Approximate Graph Coloring by Semidefinite Programming

DAVID KARGER[*]
Stanford University

RAJEEV MOTWANI[†]
Stanford University

MADHU SUDAN[‡]
IBM Research

Abstract

We consider the problem of coloring k-colorable graphs with the fewest possible colors. We give a randomized polynomial time algorithm which colors a 3-colorable graph on n vertices with $\min\{O(\Delta^{1/3}\log^{4/3}\Delta), O(n^{1/4}\log n)\}$ colors where Δ is the maximum degree of any vertex. Besides giving the best known approximation ratio in terms of n, this marks the first non-trivial approximation result as a function of the maximum degree Δ. This result can be generalized to k-colorable graphs to obtain a coloring using $\min\{\tilde{O}(\Delta^{1-2/k}), \tilde{O}(n^{1-3/(k+1)})\}$ colors. Our results are inspired by the recent work of Goemans and Williamson who used an algorithm for semidefinite optimization problems, *which generalize linear programs, to obtain improved approximations for the MAX CUT and MAX 2-SAT problems. An intriguing outcome of our work is a duality relationship established between the value of the optimum solution to our semidefinite program and the Lovász ϑ-function. We show lower bounds on the gap between the optimum solution of our semidefinite program and the actual chromatic number; by duality this also demonstrates interesting new facts about the ϑ-function.*

1 Introduction

A legal vertex coloring of a graph $G(V,E)$ is an assignment of colors to its vertices such that no two adjacent vertices receive the same color. Equivalently, a legal coloring of G by k colors is a partition of its vertices into k independent sets. The minimum number of colors needed for such a coloring is called the chromatic number of G, and is usually denoted by $\chi(G)$. Determining the chromatic number of a graph is known to be NP-hard (cf. [19]).

Besides its theoretical significance as a canonical NP-hard problem, graph coloring arises naturally in a variety of applications such as register allocation [11, 12, 13] and timetable/examination scheduling [8, 40]. In many applications which can be formulated as graph coloring problems, it suffices to find an *approximately optimum* graph coloring—a coloring of the graph with a small though non-optimum number of colors. This along with the apparent impossibility of an exact solution has led to some interest in the problem of approximate graph coloring. The analysis of approximation algorithms for graph coloring started with the work of Johnson [25] who shows that a version of the greedy algorithm gives an $O(n/\log n)$-approximation algorithm for k-coloring. Wigderson [39] improved this bound by giving an elegant algorithm which uses $O(n^{1-1/(k-1)})$ colors to legally color a k-colorable graph. Subsequently, other polynomial time algorithms were provided by Blum [9] which use $O(n^{3/8}\log^{8/5} n)$ colors to legally color an n-vertex 3-colorable graph. This result generalizes to coloring a k-colorable graph with $O(n^{1-1/(k-4/3)}\log^{8/5} n)$ colors. The best known performance guarantee for general graphs is due to Halldórsson [24] who provided a polynomial time algorithm using a number of colors which is within a factor of $O(n(\log\log n)^2/\log^3 n)$ of the optimum.

Recent results in the hardness of approximations indicate that it may be not possible to substantially improve the results described above. Lund and Yannakakis [33] used the results of Arora, Lund, Motwani, Sudan, and Szegedy [6] and Feige, Goldwasser, Lovász, Safra, and Szegedy [17] to show that there exists a (small) constant $\epsilon > 0$ such that no polynomial time algorithm can approximate the chromatic number of a graph to within a ratio of n^ϵ unless P = NP. Recently, Bellare and Sudan [7] showed that the exponent ϵ in the hardness result can be improved to $1/10$ unless NQP $\neq$ co-RQP, and to $1/13$ unless NP = co-RP. However, none of these hardness results apply to the special case of the problem where the input graph is guaranteed to be k-colorable for some small k. The best hardness result in this direction is due to Khanna, Linial,

[*] Current address: AT&T Bell Laboratories, Murray Hill, NJ 07974 (karger@@lcs.mit.edu). This work was done while the author was at Stanford University and supported by a Hertz Foundation Graduate Fellowship and by NSF Young Investigator Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation and Xerox Corporation.

[†] Department of Computer Science, Stanford University, Stanford, CA 94305 (rajeev@cs.stanford.edu). Supported by an IBM Faculty Development Award, grants from Mitsubishi and OTL, NSF Grant CCR-9010517, and NSF Young Investigator Award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

[‡] IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (madhu@watson.ibm.com).

and Safra [26] who show that it is not possible to color a 3-colorable graph with 4 colors in polynomial time unless P $=$ NP.

In this paper we present improvements on the result of Blum. In particular, we provide a randomized polynomial time algorithm which colors a 3-colorable graph of maximum degree Δ with $\min\{\tilde{\mathrm{O}}(\Delta^{1/3}), O(n^{1/4}\log n)\}$ colors; moreover, this can be generalized to k-colorable graphs to obtain a coloring using $\tilde{\mathrm{O}}(\Delta^{1-2/k})$ or $\tilde{O}(n^{1-3/(k+1)})$ colors. Besides giving the best known approximations in terms of n, our results are the first non-trivial approximations given in terms of Δ. Our results are based on the recent work of Goemans and Williamson [20] who used an algorithm for *semidefinite optimization problems* (cf. [22, 2]) to obtain improved approximations for the MAX CUT and MAX 2-SAT problems. We follow their basic paradigm of using algorithms for semidefinite programming to obtain an optimum solution to a relaxed version of the problem, and a randomized strategy for "rounding" this solution to a feasible but approximate solution to the original problem. Motwani and Naor [35] have shown that the approximate graph coloring problem is closely related to the problem of finding a CUT COVER of the edges of a graph. Our results can be viewed as generalizing the MAX CUT approximation algorithm of Goemans and Williamson to the problem of finding an approximate CUT COVER. In fact, our techniques also lead to improved approximations for the MAX k-CUT problem [18]. We also establish a duality relationship between the value of the optimum solution to our semidefinite program and the Lovász ϑ-function [22, 23, 31]. We show lower bounds on the gap between the optimum solution of our semidefinite program and the actual chromatic number; by duality this also demonstrates interesting new facts about the ϑ-function.

Alon and Kahale [4] use related techniques to devise a polynomial time algorithm for 3-coloring random graphs drawn from a "hard" distribution on the space of all 3-colorable graphs. Recently, Frieze and Jerrum [18] have used a semidefinite programming formulation and randomized rounding strategy essentially the same as ours to obtain improved approximations for the MAX k-CUT problem with large values of k. Their results required a more sophisticated version of our analysis, but for the coloring problem our results are tight up to poly-logarithmic factors and their analysis does not help to improve our bounds.

Semidefinite programming relaxations are an extension of the linear programming relaxation approach to approximately solving NP-complete problems. We thus present our work in the style of the classical LP-relaxation approach. We begin in Section 2 by defining a relaxed version of the coloring problem. Since we use a more complex relaxation than standard linear programming, we must show that the relaxed problem can be solved; this is done in Section 3. We then show relationships between the relaxation and the original problem. In Section 4, we show that (in a sense to be defined later) the value of the relaxation bounds the value of the original problem. Then, in Sections 5, 6, 7, and 8 we show how a solution to the relaxation can be "rounded" to make it a solution to the original problem. Combining the last two arguments shows that we can find a good approximation. Section 3, Section 4, and Sections 5–8 are in fact independent and can be read in any order after the definitions in Section 2. In Section 9, we investigate the relationship between vector colorings and the Lovász ϑ-function, showing that they are in fact dual to one another. We investigate the approximation error inherent in our formulation of the chromatic number via semi-definite programming in Section 10.

