Colouring $G_{n,p}$ and Spectral Techniques

Diploma Thesis

Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik

Lehrstuhl für Algorithmen und Komplexität

Submitted by: Lars Kuhtz
Supervisor: Dr. Amin Coja-Oghlan
Berlin, 31st July 2008

Abstract

In consideration of the NP-hardness of the graph colouring problem Karp asked in 1984 if there is an algorithm that colours a graph optimally in *expected* polynomial time thus directing the focus from the worst case complexity to the average case setting. For the $G_{n,p}$ model Krivelevich and Vu in 2002 presented an algorithm that, for relatively dense graphs, approximates the chromatic number within a ratio of $O((np)^{1/2}/\ln(np))$ in expected polynomial time. They asked if a similar result holds for sparse graphs, too. This was recently proved by Coja-Oghlan, Moore, Sanwalani and Taraz, although for a slightly worse approximation ratio. We present a colouring algorithm that, on sparse graphs, has the same approximation ratio as the algorithm of Krivelevich and Vu and has expected polynomial running time over $G_{n,p}$. Considering the list-colouring problem, we use the previous result to obtain an algorithm that approximates the choice number of a graph within an approximation ratio of $O((np)^{1/2}\ln(n)/\ln(np))$ in expected polynomial time over $G_{n,p}$.

In a way, all of the above algorithms use graph properties that are closely related to the eigenvalue spectrum of the adjacency matrix of the input graph. Hence, in an average case setting, the distribution of the eigenvalues of the adjacency matrix of a random graph attain special attention. We improve a result of Feige and Ofek from 2003 concerning the second eigenvalue of the adjacency matrix of $G_{n,p}$.

Zusammenfassung

Angesichts dessen, dass das Färbungsproblem für Graphen NP-schwer ist, fragte Karp 1984, ob es möglich sei, einen Graphen in *erwartet* polynomieller Zeit optimal zu färben. Auf diese Weise verschob er den Fokus von der *Worst-Case*-Komplexität hin zur *Average-Case*-Komplexität. Für das $G_{n,p}$-Model präsentierten 2002 erstmals Krivelevich und Vu einen Algorithmus, der auf relativ dichten Graphen die chromatische Zahl mit einer Güte von $O((np)^{1/2}/\ln(np))$ approximiert und erwartet polynomielle Laufzeit hat. Krivelevich und Vu fragten, ob ein ähnliches Resultat auch für dünne Graphen gilt. Dies wurde kürzlich von Coja-Oghlan, Moore, Sanwalani und Taraz gezeigt. Ihr Algorithmus weist jedoch eine etwas schlechtere Approximationsgüte als der Algorithmus von Krivelevich und Vu auf. Wir präsentieren einen Färbungsalgorithmus, der auch auf dünnen Graphen die selbe Approximationsgüte wie der Algorithmus von Krivelevich und Vu hat und und dessen Laufzeit auf $G_{n,p}$ erwartet polynomiell ist. Weiterhin leiten wir von diesem Resultat einen Algorithmus ab, der die listchromatische Zahl eines Graphen mit einer Güte von $O((np)^{1/2}\ln(n)/\ln(np))$ approximiert und auf $G_{n,p}$ erwartet polynomielle Laufzeit hat.

Alle oben erwähnten Algorithmen verwenden auf irgendeine Weise Eigenschaften des Eingabegraphen, die im engen Zusammenhang zum Eigenwertspektrum der Adjazenzmatrix des Graphen stehen. Daher kommt im Kontext von Average-Case-Komplexität der Wahrscheinlichkeitsverteilung der Eigenwerte der Adjanzenmatrix eines zufälligen Graphen eine besondere Bedeutung zu. Wir verbessern ein Resultat bzgl. des zweiten Eigenwertes der Adjazenzmatrix von $G_{n,p}$ von Feige und Ofek aus dem Jahr 2003.

Declaration

Herewith I declare that this thesis was written independently and using only the stated resources and the literature referred to. Furthermore, I declare that the thesis was presented neither literally nor in a similar form to another examination office.

Moreover, I agree that this thesis is going to be made public in the Branch Library of Mathematics/ Computer Science, University Library of the Humboldt University Berlin.

Berlin, 31st July 2008

Lars Kuhtz

Contents

Chapter 1

Introduction

A graph is a binary, symmetric, and irreflexive relation over a finite space V (i.e., a subset of all subsets of cardinality two of V). An element of V is called a vertex. A related pair of vertices is called an edge.

A colouring of a graph is a partition or a labelling of the vertices under certain constraints. A stable set is a subset of V of unrelated vertices. The most basic notion of graph colouring is the partition of the graph in stable sets (i.e., a labelling without monochromatic edges). If not stated otherwise, this kind of colouring is referred to.

Beside this basic concept this thesis considers a further kind of graph colouring. A list-colouring of a graph is a colouring without monochromatic edges, where furthermore to each vertex is assigned a list of allowed colours, from which the colour for the vertex has to be chosen.

Associated with the concept of graph colouring is the optimisation problem of finding a minimum colouring. A minimum colouring is a colouring that uses as few different colours as possible or —in terms of stable sets— a partition into as few stable sets as possible. The number of colours used by a minimum colouring is called the chromatic number.

The optimisation problem that corresponds to the concept of list-colouring is the problem of determining the so-called list-chromatic number. The list-chromatic number is the smallest integer i such that a list-colouring exists for any assignment of colour-lists of length i to the vertices. The list-chromatic number is commonly referred to as the choice number of a graph.

1.1 Relevance and Applications

The labelling of structured objects under certain constraints is a rather fundamental concept, which serves as an abstraction of many applications. In practise the optimisation problem corresponds to the task of finding a best or cheapest colouring, where the cost function is the number of different colours used.

Probably the most famous though quite theoretical example of an application of graph colouring is the colouring of maps. The question if every map can be coloured with four colours is equivalent to the question if every planar graph can be coloured with four colours. The famous four-colour theorem was conjectured about a hundred years before it could be proved.

But there are several further "real" applications. There is for example the problem of optimising timetables, where two events are adjacent if they share a common resource. The task is to colour all events with dates. Using fewer colours means to find a more compact schedule. The timetable problem was studied by Leighton among others (cf. [24]). Another application in the field of compiler technology and code-optimisation is the assignment of variables to CPU registers (cf. [6]). For further examples and references see [26].

1.2 Hardness of Graph Colouring

Two parameters determine the performance of an optimisation algorithm. The complexity (here we will consider time-complexity only) and the quality of the result. Both parameters can be considered in the worst case setting or in an average case setting.

In the worst case complexity already the basic case of a graph colouring avoiding monochromatic edges is a hard task, for determining the chromatic number of a graph is an NP-hard problem (cf. [18]). To determine the choice number of a graph is even harder. This problem is Π_2^p-complete (cf. [11]).

There are two natural strategies to treat NP-hard problems: first to move from the worst case complexity to the average case complexity by asking if a random instance in an appropriate random model can be solved efficiently and secondly to move from optimally solving the problem to an approximate solution. The first approach involves asking for algorithms that in terms of complexity perform well with high probability and the second approach means to search for heuristics that are expected to perform well in terms of quality of the result. Of special interest are approximation algorithms that guarantee on the worst case quality of the result.

Feige and Kilian have shown that as a consequence of the PCP-theorem (cf. [4]) the chromatic number can not be approximated in polynomial time within an approximation ratio less than $n^{1-\epsilon}$ unless $\mathcal{NP} = \mathcal{ZPP}$ (cf. [12]). Therefore there is no hope that the second of the above mentioned strategies alone can lead to useful results in the field of graph colouring. In this thesis we will combine both approaches and ask for approximation algorithms within the framework of the average case complexity. We will focus on the random graph model that is known as the $G_{n,p}$ model. For a graph in the $G_{n,p}$ model each edge is present independently with probability p. For instance $G_{n,1/2}$ is the uniform distribution over all graphs on n vertices.

1.3 Colourings and Eigenvalues

Approximation algorithms guarantee for the worst case quality of the result. To compute the approximation ratio, it is necessary to estimate upper and lower bounds for the result. In the case of graph colouring the upper bound might "simply" be a valid colouring of the graph.

Obtaining the lower bound is more complicated than "simply" giving a colouring of the graph since for $k \geq 3$ it is co-$\mathcal{NP}$-hard to decide whether a graph is not k-colourable. Thus, one has to certify the non-existence of a colouring of a certain quality, which requires some indirect arguing about some properties of the graph. For the list-colouring problem, the situation is even more difficult to obtain the lower as well as the upper bound.

In this thesis we deal with approximation algorithms that make use of the eigenvalue spectrum of a graph as well as the so called *Lovász ϑ-function* to compute lower bounds for the chromatic number. These properties are algebraic invariants of the graph. The eigenvalues of a graph are obtained by representing the graph as a matrix. The ϑ-function is the result of an optimisation problem over an appropriate algebraic representation of the graph. Both values can be computed in polynomial time, e.g., by formulating them as semidefinite optimisation problems and then applying the ellipsoid method (cf. [14]). As we consider approximation algorithms in the average case complexity setting we ask for the behaviour of these invariants on a random graph space.

1.4 Acknowledgements

I want to extend my gratitude to my supervisor Dr. Amin Coja-Oghlan for instructing me in the field of random graphs and spectral analysis, as well as for his patience and his guidance to find the right answers.

I also want to thank Dr. Mihyun Kang for her comments as second supervisor of this thesis and for the time she spent proofreading the text.

Chapter 2

Notation and Preliminaries

In this chapter some basic definitions and notation are given in a compact form. Most of them are of common use. Further definitions are given in the place where they are used for the first time. Readers who are familiar with the subject may skip this chapter and only refer to it in case of doubt.

The natural logarithm is denoted by ln and the logarithm to the basis 2 by log. The operators min and max bind stronger than the additive operators and weaker than the multiplicative operators. $f(x) \ll g(x)$ is true if and only if $\lim_{x\to\infty} f(x)/g(x) = 0$.

We employ the usual notation for asymptotic formulas. $f(x) \sim g(x)$ if and only if $\lim_{x\to\infty} f(x)/g(x) = 1$. Further $f(x) = O(g(x))$ if and only if there are positive constants c and x_0 such that for all $x \geq x_0$ it holds that $|f(x)| \leq c|g(x)|$; $f(x) = o(g(x))$ if and only if $f(x) \ll g(x)$; $f(x) = \Omega(g(x))$ if and only if $g(x) = O(f(x))$; and finally $f(x) = \Theta(g(x))$ if and only if $f(x) = O(g(x))$ and $g(x) = O(f(x))$. Where it is appropriate we restrict O and Ω to positive valued functions. In particular this is useful to express the intuitive notion of $-O(f(x))$ and $-\Omega(f(x))$.

The notions of probability, expectation, and variance of a random variable are defined in the usual way and denoted by $\mathbb{P}$, $\mathbb{E}$, and Var. Let $S = (S_n)_{n\in\mathbb{N}}$ be a sequence of probability spaces. An event is said to occur *almost surely* or *with high probability* on S if its probability over S_n tends to 1 as n tends to infinity.

A *graph* G is a tuple (V, E) where V is the *vertex set* and $E \subseteq \binom{V}{2}$ is the *set of edges*. Two vertices u, v are called *adjacent* if and only if $\{u, v\} \in E$. A vertex $v \in V$ is called *incident* with an edge e if and only if $v \in e$. The *neighbourhood of a vertex* $v \in V$, denoted by $N(v)$, is the set of all vertices that are adjacent to v. Let U, W be subsets of V. The number of edges between U and W is denoted by $e(U, W)$.

