NP-hard since the seminal paper of Karp [16].

Given these hardness results, one still hopes to approximate the above parameters within a reasonable factor in polynomial time. For a number $f \geq 1$, we say that an algorithm A approximates the independence number within factor f over graphs on n vertices, if for every such graph G A outputs an independent set I, whose size satisfies $|I| \geq \alpha(G)/f$. Similarly, A approximates the chromatic

number within factor f for graphs on n vertices if for every such graph it outputs a coloring with the number of colors k satisfying $k \leq f\chi(G)$. We refer the reader to a survey book [15] for a detailed discussion of approximation algorithms.

However, recent results have shown that in the worst case both computational problems are also hard to approximate. Håstad [14] showed that unless $coPR = NP$, there is no approximation algorithm for the independence number whose approximation ratio over graphs on n vertices is less than $n^{1-\epsilon}$ for any fixed $\epsilon > 0$. Also, Feige and Kilian proved in [9] the same hardness result for the chromatic number. In this paper we aim to show that the situation is significantly better when one considers the average case and not the worst case. When discussing the performance of an algorithm A in the average case, it is usually assumed that a probability distribution on the set of all inputs of A is defined. The most widely used probability measure on the set of all graphs on n vertices is the random graph $G(n,p)$. For an integer n and a function $0 \leq p = p(n) \leq 1$, the *random graph* $G(n,p)$ is a graph on n labeled vertices $1, \ldots, n$, where each pair of vertices (i,j) is chosen to be an edge of G independently and with probability p. We say that a graph property P holds *almost surely*, or a.s. for brevity, in $G(n,p)$ if the probability that a graph G, drawn according to the distribution $G(n,p)$, has P tends to 1 as the number of vertices n tends to infinity.

As with many other graph parameters, it turns out that the average case is much simpler to handle than the worst case for both independence and chromatic numbers. Bollobás [5] and Łuczak [19] showed that a. s. the chromatic number of $G(n,p)$ satisfies $\chi(G) = (1+o(1))n\log_2(1/(1-p))/\log_2 n$ for a constant p, and $\chi(G) = (1+o(1))np/(2\ln(np))$ for $C/n \leq p(n) \leq o(1)$. It follows easily from these results that a.s. $\alpha(G(n,p)) = (1-o(1))\log_2 n/\log_2(1/(1-p))$ for a constant p, and $\alpha(G(n,p) = (1-o(1))2\ln(np)/p$ for $C/n \leq p \leq o(1)$. Also, the greedy algorithm, coloring vertices of G one by one and picking each time the first available color for a current vertex, is known to produce a.s. in $G(n,p)$ with $p \geq n^{\epsilon-1}$ a coloring whose number of colors is larger than the optimal one by only a constant factor (see Ch. 11 of the monograph of Bollobás [4]). Hence the largest color class produced by the greedy algorithm is a.s. smaller than the independence number only by a constant factor.

Note however that the above positive statement about the performance of the greedy algorithm hides in fact one quite significant point. While being very successful in approximating the independence/chromatic number for *most* of the graphs in $G(n,p)$, the greedy algorithm may fail miserably for some "hard" graphs on n vertices. It is quite easy to fool the greedy algorithm by constructing an example of a graph on n vertices, for which the ratio of the number of colors used by the greedy algorithm and the chromatic number will be close to n. Moreover, it has been shown by Kučera [17] that for any fixed $\epsilon > 0$ there exists a graph G on n vertices for which, even after a random permutation of vertices, the greedy algorithm produces a. s. a coloring in at least $n/\log_2 n$ colors, while the chromatic number of G is at most n^{ϵ}. Thus, we cannot say that the greedy algorithm is *always* successful.

In contrast, here our goal is to develop approximation algorithms which will work relatively well for *all* graphs on n vertices and whose *expected* running time will be polynomial in n. Given an algorithm A, whose domain is the set of all graphs on n vertices, and a probability space $G(n,p)$, the *expected running time* of A over $G(n,p)$ is defined as $\sum_G Pr[G]R_A(G)$, where the sum runs over all labeled graphs on n vertices, $Pr[G]$ is the probability of G in $G(n,p)$, and $R_A(G)$ stands for the running time of A on G. Thus, while looking for an algorithm A whose expected running time is polynomial, we can allow A to spend a superpolynomial time on some graphs on n vertices, but it should be effective on average.