2 A Vector Relaxation of Coloring

In this section, we describe the relaxed coloring problem whose solution is in turn used to approximate the solution to the coloring problem. Instead of assigning colors to the vertices of a graph, we consider assigning (n-dimensional) unit *vectors* to the vertices. To capture the property of a coloring, we aim for the vectors of adjacent vertices to be "different" in a natural way. The *vector k-coloring* that we define plays the role that a hypothetical "fractional k-coloring" would play in a classical linear-programming relaxation approach to the problem. Our relaxation is related to the concept of an orthonormal representation of a graph [31, 22].

Definition 2.1 *Given a graph $G = (V, E)$ on n vertices, a* vector k-coloring *of G is an assignment of unit vectors u_i from the space $\Re^n$ to each vertex $i \in V$, such that for any two adjacent vertices i and j the dot product of their vectors satisfies the inequality*

$$\langle u_i, u_j\rangle \leq -\frac{1}{k-1}.$$

The definition of an *orthonormal representation* [31, 22] requires that the given dot products be equal to zero, a weaker requirement than the one above.

3 Solving the Vector Coloring Problem

In this section we show how the vector coloring relaxation can be solved using semidefinite programming. The methods in this section closely mimic those of Goemans and Williamson [20].

To solve the problem, we need the following auxiliary definition.

Definition 3.1 *Given a graph $G = (V, E)$ on n vertices, a* matrix k-coloring *of the graph is an $n \times n$ symmetric positive semidefinite matrix M, with $m_{ii} = 1$ and $m_{ij} \leq -1/(k-1)$ if $\{i, j\} \in E$.*

We now observe that matrix and vector k-colorings are in fact equivalent (cf. [20]). Thus, to solve the vector coloring relaxation it will suffice to find a matrix k-coloring.

Fact 3.1 *A graph has a vector k-coloring if and only if it has matrix k-coloring. Moreover, a vector $(k+\epsilon)$-coloring can be constructed from a matrix k-coloring in time polynomial in n and $\log(1/\epsilon)$ time.*

Proof: Given a vector k-coloring $\{v_i\}$, the matrix k-coloring is defined by $m_{ij} = \langle v_i, v_j \rangle$. For the other direction, it is well known that for every symmetric positive definite matrix M there exists a square matrix U such that $UU^T = M$ (where U^T is the transpose of U). The rows of U are vectors $\{u_i\}_{i=1}^n$ that form a vector k-coloring of G.

An δ-close approximation to the matrix U can be found in time polynomial in n and $\log(1/\delta)$ can be found using the *Incomplete Cholesky Decomposition* [20, 21]. (Here by δ-close we mean a matrix U' such that $U'U'^T - M$ has L_∞ norm less than δ.) This in turn gives a vector $(k+\epsilon)$-coloring of the graph, provided δ is chosen appropriately. ■

Lemma 3.2 *If a graph G has a vector k-coloring then a vector $(k+\epsilon)$-coloring of the graph can be constructed in time polynomial in k, n, and $\log(1/\epsilon)$.*

Proof: Our proof is similar to those of Lovász [31] and Goemans-Williamson [20]. We construct a semidefinite optimization problem (SDP) whose optimum is $-1/(k-1)$ when k is the smallest real number such that a matrix k-coloring of G exists. The optimum solution also provides a matrix k-coloring of G.

$$\begin{array}{lllll}
\text{minimize} & \alpha & & & \\
\text{where} & \{m_{ij}\} \text{ is positive semidefinite} & & & \\
\text{subject to} & m_{ij} & \leq & \alpha & \text{if } (i,j) \in E \\
 & m_{ij} & = & m_{ji} & \\
 & m_{ii} & = & 1. &
\end{array}$$

Consider a graph which has a vector (and matrix) k-coloring. This means there is a solution to the above semidefinite program with $\alpha = -1/k - 1$. The ellipsoid method or other interior point based methods [22, 2] can be employed to find a feasible solution where the value of the objective is at most $-1/(k-1) + \delta$ in time polynomial in n and $\log 1/\delta$. This implies that for all $\{i, j\} \in E$, m_{ij} is at most $\delta - 1/(k-1)$, which is at most $-1/(k+\epsilon-1)$ for $\epsilon = 2\delta(k-1)^2$, provided $\delta \leq 1/2(k-1)$. Thus a matrix $(k+\epsilon)$-coloring can be found in time polynomial in k, n and $\log(1/\epsilon)$. From the matrix coloring, the vector coloring can be found in polynomial time as was noted in the previous lemma ■

4 Relating Original and Relaxed Solutions

In this section, we show that our vector coloring problem is a useful relaxation because the solution to it is related to the solution of the original problem. In order to understand the quality of the relaxed solution, we need the following geometric lemma:

Lemma 4.1 *For all positive integers k and n such that $k \leq n+1$, there exist k unit vectors in $\Re^n$ such that the dot product of any distinct pair is $-1/(k-1)$.*

Proof: We prove the claim by induction on k. The base case with $k = 2$ is proved by the one-dimensional vectors (1) and (-1). Now assume that we can find k vectors $v_1, \ldots, v_k$ such that $\langle v_i, v_j \rangle \leq -1/(k-1)$ for $i \neq j$. We use these vectors to create $u_1, \ldots, u_{k+1}$ as follows. For $i \leq k$, let

$$u_i = \left(\frac{\sqrt{(k-1)(k+1)}}{k} v_i^1, \ldots, \frac{\sqrt{(k-1)(k+1)}}{k} v_i^k, -\frac{1}{k} \right),$$

where v_i^j denotes the jth component of the vector v_i. In other words, u_i contains $-1/k$ in the new coordinate and looks like v_i (scaled to make u_i a unit vector) in the old coordinates. The final vector $u_{k+1} = (0, \ldots, 0, 1)$.

Observe that the dot-product of any vector u_i with u_{k+1} is $-1/k$. Moreover, for distinct $i, j \leq k$,

$$\begin{aligned}
\langle u_i, u_j \rangle &= \frac{(k-1)(k+1)}{k^2} \langle v_i, v_j \rangle + \frac{1}{k^2} \\
&= \frac{-(k-1)(k+1)}{k^2(k-1)} + \frac{1}{k^2}
\end{aligned}$$

which is also equal to $-1/k$. ■

Corollary 4.2 *Every k-colorable graph G has a vector k-coloring.*

Proof: Bijectively map the k colors to the k vectors defined in the previous lemma. ■

Note that a graph is vector 2-colorable if and only if it is 2-colorable. Lemma 4.1 is tight in that it provides the best possible value for minimizing the mutual dot-product of k unit vectors. This can be seen from the following lemma.

Lemma 4.3 *Let G be vector k-colorable and let i be a vertex in G. The induced subgraph on the vertices $\{j \mid$ j is a neighbor of i in $G\}$ is vector $(k-1)$-colorable.*

Proof: Let $v_1, \ldots, v_n$ be a vector k-coloring of G and assume without loss of generality that $v_i = (1, 0, 0, \ldots, 0)$. Associate with each neighbor j of i a vector v'_j obtained by projecting v_j onto coordinates 2 through n and then scaling it up so that v'_j has unit length. It suffices to show that for any two adjacent vertices j and j' in the neighborhood of i, $\langle v'_j, v'_{j'} \rangle \leq -1/(k-2)$.