The *degree of a vertex* v is defined as the number of edges with which v is incident and denoted by $d(v)$. $\Delta(G)$ is defined as the maximum and $\delta(G)$ as the minimum degree over all vertices of G. The average degree of all vertices of G is denoted by $d(G)$. If all vertices of G are of degree d, then G is called d-regular.

A set of vertices $W \subseteq V$ is called *closed* if every vertex in $V \setminus W$ is adjacent to a vertex of W. A set of pairwise adjacent vertices is called a *clique*. $\omega(G)$ denotes the size of a maximum (i.e., globally maximal) clique of G. A set of pairwise non-adjacent vertices is called a *stable set* or *independent set*. The *independence number* of G, denoted by $\alpha(G)$, is the size of a maximum (i.e., globally maximal) stable set of G.

A graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E \cap \binom{V'}{2}$ is called a subgraph of $G = (V, E)$, which is denoted by $G' \subseteq G$. If $E' = \{e \in E \mid e \subseteq V'\}$, then G' is called *the subgraph of* G *that is induced by* V' and is denoted by $G[V']$.

Throughout this paper the *expected running time* of an algorithm over a random graph model M is defined as $\sum_{G \in M}(\text{running time on } G \cdot \mathbb{P}_M[G])$.

Chapter 3

Results

A *random graph* in the $G_{n,p}$ model is a graph on n vertices where each edge occurs independently at random with probability p. The expectation of the degree of a vertex is therefore $(n-1)p \sim np$. Because the number of edges of a random graph is concentrated about its expectation, the average degree of a random graph does not differ too much from np. The $G_{n,p}$ model was introduced by Erdős and Rényi and is widely studied. As an introduction and for further references see [17].

A *colouring* of a graph G is a labelling c of the vertices of G under certain constraints. If not stated otherwise c is assumed to prohibit monochromatic edges (i.e., c is constrained to satisfy that if $\{u, v\} \in E$, then $c(u) \neq c(v)$ for all $u, v \in V$). A *k-colouring* of G is a colouring that uses only k different colours to label the vertices of G. Generally the colours are assumed to be taken from a prefix of $\mathbb{N}$. An *optimal colouring* is a colouring that uses as few colours as possible. The *chromatic number* $\chi(G)$ is defined as the minimum k such that G is k-colourable. In Karp [18] it is shown that for $k \geq 3$ it is NP-hard to decide whether a graph is k-colourable.

A colouring algorithm is said to achieve an *approximation ratio of* r or *r-approximation* if on every input graph G the returned colouring uses at most $\chi(G) \cdot r$ colours.

The chromatic number of a graph in the $G_{n,p}$ model is known by results of Bollobás [5] and Łuczak [17]. They prove that for $G \in G_{n,p}$ with high probability

$$\chi(G) \sim \frac{np}{2\ln(np)}, \quad \text{for } c/n < p = o(1)$$

and

$$\chi(G) \sim \frac{n}{2\log_{\frac{1}{1-p}}(n)}, \quad \text{for constant } p.$$

A simple greedy heuristic that colours subsequently all vertices by assigning each vertex the smallest available colour uses almost surely approximately $np/\ln(np)$ colours on $G_{n,p}$ (where $p = o(1)$) and therefore almost surely achieves a 2-approximation. Though there are several other algorithms with polynomial running time, none of these is known to colour $G_{n,p}$ with at most $(1-\epsilon)np/\ln(np)$ colours with high probability. Even though the greedy algorithm in most cases returns twice the number of colours of an optimal colouring it does not guarantee for the worst case quality. The number of colours that the greedy heuristic uses only depends on the order in which the vertices are coloured. However, there are instances in $G_{n,p}$ for which on almost all permutations on the vertices the greedy algorithm fails to find a colouring that differs from the chromatic number less than a factor of order n. It is thus impossible to achieve substantially better results by enhancing the greedy heuristic by any kind of randomisation. Feige and Kilian [12] show that this deficiency is not peculiar to the greedy heuristic. They prove that there is no algorithm with polynomial running time that approximates the chromatic number within a ratio less then $n^{1-\epsilon}$ unless $\mathcal{NP} = \mathcal{ZPP}$.[1]

In view of the NP-hardness of graph colouring Karp [19] asks if there is an algorithm that colours a graph G in expected polynomial time, where expected polynomial time denotes the mean of the running time over all graphs in an appropriate random graph model. Considering the hardness of approximation of graph colouring even a weaker question is interesting: is there an algorithm that approximates the chromatic number of a graph within a ratio of less than $n^{1-\epsilon}$ and has expected polynomial running time?

For the $G_{n,p}$ model Krivelevich and Vu [22] first give an answer to this question, but only for relatively dense graphs. They describe an algorithm, which has expected polynomial running time and achieves an approximation ratio of $O((np)^{1/2}/\ln(np))$ for $n^{-1/2+\epsilon} \leq p \leq 0.99$. The algorithm upper bounds the chromatic number by simply applying the greedy heuristic to the input graph. For the lower bound it first bounds the independence number by the smallest eigenvalue of the input graph. The chromatic number then is bounded by n/α. Krivelevich and Vu can show that with high probability these bounds are good enough to guarantee the claimed approximation ratio. The chromatic number is approximated by stronger algorithms only in the case that the upper and the lower bounds computed so far differ too much. However, these algorithms exceed polynomial running time. The authors show that this case is unlikely enough to remain in expected polynomial running time for the whole algorithm. They ask if a similar result holds for sparse random graphs, too.

[1] Using the slightly stronger assumption $\mathcal{NP} \not\subseteq \mathcal{ZP}\mathrm{TIME}(2^{O(\log(n)(\log\log(n))^{2/3})})$ Holmerin and Engebretsen [10] prove that χ can not be approximated up to a factor of $n^{1-O(1/\sqrt{loglog(n)})} = n^{1-o(1)}$.

For $\ln^6(n)/n \ll p \leq 3/4$ a first algorithm is given by Coja-Oghlan and Taraz [9]. It computes an upper bound by applying a colouring algorithm called CoreColour (cf. Chapter 4). As a lower bound on the chromatic number serves the value of the so-called Lovaśz ϑ function of the complement graph. The restriction $p \gg \ln^6(np)/n$ is due to the analysis of the probable value of $\vartheta(\overline{G})$. This result is improved to hold for essentially all values of p ($1/n \leq p \leq 0.99$) by Coja-Oghlan, Moore, and Sanwalani [8] by replacing ϑ by a semidefinite program called SDP_k, which is a relaxation of the *MAX-k-CUT* problem. However, these algorithms achieve an approximation ratio of only $O((np)^{1/2})$.

The algorithm of Krivelevich and Vu fails on sparse graphs because the upper as well as the lower bound will not give the needed approximation ratio with a sufficiently high probability. Coja-Oghlan, Moore, and Sanwalani compute a lower bound of the same quality as the algorithm of Krivelevich and Vu that works for all values of p. The CoreColour algorithm that they employ for the upper bound achieves a ratio of $O(np)$. However, to guarantee the same approximation ratio as the algorithm of Krivelevich and Vu for all values of p a ratio of $O(np/\ln(np))$ for the upper bound is needed.

In this thesis we give an algorithm that bounds the chromatic number of a graph from above with an approximation ratio of $O(np/\ln(np))$ for (almost) all values of p and has linear expected running time over $G_{n,p}$:

Theorem 1. *Let c be a sufficiently large constant. For $c/n \leq p \leq 0.99$ there is a colouring algorithm with approximation ratio $O(np/\ln(np))$, which has linear expected running time over $G_{n,p}$.*

Combining this algorithm with Coja-Oghlan, Moore, and Sanwalani [8] immediately leads to the following theorem, which extends the result of Krivelevich and Vu [22] to all values of p and thus gives a positive answer to their question.

Theorem 2. *Let c be a sufficiently large constant. For $c/n \leq p \leq 0.99$ there exists an algorithm that colours $G \in G_{n,p}$ in expected polynomial time and achieves an approximation ratio of $O(\sqrt{np}/\ln(np))$.*

The concept of list-colouring was introduced by Erdős, Rubin, and Taylor [11] and independently by Vizing [28]. Given a set of colours $\mathfrak{L}$ a *list assignment* of a graph $G = (V, E)$ is an assignment $L\colon V \to \wp(\mathfrak{L})$ that assigns each vertex v of G a list $L(v)$ of colours. An *L-list-colouring* of G is a colouring c of G such that $c(v) \in L(v)$ for all $v \in V$. A k-list assignment of a graph G is a list assignment of G where all lists contain k different colours (i.e., $|L(v)| = k$ for all $v \in V$). The *choice number* of G (denoted by $ch(G)$) is the smallest k such that for every k-list assignment L of G there is a L-list-colouring of G . Erdős, Rubin, and Taylor show in their seminal paper [11] that it is Π_2^p-hard to determine the choice number of a graph.

Molloy and Reed [25] conjecture that the choice number of a graph G is bounded by $\chi(G) \cdot \ln(\Delta(G))$. While a proof on this conjecture is not known, the choice number of G can be bounded by $\chi(G) \cdot \ln(n)$ using simple probabilistic techniques (cf. Chapter 5). $\chi(G)$ itself is a lower bound on the choice number. Therefore the algorithm mentioned in Theorem 2 immediately implies an approximation algorithm for the choice number.

Theorem 3. *Let c be a sufficiently large constant. For $c/n \leq p \leq 0.99$ there exists an algorithm that approximates the choice number of a graph $G \in G_{n,p}$ in expected polynomial within an approximation ratio of $O(\sqrt{np}\ln(n)/\ln(np))$.*

The adjacency matrix of G is the matrix $A(G) \in \mathrm{Mat}(V, V)$ with $A_{u,v} = 1$ if and only if $\{u, v\} \in E$. The eigenvalues of $A(G)$ are denoted by $\lambda_1 \geq \cdots \geq \lambda_n$. We define $\lambda(G) = \max\{|\lambda_2|, \ldots, |\lambda_n|\}$. The difference between λ_1 and $\lambda(G)$ is commonly called the *eigenvalue gap of G*.

There are several examples for the tight relation between colourings and the eigenvalue spectrum (i.e., the distribution of the eigenvalues) of a graph. An upper bound on the chromatic number can simply be proved by giving a colouring of the graph. It is much more difficult to prove that one cannot get along with a certain number of colours. All of the previously mentioned approaches to colouring as far as they do not employ the trivial lower bound 2 use some kind of properties related to the eigenvalue spectrum of the graph in order to lower bound the chromatic number.

The eigenvalue gap is salient in the context of list colouring. It seems that there is a tight connection between the choice number and expansion properties a graph, for which the eigenvalue gap is a measure (cf. [27]). As an example for the relation of the eigenvalues and the choice number of a graph, Alon and Krivelevich [2] bound the choice number of a regular bipartite graph from below using an expansion property, which can be estimated by the eigenvalue gap.

In this thesis we study the value of the eigenvalue gap of a graph in $G_{n,p}$. We give two improvements on a paper by Feige and Ofek [13], in which λ of a sparse random graph is bounded by $O(\sqrt{np})$.

The following theorem is a slightly improved version of theorem 1 from [13]. Feige and Ofek prove the result under the stronger assumption $p = c_0 \ln(n)/n$, where c_0 is a sufficiently large constant. We enhance their proof to hold for an arbitrary (positive) constant c_0.