The approach of devising deterministic algorithms with expected polynomial time over some probability spaces has been undertaken by quite a few papers. Some of them, e.g., [8], [20], [12] discuss coloring algorithms with expected polynomial time. We wish to stress that the underlying probability spaces in all these papers are different from $G(n,p)$.

In this paper we present approximation algorithms for the independence number and the chromatic number, whose expected running time is polynomial over the probability spaces $G(n,p)$, when the edge probability $p(n)$ is not too low. We have the following results.

Theorem 1. *For any constant $\epsilon > 0$ the following holds. If the edge probability $p(n)$ satisfies $n^{-1/2+\epsilon} \leq p(n) \leq 3/4$, then there exists a deterministic algorithm, approximating the independence number $\alpha(G)$ within a factor $O((np)^{1/2}/\log n)$ and having polynomial expected running time over $G(n,p)$.*

Theorem 2. *For any constant $\epsilon > 0$ the following holds. If the edge probability $p(n)$ satisfies $n^{-1/2+\epsilon} \leq p(n) \leq 3/4$, then there exists a deterministic algorithm, approximating the chromatic number $\chi(G)$ within a factor $O((np)^{1/2}/\log n)$ and having polynomial expected running time over $G(n,p)$.*

Thus, in the most basic case $p = 1/2$ we get approximation algorithms with approximation ratio $O(n^{1/2}/\log n)$ – a considerable improvement over best known algorithms for the worst case [7], [13], whose approximation ratio is only $O(n/polylog(n))$. Note also that the smaller the edge probability $p(n)$, the better the approximation ratio is in both our results.

Before turning to descriptions of our algorithms, we would like to say a few words about combinatorial ideas forming the basis of their analysis. As is typically the case with developing algorithms whose expected running time is polynomial, we will need to distinguish efficiently between "typical" graphs in the probability space $G(n,p)$, for which it is relatively easy to provide a good approximation algorithm, and "non-typical" ones, which are rare but may be hard for approximating a desired quantity. As these rare graphs will have an exponentially small probability in $G(n,p)$, this will allow us to spend an exponential time on each of them. This in turn will enable to approximate the independence/chromatic number within the desired factor even for these graphs.

A separation between typical and non-typical instances will be made based on the first eigenvalue of an auxiliary matrix, to be defined later. Thus we may

say that our algorithms exploit spectral properties of random graphs. Spectral techniques have proven very successful in many combinatorial algorithms. The ability to compute eigenvalues and eigenvectors of a matrix in polynomial time combined with understanding of the information provided by these parameters can constitute a very powerful tool, capable of solving algorithmic problems where all other methods failed. This is especially true for randomly generated graphs, several successful examples of spectral techniques are [6], [2], [3]. A survey [1] discusses several applications of spectral techniques to graph algorithms.

In order to show that bad graphs have an exponentially small probability in $G(n,p)$, we will prove a new large deviation result for eigenvalues of random symmetric matrices. This result, bounding the tails of the distribution of the first eigenvalue of a random symmetric matrix, is proven by applying the inequality of Talagrand [22] and may be of an independent interest.

The rest of the paper is organized as follows. In Section 2 we provide technical tools to be used in the the proof of correctness of our algorithms. In Section 3 we present an algorithm for approximating the independence number. In Section 4 an algorithm for approximating the chromatic number is described. Section 5 is devoted to concluding remarks.

We wish to note that our results are asymptotic in nature. Therefore all usual asymptotic assumptions apply. In particular, we assume n to be large enough whenever needed. We omit routinely all ceiling and floor signs. No serious attempt is made to optimize constants involved.

2 Preliminaries

In this section we prove technical results needed for the analysis of our approximation algorithms, to be proposed in the next two sections. In the first subsection we analyze the performance of the greedy algorithm on random graphs, the second subsection is devoted to bounding the tails of the first eigenvalue of a random matrix.

2.1 Greedy algorithm on random graphs

Given a graph $G = (V, E)$ and some fixed ordering of its vertices, the greedy coloring algorithm proceeds by scanning vertices of G in the given order and assigning the first available color for a current vertex. The greedy algorithm has long been known to be a quite successful algorithm for almost all graphs in $G(n,p)$, if the edge probability p is not too small. For our purposes, we need to prove that it is also extremely robust, i.e., uses an optimal up to a constant factor number of colors with probability extremely close to 1. We will also prove that the largest color class of the output of the greedy algorithm is of order of the independence number with even higher probability. Throughout this section we assume that the $p(n)$ falls in the range of Theorems 1-2, i.e., satisfies $n^{-1/2+\epsilon} \leq p(n) \leq 3/4$ for some positive constant ϵ. We fix the natural order of the n vertices, i.e., $1, 2, \ldots, n$.