Observe first that the projection of v_j onto the first coordinate is negative and has magnitude at least $1/(k-1)$. This implies that the scaling factor for v'_j is at least $\sqrt{(k-1)/(k-2)}$. Thus,

$$\langle v'_j, v'_{j'} \rangle \leq \frac{k-1}{k-2}(\langle v_j, v_{j'} \rangle - \frac{1}{(k-1)^2}) \leq \frac{-1}{k-2}.$$

■

A simple induction using the above lemma shows that any graph containing a $(k+1)$-clique is not k-vector colorable. Thus the "vector chromatic number" lies between between the clique and chromatic number. This also shows that the analysis of Lemma 4.1 is tight in that $-1/(k-1)$ is the minimum possible value of the maximum of the dot-products of k vectors.

In the next few sections we prove the harder part, namely, if a graph has a vector k-coloring then it has an $\tilde{O}(n^{1-3/(k+1)})$-coloring.

5 Semicolorings

Given the solution to the relaxed problem, our next step is to show how to "round" the solution to the relaxed problem in order to get a solution to the original problem. Both of the rounding techniques we present in the following sections produce the coloring by working through an almost legal *semicoloring* of the graph, as defined below.

Definition 5.1 *A* k-semicoloring *of a graph G is an assignment of k colors to the vertices such that at most $|V(G)|/4$ edges are incident on two vertices of the same color.*

Any constant larger than 2 can replace 4 in the denominator in the above definition. An algorithm for semicoloring leads naturally to a coloring algorithm.

Lemma 5.1 *If an algorithm A can k_i-semicolor any i-vertex subgraph of graph G in polynomial time, where k_i increases with i, then A can be used to $O(k_n \log n)$-color G. Furthermore, if there exists $\epsilon > 0$ such that for all i, $k_i = \Omega(i^\epsilon)$, then A can be used to color G with $O(k_n)$ colors.*

Proof: We show how to construct a coloring algorithm A' to color any subgraph H of G. A' starts by using A to semicolor H. Let S be the subset of vertices which have at least one improperly colored edge incident to them. Observe that $|S| \leq |V(H)|/2$. A' fixes the colors of vertices not in S, and then recursively colors the induced subgraph on S using a new set of colors.

Let c_i be the maximum number of colors used by A' to color any i-vertex subgraph. Then c_i satisfies the recurrence

$$c_i \leq c_{i/2} + k_i$$

It is easy to see that this any c_i satisfying this recurrence, must satisfy $c_i \leq k_i \log i$. In particular this implies that $c_n \leq O(k_n \log n)$. Furthermore for the case where $k_i = \Omega(i^\epsilon)$ the above recurrence is satisfied only when $c_i = \Theta(k_i)$. ■

Using the above lemma, we devote the next few sections to algorithms for transforming vector colorings into semicolorings.

6 Rounding via Hyperplane Partitions

We now focus our attention on vector 3-colorable graphs, leaving the extension to general k for later. Let Δ be the maximum degree in a graph G. In this section, we outline a randomized rounding scheme for transforming a vector 3-coloring of G into an $O(\Delta^{\log_3 2})$-semicoloring, and thus into an $O(\Delta^{\log_3 2} \log n)$-coloring of G. Combining this method with Wigderson's technique yields an $O(n^{0.386})$-coloring of G. The method is based on [20] and is weaker than the method we describe in the following section; however, it introduces several of the ideas we will use in the more powerful algorithm.

Assume we are given a vector 3-coloring $\{u_i\}_{i=1}^n$. Recall that the unit vectors u_i and u_j associated with an adjacent pair of vertices i and j have a dot product of at most $-1/2$, implying that the angle between the two vectors is at least $2\pi/3$ radians or 120 degrees.

Definition 6.1 *Consider a hyperplane H. We say that H* separates *two vectors if they do not lie on the same side of the hyperplane. For any edge $\{i, j\} \in E$, we say that the hyperplane H* cuts *the edge if it separates the vectors u_i and u_j.*

In the sequel, we use the term *random hyperplane* to denote the unique hyperplane containing the origin and having as its normal a random unit vector v uniformly distributed on the unit sphere S_n. The following lemma is a restatement of Lemma 1.2 of Goemans-Williamson [20].

Lemma 6.1 (Goemans-Williamson [20]) *Given two vectors at an angle of θ, the probability that they are separated by a random hyperplane is exactly θ/π.*

We conclude that for any edge $\{i, j\} \in E$, the probability that a random hyperplane cuts the edge is exactly $2/3$. It follows that the expected fraction of the edges in G which are cut by a random hyperplane is exactly $2/3$. Suppose that we pick r random hyperplanes independently. Then, the probability that an edge is not cut by one of these hyperplanes is $(1/3)^r$, and the expected fraction of the edges not cut is also $(1/3)^r$.

We claim that this gives us a good semicoloring algorithm for the graph G. Notice that r hyperplanes can partition $\Re^n$ into at most 2^r distinct regions. (For $r \leq n$ this is tight since r hyperplanes create exactly 2^r regions.) An edge is cut by one of these r hyperplanes if and only if the vectors associated with its end-points lie in distinct regions. Thus, we can associate a distinct color with each of the 2^r regions and give each vertex the color of the region containing its vector. The expected number of edges whose end-points have the same color is $(1/3)^r m$, where m is the number of edges in E.

Theorem 6.2 *If a graph has a vector* 3*-coloring, then it has an* $O(\Delta^{\log_3 2})$*-semicoloring which can be constructed from the vector* 3*-coloring in polynomial time with high probability.*

Proof: We use the random hyperplane method just described. Fix $r = 2 + \lceil \log_3 \Delta \rceil$, and note that $(1/3)^r \leq 1/9\Delta$ and that $2^r = O(\Delta^{\log_3 2})$. As noted above, r hyperplanes chosen independently at random will cut an edge with probability $1/9\Delta$. Thus the expected number of edges which are not cut is $m/9\Delta \leq n/18 \leq n/8$, since the number of edges is at most $n\Delta/2$. By Markov's inequality, the probability that the number of uncut edges is more than twice the expected value is at most $1/2$. But if the number of uncut edges is less than $n/4$ then we have a semicoloring.

Repeating the entire process t times means that we will find a $O(\Delta^{\log_3 2})$-semicoloring with probability at least $1 - 1/2^t$. ■

Noting that $\log_3 2 < 0.631$ and that $\Delta \leq n$, this theorem and Lemma 5.1 implies a semicoloring using $O(n^{0.631})$ colors. However, this can be improved using the following idea due to Wigderson [39]. Fix a threshold value δ. If there exists a vertex of degree greater than δ, pick any one such vertex and 2-color its neighbors (its neighborhood is vector 2-colorable and hence 2-colorable). The colored vertices are removed and their colors are not used again. Repeating this as often as possible (or until half the vertices are colored) brings the maximum degree below δ at the cost of using at most $2n/\delta$ colors. Thus, we can obtain a semicoloring using $O(n/\delta + \delta^{0.631})$ colors. The optimum choice of δ is around $n^{0.613}$, which implies a semicoloring using $O(n^{0.387})$ colors. This semicoloring can be used to legally color G using $O(n^{0.387})$ colors by applying Lemma 5.1.