Theorem 4. *Let G be a random graph in the $G_{n,p}$ model with $p = \Omega(\ln(n)/n)$. Then almost surely $\lambda(G) = O(\sqrt{np})$.*

This result does not hold if np is a constant. In this case Feige and Ofek bound the value of λ after removing the vertices of high degree. They state the following theorem.

Theorem 5. *Let G be a random graph taken from $G_{n,p}$ where np is a sufficiently large constant. Let $\sqrt{\ln(np)/np} < \epsilon < 0.9$ be a constant. The subgraph G' induced by removing from G all the vertices of degree greater than $(1+\epsilon)np$ has $\lambda = O(\sqrt{np})$* almost surely.

The proof of this theorem that is presented in [13] assumes that the term *almost surely* is used for probabilities of at least $1 - e^{-\Omega(\epsilon^2 np)}$, which is constant for $n \to \infty$. This notion differs from the common definition of the term *almost surely* for probabilities which are at least $1 - o(1)$ as $n \to \infty$. We extend the proof given in [13] so that it holds for this common notion of the term *almost surely*.

The remaining parts of this thesis are organised as follows. Chapter 4 introduces the algorithm GreedyCoreColour and presents the proofs of Theorem 1 and Theorem 2. In Chapter 5 we present a lower bound on the choice number of bipartite r-regular graphs by its expansion due to Alon and Krivelevich. As the expansion of a graph can be bounded by the eigenvalue gap, this serves as a motivation for the results concerning the eigenvalue gap of $G_{n,p}$. Furthermore, in Chapter 5 an algorithm for approximating the choice number is given, which complies with the requirements of Theorem 3. Chapter 6 is merely technical and contains the proofs of Theorem 4 and Theorem 5 concerning the eigenvalue gap of $G_{n,p}$. The thesis is completed by some concluding remarks in Chapter 7 and supplemented by an appendix containing some numerical results concerning the value of ϑ and *MAXCUT* on $G_{n,p}$.

Definitions of symbols and concepts are given in three places throughout this thesis. Notational conventions and most basic and common definitions are given in Chapter 2. Concepts needed for the presentation of the results are defined together with the results in Chapter 3. Notation, symbols, and specific notions of merely technical use are introduced and defined in the place where they appear.

Chapter 4

Colouring $G_{n,p}$ in Expected Polynomial Time

This chapter treats the problem of colouring a graph in the setting of the average case complexity. First we give a colouring algorithm that has linear expected running time over $G_{n,p}$ and meets the requirements of Theorem 1. Adding an appropriate lower bound on the chromatic number to the upper bound returned by that colouring algorithm gives an algorithm that approximates the chromatic number and proves Theorem 2.

4.1 Upper Bounds on the Chromatic Number

The colouring algorithm presented in this section combines a simple greedy heuristic and the core colour approach from [9]. Since the strategies of these algorithms are independent of each other their combination leads to a better approximation ratio. In the following we explain the way these algorithms work and how they can be combined.

4.1.1 A Greedy Heuristic

The greedy heuristic for stable sets works by "greedily" collecting as many vertices as possible. "Greedily" here means that the vertices are called upon in some arbitrary order and that a vertex is included in the stable set if it is not adjacent to any of the vertices already included in the stable set.

```
STABLESETGREEDY
Input:  a graph G
Output:  a stable set in G

  1. S = ∅

  2. For all v in V(G) do:

     (a) if S ∪ {v} is stable in G then S := S ∪ {v}.

  3. return S
```

A colouring algorithm is obtained by iteratively applying STABLESETGREEDY: each returned stable set becomes a colour-class and is removed from the graph before the next iteration step is executed. Obviously in this way a graph can be coloured in linear time ($O(n+m)$).

The STABLESETGREEDY heuristic has one nice property, which first of all makes it easy to analyse. Let $G \in G_{n,p}$ and let I be a stable set in G produced by STABLESETGREEDY. Then the graph $G[V(G) \setminus I]$ equals $G_{n-|I|,p}$. This is so because the decision if a vertex is included in the stable set only depends on the edges between this vertex and already collected vertices. All these edges, however, will be removed from G together with the stable set. The edges between the remaining vertices are therefore independent from the removed vertices. This implies that when colouring G with STABLESETGREEDY by subsequently removing stable sets, in every step the behaviour of STABLESETGREEDY depends on the previous steps only by the number of the already removed vertices. Furthermore, it is possible to colour the graph partially using STABLESETGREEDY. Switching then to another algorithm, the behaviour of this algorithm will only depend on the size of the remaining uncoloured graph because it looks completely random.

Another property of the STABLESETGREEDY heuristic is that a found stable set (respective a colour-class) is closed (i.e., all vertices outside the set are adjacent to some vertex inside the set). If there were a vertex outside that is not adjacent to some vertex inside it would have been collected by STABLESETGREEDY.

4.1.2 The Core Colouring Approach

The k-core of a graph is obtained by removing subsequently all vertices of degree at most $k-1$. Therefore, the k-core of a graph G is an induced subgraph of G and has minimal degree at least k. During the process of removing vertices the degree of a vertex of G only decreases. Hence the k-core of a graph is uniquely determined. Given that the k-core is already coloured, this colouring can be extended to the vertices outside the core using at most k colours. At the moment when one of these vertices was removed it had at most $k-1$ neighbours, which are coloured with at most $k-1$ colours and hence one of the k colours can be used to colour the vertex itself. Thus, the task of k-colouring G is reduced to the task of k-colouring the k-core of G. To find an optimal colouring on the k-core Lawler's algorithm (cf. [23]) can be used, which on n vertices has running time of less than e^n.

```
CoreColour
Input:  a graph G, an integer k
Output:  a colouring of G

  1. Determine the k-core of G by removing subsequently all
     vertices of degree less than k from G.

  2. Colour the k-core of G optimally using Lawler's
     algorithm (cf. [23]).

  3. Colour the vertices outside the k-core in the reversed
     order in which they were removed from G with k colours.
```

Notice that the k colours that are used in Step 3 intersect with the colours used in Step 2. Hence the returned colouring employs at most $\max\{\chi(G), k\}$ colours. Step 1 and Step 3 have running time $O(n+m)$. The expected running time of the algorithm depends primarily on the size of the k-core. Coja-Oghlan and Taraz [9] have shown that for $k \geq e^2 np$ and $p \geq 1/n$ the probability that the k-core of $G_{n,p}$ contains at least ν vertices is bounded by $e^{-\nu}$. Therefore the expected running time of CoreColour is bounded by $O(n+m)+\sum_{\nu=0}^{n} e^{\nu}e^{-\nu} = O(n+m)$. Altogether this yields an approximation ratio of $O(np)$ in expected linear time. To achieve a better approximation ratio it might be necessary to achieve expected linear (or polynomial) time for even smaller values of k. But since in this case almost surely the k-core is large this approach cannot achieve a better approximation ratio.

4.1.3 Combining the Greedy and the Core Colouring Approach

The crucial observation for the combination of the StableSetGreedy and the CoreColour approach is that as long as a graph $G \in G_{n,p}$ still contains many uncoloured vertices the StableSetGreedy heuristic finds with high probability a big stable set on the uncoloured vertices. This means that the StableSetGreedy heuristic performs well during the first steps by finding large colour classes and gets worse when it needs many colours on the last remaining vertices. The idea is to apply the StableSetGreedy heuristic as long as it performs well and to switch then to another algorithm that performs well on the remaining smaller graph. For this purpose we will use CoreColour. As seen above, in the moment of switching from the StableSetGreedy heuristic to CoreColour the still uncoloured part of G on, say, ν vertices, is distributed as $G_{\nu,p}$. Hence the probability distribution of the size of the core of the original random graph on n vertices is induced to the probability distribution of the size of the core of the random graph on ν vertices. Now, as seen above, CoreColour$(\nu, e^2\nu p)$ has expected linear running time and achieves an $O(\nu p)$ approximation. Setting ν to $n/\ln(np)$ would give the approximation ratio stated in Theorem 1. In the sequel our goal is to show how the StableSetGreedy heuristic can achieve a $O(np/\ln(np))$ approximation on the first but ν vertices. Here is the algorithm:

```
GreedyCoreColour
Input:  a graph G = (V, E), an integer k ∈ ℕ
Output:  a colouring of G

  1. G* := G.

  2. while |G*| > n/ln(np) do

     (a) Use the StableSetGreedy heuristic to choose a
         stable set S ⊆ V* and colour S with a new colour.

     (b) G* := G* \ S;

  3. If less than 20np/ln(np) colours have been used until
     now,

     (a) then colour G* with new colours using
         CoreColour(G*,k),

     (b) else reject the present colouring of G and colour G
         exactly using Lawler's algorithm (cf. [23]).
```

Theorem 1 is an immediate consequence of the following proposition.

Proposition 1. *Let c be a sufficiently large constant. Suppose that $c/n \leq p \leq 0.99$. Then the algorithm* GreedyCoreColour$(G_{n,p}, e^2np/\ln(np))$ *has linear expected running time and achieves an approximation ratio of* $O(np/\ln(np))$.

The remaining part of this section contains the proof of Proposition 1. For the proof the following merely technical lemma is needed, which guarantees that the test in Line 3 of GreedyCoreColour is sufficiently unlikely to fail.

Lemma 1. *Let c be a sufficiently large constant. Suppose that $c/n \leq p(n) \leq 0.99$ and $\nu \geq n/\ln(np)$. Then the probability that a graph in $G_{\nu,p}$ does contain a closed set of less than $\ln(np)/(10p)$ vertices is at most $\exp(-n/(np)^{0.6})$.*

Proof of Lemma 1. Let $G=(V,E)$ be a random graph from $G_{\nu,p}$. Let $k = \ln(np)/(10p)$. The probability that for a fixed vertex u and a fixed set of vertices $M \subseteq V$ with $|M|=k$ and $u \notin M$, u is connected to some vertex of M is

$$1-(1-p)^k.$$

The probability that a fixed set $M \subseteq V$ with $|M|=k$ is closed (i.e., every vertex outside from M is connected to some vertex of M), is

$$(1-(1-p)^k)^{\nu-k}.$$

Furthermore the probability that there exists a closed set $M \subseteq V$ with $|M|=k$, is at most

$$\binom{\nu}{k}(1-(1-p)^k)^{\nu-k}. \tag{4.1}$$

Notice that the probability for the existence of closed set of a certain size increases as $|V|$ decreases. Hence, a minimal value of ν maximises (4.1).