Lemma 1. *The probability in $G(n,p)$ that the largest color class, produced by the greedy algorithm, has size less than $\ln n/(2p)$, is less than 2^{-n}.*

Proof. Set $\alpha_0 = \frac{\ln n}{2p}$, $t = \frac{n}{2\alpha_0}$. We call a family $\mathcal{C} = \{C_1, \ldots, C_t\}$ of t subsets of V *bad* if

1. All C_i are pairwise disjoint;

2. For every vertex $v \in V \setminus \bigcup_{i=1}^{t} C_i$ and for every $1 \leq i \leq t$, there is an edge of G connecting v and C_i;

3. For every $1 \leq i \leq t$, $|C_i| < \alpha_0$.

It is easy to see that if $C_1, \ldots, C_t$ are the first t colors produced by the greedy algorithm, then the family $\mathcal{C} = \{C_1, \ldots, C_t\}$ satisfies requirements 1, 2 above. Thus, if the greedy algorithm fails to produce a color class of size at least α_0 while running on a graph G, the first t colors of its output form a bad family in G.

Fix a collection $\mathcal{C} = \{C_1, \ldots, C_t\}$ with all C_i being pairwise disjoint and of size $|C_i| < \alpha_0$. A vertex $v \in V \setminus \bigcup_{i=1}^{t} C_i$ is *stuck* with respect to $\mathcal{C}$ if it satisfies condition 2 above. The probability that v is stuck is

$$\prod_{i=1}^{t}(1-(1-p)^{|C_i|}) \leq \exp\{-\sum_{i=1}^{t}(1-p)^{|C_i|}\} \leq e^{-t(1-p)^{\alpha_0}}.$$

As the events that different vertices outside $\mathcal{C}$ are stuck are mutually independent, we get

$$Pr[\mathcal{C} \text{ bad}] \leq \exp\{-t(1-p)^{\alpha_0}|V \setminus \bigcup_{i=1}^{t} C_i|\} \leq e^{-t(1-p)^{\alpha_0} n/2} = (1+o(1))e^{-tn^{1/2}/2}.$$

Therefore the probability that $G(n,p)$ contains a bad collection is at most

$$\left(\sum_{i=1}^{\alpha_0-1}\binom{n}{i}\right)^t (1+o(1))e^{-tn^{1/2}/2} \leq (1+o(1))\binom{n}{\alpha_0}^t e^{-tn^{1/2}/2}$$

$$\leq n^n e^{-tn^{1/2}/2} \leq n^n e^{-\frac{n^{3/2}p}{2\ln n}} < 2^{-n}. \qquad \square$$

Lemma 2. *The probability in $G(n,p)$ that the greedy algorithm uses at least $4np/\ln n$ colors it at most $2^{-2np/\ln n}$.*

Proof. The proof presented here is essentially identical to the proof of Theorem 11.14 of the monograph of Bollobás [4], where a somewhat stronger result is proven for the case of a constant p.

Set $k_0 = \frac{2np}{\ln n}$,$k_1 = 2k_0 = \frac{4np}{\ln n}$. Denote by A^k the event that at least k colors are used by the greedy algorithm in coloring $G(n,p)$. Let also, for $k \leq j \leq n$, B_j^k denote the event that vertex j gets color k. As obviously $A^{k+1} = \cup_{j=k+1}^{n} B_j^{k+1}$, we get for all $k_0 \leq k \leq k_1$

$$Pr[A^{k+1}|A^k] \leq \sum_{j=k+1}^{n} Pr[B_j^{k+1}|A^k].$$

Let us estimate now $Pr[B_j^{k+1}|A^k]$ for $j \geq k+1$. Suppose $C_1, \ldots, C_k$ are the color classes produced by the greedy algorithm before coloring vertex j. Then

$$Pr[B_j^{k+1}|A^k] = \prod_{i=1}^{k}(1-(1-p)^{|C_i|}) \leq (1-(1-p)^{j/k})^k \leq e^{-(1-p)^{j/k}k} \leq e^{-(1-p)^{n/k}k} .$$

Hence $Pr[A^{k+1}|A^k] \leq ne^{-(1-p)^{n/k_0}k_0} = ne^{-(1-p)^{\frac{\ln n}{2p}}k_0} = (1+o(1))ne^{-n^{-1/2}k_0} \leq \frac{1}{2}$. We derive

$$Pr[A^{k_1}] \leq \prod_{k=k_0}^{k_1-1} Pr[A^{k+1}|A^k] \leq \left(\frac{1}{2}\right)^{k_1-k_0} = 2^{-k_0} . \qquad \Box$$

2.2 Large deviation result

In this subsection we present a new large deviation result, which will be needed in the analysis of the algorithms. This result is also of independent interest. The proof in this subsection will make use of the following powerful result, due to Talagrand [22].