Corollary 6.3 *A 3-colorable graph with n vertices can be colored using $O(n^{0.387})$ colors by a polynomial time randomized algorithm.*

By varying the number of hyperplanes, we can arrange for a tradeoff between the number of colors used and the number of edges that violate the resulting coloring. This may be useful in some applications where a nearly legal coloring is good enough.

The bound just described is (marginally) weaker than the guarantee of a $O(n^{0.375})$ coloring due to Blum [9]. We now improve this result by constructing a semicoloring with fewer colors.

7 Rounding via Vector Projections

This section is dedicated to proving the following more powerful version of Theorem 6.2.

Theorem 7.1 *If a graph has a vector k-coloring, then it has an $\tilde{O}(\Delta^{1-2/k})$-semicoloring that can be constructed from the vector coloring with high probability in polynomial time.*

As in the previous section, this has immediate consequences for approximate coloring.

We prove this theorem by analyzing a new method for assigning colors to vertices which provides a significantly better semicoloring than the hyperplane partition method. The idea is to pick t random *centers* $c_1, \ldots, c_t \in \Re^n$ and use them to define a set of t colors, say $1, \ldots, t$. Consider any vertex i and let u_i be its associated unit vector from a vector coloring. We color vertex i according to the center "nearest" to vector u_i, i.e. the center with the largest projection onto u_i.

Definition 7.1 *Given any fixed vector a, we say that a center c_j* captures *a if for all $i \neq j$,*

$$\langle c_i, a \rangle < \langle c_j, a \rangle.$$

Note that this definition allows for some vertices not to be captured by any vector, but this happens with probability approaching 0 in the limit.

Observe that the centers need not be of equal length and thus the nearest center to a may not be the one of minimum angle displacement from a. Each vector u_i is captured by a unique center and the index of that center is assigned to vertex i as its color. Clearly, this gives a t-coloring of the vertices of G, but this need not be a legal coloring or even a good partial coloring in general. However, it is intuitive that since the vectors corresponding to the endpoints of an edge are "far apart," it is unlikely that both are captured by the same center; thus, as in the hyperplane rounding method, an edge is likely to be cut by the coloring. We formalize this intuition and show how to pick centers so that the resulting coloring is indeed a semicoloring with high probability.

Our basic plan for choosing centers is to give each center a "direction" selected uniformly at random in $\Re^n$. The most obvious method for doing this might be to choose the vector uniformly from the points on the unit sphere in $\Re^n$. Instead, we choose each center c_j independently at random from the n-dimensional normal distribution. This means that each of the n components of c_j is independently chosen from the standard normal distribution with expectation 0 and variance 1. The reason for this choice of the distribution will become clear shortly. Notice that the lengths of these vectors are random, and so they are not unit vectors. It turns out that the limiting behavior of the random unit vector approach is exactly the same as for the one we use, but it is more difficult to analyze.

We now give an overview of how and why this assignment of centers gives a semicoloring. As before, the problem reduces to showing that the probability that an edge is cut by the assignment of colors is high, which in turn reduces to showing that the two endpoints of an edge are unlikely to be captured by the same center. In particular, suppose we have a graph with an n-dimensional vector k-coloring. Suppose we throw in t random centers and use them to assign colors as described above. By definition, the dot product between the unit vectors assigned to the endpoints of an edge is $-1/(k-1)$. Let $P_k(n,t)$ be the probability that two such widely separated vectors are captured by the same center. The technical work of this section shows that

$$P_k(n,t) \approx t^{-k/(k-2)}.$$

Given this fact, we can use the same techniques as the hyperplane rounding scheme to construct a semicoloring. Take t to be about $\Delta^{1-2/k}$. Then $P_k(n,t)$ is about $1/\Delta$. Using the same approach as with the hyperplane rounding method, this gives us a semicoloring with t colors.

We now discuss the analysis of $P_k(n,t)$. This probability is just t times the probability that both endpoints of an edge are captured by a particular center, say the first. To show this probability is small, note that regardless of the orientation of the first center it must be "far" from at least one of the two vectors it is trying to capture, since these two vectors are far from each other. For example, in the case of a vector 3-coloring any center must be at an angle of at least 60° from one of the endpoints of an edge. The center's projection onto this distant vector is very small, making it likely that some other nearer center will have a larger projection, thus preventing the center from capturing that far away vector.

We have therefore reduced our analysis to the problem of determining the probability that a center at a large angle from a given vector captures that vector. We start by deriving some useful properties of the normal distribution. In particular, we show that the properties of the normal distribution allow us to reduce the n-dimensional problem under consideration to a two dimensional one. But first, we develop some technical tools which will be applied to the two-dimensional analysis.

7.1 Probability Distributions in $\Re^n$

Recall that the *standard normal distribution* has the density function $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ with distribution function $\Phi(x)$, mean 0, and variance 1. A random vector $r = (r_1, \ldots, r_n)$ is said to have the *n-dimensional standard normal distribution* if the components r_i are independent random variables, each component having the standard normal distribution. It is easy to verify that this distribution is spherically symmetric, in that the direction specified by the vector r is uniformly distributed. (Refer to Feller [15, v. II], Knuth [29, v. 2], and Rényi [36] for further details about the higher dimensional normal distributions.)

Subsequently, the phrase "random d-dimensional vector" will always denote a vector chosen from the d-dimensional standard normal distribution. A crucial property of the normal distribution which motivates its use in our algorithm is the following theorem paraphrased from Rényi [36] (see also Section III.4 of Feller [15, v. II]).

Theorem 7.2 (Theorem IV.16.3 [36])
Let $r = (r_1, \ldots, r_n)$ be a random n-dimensional vector. The projections of r onto two lines ℓ_1 and ℓ_2 are independent (and normally distributed) if and only if ℓ_1 and ℓ_2 are orthogonal.

Alternatively, we can say that under any rotation of the coordinate axes, the projections of r along these axes are independent standard normal variables. In fact, it is known that the only distribution with this strong spherical symmetry property is the n-dimensional standard normal distribution. The latter fact is precisely the reason behind this choice of distribution[1] in our algorithm. In particular, we

[1] Readers familiar with physics will see the connection to Maxwell's law on the distribution of velocities of molecules in $\Re^3$. Maxwell started with the assumption that in *every* Cartesian coordinate system in $\Re^3$, the

will make use of the following corollary to the preceding theorem.

Corollary 7.3 *Let $r = (r_1, \ldots, r_n)$ be a random vector (of i.i.d. standard normal variables). Suppose we fix two orthogonal unit vectors u_1 and u_2 in $\Re^n$. The projections of r along these two directions, given by the dot products $\langle u_1, r\rangle$ and $\langle u_2, r\rangle$, are independent random variables with the standard normal distribution.*

It turns out that even if r is a random n-dimensional *unit* vector, the above lemma still holds in the limit: as n grows, the projections of r on orthogonal lines approach (scaled) independent normal distributions. Thus using random unit vectors for centers turns out to be equivalent to using random normal vectors in the limit, but is much more difficult to analyze.

The following two lemmas are also useful in our analysis. The first lemma states that the square of the length of a random vector in two dimensions has the exponential distribution with parameter $1/2$. Recall that the exponential distribution with parameter λ has density function $f(x) = \lambda e^{-\lambda x}$, distribution function $F(x) = 1 - e^{-\lambda x}$ and expectation $1/\lambda$.