Since the event that there is a closed set of size at most k implies that there is a closed set of size k, it suffices to bound (4.1). We get

$$\begin{aligned}
&\binom{\nu}{k}(1-(1-p)^k)^{\nu-k}\\
\leq &\left(\frac{e\nu}{k}\right)^k \exp(-(1-p)^k(\nu-k))\\
= &\exp\left(k\ln(e\nu/k)-(1-p)^k(\nu-k)\right)\\
= &\exp\left(k+k\ln(\nu/k)-(1-p)^k\nu+(1-p)^k k\right)\\
= &\exp\left(k\left(1+(1-p)^k+\ln(\nu/k)-(1-p)^k\frac{\nu}{k}\right)\right)\\
\leq &\exp\left(k\left(2+\ln(\nu/k)-(1-p)^k\frac{\nu}{k}\right)\right).
\end{aligned} \tag{4.2}$$

Recall that $k = \ln(np)/(10p)$. Setting ν to its minimum value (i.e. $\nu = n/\ln(np)$) we get $\nu/k = 10np/\ln^2(np)$. Furthermore we use that $(1-p)^k = (1-p)^{\ln(np)/(10p)} = (np)^{\ln(1-p)/(10p)}$. Now inequality (4.2) can be continued by

$$(4.2) \leq \exp\left(k\left(2 + \ln(\frac{10np}{\ln^2(np)}) - \frac{10(np)^{1+(\ln(1-p)/(10p))}}{\ln^2(np)}\right)\right).$$

Notice that $(np)^{\ln(1-p)/(10p)} \geq (np)^{\ln(1-0.99)/(10 \cdot 0.99)} > (np)^{-0.5}$. For $np \geq c$ with c being a sufficiently large constant we get

$$\begin{aligned}
&\exp\left(k\left(2 + \ln(\frac{10np}{\ln^2(np)}) - \frac{10(np)^{1+(\ln(1-p)/(10p))}}{\ln^2(np)}\right)\right) \\
&\leq \exp\left(k\left(-(np)^{1-0.5)}\right)\right) = \exp\left(-\frac{\ln(np)(np)^{1-0.5}}{10p}\right) \\
&\qquad \leq \exp\left(-\frac{(np)^{1-0.6}}{p}\right) \leq \exp\left(-\frac{n}{(np)^{0.6}}\right).
\end{aligned}$$

□

Proof of Proposition 1. Running time: To determine the expected running time of GreedyCoreColour one has to calculate the probability that Line 3b is executed (i.e., the probability that the StableSetGreedy heuristic fails to use at most $20np/\ln(np)$ colours). In the following we assume that the StableSetGreedy heuristic all together uses more than $20np/\ln(np)$ colours. If in every step the StableSetGreedy heuristic produces a stable set of size at least $\ln(np)/(10p)$, then altogether it would have used at most $10np/\ln(np)$ colours. Therefore among the first $20np/\ln(np)$ colour classes at least $10np/\ln(np)$ classes are of size less than $\ln(np)/(10p)$ vertices. The probability that in one step the StableSetGreedy heuristic fails to find a stable set of size $\ln(np)/(10p)$ is bounded by the probability that there is a closed set of this size, which by Lemma 1 is bounded by $\exp(-n/(np)^{0.6})$. Notice that this bound does not depend on the size $|V^*|$ of the current Graph G^* and recall that G^* in every step i is distributed as $G_{|V^*|,p}$. Hence the probability that StableSetGreedy fails to find a stable set of size $\ln(np)/(10p)$ is bounded for all steps independently of each other, and the number of failures is bounded by the binomial distribution. Therefore the probability that during the first $20np/\ln(np)$ steps the StableSetGreedy heuristic fails more than $10np/\ln(np)$ times is at most

$$\begin{aligned}
&\binom{\frac{20np}{\ln(np)}}{\frac{10np}{\ln(np)}} \exp\left(-\frac{n}{(np)^{0.6}}\right)^{\frac{10np}{\ln(np)}} \\
&\leq 2^{\frac{20np}{\ln(np)}} \exp\left(-\frac{10n(np)^{0.4}}{\ln(np)}\right)
\end{aligned}$$

$$\begin{aligned}
&\leq \exp\left(\frac{\ln(2)20np}{\ln(np)}\right)\exp\left(-\frac{10n(np)^{0.4}}{\ln(np)}\right)\\
&\leq \exp\left(\frac{\ln(4)10np}{\ln(np)}-\frac{10n(np)^{0.4}}{\ln(np)}\right)\\
&\leq \exp\left(-10n(\frac{-\ln(4)p+(np)^{0.4}}{\ln(np)})\right)\\
&\leq \exp(-10n)\leq \exp(-n).
\end{aligned}$$

Lawler's algorithm has a running time of $(1+\sqrt[3]{3})^n < e^n$ (cf. [23]). During the execution of the while-loop every vertex and every edge are treated at most once. So the while-loop takes at most $O(n+m)$ steps. CORE-COLOUR(G,k) on at most $n/\ln(np)$ vertices has linear expected running time (cf. [9]). Therefore the expected running time of the whole algorithm is

$$\begin{aligned}
&\text{running time of lines 1 and 2}\\
&+\text{(expected) running time of Line 3a}\cdot\mathbb{P}[\text{Line 3a is executed}]\\
&+\text{running time of Line 3b}\cdot\mathbb{P}[\text{Line 3b is executed}],
\end{aligned}$$

which is

$$O(n+m)+O(n+m)(1-e^{-n})+e^n e^{-n}=O(n+m).$$

Approximation ratio: If Line 3a is executed, then the algorithm uses at most $20np/\ln(np)$ colours in step 2 and at most $\max\{\chi(G^*), e^2np/\ln(np)\}$ colours in line 3a. Altogether

$$\begin{aligned}
&20np/\ln(np)+\max\{\chi(G^*),e^2np/\ln(np)\}\\
&\leq \max\{28np/\ln(np),\chi(G)+10np/\ln(np)\}
\end{aligned}$$

colours are used. If $28np/\ln(np)$ colours are returned the approximation ratio is $O(np/\ln(np))$. Else, if $\chi(G)+10np/\ln(np)$ colours are returned then $\chi(G)$ is at least $e^2np/\ln(np)$, and thus an approximation ratio of $(20np/\ln(np)+\chi(G))/\chi(G)\leq 28np/e^2np\leq 4$ is achieved.

If Line 3b is executed, then G is coloured with $\chi(G)$ colours. □

4.2 Approximating the Chromatic Number

Now we present an algorithm that is obtained by combining the algorithm GREEDYCORECOLOUR with the approach from Coja-Oghlan, Moore, and

Sanwalani [8] to approximate the chromatic number of $G_{n,p}$ in expected polynomial time. The lower bound is taken from their approach while GreedyCoreColour serves as upper bound on χ. The semidefinite programming procedure SDP_k, which is used in [8] to estimate the lower bound, is a relaxation of the *MAX-k-CUT*-Problem and can be computed in polynomial time using the ellipsoid method (cf. [14]). For a description of SDP_k and its relation to graph colouring refer to [8].

The algorithm ApproxColour is obtained from the algorithm ApproxColour in [8] by replacing the procedure CoreColour by GreedyCoreColour. Remember that m denotes the number of edges of a graph G.

ApproxColour
Input: a graph G
Output: a colouring of G

1. Let $k = \gamma(np)^{1/2}$ for a sufficiently small constant γ and let $l = e^2 np/\ln(np)$.
2. If $SDP_k(G) < m$,
   (a) then return GreedyCoreColour(G, l),
   (b) else return an optimal colouring of G using Lawler's algorithm (cf. [23]).

Theorem 2 is an immediate consequence of the following proposition.

Proposition 2. *Let c be a sufficiently large constant. Supposed that $c/n \leq p \leq 1/2$ the algorithm* ApproxColour *approximates the chromatic number of a graph within a ratio of $O\left(\sqrt{np}/\ln(np)\right)$ and has expected polynomial running time over $G_{n,p}$.*

Proof of Proposition 2. Running time: As shown in [8] the probability that Line 2b is executed is at most $\exp(-2n)$. Taking into account that GreedyCoreColour has linear expected running time and that Lawler's algorithm has running time $(1+\sqrt[3]{3})^n < e^n$ ApproxColour has expected polynomial running time over $G_{n,p}$.

Approximation ratio: If $\chi(G) \leq k$ then obviously a maximum k-Cut of G includes all edges of G. As SDP_k is a relaxation (i.e., an upper bound) of *MAX-k-CUT* it holds that if $m > SDP_k(G)$, then a maximum k-Cut does not include all edges and k is a lower bound on $\chi(G)$. Proposition 1 assures that GreedyCoreColour achieves an approximation ratio of $O(np/\ln(np))$ as an upper bound on $\chi(G)$. Hence, if line 2a is executed, ApproxColour guarantees an approximation ratio of $O(\sqrt{np}/\ln(np))$. Otherwise the returned colouring is optimal. $\square$

Chapter 5

The Choice Number of Random Graphs

Alon, Krivelevich, and Sudakov [3] show that for a graph G in $G_{n,p}$ the choice number $ch(G)$ and the chromatic $\chi(G)$ number are almost surely within a constant factor. Moreover Krivelevich [21] shows that for dense random graphs ($n^{-1/4+\epsilon} \leq p(n) \leq 3/4$) the chromatic number and the choice number of a random graph in the $G_{n,p}$ model are almost surely asymptotically equal. Furthermore, he conjectures that this holds for all values of $p = p(n)$. Still, these two numbers can differ significantly. Bipartite (i.e., 2-colourable) graphs are the most popular example of a family of graphs where the value of the choice number can be much higher than the chromatic number. Erdős, Rubin, and Taylor [11] show that the choice number of the complete bipartite graph $K_{n,n}$ is $(1 + o(1)) \log(n)$.

The choice number is trivially bounded from below by the chromatic number. As we show below in Proposition 4 the choice number of a Graph G can be bounded from above by $\chi(G) \ln(n)$. A much stronger bound is conjectured by Molloy and Reed [25]: $ch(G) \leq \chi \ln(\Delta)$.

An interesting (though settled) question in this context is the value of the choice number of a random bipartite graph. A random bipartite graph in the $G_{n,n,p}$ model is obtained by taking two classes, each of n vertices, and including each edge between two vertices from different classes independently at random with probability p. The two classes themselves remain stable. Alon and Krivelevich [2] show that $ch(G_{n,n,p}) = (1 + o(1)) \log(np)$. They bound the choice number of a bipartite graph from below using an expansion property. They remark that the following theorem holds because the expansion of a regular bipartite graph can be estimated by the eigenvalue gap of the graph:

Theorem 6 (Alon, Krivelevich (1998))**.** *For a bipartite d-regular graph the choice number is at least* $(1 + o(1)) \log(d/\lambda)$.

In this chapter we first give a proof of Theorem 6. Then we shall bound the choice number with the chromatic number and apply these bounds to the colouring algorithm ApproxColour. The obtained algorithm approximates the choice number of a graph within a ratio of $(np)^{1/2} \ln(n)/\ln(np)$ and has expected polynomial running time over $G_{n,p}$ thus proving theorem 3.

5.1 The Choice Number of Regular Bipartite Graphs and the Eigenvalue Gap

Theorem 6 motivates the interest in the so-called eigenvalue gap in the following chapter. The expression *eigenvalue gap* refers to the gap between the largest eigenvalue $\lambda_1(G)$ and the —in absolute value— second largest eigenvalue λ(G) of the adjacency matrix of a graph G.

In the context of bipartite graphs, however, we define $\lambda(G)$ as the maximum —in absolute value— of all eigenvalues except for the largest and smallest one. This is due to the symmetry of the eigenvalue spectrum of a bipartite graph. In this chapter $\lambda(G)$ is therefore defined as $\lambda_2(G)$ (we assume that G has more than two vertices).

In the following we demonstrate the connection between the choice number and $\lambda(G)$ by proving Theorem 6. First we adapt a result from Tanner [27] concerning the connection between $\lambda(G)$ and the expansion of a d-regular graph to the bipartite case. Then we recall the proof of the lower bound on the choice number from [2].