Let $t_1, \ldots, t_m$ be independent random variables and let $\mathcal{S}$ be the product space (with the product measure) generated by $t_1, \ldots, t_m$. Fix a set $\mathcal{B} \subset \mathcal{S}$. For a non-negative number t, define $\mathcal{B}_t$ as follows

$$\mathcal{B}_t = \{x \in \mathcal{S} | \forall \alpha = (\alpha_1, \ldots, \alpha_m), \exists y \in \mathcal{B} \text{ s.t. } \sum_{x_i \neq y_i} |\alpha_i| \leq t(\sum_{i=1}^{n} \alpha_i^2)^{1/2}\}.$$

Then Talagrand's inequality gives:

$$Pr[\overline{\mathcal{B}_t}]Pr[\mathcal{B}] \leq e^{-t^2/4} .$$

Given a graph G on n vertices and a number $0 < p < 1$, we define a matrix $M = M(G,p) = (m_{ij})_{i,j=1}^n$ as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } i,j \text{ are non-adjacent in } G, \\ -q/p, & \text{otherwise} , \end{cases} \tag{1}$$

where $q = 1-p$. Let $\lambda_1(M) \geq \lambda_2(M) \geq \ldots \geq \lambda_n(M)$ denote the eigenvalues of M.

Lemma 3.

$$Pr[\lambda_1(M) \geq 4(n/p)^{1/2}] \leq 2^{-np/8} .$$

Proof. First notice that $M(G,p)$ is a random symmetric matrix which can be generated as follows. Consider $\binom{n}{2}$ random variables m_{ij}, $1 \leq i < j \leq n$, where $m_{ij} = 1$ with probability q, and $-q/p$ with probability p. Set $m_{ji} = m_{ij}$ and

$m_{ii} = 1$. This observation enables us to apply known results on eigenvalues of random matrices. Füredi and Komlós proved implicitly in [11] that

$$E[\lambda_1(M)] = 2\left(\frac{qn}{p}\right)^{1/2}(1+o(1)) \ .$$

Thus we need to bound the probability of deviation of the first eigenvalue from its mean. Denote by m the median of $\lambda_1(M)$ and set $\mathcal{B}$ to be the set of all matrices M with $\lambda_1(M) \leq m$. Clearly, $Pr[\mathcal{B}] = 1/2$. Assume that for a positive t, a matrix M^0 satisfies $\lambda_1(M^0) \geq m + t$. Then by Courant-Fisher's theorem, there is a vector $\mathbf{x} = (x_1, \ldots, x_n) \in R^n$ with norm 1 such that

$$m + t \leq \mathbf{x}^t M^0 \mathbf{x} = \sum_{1 \leq i < j \leq n} 2x_i x_j m^0_{ij} + \sum_{i=1}^{n} x_{ii}^2 m^0_{ii} = 1 + \sum_{1 \leq i < j \leq n} 2x_i x_j m^0_{ij} \ .$$

On the other hand, for any matrix $M^1 \in \mathcal{B}$ we have

$$m \geq \mathbf{x}^t M^1 \mathbf{x} = \sum_{1 \leq i < j \leq n} 2x_i x_j m^1_{ij} + \sum_{i=1}^{n} x_{ii}^2 m^1_{ii} = 1 + \sum_{1 \leq i < j \leq n} 2x_i x_j m^1_{ij} \ .$$

It follows that

$$\sum_{1 \leq i < j \leq n} 2x_i x_j (m^0_{ij} - m^1_{ij}) \geq t \ .$$

Set $\alpha_{ij} = 2x_i x_j$ for $1 \leq i < j \leq n$. Since $\mathbf{x}$ has norm 1, we get $\sum_{1 \leq i < j \leq n} \alpha_{ij}^2 \leq 2(\sum_{i=1}^{n} x_i^2)^2 = 2$. Moreover, since $|m^0_{ij} - m^1_{ij}| \leq 1 + q/p = 1/p$,

$$\sum_{ij, m^0_{ij} \neq m^1_{ij}} |\alpha_{ij}| \geq tp \geq \frac{tp}{\sqrt{2}}\Big(\sum_{1 \leq i \leq j \leq n} \alpha_{ij}^2\Big)^{1/2} \ .$$