Lemma 7.4 *Let X and Y be standard normal random variables. Then, the random variable $S = X^2 + Y^2$ has the exponential distribution with parameter $\lambda = 1/2$.*

Lemma 7.5 *Let Y_1, ..., Y_r, and X have the exponential distribution with parameter $\lambda = 1/2$. Then the probability of the event $\mathcal{E}$ that $\{X \geq q \times \max_i Y_i\}$ is*

$$\binom{r+q}{r}^{-1},$$

where $\binom{r+q}{r}$ is the generalized binomial coefficient when q is not necessarily an integer.

Notice that the probability bound is essentially r^{-q} for large r. In our application, $q = 1/\cos^2\omega$ where ω is half the angle between the endpoints of an edge. Since for vector 3-colorings $\omega = \pi/3$, we have $\cos\omega = 1/2$, $q = 4$ and the probability bound is $1/r^4$.

7.2 Analyzing the Vector Projection Algorithm

We are now ready to analyze the quality of the partial coloring obtained by using the projections of random vectors to color the vertices of G. The first step in the analysis is to determine a tight bound on the probability that for a specific edge $\{x, y\}$ the two endpoints receive the same color. Let u_x and u_y denote the unit vectors associated with the two vertices. Recall that the angle between these two vertices is at least $2\pi/3$. Note that the bad event happens when the same random center, say c_1, captures both u_x and u_y. We will show that this is unlikely to happen if the number of centers is large.

Fix any two unit vectors a and b in $\Re^n$ such that they subtend an angle of $2\pi/3$ (as do the vectors of adjacent vertices in a vector 3-coloring). We will study the probability of the bad event with respect to these vectors, and by the spherical symmetry of the normal distribution our analysis will apply to the case of two vertex vectors u_x and u_y. The crucial step in this analysis is a reduction to a two-dimensional problem, as follows. Note that the use of the n-dimensional normal distribution was motivated entirely by the need to facilitate the following lemma.

Lemma 7.6 *Let θ be such that $\cos\theta = -1/(k-1)$. Let $P_k(d,t)$ denote the probability of the event that, given any two vectors a, $b \in \Re^d$ subtending an angle of θ, they are both captured by the same member of a collection of t random centers in $\Re^d$. Then, for all $d \geq 2$ and all $t \geq 1$,*

$$P_k(d,t) = P_k(2,t).$$

Proof: Let $H(a,b)$ be the plane determined by the two vectors a and b. Rotate the coordinate axes so that the first two axes lie in this plane and all other axes are perpendicular to it. By Corollary 7.3, we can still view the random vectors as having been chosen by picking their components along the new axes as standard normal random variables. Now, the projection of any vector in $\Re^d$ onto any line of this plane depends only on its components along the two coordinate axes lying in the plane. In other words, any event depending only on the projection of the random vectors onto the lines in this plane does not depend on the components of the vectors along the remaining $d-2$ axes. In particular, the probability $P_k(d,t)$ is the same as $P_k(2,t)$. ■

In the rest of this section, we will assume that all vectors are in $\Re^2$, and by the preceding lemma the resulting analysis will apply to the case of n-dimensional vectors. We focus on the case where the angle between the vectors a and b is $2\pi/3$ and thus bound $P_3(n,t)$, but the analysis generalizes easily to other values of k as well.

Theorem 7.7 *For $0 < \epsilon < \pi/3$, let $p = \epsilon/\pi$, $\theta = \pi/3 - \epsilon$, and $q = 1/\cos^2\theta$. Then,*

$$P_3(n,t) = P_3(2,t) = O(tp^{q-\lceil q\rceil}(pt)^{-q}).$$

three components of the velocity vector are mutually independent and had expectation zero. Applying this assumption to rotations of the axes, we conclude that the velocity components must be independent normal variables with identical variance. This immediately implies Maxwell's distribution on the velocities.

Proof: We will concentrate on bounding the probability that the first random vector, c_1, captures both a and b; clearly, multiplying this by t will give the desired probability. Note that any vector must subtend an angle of at least $\pi/3$ with one of the two vectors a and b. Assume that c_1 subtends a larger angle with a, and hence is at least $\pi/3$ radians away from it. Now, c_1 captures a only if none of the remaining $t-1$ vectors has a larger projection onto a. We will bound the probability of this event from above. A similar argument applies in the case where b is further away from c_1.

Let R denote the wedge of the plane within an angle of ϵ from a, and suppose that r centers fall in this region. If c_1 captures a, then its projection onto a must be larger than that of the r centers in R. In fact, it is easy to see that the projection of c_1 onto the nearer of the two lines bounding R must be larger than the lengths of all the centers in R. (Observe that the latter is a necessary, but not sufficient, condition for c_1 to capture a.) Essentially this corresponds to the event $\mathcal{F}$ that the projection of c_1 onto a line at an angle of $\theta = \pi/3 - \epsilon$ is longer than the lengths of all the centers lying in R.

We will upper bound the probability of the event $\mathcal{F}$. If r random vectors fall into the region R, then by Lemma 7.5 we know that the probability of $\mathcal{F}$ is given by $\binom{r+q}{r}^{-1}$, where $q = 1/\cos^2\theta$. Since the random vectors have a spherically symmetric distribution, the number of random vectors lying in R has the binomial distribution $B(t,p)$ with $p = \epsilon/\pi$. Thus, we obtain the following bound on the probability of $\mathcal{F}$. In the first step of the derivation, we use an identity given in Exercise 1.2.6 (20) of Knuth's book [29, v. 1], which applies to generalized binomial coefficients.

$$
\begin{aligned}
\mathrm{P}[\mathcal{F}] &= \sum_{r=0}^{t}\binom{t}{r}p^r(1-p)^{t-r}\cdot\binom{r+q}{r}^{-1}\\
&= \binom{t+q}{t}^{-1}\sum_{r=0}^{t}\binom{t+q}{t-k}p^r(1-p)^{t-r}\\
&= \binom{t+q}{t}^{-1}\sum_{u=0}^{t}\binom{t+q}{u}p^{t-u}(1-p)^u\\
&\le \binom{t+q}{t}^{-1}\sum_{u=0}^{t}\binom{t+\lceil q\rceil}{u}p^{t-u}(1-p)^u\\
&= p^{-\lceil q\rceil}\binom{t+q}{t}^{-1}\sum_{u=0}^{t}\binom{t+\lceil q\rceil}{u}p^{t+\lceil q\rceil-u}(1-p)^u\\
&\le p^{q-\lceil q\rceil}\left(p^q\binom{t+q}{t}\right)^{-1}(p+(1-p))^{t+\lceil q\rceil}\\
&= O(p^{q-\lceil q\rceil}(pt)^{-q})
\end{aligned}
$$

By the preceding argument, multiplying this by t gives a bound on the probability $P_k(n,t)$.

The reason for introducing $\lceil q\rceil$ is that there are two problems with directly applying the binomial theorem of calculus: for one, we are outside the radius of convergence of the infinite sum; and for the other, the infinite sum has negative terms so we cannot immediately make claims about the first few terms being less than the whole sum. ■

The above theorem applies regardless of how we choose ϵ (thus determining p and q). We now show how t and ϵ should be chosen so as to ensure that we get a semicoloring.