5.1.1 The Second Eigenvalue and Expansion

Theorem 7 (Tanner (1982))**.** *Let* $G = (A \dot\cup B, E)$ *be a d-regular bipartite graph. Let* $n = |A| = |B|$ *and* $\lambda = \lambda(G)$*. Let* $X \subseteq A$ *and* $\rho = |X|/n$*. Then*

$$\frac{|N(X)|}{n} \geq \frac{\rho}{\rho + \frac{\lambda^2}{d^2}(1-\rho)}.$$

Proof. Let $M = (a_{i,j})_{1 \leq i,j \leq 2n}$ be the adjacency matrix of G and let $X \subseteq A$ as in the theorem. The characteristic vector s of a vertex set $S \subseteq V$ is defined as the vector $(s_i)_{i \in V}$ where $s_i = 1$ if $v_i \in S$ and $s_i = 0$ otherwise. Let x be the characteristic vector of X. Let $e_1, \ldots, e_n$ be an orthonormal basis of eigenvectors of M corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$.

Notice that for a d-regular bipartite graph $|(e_1)_i| = |(e_{2n})_i| = \frac{1}{\sqrt{2n}}$, for $1 \le i \le 2n$ and $\lambda_1 = d$, $\lambda_{2n} = -d$, and $\lambda_i = -\lambda_{2n+1-i}$.

There exists $\alpha \in \mathbb{R}^{2n}$ such that x can be represented as $x = \sum_{i=1}^{2n} \alpha_i e_i$. Let $\beta_i = \alpha_i^2 + \alpha_{2n+1-i}^2$. Then

$$Mx = M\sum_{i=1}^{2n} \alpha_i e_i = \sum_{i=1}^{2n} \alpha_i M e_i = \sum_{i=1}^{2n} \alpha_i \lambda_i e_i$$

and

$$\begin{aligned}\langle Mx, Mx\rangle =&(\sum_{i=1}^{2n} \alpha_i\lambda_i e_i)^T(\sum_{i=1}^{2n} \alpha_i\lambda_i e_i)\\ =&\sum_{i=1}^{2n} \alpha_i^2\lambda_i^2 e_i^T e_i + \sum_{\substack{i\neq j\\ 1\le i,j\le 2n}} \alpha_i\alpha_j\lambda_i\lambda_j e_i^T e_j = \sum_{i=1}^{2n} \alpha_i^2\lambda_i^2 = \sum_{i=1}^{n} \beta_i\lambda_i^2.\end{aligned}$$

Thus, we have

$$\begin{aligned}&\|Mx\|^2\\ =&\beta_1\lambda_1^2 + \sum_{i=2}^{n} \beta_i\lambda_i^2\\ \le&\beta_1\lambda_1^2 + \lambda_2^2\sum_{i=2}^{n} \beta_i\\ =&\beta_1(\lambda_2^2 + \lambda_1^2 - \lambda_2^2) + \lambda_2^2\sum_{i=2}^{n} \beta_i\\ =&\beta_1(\lambda_1^2 - \lambda_2^2) + \lambda_2^2\sum_{i=1}^{n} \beta_i\\ =&(\alpha_1^2 + \alpha_{2n}^2)(\lambda_1^2 - \lambda_2^2) + \lambda_2^2\sum_{i=1}^{2n} \alpha_i^2\\ =&(\langle x, e_1\rangle^2 + \langle x, e_{2n}\rangle^2)(d^2 - \lambda_2^2) + \lambda_2^2\langle x, x\rangle\\ =&2|X|^2(\frac{1}{\sqrt{2n}})^2(d^2 - \lambda_2^2) + \lambda_2^2|X|\\ =&\frac{|X|^2}{n}(d^2 - \lambda_2^2) + \lambda_2^2|X|.\end{aligned}$$

Let y be the characteristic vector of $N(X)$. We have that

$$d^2|X|^2 = \langle Mx, \mathbb{1}\rangle = \langle Mx, y\rangle \le \|Mx\|^2\|y\|^2 \le \left(\frac{|X|^2}{n}(d^2 - \lambda_2^2) + \lambda_2^2|X|\right)|N(X)|.$$

From this the claim directly follows:

$$\frac{|N(X)|}{n} \geq \frac{d^2|X|^2}{|X|^2(d-\lambda^2)+n\lambda^2|X|} = \frac{d^2}{(d^2-\lambda^2)+\frac{\lambda^2}{\rho}} = \frac{\rho}{\rho+\frac{\lambda^2}{d^2}(1-\rho)}.$$

□

In order to apply Theorem 7 to the proof in [2] the following corollary is needed.

Corollary 1. *Let $G = (A \dot{\cup} B, E)$ be a d-regular bipartite graph. Let $n = |A| = |B|$ and $\lambda = \lambda(G)$. Then there is an edge between every two subsets $X \subseteq A$ and $Y \subseteq B$, provided that X and Y are of cardinality of at least $n\lambda/d$.*

Proof of Corollary 1. We show that $|N(X)|+|Y| > n$. Therefore $N(X)$ and Y share a common vertex. We have

$$\begin{aligned}|N(X)|+|Y| \geq n\frac{\frac{|X|}{n}}{\frac{|X|}{n}+\frac{\lambda^2}{d^2}(1-\frac{|X|}{n})}+|Y| &\geq n\frac{\frac{\lambda}{d}}{\frac{\lambda}{d}+\frac{\lambda^2}{d^2}(1-\frac{\lambda}{d})}+n\frac{\lambda}{d} \\ &= n\left(\frac{1}{1+\frac{\lambda}{d}-\frac{\lambda^2}{d^2}}+\frac{\lambda}{d}\right).\end{aligned}$$

It suffices to show that $\frac{1}{1+\frac{\lambda}{d}-\frac{\lambda^2}{d^2}}+\frac{\lambda}{d} > 1$:

$$\begin{aligned}\frac{1}{1+\frac{\lambda}{d}-\frac{\lambda^2}{d^2}}+\frac{\lambda}{d} = \frac{d^2}{d^2+\lambda d-\lambda^2}+\frac{\lambda}{d} &= \frac{d^3+d^2\lambda-\lambda^3+d\lambda^2}{d^3+d^2\lambda-d\lambda^2} \\ &\geq \frac{d^3+d^2\lambda-\lambda^3+d\lambda^2}{d^3+d^2\lambda-\lambda^3} > 1.\end{aligned}$$

□

5.1.2 Expansion and the Choice Number

A k-uniform hypergraph $H = (V, F)$ is an irreflexive symmetric relation F over a domain V, where exactly k elements are related at a time. Thus, the concept of hypergraphs is a generalisation of the concept of graphs. In this sense a colouring of a hypergraph is a labelling of all vertices of V such that there are no monochromatic edges.

Now we present the proof of the lower bound on the choice number from [2] with Corollary 1 plugged in, thus proving Theorem 6 at the beginning of this chapter.

Proof of Theorem 6. Let $G = (A \dot{\cup} B, E)$ be a d-regular bipartite graph with $n = |A| = |B|$. Set $\alpha = n\lambda/d$. Let $t(k)$ be the minimal number of edges in a k-uniform non-2-colourable hypergraph. It is shown in [11] that

$$t(k) \leq (1 + o(1))(e \ln(2)/4)2^k k^2.$$

Let

$$k = k(n, d) = \max\left\{ i \mid i\alpha \leq \left\lfloor \frac{n}{t(i)} \right\rfloor \right\}.$$

To simplify the notation we omit ceiling- and flooring-signs. We have

$$\begin{aligned} k &\geq \max\left\{ i \mid i \leq \frac{d}{\lambda\,(1+o(1))(\frac{e\ln(2)}{4})\,2^i\,i^2} \right\} \\ &= \max\left\{ i \mid i \leq \log\left(\frac{d}{\lambda}\right) - 3\log(i) - \log\left((1+o(1))\left(\frac{e\ln(2)}{4}\right)\right) \right\} \\ &\geq \max\left\{ i \mid i \leq \log\left(\frac{d}{\lambda}\right) - 3\log(i) \right\} = (1+o(1))\log\left(\frac{d}{\lambda}\right). \end{aligned}$$

Therefore it suffices to show that $ch(G) \geq k$. Let $t = t(k)$. Let $A = A_1 \dot{\cup} \ldots \dot{\cup} A_t$ ($B = B_1 \dot{\cup} \ldots \dot{\cup} B_t$) a partition of A (B), with $|A_i| \geq \lfloor \frac{n}{t} \rfloor$ ($|B_i| \geq \lfloor \frac{n}{t} \rfloor$) for $1 \leq i \leq t$. Let $H = (V, F = \{S_1, \ldots, S_t\})$ be a k-uniform hypergraph. Notice that H is not 2-colourable. Now we regard $V(H)$ as a set of colours and the edges of H as lists of colours that will be assigned to the vertices of $V(G)$. Let $L : V(G) \to F$ be a list-assignment to G, where for $v \in A_i \cup B_i$ $L(v) = S_i$ for all $1 \leq i \leq t$. We claim that there is no proper L-list-colouring of G. Assume a L-list-colouring $f : V(G) \to V(H)$ of G. Let C be defined as the set of all colours that appear at least α times as colour of a vertex in A. From the definition of k it follows that $|A_i| \geq \lfloor \frac{n}{t} \rfloor \geq k\alpha$. Because of $|S_i| = k$ there is at least one colour in S_i which is chosen at least α times on A_i by f for all $1 \leq i \leq t$. Therefore C intersects all edges of H. Consider the partition $V(H) = C \dot{\cup} (V(H) \setminus C)$. As H is not 2-colourable and every edge intersects C, there must be a monochromatic edge S^* with $S^* \subseteq C$. All colours in S^* form colour classes of at least α vertices in A. Conforming to the definition of the list-assignment L there is a set B^* in B with $L(B^*) = S^*$. Following the same argumentation as above for the classes $A_i (1 \leq i \leq t)$, there must be a colour c^* in S^* that is chosen more than α times on B^* by f. Hence, there is a set of at least α vertices coloured with c^* in B. On the other hand, as $c^* \in S^*$, there is a set of at least α vertices coloured with c^* in A. From Corollary 1 it follows that there is an edge between these two sets, thus preventing that f is a proper colouring. This shows that there is no proper L-list-colouring of G. Therefore $ch(G) > k$. □

5.2 Approximating the Choice Number in Expected Polynomial Time

To prove Theorem 3, we first need an upper and a lower bound on the choice number. The following two simple propositions bound the choice number by the chromatic number.

5.2.1 Bounding the Choice Number by the Chromatic Number

Proposition 3. *For every graph G the following holds.*

$$\chi(G) \leq ch(G) \tag{5.1}$$

Proof of Proposition 3. $ch(G) = k$ implies that for every assignment of lists of length k to the vertices of G there is a valid colouring of G. As a special case this holds if the same list is assigned to all vertices of G. □

Proposition 4. *For every graph G the following holds.*

$$ch(G) \leq \chi(G)\ln(n) + \epsilon, \tag{5.2}$$

where ϵ is an arbitrary (small) constant.