This implies that $M^0 \in \overline{\mathcal{B}_{tp/\sqrt{2}}}$. By Talagrand's inequality

$$Pr[\lambda_1(M) \geq m + t] \leq Pr[\overline{\mathcal{B}_{tp/\sqrt{2}}}] \leq \frac{1}{Pr[\mathcal{B}] e^{(tp)^2/8}} \ .$$

Given that $Pr[\mathcal{B}] = 1/2$, it follows that

$$Pr[\lambda_1(M) \geq m + t] \leq 2e^{-(tp)^2/8}.$$

Now set $\mathcal{B} = \{M | \lambda_1(M) \leq m - t\}$. By a similar argument, we can show that if $\lambda_1(M^0) \geq m$, then $M^0 \in \overline{\mathcal{B}_{tp/\sqrt{2}}}$. This, again by Talagrand's inequality, yields

$$Pr[\lambda_1(M) \leq m - t] \leq 2e^{-(tp)^2/8}.$$

Together, we have

$$Pr[|\lambda_1(M) - m| \geq t] \leq 4e^{-(tp)^2/8} \ ,$$

or equivalently

$$Pr[|\lambda_1(M) - m| \geq \sqrt{8}t/p] \leq 4e^{-t^2} \ ,$$

The problem here is that the median m is not exactly the mean. However, from what we have already proved, it is simple to show that they differ only by $O(1/p)$. Indeed, let $X = \lambda_1(M)$, $Y = pX$, let also $m_1 = pm$ be the median of Y. Then we have $Pr[|m_1) - Y| \geq t] \leq 4e^{-t^2/8}$. Therefore,

$$|m_1 - E[Y]| \leq E[|m_1 - Y|] \leq \int_0^\infty Pr[|Y - m_1| \geq t]dt \leq \int_0^\infty 4e^{-t^2/8}dt = 4\sqrt{2\pi} \ ,$$

implying $|E[X] - m| \leq O(1/p)$. Recalling our assumption about $p(n)$, the claim of the lemma follows directly. □

What is the connection between $\lambda_1(M(G))$ and $\alpha(G)$? The answer is given by the following simple lemma.

Lemma 4. *Let $M = M(G,p)$ be as defined in (1). Then $\lambda_1(M) \geq \alpha(G)$.*

Proof. Let $k = \alpha(G)$. Then M contains a k by k block of all 1's, indexed by the vertices of an independent set of size k. It follows from interlacing that $\lambda_1(M) \geq \lambda_1(1_{k\times k}) = k$. □

The reader has possibly noticed that $\lambda_1(M(G))$ is an upper bound not only for the independence number of G, but for its Lovász Theta-function ([18]). Therefore our Lemma 3 provides in fact a large deviation result also for the Theta-function. We get the following bound:

Lemma 5. *Let $p = p(n)$ satisfy $p(n) = \omega(1)/n$. Then in $G(n,p)$*

$$Pr[\alpha(G) \geq 4(n/p)^{1/2}] \leq Pr[\theta(G) \leq 4(n/p)^{1/2}] \leq 2^{-np/8} \ .$$

3 Approximating the independence number

We are now in position to present both of our approximation algorithms. In this section we describe an algorithm for approximating the independence number, while an algorithm for the chromatic number is described in the next section. We assume that the algorithm is given a graph G on n vertices and the value of the edge probability $p(n)$. For a subset $W \subset V$ we denote $\overline{N}(W) = \{v \in V \setminus W : \forall w \in W, (v,w) \notin E(G)\}$.

Step 1. Run the greedy algorithm on G. Let I be a largest color class produced by the greedy algorithm. If $|I| < \ln n/(2p)$, goto *Step 5*;
Step 2. Define matrix $M = M(G,p)$ as given by (1). Compute $\lambda_1(M)$. If $\lambda_1(M) \leq 4(n/p)^{1/2}$, output I;
Step 3. For each $W \subset V$ of size $|W| = \ln n/p$, compute $|\overline{N}(W)|$. If for no W, $|\overline{N}(W)| > (n/p)^{1/2}$, output I;
Step 4. Check all subsets of V of size $(2n/p)^{1/2}$. If none of them is independent, output I;

Step 5. Check all subsets of V and set I to be an independent set of the largest size. Output I.