Corollary 7.8 $P_3(2,t) = O(t^{-3}\log^4 t)$.

Proof: We set $\epsilon = 1/\log t$. Thus $p = 1/(\pi\log t)$. To get q, we use the Taylor expansions for sines and cosines. In fact, the particular constants do not matter: it suffices to note that $q = 1/\cos^2(\pi/3-\epsilon) = 4 - O(\epsilon)$. Thus, $q - \lceil q\rceil = O(\epsilon)$ and

$$p^{q-\lceil q\rceil} = \epsilon^{-\Theta(\epsilon)} = \log^{-\Theta(1/\log t}t = \Theta(1).$$

By Theorem 7.7 we have

$$
\begin{aligned}
P_3(2,t) &= O(t(pt)^{-q})\\
&= O\left(t(t\log t)^{-4(1-O(1/\log t))}\right)\\
&= O(t^{-3}\log^4 t).
\end{aligned}
$$

■

Lemma 7.9 *With high probability, the vector projection method provides an $O(\Delta^{1/3}\log^{4/3}\Delta)$-semicoloring of a 3-colorable graph G with maximum degree Δ.*

Proof: We use $t = \Delta^{1/3}\log^{4/3}\Delta$ random vectors and apply the above corollary. It follows that the probability that a particular edge is not legally colored is at most $O(1/\Delta)$. Thus the expected number of edges which are not legally colored is at most $O(n)$, and can be made less than $n/4$ by proper choice of constants. ■

As in Theorem 6.2, we now apply the idea of finding an independent set of linear size and recursively coloring the remaining graph.

Theorem 7.10 *A vector 3-colorable graph G with n vertices and maximum degree Δ can be colored with $O(\Delta^{1/3}\log^{4/3}\Delta\log n)$ colors by a polynomial time randomized algorithm (with high probability).*

As in Corollary 6.3, we now use Wigderson's technique (with $\Delta = n^{3/4}/\log n$) to get a $O(n^{1/4}\log n)$-semicoloring of any vector 3-colorable graph. The next result follows from an application of Lemma 5.1.

Theorem 7.11 *A vector 3-colorable graph G with n vertices can be colored with $O(n^{1/4}\log n)$ colors by a polynomial time randomized algorithm (with high probability).*

The analysis of the vector projection algorithm given above is tight to within polylogarithmic factors. A tighter analysis, due to Coppersmith [14], shows that the number of colors used by this algorithm is $\Theta((n\log n)^{1/4})$.

8 Approximation for k-Colorable Graphs

An easy generalization of the above shows that for any constant vector-chromatic number χ, we can color a graph of maximum degree Δ using $\Delta^{1-2/\chi+o(1)}$ colors. The only change is in the degree of separation between the vectors of the endpoints of an edge. Suppose a graph is χ-colorable. Then it is vector χ-colorable, meaning we can assign unit vectors so that the vectors on the endpoints of an edge have dot-product at most $-1/(\chi-1)$. We round these vectors with the same approach of using random centers. The only change in the analysis is in determining the probability that with t random centers, the same center will capture both endpoints of an edge. This analysis is a generalization of Theorem 7.7, where now $\theta = \frac{1}{2}\arccos(1/(\chi-1)) - \epsilon$, so that $q = 1/\cos^2\theta \approx 2(\chi-1)/(\chi-2)$. We deduce that the probability that an edge is cut is approximately $t^{-\chi/(\chi-2)}$ so that $\Delta^{1-2/\chi+o(1)}$ centers suffice to give a semicoloring.

Ignoring the $o(1)$ term, we determine absolute approximation ratios independent of Δ. We identify a positive real function $r(\chi)$ such that we can color a vector χ-chromatic graph with at most $n^{r(\chi)}$ colors. For each χ, we establish a degree threshold $\Delta_\chi = \Delta_\chi(n)$. While the degree exceeds Δ_χ, we take a neighborhood of a vertex of degree $d \geq \Delta_\chi$ and recursively $d^{r(\chi-1)}$-color it and discard it (by Lemma 4.3 the neighborhood is vector $(\chi-1)$-chromatic). The average number of colors used per vertex in this process is $d^{r(\chi-1)-1} \leq \Delta_\chi^{r(\chi-1)-1}$. Thus the total number of colors used up in this process is at most $n\Delta_\chi^{r(\chi-1)-1}$ colors. Once the degree is less than Δ_χ, we use our coloring algorithm directly to use an additional $\Delta_\chi^{1-2/\chi}$ colors. We balance the colors used in each part by setting

$$n\Delta_\chi^{r(\chi-1)-1} = \Delta_\chi^{1-2/\chi}$$

which implies that

$$\begin{aligned} n &= \Delta_\chi^{2-2/\chi-r(\chi-1)}, \\ \Delta_\chi &= n^{1/(2-2/\chi-r(\chi-1))}. \end{aligned}$$

We obtain a coloring with $n^{(1-2/\chi)/(2-2/\chi-r(\chi-1))}$ colors, in other words

$$r(\chi) = (1-2/\chi)/(2-2/\chi-r(\chi-1)).$$

By substitution, $r(\chi) = 1 - 3/(\chi+1)$.

Theorem 8.1 *A vector χ-colorable graph can be colored using $\tilde{O}(\Delta^{1-2/\chi})$ or $\tilde{O}(n^{1-3/(\chi+1)})$ colors.*

9 Duality Theory

The most intensively studied relaxation of a semidefinite programming formulation to date is the Lovász ϑ-function [22, 23, 31]. This relaxation of the clique number of a graph led to the first polynomial-time algorithm for finding the clique and chromatic numbers of perfect graphs. We now investigate a connection between ϑ and a close variant of the vector chromatic number.

Intuitively, the clique and coloring problems have a certain "duality" since large cliques prevent a graph from being colored with few colors. Indeed, it is the equality of the clique and chromatic numbers in perfect graphs which lets us compute both in polynomial time. We proceed to formalize this intuition. The duality theory of linear programming has an extension to semidefinite programming. With the help of Eva Tardos and David Williamson, we have shown that in fact the ϑ-function and a close variant of the vector chromatic number are semidefinite programming duals to one another and are therefore equal.

We first define the variant.

Definition 9.1 *Given a graph $G = (V, E)$ on n vertices, a* strict vector k-coloring *of G is an assignment of unit vectors u_i from the space $\Re^n$ to each vertex $i \in V$, such that for any two adjacent vertices i and j the dot product of their vectors satisfies the equality*

$$\langle u_i, u_j \rangle = -\frac{1}{k-1}.$$

As usual we say that a graph is strictly vector k-colorable if it has a strict vector k-coloring. The strict vector chromatic number of a graph is the smallest real number k for which it has a strict vector k-coloring. It follows from the definition that the strict vector chromatic number of any graph is lower bounded by the vector chromatic number.