Proof of Proposition 4. To prove this proposition we use a probabilistic approach from [1]. Let $\chi = \chi(G)$. Because G is χ-colourable, there is a partition $V = V_1 \cup \cdots \cup V_\chi$ of G in χ colour classes. Let L be an arbitrary assignment of lists of length $\chi\ln(n) + \epsilon$ over a colour set $\mathfrak{L}$. Using the probabilistic method we show that there is a L-list-colouring of G. To do so $\mathfrak{L}$ is partitioned uniformly at random into classes $\mathfrak{L}_1, \ldots, \mathfrak{L}_\chi$. Now we show that with positive probability all vertices of a colour class V_i can be coloured with colours from $\mathfrak{L}_i$ for all $1 \leq i \leq \chi$. The probability that a colour class V_i contains a vertex $v \in V_i$ such that $\mathfrak{L}_i \cap L(v) = \emptyset$ is at most

$$n(1 - \frac{1}{\chi})^{\chi\ln(n)+\epsilon} \leq ne^{-\frac{1}{\chi}(\chi\ln(n)+\epsilon)} = e^{-\epsilon/\chi} < 1.$$

Therefore with positive probability all partitions can be coloured. □

5.2.2 The Algorithm ApproxCH

The combination of Theorem 2 with the above bounds on the choice number immediately yields an approximation algorithm for the choice number of a graph.

ApproxCH

```
Input:  a graph G
Output:  a number that approximates the choice number of G

  1. Execute the algorithm referred to in Theorem 2.

  2. Set l to the number of colours used by the obtained
     colouring and return l · (ln(n) + 1).
```

Proposition 5. *Let c be a sufficiently large constant. For $c/n < p \leq 0.99$ the algorithm* ApproxCH *approximates the choice number of a graph $G \in G_{n,p}$ within a ratio of $O(\sqrt{np}\ln(n)/\ln(np))$ and has expected polynomial running time over $G_{n,p}$.*

Proof of Proposition 5. Theorem 2 assures that there is an algorithm with expected polynomial running time over $G_{n,p}$ that returns a colouring within an approximation ratio of $O(\sqrt{np}/\ln(np))$. Therefore ApproxCH has polynomial expected running time over $G_{n,p}$. Proposition 4 assures that ApproxCH returns in any case an upper bound on ch. Together with the lower bound from Proposition 3 the achieved approximation ratio is bounded by

$$\frac{\text{ApproxCH}(G)}{ch(G)} \leq O\left(\chi(G)\frac{\sqrt{np}(\ln(n)+1)}{\ln(np)}\right) \cdot \frac{1}{\chi(G)} = O\left(\frac{\sqrt{np}\ln(n)}{\ln(np)}\right).$$

□

Chapter 6

The Eigenvalue Gap of $G_{n,p}$

In this chapter we give the proofs of Theorem 4 and Theorem 5. The chapter is rather technical and recommended only for a reader who is interested in the details. For the statements of the theorems and their relevance see Chapter 3.

6.1 The Eigenvalue Gap of $G_{n,p}$ for $np \gg 1$ (Proof of Theorem 4)

Proof of Theorem 4. The proof reduces the more general case $p = \Omega(\ln(n)/n)$ to the case for which Theorem 1 from [13] can be applied. For this reason we recall that theorem here:

Theorem 8 (Feige, Ofek (2003))**.** *Let G be a random graph in the $G_{n,p}$ model with $p = c_0 \ln(n)/n$, where c_0 is a large enough constant. Then $\lambda(G) = O(\sqrt{np})$ almost surely.*

Let $G \in G_{n,p}$ with $p = c\ln(n)/n$ for some $c > 0$ and let $A = A(G)$ be the adjacency matrix of G. We define $d = np = c\ln(n)$. Let c_0 be the constant from Theorem 1 in [13] and let $G_2 \in G_{n,p_2}$, $A_2 = A(G_2)$ with $p_2 = c_2 \ln(n)/n$, $c_2 \geq c_0$. Further we define $d_2 = np_2 = c_2 \ln(n)$. Finally let $A_3 = A + A_2$. The probability for an entry of A_3 to be 2 is $pp_2 = cc_2 \ln^2(n)/n^2$. Denote with D the graph $G \cap G_2$ and let $A_D = A(D)$ be the adjacency matrix of D. Then $A_D = 1$ if and only if $A_3 = 2$. Furthermore $D \in G_{n,pp_2}$.

We show that with high probability $\Delta(D) \leq 1$. For $v \in V(D)$ we have

$$\mathbb{P}[d(v) > 1] = \mathbb{P}[d(v) \geq 2] \leq \binom{n}{2}(pp_2)^2 \leq n^2 \cdot \frac{c^2 c_2^2 \ln^4(n)}{n^4} = \frac{c^2 c_2^2 \ln^4(n)}{n^2}.$$

For the probability that there is a vertex of degree greater than 1 the union bound yields an upper bound of

$$n \cdot \frac{c^2 c_2^2 \ln^4(n)}{n^2} = \frac{c^2 c_2^2 \ln^4(n)}{n} = o(1).$$

Hence, almost surely D is a matching and all eigenvalues of D are bounded by 1 in absolute value.

Let $A_3' = A_3 - A_D$ and let G_3' be the graph described by A_3'. Then $G_3' \in G_{n,p_3'}$ where $p_3' = p + p_2 - pp_2$. Thus, $p_2 < p_3' < p + p_2$. Since $c_2 \geq c_0$, the graphs G_2 and G_3' both satisfy the assumptions of Theorem 1 in [13]. In the following we apply that theorem as well as some lemmas from its proof to G_2 and G_3'. Let $d_3' = p_3' n$. Notice that $d_2 < d_3' < d + d_2$ and $O(\sqrt{d_3'}) = O(\sqrt{d_2}) = O(\sqrt{d})$.

In the following we show that $\lambda_2(A) \leq O(\sqrt{d})$ and $\lambda_n(A) \geq O(\sqrt{d})$. Since $\lambda_n(A) \leq \lambda_2(A)$ we thus proof $\lambda(A) = O(\sqrt{d})$.

First we bound $\lambda_2(A)$. Using the Courant-Fischer Theorem results in

$$\begin{aligned}
&\lambda_2(A) = \lambda_2(A_3' + D - A_2) \\
&= \max_{\substack{V \in \mathbb{R}^n \\ \dim V = 2}} \min_{\substack{x \\ x \in V \\ \|x\|=1}} x^t (A_3' + D - A_2) x \\
&\leq \max_{\substack{V \in \mathbb{R}^n \\ \dim V = 2}} (\min_{\substack{x \\ x \in V \\ \|x\|=1}} x^t A_3' x - \min_{\substack{x \\ x \in V \\ \|x\|=1}} x^t A_2 x) + \lambda_1(D) \\
&\leq \max_{\substack{V \in \mathbb{R}^n \\ \dim V = 2}} \min_{\substack{x \\ x \in V \\ \|x\|=1}} x^t A_3' x - \min_{\substack{V \in \mathbb{R}^n \\ \dim V = 2}} \min_{\substack{x \\ x \in V \\ \|x\|=1}} x^t A_2 x + \lambda_1(D) \\
&= \lambda_2(A_3') - \lambda_n(A_2) + \lambda_1(D) = O(\sqrt{d_3'}) + O(\sqrt{d_2}) + O(1) = O(\sqrt{d}).
\end{aligned}$$

To bound $\lambda_n(A)$ we use the same argument as in the proof of Lemma 2.1 in [13]. Let e be the eigenvector corresponding to $\lambda_n(A)$ in an orthonormal basis of eigenvectors of A. Let $u = \frac{1}{\sqrt{n}} \mathbb{1}$. There are a vector $w \perp u$, $\|w\| = 1$ and constants α and β such that $\alpha^2 + \beta^2 = 1$ and $\alpha u + \beta w = e$. Using the fact that all considered matrices are symmetric we get

$$\begin{aligned}
&\lambda_n(A) = \lambda_n(A_3' + D - A_2) = e^t (A_3' + D - A_2) e \\
&= (\alpha u + \beta w)^t (A_3' + D - A_2)(\alpha u + \beta w) \\
&= \alpha^2 (u^t (A_3 + D - A_2) u) \\
&\quad + 2\alpha\beta(u^t A_3' w + u^t D w - u^t A_2 w) \\
&\quad + \beta^2 (w^t A_3' w + w^t D w - w^t A_2 w) \\
&= \alpha^2 (u^t A u) \\
&\quad + O(u^t A_3' w) + O(1) + O(u^t A_2 w) \ . \\
&\quad + O(w^t A_3' w) + O(1) + O(w^t A_2 w)
\end{aligned} \tag{6.1}$$

Now we apply Lemma 2.2, Claim 2.4, and Theorem 2.5 from [13] to A_2 and A_3'. Further we use the fact that for $G_{n,p}$ the average degree and np are almost surely asymptotically equal. We get

$$\begin{aligned} u^t A_3' w &= O(\sqrt{d_3'}) \qquad & u^t A_2 w &= O(\sqrt{d_2}) \\ w^t A_3' w &= O(\sqrt{d_3'}) \qquad & w^t A_2 w &= O(\sqrt{d_2}). \end{aligned} \tag{6.2}$$

$u^t A u$ is the average degree of G and therefore it holds that $u^t A u \geq 0 = O(\sqrt{d})$. Applying this and equations (6.2) to equation (6.1) results in $\lambda_n(A) \geq O(\sqrt{d})$, thus finishing the proof. □

6.2 The Eigenvalue Gap of $G_{n,p}$ for $np = O(1)$ (Proof of Theorem 5)

As assumed in Theorem 5, during the whole proof I suppose that np is a sufficiently large constant. Two further random graph models are needed for the proof.

The $G_{n,m}$ and the $G^*_{n,m}$ Model

$G_{n,m}$ is the probability space of all graphs on n vertices with m edges (i.e., all subsets of $\binom{\{1,\ldots,n\}}{2}$ of cardinality m) together with the uniform distribution. Because in $G_{n,p}$ the probability of a graph depends only on the number of edges, $G_{n,m}$ is equal to $G_{n,p}$ under the condition $|E(G_{n,p})| = m$.

$G^*_{n,m}$ denotes the probability space of all m-tuples over the set $\binom{\{1,\ldots,n\}}{2}$, where every tuple M takes the probability $p(M) = 1/m^{\binom{n}{2}}$. Because an edge can appear more than once, $G^*_{n,m}$ is the random multigraph model, where a multigraph M is chosen uniformly at random from the set of all multigraphs on n vertices that collectively include m edges.

Two lemmas are needed, which relate the $G_{n,p}$ model to the $G^*_{n,m}$ model.

