Coloring 2-colorable Hypergraphs with a Sublinear Number of Colors

Noga Alon
School of Mathematical Sciences
The Raymond and Beverly Sacker Faculty of Exact Sciences
Tel Aviv University
Ramat Aviv, Tel Aviv 69978
Israel
alon@math.tau.ac.il

Pierre Kelsen, Sanjeev Mahajan
Max–Planck–Institut für Informatik
D-66123 Saarbrücken, Germany
mahajan@mpi-sb.mpg.de

Hariharan Ramesh
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India ramesh@csa.iisc.ernet.in

November 29, 1996

Abstract

A coloring of a hypergraph is a mapping of vertices to colors such that no hyperedge is monochromatic. We are interested in the problem of coloring 2-colorable hypergraphs. For the special case of graphs (hypergraphs of dimension 2) this can easily be done in linear time. The problem for general hypergraphs is much more difficult since a result of Lovasz implies that the problem is NP-hard even if all hyperedges have size three.

In this paper we develop approximation algorithms for this problem. Our first result is an algorithm that colors any 2-colorable hypergraph on n vertices and dimension d with $O(n^{1-1/d}\log^{1-1/d} n)$ colors. This is the first algorithm that achieves a sublinear number of colors in polynomial time. This algorithm is based on a new technique for reducing degrees in a hypergraph that should be of independent interest. For the special case of hypergraphs of dimension three we improve on the previous result by obtaining an algorithm that uses only $O(n^{2/9}\log^{\frac{17}{8}} n)$ colors. This result makes essential use of semidefinite programming. This last result is rather surprising because we show that semidefinite programming will fail for any larger dimension.

Key words. Approximation algorithms, hypergraphs, semidefinite programming.

1 Introduction

A hypergraph $H = (V, E)$ consists of a finite set V of *vertices (or nodes)* (whose number is denoted by n) and a collection E of non-empty subsets of V called *edges*. The *dimension* of H, denoted by d, is the maximum size of an edge in E. Thus, a graph is a hypergraph of dimension 2. A k-coloring of a hypergraph is a mapping $\phi : V \to \{1, \ldots, k\}$ such that no edge of H has all vertices of the same color. A hypergraph is k-colorable if it admits a k-coloring.

We consider the problem of coloring hypergraphs that are known to be 2-colorable. Although we can 2-color a 2-colorable (or bipartite) graph in linear time using a straightforward depth-first search algorithm, the problem is NP-hard for hypergraphs even if all edges have size at most 3. (This problem is a special case of *Set Splitting* and was shown to be NP-hard by Lovasz ([?]; see also [?])).

In this paper we give the first polynomial time algorithm for coloring 2-colorable hypergraphs. Our algorithm runs in polynomial time and uses $O(n^{1-1/d} \log^{1-1/d} n)$ colors. This result is obtained by combining a simple randomized algorithm with a new technique for reducing the degrees in a hypergraph. The latter technique is inspired by a similar method that was developed by Wigderson ([?]) for coloring 3-colorable graphs.

For the special case of hypergraphs of dimension 3 we reduce the number of colors to $O(n^{2/9} \log^{\frac{17}{8}} n)$. We achieve this bound by using a powerful new technique based on semidefinite programming and pioneered by Goemans and Williamson ([?]). The technique involves relaxing an integer program (whose solution is NP-hard) into a semidefinite program (for which an approximate solution can be found in polynomial time). Using an approach based on semidefinite programming we get an algorithm that uses only $O(n^{2/9} \log^{\frac{17}{8}} n)$ colors. The degree reduction technique is also used for this result.

The semidefinite programming technique was first successfully used by Goemans and Williamson to obtain a solution to the Max-Cut problem with approximation ratio .878. The best previous ratio was 1/2 obtained by a straightforward random cut algorithm. Using the same technique Goemans and Williamson also derived improved approximations for the MAX-2SAT problem.