Theorem 9.1 *The strict vector chromatic number of G is equal to $\vartheta(\overline{G})$.*

Proof: The dual of our strict vector coloring semidefinite program is as follows (cf. [2]):

$$\begin{array}{lll} \text{maximize} & -\sum p_{ii} & \\ \text{where} & \{p_{ij}\} \text{ is positive semidefinite} & \\ \text{subject to} & \displaystyle\sum_{i \neq j} p_{ij} \leq 1 & \\ & p_{ij} = p_{ji} & \\ & p_{ij} = 0 & \text{for } (i,j) \notin E \text{ and } i \neq j \end{array}$$

By duality, the value of this SDP is $-1/(k-1)$ where k is the strict vector chromatic number. Our goal is to prove $k = \vartheta$. As before, the fact that $\{p_{ij}\}$ is positive semidefinite means we can find vectors v_i such that $p_{ij} = \langle v_i, v_j \rangle$. The last constraint says that the vectors v form an *orthogonal labeling* [23], i.e. that $\langle v_i, v_j \rangle = 0$ for $(i,j) \notin E$. We now claim that optimization problem can be reformulated as follows:

$$\text{maximize } \frac{-\sum \langle v_i, v_i \rangle}{\sum_{i \neq j} \langle v_i, v_j \rangle}$$

over all orthogonal labelings $\{v_i\}$. To see this, consider an orthogonal labeling and define $\mu = \sum_{i \neq j} \langle v_i, v_j \rangle$. Note this is the value of the first constraint in the first formulation of the dual (so $\mu \leq 1$) and of the denominator in the second formulation. Then in an optimum solution to the first formulation, we must have $\mu = 1$, since otherwise we can divide each v_i by $\sqrt{\mu}$ and get a feasible solution with a larger objective value. Thus the optimum of the second formulation is at least as large as that of the first. Similarly, given any optimum $\{v_i\}$ for the second formulation, $v_i/\sqrt{\mu}$ forms a feasible solution to the first formulation with the same value. Thus the optima are equal. We now manipulate the second formulation.

$$\begin{aligned}
\max \frac{-\sum \langle v_i, v_i \rangle}{\sum_{i \neq j} \langle v_i, v_j \rangle} &= \max \frac{-\sum \langle v_i, v_i \rangle}{\sum_{i,j} \langle v_i, v_j \rangle - \sum \langle v_i, v_i \rangle} \\
&= \left(\min \frac{\sum_{i,j} \langle v_i, v_j \rangle - \sum \langle v_i, v_i \rangle}{-\sum \langle v_i, v_i \rangle} \right)^{-1} \\
&= \left(\min - \frac{\sum_{i,j} \langle v_i, v_j \rangle}{\sum \langle v_i, v_i \rangle} + 1 \right)^{-1} \\
&= -\left(\max \frac{\sum_{i,j} \langle v_i, v_j \rangle}{\sum \langle v_i, v_i \rangle} - 1 \right)^{-1}.
\end{aligned}$$

It follows from the last equation that the vector chromatic number is

$$\max \frac{\sum_{i,j} \langle v_i, v_j \rangle}{\sum \langle v_i, v_i \rangle}.$$

However, by the same argument as used to reformulate the dual, this is equal to problem of maximizing $\sum_{i,j} \langle v_i, v_j \rangle$ over all orthogonal labelings such that $\sum \langle v_i, v_i \rangle \leq 1$. This is simply Lovász's ϑ_3 formulation of the ϑ-function [23, page 287]. ■

10 The Gap between Vector Colorings and Chromatic Numbers

The performance of our randomized rounding approach seems far from optimum. In this section we ask why, and show that the problem is not in the randomized rounding but in the gap between the original problem and its relaxation. We investigate the following question: given a vector k-colorable graph G, how large can its chromatic number be in terms of k and n? We will show that a graph with chromatic number $n^{\Omega(1)}$ can have bounded vector chromatic number. This implies that our technique is tight in that it is not possible to guarantee a coloring with $n^{o(1)}$ colors on all vector 3-colorable graphs.

Definition 10.1 *The Kneser graph $K(m,r,t)$ is defined as follows: the vertices are all possible r-sets from a universe of size m; and, the vertices v_i and v_j are adjacent if and only if the corresponding r-sets satisfy $|S_i \cap S_j| < t$.*

We will need following theorem of Milner [34] regarding intersecting hypergraphs. Recall that a collection of sets is called an antichain if no set in the collection contains another.

Theorem 10.1 (Milner) *Let $S_1, \ldots, S_\alpha$ be an antichain of sets from a universe of size m such that, for all i and j,*

$$|S_i \cap S_j| \geq t.$$

Then, it must be the case that

$$\alpha \leq \binom{m}{\frac{m+t+1}{2}}.$$

Notice that using all q-sets, for $q = (m+t+1)/2$, gives a tight example for this theorem.

The following theorem establishes that the Kneser graphs have a large gap between their vector chromatic number and chromatic numbers.

Theorem 10.2 *Let $n = \binom{m}{r}$ denote the number of vertices of the graph $K(m,r,t)$. For $r = m/2$ and $t = m/8$, this graph is 3-vector colorable but has chromatic number $n^{0.0113}$.*

Proof: We prove a lower bound on the Kneser graph's chromatic number χ by establishing an upper bound on its independence number α. It is easy to verify that the α in Milner's theorem is exactly the independence number of the Kneser graph. We can bound χ as follows, using the standard equality that

$$\binom{a}{b} = \Theta\left(\left(\frac{a}{b}\right)^b \left(\frac{a}{a-b}\right)^{a-b} \right)$$

for b linearly related to a. For the purposes of determining the exponent in the chromatic number, the constant factor hidden in the Θ-notation can and will be ignored. We now observe that

$$\begin{aligned}\chi &\geq \frac{n}{\alpha}\\ &\geq \frac{\binom{m}{r}}{\binom{m}{(m+t)/2}}\\ &= \left[\frac{(2)^{1/2}(2)^{1/2}}{(16/9)^{9/16}(16/7)^{7/16}}\right]^{m}\\ &= (1.007864)^{m}.\end{aligned}$$

Again using the approximation,

$$n = \binom{m}{r} = \left[(2)^{1/2}(2)^{1/2}\right]^{m} \approx 2^{m}.$$

Since $n \approx \lg m$, it follows that

$$\chi \geq (1.007864)^{\lg n} = n^{\lg 1.007864} \approx n^{0.0113}.$$

Finally, it remains to show that the vector chromatic number of this graph is 3. This follows by associating with each vertex v_i an m-dimensional vector obtained from the characteristic vector of the set S_i. In the characteristic vector, $+1$ represents an element present in S_i and -1 represents elements absent from S_i. The vector associated with a vertex is the characteristic vector of S_i scaled down by a factor of $\sqrt{m}$ to obtain a unit vector. It is easy to see that the dot product of adjacent vertices, or sets with intersection at most t, is bounded from above by

$$-\frac{4r - 4t - m}{m} = -1/2.$$

This implies that the vector chromatic number is 3. ■

More refined calculations can be used to improve this bound somewhat. The basic idea is to improve the bound on the vector chromatic number of the Kneser graph using an appropriately weighted version of the characteristic vectors.

Theorem 10.3 *There exists a Kneser graph $K(m, r, t)$ which is 3-vector colorable but has chromatic number exceeding $n^{0.016101}$, where $n = \binom{m}{r}$ denotes the number of vertices in the graph. Further, for large k, there exists a Kneser graph $K(m, r, t)$ which is k-vector colorable but has chromatic number exceeding $n^{0.0717845}$.*

11 Conclusions

The Lovász number of a graph has been a subject of active study due to the close connections between this parameter and the clique and chromatic numbers. In particular, the following "sandwich theorem" was proved by Lovász [31] (see Knuth [30] for a survey).

$$\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G). \qquad (1)$$

This has led to the hope that the following extended version may be true.