Let us first check that the output always approximates $\alpha(G)$ within a factor of $O((np)^{1/2}/\log n)$. If an independent set is output at Step 2, its size satisfies $|I| \geq \ln n/(2p)$, and due to Lemma 4 we know that $\alpha(G) \leq \lambda_1(M) \leq 4(n/p)^{1/2}$. Hence the approximation ratio for this case is $O((n/p)^{1/2}/(\ln n/p)) = O((np)^{1/2}/\ln n)$. If I is output at Step 3, then G does not contain an independent set of size $(n/p)^{1/2} + \ln n/p \leq 2(n/p)^{1/2}$, since no W has many non-neighbors. If I is output at Step 4, we get that $\alpha(G) \leq (2n/p)^{1/2}$, thus giving the desired approximation ratio. Finally, if the output is produced at Step 5, it is the result of the exhaustive search over all subsets of V, and thus its size is equal to $\alpha(G)$.

Now it remains to prove that the expected running time of the algorithm is polynomial. We assume that the exhaustive search has running time 2^n (in fact better exponential bound is known, but we do not need it). Notice that eigenvalues of an n by n matrix are computable in time polynomial in n. Clearly, the cost of performing Steps 1 and 2 of the above algorithm is polynomial. The only chance to get to Step 3 is to have a graph G with $\lambda_1(M(G,p)) > 4(n/p)^{1/2}$. The probability of this event is at most $2^{-np/8}$ by Lemma 3. The complexity of Step 3 is $O(\binom{n}{\ln n/p})$. Therefore the expected amount of calculations performed at Step 3 is of order at most

$$\binom{n}{\ln n/p} 2^{-np/8} \leq \left(\frac{enp}{\ln n}\right)^{\ln n/p} 2^{-np/8} = o(1)\,,$$

due to our assumption on $p(n)$. Similarly, we get to Step 4 only if there exists a set W of size $|W| = \ln n/p$ with $\overline{N}(W) \geq (n/p)^{1/2}$. The probability of this event in $G(n,p)$ is at most

$$\binom{n}{\ln n/p}\binom{n}{(n/p)^{1/2}}(1-p)^{(\ln n/p)(n/p)^{1/2}} = o(\binom{n}{(n/p)^{1/2}}^{-1})\,.$$

As executing Step 4 requires $\binom{n}{(n/p)^{1/2}}$ operations, the expected number of operations performed at Step 4 is $o(1)$. Finally, we get to Step 5 if either the greedy algorithm outputs no color class of size at least $\ln n/(2p)$ (and the probability of this event is at most 2^{-n} by Lemma 1) or if G contains an independent set of size $(n/p)^{1/2}$, this happens with probability at most

$$\binom{n}{(2n/p)^{1/2}}(1-p)^{\binom{(2n/p)^{1/2}}{2}} = o(2^{-n})\,.$$

Thus the expected complexity of Step 5 is also $o(1)$. We can conclude that the expected running time of the above algorithm over $G(n,p)$ is dominated by the cost of performing its first two steps (in fact, Step 2) and is therefore polynomial in n. This proves Theorem 1.

4 Approximating the chromatic number

In this section we present an approximation algorithm for the chromatic number. We assume again that the algorithm is given a graph G on n vertices and the value of p as an input.

Step 1. Run the greedy algorithm on G. Let $\mathcal{C}_1$ be the resulting coloring. If the number of colors in $\mathcal{C}_1$ is at least $4np/\ln n$ goto *Step 3*;
Step 2. Define $M = M(G,p)$ according to (1) and compute $\lambda_1(M)$. If $\lambda_1(M) \leq 4(n/p)^{1/2}$, output $\mathcal{C}_1$;
Step 3. If the number of vertices of G of degree at least $4np$ exceeds np goto *Step 7*. Otherwise, color G by first coloring each vertex of degree at least $4np$ by a separate color, and then running the greedy algorithm on the rest of the graph and using fresh colors. Let $\mathcal{C}_2$ denote the obtained coloring.
Step 4. For each $W \subset V$ of size $|W| = \ln n/p$, compute $|\overline{N}(W)|$. If for no W, $|\overline{N}(W)| \geq n^{1/2}/(p^{1/2}\ln n)$ output $\mathcal{C}_2$;
Step 5. Check all subsets of V of size $n^{1/2}/(4p^{1/2}\ln n)$. If none of them is independent, output $\mathcal{C}_2$;
Step 6. Check whether there exist $\ln^4 n$ pairwise disjoint independent sets of size $n^{1/2}/(p^{1/2}\ln n)$ each. If there is no such collection output $\mathcal{C}_2$;
Step 7. Find an optimal coloring by the exhaustive search and output it.