Lemma 2. *Let np be a constant. For $G^*_{n,m}$ with $m \leq \binom{n}{2}p + O(\sqrt{n}\ln(n))$ it holds that*

$$\mathbb{P}[G^*_{n,m} \textit{has only single edges}] = \Omega(1).$$

Proof of Lemma 2. Since there is a constant c such that $m \leq \binom{n}{2}p + cn$ for $n \geq 2$ it holds that

$$\begin{aligned}\frac{m^2}{\binom{n}{2}} \leq& \frac{(\binom{n}{2}p + cn)^2}{\binom{n}{2}} = \binom{n}{2}p^2 + 2pcn + \frac{n^2c^2}{\binom{n}{2}} \\ =& \frac{n(n-1)p^2}{2} + 2cnp + 2c^2\frac{n}{n-1} \leq \frac{(np)^2}{2} + 2cnp + 4c^2 = O(1).\end{aligned} \tag{6.3}$$

Using equation (6.3) the probability that $G^*_{n,m}$ has only single edges can be bounded by

$$\begin{aligned}&\mathbb{P}[G^*_{n,m}\text{has only single edges}] \\ =& \prod_{i=0}^{m-1}\frac{\binom{n}{2}-i}{\binom{n}{2}} = \prod_{i=1}^{m-1}\frac{\binom{n}{2}-i}{\binom{n}{2}} = \prod_{i=1}^{m-1}(1-\frac{i}{\binom{n}{2}}) \\ \geq& \left(1-\frac{m-1}{\binom{n}{2}}\right)^{m-1} \geq \exp\left(\left(-\frac{m-1}{\binom{n}{2}} - \left(\frac{m-1}{\binom{n}{2}}\right)^2\right)(m-1)\right) \\ =& \exp\left(-\frac{(m-1)^2}{\binom{n}{2}} - \frac{(m-1)^3}{\binom{n}{2}^2}\right) \geq \exp\left(-\frac{m^2}{\binom{n}{2}} - \frac{m^3}{\binom{n}{2}^2}\right) \\ \geq& \exp\left(-O(1) - O(1)\frac{m}{\binom{n}{2}}\right) \geq \exp\left(-O(1) - O(1)\frac{\binom{n}{2}p + cn}{\binom{n}{2}}\right) \\ =& \exp\left(-O(1) - O(1)(p + 2c\frac{1}{n-1})\right) = \exp(-O(1) - o(1)) \\ =& \Omega(1).\end{aligned}$$

□

Lemma 3. *Let np be a constant. Almost surely $G \in G_{n,p}$ has less than $\binom{n}{2}p + O(\sqrt{n}\ln(n))$ edges.*

Proof of Lemma 3. The number of edges of $G \in G_{n,p}$ is binomially distributed with parameters $\binom{n}{2}$ and p. Let c be an arbitrary constant. Applying a Chernoff-like bound for the binomial distribution (cf. [17]) yields

$$\begin{aligned}
&\mathbb{P}[|E(G_{n,p})| \geq \mathbb{E}[|E(G_{n,p})|] + c\sqrt{n}\ln(n)] \\
&\leq \exp\left(-\frac{(c\sqrt{n}\ln(n))^2}{2(\mathbb{E}[|E(G_{n,p})|] + \frac{c\sqrt{n}\ln(n)}{3})}\right) \\
&= \exp\left(-\frac{c^2 n \ln^2(n)}{2\binom{n}{2}p + \frac{2}{3}c\sqrt{n}\ln(n)}\right) \\
&\leq \exp\left(-\frac{c^2 n \ln^2(n)}{2n^2 p + \frac{2}{3}c\sqrt{n}\ln(n)}\right) \\
&\leq \exp\left(-\frac{c^2 \ln^2(n)}{2np + \frac{2}{3}c\frac{1}{\sqrt{n}}\ln(n)}\right) \\
&= \exp(-\Omega(\ln^2(n))) = o(1).
\end{aligned}$$

□

Proof of Theorem 5. Let np be a sufficiently large constant. As defined in [13] for a graph G and a vertex $v \in V(G)$ the predicate $bad(v)$ is true if and only if $d(v) \geq (1+\epsilon)np$.

Let G and G' be as defined in the theorem. Then G' is the graph obtained from G by removing all bad vertices. To prove the theorem one has to show that the number of bad vertices is small. In detail the proof of Theorem 1.2 in [13] turns out that the probability of $\lambda(G') = O(\sqrt{np})$ relies on the probability that the fraction of bad vertices in G is less than $e^{-\Omega(\epsilon^2 np)}$. In [13] this probability is bounded by $1 - e^{-\Omega(\epsilon^2 np)}$ using Markov's inequality. In the following we prove a stronger bound of $1 - o(1)$ on that probability. Plugging this bound into the proof of Theorem 1.2 in [13] immediately yields the result stated in Theorem 5.

Let $\gamma(G) := |\{v \in V(G) \mid bad(v)\}|/n$ describe the fraction of bad vertices of G. We have to show that $\mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np))] = o(1)$.

Let $m = \binom{n}{2}p + O(\sqrt{n}\ln(n))$. Let M be taken from $G^*_{n,m}$. In the following we will bound $\mathbb{P}[\gamma(M) \geq \exp(-O(\epsilon^2 np))]$ in the $G^*_{n,m}$ model.

In the $G^*_{n,m}$ model for a vertex v the random variable $d(v)$ is binomially distributed with parameters m and $(n-1)/\binom{n}{2}$. (There are m Bernoulli experiments where an experiment succeeds, when one of the $n-1$ possibly with v incident edges is chosen from all $\binom{n}{2}$ edges.) Therefore

$$\mathbb{E}[d(v)] = \frac{m(n-1)}{\binom{n}{2}} = (n-1)p + \frac{2O(\sqrt{n}\ln(n))}{n} = np + o(1).$$

Applying a Chernoff-like bound for the binomial distribution (cf. [17]) yields that a vertex $v \in V(M)$ is bad with a probability of $\exp(-\Omega(\epsilon^2 np))$:

$$\begin{aligned}
&\mathbb{P}[d(v) \geq np + \epsilon np] \leq \mathbb{P}[d(v) \geq (np + o(1)) + \epsilon np] \\
&= \mathbb{P}[d(v) \geq \mathbb{E}[d(v)] + \epsilon np] \leq \exp(-\frac{\epsilon^2 (np)^2}{2(np + o(1)) + 2/3\epsilon np}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad = \exp(-\Omega(\epsilon^2 np)).
\end{aligned}$$

(Notice that $o(1) = -o(1)$.) Thus, we have

$$\mathbb{E}[|\{v \in V(M) \mid bad(v)\}|] = n \exp(-\Omega(\epsilon^2 np))$$

and

$$\mathbb{E}[\gamma(M)] = \exp(-\Omega(\epsilon^2 np)) = O(1).$$

Now the Azuma-inequality can be applied to the edge exposure martingale of $G^*_{n,m}$. m edges are independently chosen from $\binom{n}{2}$. With each choice the value of the random variable γ changes at most $2/n$. The Azuma-inequality in its combinatorial formulation in [17] chapter 2 Corollary 2.27 yields:

$$\mathbb{P}[\gamma(M) \geq \mathbb{E}[\gamma] + t] \leq \exp(-\frac{2t^2}{m(2/n)^2}). \tag{6.4}$$

We want to show that $\mathbb{P}[\gamma(M) \geq \exp(-c\epsilon^2 np)] = o(1)$ for an arbitrarily large constant c. Let $\delta = \exp(-c\epsilon^2 np) - \mathbb{E}[\gamma(M)]$. Since $\epsilon^2 np$ is a constant we have $\delta = \Omega(1)$. The inequality (6.4) implies

$$\begin{aligned}
&\mathbb{P}[\gamma(M) \geq \exp(-c\epsilon^2 np)] \\
&= \mathbb{P}[\gamma(M) \geq \exp(-c\epsilon^2 np) + \mathbb{E}[\gamma(M)] - \mathbb{E}[\gamma(M)]] \\
&= \mathbb{P}[\gamma(M) \geq \mathbb{E}[\gamma(M)] + \delta] \\
&\leq \exp(-\frac{2\delta^2}{m(2/n)^2}) \leq \exp(-\frac{\delta^2 n^2}{2(\binom{n}{2}p + O(\sqrt{n}\ln(n)))}) \\
&\qquad\qquad \leq \exp(-\frac{\delta^2 n^2}{2(n^2 p + O(n))}) \leq \exp(-\frac{\Omega(1)n^2}{O(n)}) = o(1).
\end{aligned}$$

Using Lemma 2, the probability that $\gamma(G_{n,m})$ is larger than $\exp(-O(\epsilon^2 np))$

is bounded by $o(1)$:

$$
\begin{aligned}
&\mathbb{P}[\gamma(G_{n,m}) \geq \exp(-O(\epsilon^2 np))] \\
=&\mathbb{P}[\gamma(G^*_{n,m}) \geq \exp(-O(\epsilon^2 np)) \mid G^*_{n,m}\text{has only single edges}] \\
=&\frac{\mathbb{P}[\gamma(G^*_{n,m}) \geq \exp(-O(\epsilon^2 np)) \wedge G^*_{n,m}\text{has only single edges}]}{\mathbb{P}[G^*_{n,m}\text{has only single edges}]} \\
\leq&\frac{\mathbb{P}[\gamma(G^*_{n,m}) \geq \exp(-O(\epsilon^2 np))]}{\mathbb{P}[G^*_{n,m}\text{has only single edges}]} \\
=&\frac{o(1)}{\Omega(1)} = o(1).
\end{aligned}
$$

Finally Lemma 3 allows us to bound the probability of $\gamma(G) \geq \exp(-O(\epsilon^2 np))$ in the $G_{n,p}$ model. Let $F \in G_{n,m}$. Let $m(G)$ denote the number of edges of G. Observe that $\mathbb{P}[\gamma(G) \geq \exp(-O(e^2 np))|m(G) = m]$ increases as m increases. We get

$$
\begin{aligned}
&\mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np))] \\
=\ &\mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np)) \mid m(G) < m]\,\mathbb{P}[m(G) < m] \\
&+ \mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np)) \mid m(G) \geq m]\,\mathbb{P}[m(G) \geq m] \\
\leq\ &\mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np)) \mid m(G) = m]\,\mathbb{P}[m(G) < m] \\
&+ \mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np)) \mid m(G) \geq m]\,\mathbb{P}[m(G) \geq m] \\
=\ &\mathbb{P}[\gamma(F) \geq \exp(-O(\epsilon^2 np))]\,\mathbb{P}[m(G) < m] \\
&+ \mathbb{P}[\gamma(G) \geq \exp(-O(\epsilon^2 np)) \mid m(G) \geq m]\,\mathbb{P}[m(G) \geq m] \\
\leq\ &o(1) + o(1) = o(1).
\end{aligned}
$$

□

Chapter 7

Concluding Remarks

- The algorithm ApproxColour includes solving the semidefinite program SDP_k for calculating a lower bound on the chromatic number. The order of the number of constraints of SDP_k, however, is quadratic in n. Coja-Oghlan [7] shows that the *MAXCUT* relaxation SDP_2 could be used as well to lower bound the vector chromatic number $\overline{\vartheta}_{1/2}$, which itself is a lower bound on the chromatic number. This approach has the advantage that SDP_2 (in matrix notation) has only one constraint. Therefore it is possible that the semidefinite program can be solved for *reasonably* large instances. Experiments with different solvers for semidefinite programs have shown that sparse instances of up to 1000 vertices ($p < 0.1$) can be handled within a sufficient precision on a common personal computer (cf. Appendix A).

- Krivelevich and Vu [22] remark that it might be difficult to approximate the chromatic number within a ratio of less than $O(\sqrt{np}/\ln(np))$ in expected polynomial time for all values of p. They argue for the case $p = 1/2$ that an approximation ratio better than $O(\sqrt{np}/\ln(np))$ is tightly connected to the problem of distinguishing between a random graph taken from $G_{n,1/2}$ and a random graph from $G_{n,1/2}$ with an independent set of size $o(\sqrt{n})$ implanted, which they considerer a hard task.

- From the analysis of the algorithm GreedyCoreColour follows a slightly stronger approximation ratio than $O(np/\ln(np))$. Namely the algorithm GreedyCoreColour$(G, e^2np/\ln(np))$ returns a colouring that uses less than $28np/\ln(np)$ different colours or achieves an approximation ratio of 4.

- Coja-Oghlan and Taraz [9] point out that for $p \geq k/e^2n$ the algorithm CoreColour decides k-colourability in expected polynomial time over $G_{n,p}$. One might expect that a similar result applies to GreedyCoreColour, too. Unfortunately this is not true. CoreColour for $k \geq e^2 np$ returns a colouring that is either optimal or uses less than k colours. So if the algorithm returns a value greater than k the graph is definitively not k-colourable. This observation does not hold for GreedyCoreColour. As mentioned above GreedyCoreColour returns a colouring that uses *less* than $28np/\ln(np)$ colours or achieves an approximation ratio of 4.