Conjecture 11.1 *There exist ϵ, $\epsilon' > 0$ such that, for any graph G on n vertices*

$$\frac{\vartheta(\overline{G})}{n^{1-\epsilon}} \leq \omega(G) \leq \vartheta(\overline{G}) \leq \chi(G) \leq \vartheta(\overline{G}) \times n^{1-\epsilon'}. \qquad (2)$$

Our work in this paper provides reinforcement for this hope by giving an upper bound on the the chromatic number of G in terms of $\vartheta(\overline{G})$. However, this is far from achieving the bound conjectured above and it remains to be seen if this conjecture is true. In related work, Szegedy [37] studies various aspects of the parameter ϑ and, with respect to this conjecture, shows that there is such an ϵ bounded away from zero if and only if there is an ϵ' bounded away from zero. Alon, Kahale and Szegedy [5] have also been able to use the semidefinite programming technique in conjunction with our techniques to obtain algorithms for computing bounds on the clique number of a graph with linear-sized cliques, improving upon some results due to Boppana and Halldorsson [10].

In terms of disproving such a conjecture (or, proving upper bounds on ϵ and ϵ'), relevant results include the following: Lovász [32] points out that for a random graph G, $\chi(G) = n/\log n$ while $\vartheta(\overline{G}) = \sqrt{n}$; Koniagin has demonstrated the existence of a graph which has $\chi(G) \geq n/2$ and $\vartheta(\overline{G}) = O(n^{2/3}\log n)$; Alon [3] has explicit constructions matching or slightly improving both these bounds. Our constructions from Section 10 are of a similar flavor and provide graphs with vector chromatic number at most 3 but with $\chi(G) \geq n^{\epsilon}$. In fact, by using a similar construction and applying a result of Frankl and Rodl [16], we can also construct graphs with $\vartheta(\overline{G}) \leq 3$ and $\chi(G) \geq n^{\epsilon}$. Independent of our results, Szegedy [38] has also shown that a similar construction yields graphs with vector chromatic number at most 3 but which are not colorable using $n^{0.05}$ colors. Notice that the exponent obtained from his result is better than the one in Section 10. Alon [3] has obtained a slight improvement over Szegedy's bound by using an interesting variant of the Kneser graph construction.

The connection between the vector chromatic number and the clique/chromatic numbers is far from being completely understood and it is our hope that this work will motivate further study of this relationship.

Acknowledgements

Thanks to David Williamson for giving us a preview of the MAX-CUT result [20] during a visit to Stanford. We are indebted to John Tukey and Jan Pedersen for their help in understanding multidimensional probability distributions. Thanks to David Williamson and Eva Tardos for

discussions of the duality theory of SDP. Finally, we thank Noga Alon, Don Coppersmith, Laci Lovász and Mario Szegedy for useful discussions.

References

[1] D. Aldous. *Probability Approximations via the Poisson Clumping Heuristic.* Springer-Verlag, New York, 1989.

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. In *Proceedings of the 2nd MPS Conference on Integer Programming and Combinatorial Optimization*, Carnegie-Mellon University, 1992.

[3] N. Alon. Personal Communication, August 1994.

[4] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. In *Proceedings of the Twenty-Sixth ACM Symposium on Theory of Computing*, pp. 346–353, 1994.

[5] N. Alon, N. Kahale and M. Szegedy. Personal Communication, August 1994.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof Verification and Hardness of Approximation Problems. In *Proceedings 33rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 14–23, 1992.

[7] M. Bellare and M. Sudan. Improved Non-approximability Results. In *Proceedings of the Twenty-Sixth ACM Symposium on Theory of Computing*, pp. 184–193, 1994.

[8] C. Berge. **Graphs and Hypergraphs**. North-Holland, Amsterdam, 1973.

[9] A. Blum. New approximation algorithms for graph coloring. *Journal of the ACM*, 41:470–516, 1994.

[10] R.B. Boppana and M.M. Halldorsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32:180–196, 1992.

[11] P. Briggs, K.D. Cooper, K. Kennedy, and L. Torczon. Coloring heuristics for register allocation. In *Proceedings of the SIGPLAN 89 Conference on Programming Language Design and Implementation*, pp. 275–274, 1989.

[12] G.J. Chaitin. Register allocation and spilling via graph coloring. In *Proceedings of the SIGPLAN 89 Conference on Compiler Construction*, pp. 98–101, 1982.

[13] G.J. Chaitin, M.A. Auslander, A.K. Chandra, J. Cocke, M.E. Hopkins, and P.W. Markstein. Register allocation via coloring. *Computer Languages*, 6:47–57, 1981.

[14] D. Coppersmith. Personal Communication, March 1994.

[15] William Feller. **An Introduction to Probability Theory and Its Applications**. John Wiley & Sons, New York, 1968.

[16] P. Frankl and V. Rodl. Forbidden Intersections. *Transactions of the American Mathematical Society*, 300:259–286, 1994.

[17] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pp. 2–12, 1991.

[18] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. Manuscript, June 1994.

[19] Michael R. Garey and David S. Johnson. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. W.H. Freeman, 1979.

[20] Michel X. Goemans and David P. Williamson. $.878$-Approximation Algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the Twenty-Sixth ACM Symposium on Theory of Computing*, pp. 422-431, 1994.

[21] Gene H. Golub and C. F. Van Loan. **Matrix Computations**. Johns Hopkins University Press, Baltimore, 1983.

[22] M. Grötschel, L. Lovász and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[23] M. Grötschel, L. Lovász and A. Schrijver. **Geometric Algorithms and Combinatorial Optimization.** Springer-Verlag, Berlin, 1987.

[24] M.M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45:19–23, 1993.

[25] D. S. Johnson. Worst case behavior of graph coloring algorithms. *Proc. 5th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Congressus Numerantium X, 513–527, 1974.

[26] S. Khanna, N. Linial, and S. Safra. On the Hardness of Approximating the Chromatic Number. In *Proceedings 2nd Israeli Symposium on Theory and Computing Systems*, pp. 250–260, 1992.

[27] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On Syntactic versus Computational Views of Approximability. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, 1994 (to appear).

[28] M. Kneser. Aufgabe 300. *Jber. Deutsch. Math.-Verein.*, 58, 1955.

[29] Donald E. Knuth. **The Art of Computer Programming**. Addison-Wesley, Reading, MA, 1971.

[30] Donald E. Knuth. The Sandwich Theorem. *The Electronic Journal of Combinatorics*, 1:1–48, 1994.

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

[32] L. Lovász. Personal Communication, March 1994.

[33] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pp. 286–293, 1993.

[34] E.C. Milner. A combinatorial theorem on systems of sets. *Journal of the London Mathematical Society*, 43:204–206, 1968.

[35] R. Motwani and J. Naor. On Exact and Approximate Cut Covers of Graphs. Manuscript, 1993.

[36] A. Rényi. *Probability Theory*. Elsevier, New York, 1970.

[37] M. Szegedy. A note on the θ number of Lovász and the generalized Delsarte bound. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994 (to appear).

[38] M. Szegedy. Personal Communication. March 1994.

[39] A. Wigderson. Improving the Performance Guarantee for Approximate Graph Coloring. *Journal of the ACM*, 30:729–735, 1983.

[40] D.C. Wood. A Technique for Coloring a Graph Applicable to Large-Scale Optimization Problems. *Computer Journal*, 12:317, 1969.