As the reader has possibly noticed, the above algorithm is quite similar to that of Section 3. The algorithm of this section is however somewhat more complicated. This distinction is caused by the fact that the bound of Lemma 1 is much stronger than that of Lemma 2.

Let us verify that the above algorithm approximates $\chi(G)$ within a factor of $O((np)^{1/2}/\ln n)$. If coloring $\mathcal{C}_1$ is output at Step 2, we get by Lemma 4 $\chi(G) \geq n/\alpha(G) \geq n/\lambda_1(M) \geq (np)^{1/2}/4$. On the other hand, $\mathcal{C}_1$ has at most $4np/\ln n$ colors. Thus in this case the approximation ratio is $O((np)^{1/2}/\ln n)$. Observe that if G has at most np vertices of degree at least $4np$, then the coloring $\mathcal{C}_2$, produced at Step 3, uses at most $np + 4np = 5np$ colors. If $\mathcal{C}_2$ is output at Step 4, then $\alpha(G) = O(n^{1/2}/(p^{1/2}\ln n))$ and hence the approximation ratio in this case is $O(np/(n^{1/2}p^{1/2}\ln n)) = O((np)^{1/2}/\ln n)$ as well. An identical argument works for Step 5. If $\mathcal{C}_2$ is output at Step 6, we claim that $\chi(G) = \Omega((np)^{1/2}\ln n)$. Indeed, let $V = (C_1, \ldots, C_k)$ be an optimal coloring of G. Let k_0 be the number of color classes of size at least $n^{1/2}/(p^{1/2}\ln n)$, then $k_0 \leq \ln^4 n$. Also, all color classes are of size less than $n^{1/2}\ln^3 n/p^{1/2}$. Thus the k_0 large color classes cover altogether at most $n^{1/2}\ln^7 n/p^{1/2} \ll n$ vertices. The rest of the vertices are covered by $k - k_0$ color classes, each of size less than $n^{1/2}/(p^{1/2}\ln n)$, implying $k \geq k - k_0 \geq (1 - o(1))n/(n^{1/2}/(p^{1/2}\ln n)) = (1 - o(1))(np)^{1/2}\ln n$. Recalling that $\mathcal{C}_2$ has $O(np)$ colors, we get the desired approximation ratio. Finally, if we ever get to Step 7, the output is found by the exhaustive search and is thus optimal.

The expected running time of the above algorithm can be shown to be polynomial in n similarly to the algorithm for the independence number. The only notable difference is that here we use Lemma 2. We omit detailed calculations.

5 Concluding remarks

In this paper we presented approximation algorithms for the independence number and the chromatic number of a graph. These algorithms were designed as to be efficient over the probability space $G(n,p)$ of random graphs for various values of $p = p(n)$. For every $p(n)$ in the range $n^{-1/2+\epsilon} \leq p(n) \leq 0.75$, both our algorithms *always* achieve approximation ratio $O((np)^{1/2}/log(n))$ and their *average* running time is polynomial over $G(n,p)$.

How good are our results? As stated in the introduction, their approximation ratio is much better than that of the best known algorithms for the worst case. Still, the greedy algorithm, one of the simplest possible algorithms for finding an independent set or a coloring, performs much better for a typical graph than what is guaranteed by our approximation algorithms. There is some indication, however, that the approximation ratio $O((np)^{1/2}/\log n)$ may be hard to improve. Consider, for example, the basic case $p = 1/2$. Saks [21] suggested the following interesting problem. 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 following distribution: choose first a random graph $G(n,1/2)$ and then pick randomly a subset Q of size k and force it to be independent by erasing all edges inside Q. We denote the last model of random graphs by $G(n,1/2,k)$. The problem is to distinguish in polynomial time between the above two models. For the case $k = \Theta(n^{1/2})$, Alon, Krivelevich and Sudakov [3] showed how to recover the independent set of size k in $G(n,1/2,k)$, using spectral techniques, thus clearly providing a tool for distinguishing between $G(n,1/2)$ and $G(n,1/2,k)$. (See also [10] for a related result.) However, Saks' question is still open for every $k = o(n^{1/2})$. Returning to our problem of developing efficient approximation algorithms, note that if we are unable to distinguish between $G(n,1/2)$ and $G(n,1/2,k)$ in polynomial time when $k = o(n^{1/2})$, our algorithm should act the same for both models. As the independence number is a.s. of order $\ln n$ in the first model and is a.s. k in the second one, we cannot hope then to get an algorithm with approximation ratio better than $k/\ln n$. This argument shows that the question of Saks and the problem of developing good approximation algorithms for the independence/chromatic number may be tightly connected.