Appendix A

Experimental Results for ϑ and $MAXCUT$ on $G_{n,p}$

It is a difficult problem to efficiently compute lower bounds on the chromatic number. Namely, for $k \geq 3$ it is co-$\mathcal{NP}$-hard to certificate that a graph is not k-colourable. Such a certificate has to prove that there is no valid k-colouring. As there are exponentially many k-colour-assignments to the vertices, it is not applicable to exclude all such assignments explicitly. It is obvious that the clique number is a lower bound on the chromatic number. However the clique number can not be approximated within a factor of $n^{1-\epsilon}$ in polynomial time unless $\mathcal{NP} = \mathcal{ZPP}$ (cf. [16]). Thus, it is quite noteworthy that there is an entity that is *sandwiched* (cf.[20]) between the clique number and the chromatic number of a graph and that moreover can be computed in polynomial time. The refered invariant of a graph is the so called *Lovász ϑ-function* of the complementary graph. It is defined as an eigenvalue optimisation problem over a certain matrix representation of the graph. The crucial fact is that it can be formulated as a semidefinite program and therefore the ellipsoid method can be applied. For this reason the value of ϑ can be computed in polynomial time. For a comprehensive introduction to the Lovász ϑ-function and the ellipsoid method refer to [14]. The complexity of the ellipsoid method is of merely theoretical interest. A comprehensive overview of algorithmic approaches to semidefinite programming can be found on the homepage of C. Helmberg (`http://www-user.tu-chemnitz.de/~helmberg/semidef.html`). There he also maintains a list of implementations of sdp-solvers. For the theoretical background of the algorithms we refer to [15].

The graph colouring algorithm presented in this thesis relies on a variant of the ϑ-function on $G_{n,p}$. However $\vartheta(G_{n,p})$ itself could have served as well as lower bound on the chromatic number. In the following we give some

n	$p = 1/n$	$p = 5/n$	$p = n^{-2/3}$	$p = n^{-1/2}$
100	*(p=0.01)* 71.5374	*(p=0.05)* 44.8219	*(p=0.046415)* 45.2731	*(p=0.1)* 32.1712
200	*(p=0.005)* 145.852	*(p=0.025)* 90.5587	*(p=0.029240)* 85.1326	*(p=0.070710)* 59.6544
300	*(p=0.003333)* 216.709	*(p=0.016666)* 137.08	*(p=0.022314)* 123.491	*(p=0.057735)* 83.2136
400	*(p=0.00250)* 292.5	*(p=0.0125)* 183.588	*(p=0.018420)* 161.027	*(p=0.05)* 106.273
500	*(p=0.002)* 363.537	*(p=0.01)* 230.224	*(p=0.015874)* 196.176	*(p=0.044721)* 128.184

Table A.1: Numerical data concerning the mean value of ϑ on sparse random Graphs.

experimental results for the value of ϑ on random graphs. The results were obtained using three different tools: *SBmethod*, *dsdp4*, and *csdp*, where the last two are primal-dual solvers, thus providing a guarantee on the quality of the result. This guarantee comes at the price of a worse performance. Therefore these tools were used only for checking/ calibrating the tool *SBmethod*, which even on relatively large and dense input instances converge relatively fast but does not guarantee on the quality of the result.

The Table A.1 shows the data from experiments on random graphs from $G_{n,p}$ within the range of $100 \leq n \leq 1000$ and $1/n \leq p \leq n^{-1/2}$.

It is conjectured that asymptotically almost surely $\vartheta(G_{n,p}) = c\sqrt{n(1-p)/p}$ where c is a constant with $1 \leq c \leq 2$. The numerical data does not support this conjecture. Figure A.1 shows the quotient of the numerical data on sparse random graphs within the same range as above and the conjectured value where $c = 1$.

In Table A.2 we present the results of some experiments on $G_{n,1/2}$. The results for dense graphs however have to be treated somewhat carefully. Due to the heavy consumption of CPU-time only few instances could be treated using *SBmethod* to calculate an upper bound on ϑ. Nevertheless ϑ is well enough concentrated that for $n \geq 500$ $\vartheta(G_{n,1/2})$ with high probability should only differ minimally from its mean. The data should represent upper bounds of this mean within an additive error of not more than 2 percent. The numerical data is compared to the conjectured value where $c = 1$. Therefore it seems that on $G_{n,p}$ the conjecture holds for approximately $c = 1$.

The algorithm ApproxColour (cf. Chapter 4) employs the semidefinite program SDP_k, which is a *MAX-k-CUT*-relaxation. We remark that it would be possible to use the *MAXCUT*-relaxation SDP_2, which is easier to compute, as well (cf. Chapter 7). Both concepts are closely related to ϑ

PSfrag replacements

$\vartheta(G \in G_{n,p})/(\sqrt{n(1-p)/p})$

1

Figure A.1: Numerical value of ϑ of sparse random graphs compared to the conjectured value.

n	$\vartheta(G \in G_{n,1/2})$	$\sqrt{n}$	$\vartheta(G \in G_{n,1/2})/\sqrt{n}$
500	22.590592	22.3606797749979	1.01028198727926
600	24.679417	24.4948974278318	1.00753297998948
700	26.706707	26.4575131106459	1.00941864370668
800	28.689802	28.2842712474619	1.01433767725497
900	30.354620	30	1.01182066666667
1000	31.750472	31.6227766016838	1.00403808305402

Table A.2: Numerical data concerning the mean value of $\vartheta(G_{n,1/2})$.

n	$MAXCUT(G \in G_{n,1/2})$	c
100	1437.21	0.39942
200	5672.02	0.492868
300	12447.1	0.475198
400	21944.8	0.4987
500	33762.6	0.460648
600	48258.1	0.453577
700	65454.5	0.463492
800	85362.4	0.482813
900	107671	0.483963
1000	132394	0.475543
2000	521535.02	0.487128
3000	1164114.5	0.480651

Table A.3: Numerical data concerning the mean value of $SDP_2(G_{n,1/2})$. (Column c contains numerical values of the constant c from the conjectured value of $MAXCUT(G_{n,1/2})$.)

and its variants. In Table A.3 we present some experimental data obtained by computing SDP_2 on instances of $G_{n,1/2}$.

It is conjectured that $MAXCUT(G_{n,p}) = 1/2\binom{n}{2}p + cn^{3/2}p^{1/2}(1-p)^{1/2}$ where $c \leq 1$ is a constant. Our numerical data indicates that for $p = 1/2$ the constant c lies in the interval between 0.46 and 0.49.

References

[1] Noga Alon, *Restricted colorings of graphs*, Proceedings of the 14th British Combinatorical Conference, London Mathematical Society Lecture Notes, vol. 187, 1993, pp. 1–33.

[2] Noga Alon and Michael Krivelevich, *The choice number of random bipartite graphs*, Annals of Combinatorics **2** (1998), 291–297.

[3] Noga Alon, Michael Krivelevich, and Benny Sudakov, *List coloring of random and pseudo-random graphs*, Combinatorica **19** (1999), no. 4, 453–472.

[4] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy, *Proof verification and the hardness of approximation problems*, Journal of the ACM **45** (1998), no. 3, 501–555.

[5] Béla Bollobás, *Random graphs*, 2 ed., Cambridge studies in advanced mathematics, vol. 73, Cambridge University Press, 2001.

[6] Gregory J. Chaitin, *Register allocation & spilling via graph coloring*, Proceedings of the ACM SIGPLAN 82 Symposium on Compiler Construction, 1982, pp. 98–105.

[7] Amin Coja-Oghlan, *The Lovász number of random graphs*, accepted for publication (preprint available from http://www.informatik.hu-berlin.de/∼coja/); preliminary version in Proceedings of the 7th International Workshop RANDOM 2003, 228–239.

[8] Amin Coja-Oghlan, Cristopher Moore, and Vishal Sanwalani, *MAX k-CUT and approximating the chromatic number of random graphs*, submitted for publication (preprint available from http://www.informatik.hu-berlin.de/∼coja/); preliminary version in Proceedings of International Colloquium on Automata, Languages and Programming 2003, 200–211, 2003.

[9] Amin Coja-Oghlan and Anusch Taraz, *Exact and approximative algorithms for coloring G(n,p)*, Random Structures and Algorithms **24** (2004), 259–278.

[10] Lars Engebretsen and Jonas Holmerin, *Towards optimal lower bounds for clique and chromatic number*, Theoretical Computer Science **299** (2003), no. 1–3, 573–584.

[11] Paul Erdös, A. L. Rubin, and H. Taylor, *Choosability in graphs*, Proceedings of the West-Coast Conference on Combinatorics, Graph Theory and Computing (Arcata, California), Congressus Numerantium, vol. 26, 1979, pp. 125–157.

[12] Uriel Feige and Joe Kilian, *Zero knowledge and the chromatic number*, Journal of Computer and System Sciences **57** (1998), no. 2, 187–199.

[13] Uriel Feige and Eran Ofek, *Spectral techniques applied to sparse random graphs*, Tech. Report MCS03-01, Weizmann Institute of Science, 2003.

[14] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver, *Geometric algorithms and combinatorial optimization*, Algorithms and Combinatorics, vol. 2, Springer, 1988.

[15] Christoph Helmberg, *Semidefinite programming*, European Journal Of Operational Research **137** (2002), no. 3, 461–482.

[16] Johan Håstad, *Clique is hard to approximate within n to the power of one minus epsilon*, Acta Mathematica **182** (1999), 105–142.

[17] Svante Janson, Tomasz Łuczak, and Andrezej Ruciński, *Random graphs*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley and Sons, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2000.

[18] Richard Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[19] Richard Karp, *The probabilistic analysis of combinatorial optimization algorithms*, Proceedings of International Congress of Mathematicians, 1984, pp. 1601–1609.

[20] Donald E. Knuth, *The sandwich theorem*, Electronic Journal of Combinatorics **1** (1994), Article 1, approx. 48 pp. (electronic).

[21] Michael Krivelevich, *The choice number of dense random graphs*, Combinatorics, Probability & Computing **9** (2000), no. 1, 19–26.

[22] Michael. Krivelevich and Van H. Vu, *Approximating the independence number and the chromatic number in expected polynomial time*, Journal of Combinatorial Optimization **6** (2002), 143–155.

[23] Eugene L. Lawler, *A note on the complexity of the chromatic number problem*, Information Processing Letters **5** (1976), 66–67.

[24] Frank Thomson Leighton, *A graph coloring algorithm for large scheduling problems*, Journal of Reasearch of the National Bureau of Standards **84** (1979), 489–506.

[25] Michael Molloy and Bruce Reed, *Graph colouring and the probabilistic method*, Algorithms and Combinatorics, vol. 23, Springer, 2002.

[26] Jan Seidel, *3-Färbbarkeit regulärer Graphen und des 3-Core zufälliger Graphen*, Master's thesis, Humboldt-Universität zu Berlin, 2003.

[27] Gregor Tanner, *Explicit construction of concentrators from generalized n-gons*, SIAM Journal on Discrete Mathematics **5** (1984), 287–293.

[28] Vadim G. Vizing, *Coloring the vertices of a graph in prescribed colors*, Diskret. Analiz. **29** (1976), 3–10 (Russian).