An obvious open question is what can be done for smaller values of p, i.e., for $p \ll n^{-1/2}$. While our large deviation result (Lemma 3) keeps working for smaller values of p as well, Step 3 of our algorithm for approximating the independence number does not have polynomial expected time anymore (as $\binom{n}{\ln n/p}2^{-np/8}$ tends exponentially fast to infinity for $p \ll n^{-1/2}$). Using again the greedy algorithm as our main tool, we can give an algorithm whose approximation ratio is $O(np)$. We conjecture however that much better approximation algorithms exist for sparse random graphs.

Spectral techniques combined with large deviation inequalities have played an essential role in both of our algorithms. It appears that this machinery can be used successfully to develop approximation algorithms with expected polynomial time for other hard combinatorial problems as well. One possible candidate is the problem of approximating the value of a maximum cut in a graph (MAXCUT).

We hope that the ideas of this paper can be used to devise algorithms, finding almost optimal cuts in expected polynomial time for various values of the edge probability p.

References

1. N. Alon, *Spectral techniques in graph algorithms*, Lecture Notes Comp. Sci. 1380 (C. L. Lucchessi and A. V. Moura, Eds.), Springer, Berlin, 1998, 206–215.
2. N. Alon and N. Kahale, *A spectral technique for coloring random 3-colorable graphs*, Proc. 26^{th} ACM STOC, ACM Press (1994), 346-355.
3. N. Alon, M. Krivelevich and B. Sudakov, *Finding a large hidden clique in a random graph*, Proc. 9^{th} ACM-SIAM SODA, ACM Press (1998), 594–598.
4. B. Bollobás, **Random graphs**, Academic Press, New York, 1985.
5. B. Bollobás, *The chromatic number of random graphs*, Combinatorica 8 (1988), 49–55.
6. R. Boppana, *Eigenvalues and graph bisection: An average case analysis*, Proc. 28^{th} IEEE FOCS, IEEE (1987), 280–285.
7. R. Boppana and M. M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, Bit 32 (1992), 180–196.
8. M. Dyer and A. Frieze, *The solution of some random NP-hard problems in polynomial expected time*, J. Algorithms 10 (1989), 451–489.
9. U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, Proc. 11^{th} IEEE Conf. Comput. Complexity, IEEE (1996), 278–287.
10. U. Feige and R. Krauthgamer, *Finding and certifying a large hidden clique in a semi-random graph*, Random Str. Algor. 16 (2000), 195–208.
11. Z. Füredi and J. Komlós, *The eigenvalues of random symmetric matrices*, Combinatorica 1 (1981), 233–241.
12. M. Fürer, C. R. Subramanian and C. E. Veni Madhavan, *Coloring random graphs in polynomial expected time*, Algorithms and Comput. (Hong Kong 1993), Lecture Notes Comp. Sci. 762, Springer, Berlin, 1993, 31–37.
13. M. M. Halldórsson, *A still better performance guarantee for approximate graph coloring*, Inform. Process. Lett. 45 (1993), 19–23.
14. J. Håstad, *Clique is hard to approximate within* $n^{1-\epsilon}$, Proc. 37^{th} IEEE FOCS, IEEE (1996), 627–636.
15. **Approximation algorithms for NP-hard problems**, D. Hochbaum, Ed., PWS Publish. Company, Boston, 1997.
16. R. Karp, Reducibility among combinatorial problems, in: *Complexity of computer computations* (E. Miller and J. W. Thatcher, eds.) Plenum Press, New York, 1972, 85–103.
17. L. Kučera, *The greedy coloring is a bad probabilistic algorithm*, J. Algorithms 12 (1991), 674–684.
18. L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory 25 (1979), 1–7.
19. T. Łuczak, *The chromatic number of random graphs*, Combinatorica 11 (1991), 45–54.
20. H. J. Prömel and A. Steger, *Coloring clique-free graphs in polynomial expected time*, Random Str. Algor. 3 (1992), 275–402.
21. M. Saks, Private communication.
22. M. Talagrand, *A new isoperimetric inequality for product measure, and the tails of sums of independent random variables*, Geom. Funct. Analysis 1 (1991), 211–223.