Karger, Motwani and Sudan ([?]) used the semidefinite programming to obtain better coloring algorithms for 3-colorable graphs. In an elegant paper they achieve an $O(n^{1/4} \log n)$ bound on the number of colors needed to color a 3-colorable graph (in random polynomial time). The previous best bound was $O(n^{2/5})$ colors achieved by a deterministic polynomial time algorithm of Blum ([?]).

Frieze and Jerrum ([?]) gave a .65 approximation algorithm for MaxBisection based on semidefinite programming, improved form the .5 ratio achieved by the random bisection algorithm. Finally, Alon and Kahale ([?]) use semidefinite programming to achieve better approximations to the maximum independent set problem.

In this paper we apply the technique for the first time to a problem on hypergraphs. This application is rather surprising because we show that the technique will fail for any dimension higher than 3.

The paper is organized as follows: in the next section we present a new technique for reducing the maximum degree of a hypergraph. We present this result first because it will be used throughout the rest of the paper. In section ?? we describe an algorithm that colors 2-colorable hypergraphs with $o(n)$ colors in polynomial time. Only elementary techniques are used for this result: the key ingredients are the degree reduction technique from section ?? and a simple randomized algorithm. In sections ?? and ?? we show how semidefinite programming can be used in conjunction with the degree reduction technique to get better colorings for 2-colorable hypergraphs of dimension 3. In the concluding section ?? we argue that this algebraic technique cannot be extended to hypergraphs of higher dimensions and show the limitations of semidefinite programming based approach for dimension 3 hypergraphs.

Since the conference version of this paper appeared [?], we found out that similar results have been obtained independently by Chen and Frieze in [?].

2 Degree Reduction

Wigderson ([?]) showed that by coloring some subset of vertices in a 3-colorable graph with $O(\sqrt{n})$ colors, the subgraph induced by the uncolored vertices can be made to have maximum degree $O(\sqrt{n})$. In this section we shall show a similar type of result for hypergraphs of arbitrary dimension. As the number of edges in hypergraphs of arbitrary dimension can be exponential in n, we do not represent them explicitly, but with the means of an independent set oracle, that is, given a subset of vertices S, the oracle answers whether it is independent or not. Our algorithm may be viewed as an extension of Wigderson's technique to hypergraphs. Together with the results of the following sections it will yield polynomial time algorithms that color 2-colorable hypergraph with a sublinear number of colors.

Fix a 2-colorable hypergraph $H = (V, E)$ with n vertices. For a subset S of vertices let $N_1(S) = \{w \in V : S \cup \{w\} \in E\}$ and $d_1(S) = |N_1(S)|$. Fix an integer value $t > 0$. Consider the following procedure:

Algorithm Reduce.
For all $S \subseteq V$ with $d_1(S) > t$:

(1) If $N_1(S)$ is an independent set (i.e., it does not completely contain an

edge of E), then color the vertices of $N_1(S)$ with a new color; remove the vertices from H and all edges that contain at least one of these vertices.

(2) If $N_1(S)$ is not an independent set (i.e., it contains some edge of E), then replace all edges of H containing S with the hyperedge S.

We note that this algorithm can be made to run in polynomial time since we only have to look at those subsets S obtained by removing a single vertex from an edge (total number at most n times the number of hyperedges). After running algorithm Reduce on hypergraph H, we obtain a hypergraph H' with the following properties: (1) in H' each subset S of vertices satisfies $d_1(S) \leq t$; (2) H' is 2-colorable; (3) any k-coloring of H' yields a $k + n/t$-coloring of H (in polynomial time). *Proof.* Property 1 is immediate from the algorithm. To verify property 2, note that step (1) of Reduce preserves 2-colorability. Now assume that the hypergraph just before an application of step (2) is 2-colorable. We observe that the fact that $N_1(S)$ is not independent implies that some two vertices in $N_1(S)$ have different colors under any 2-coloring. Thus, the subset S cannot be monochromatic under any two-coloring. Hence, replacing all edges containing S by the subset S will preserve 2-colorability. Finally, for property 3 we note that the total number of colors used by Reduce is at most n/t since each color takes care of at least t vertices. Now suppose that we color the remaining hypergraph H' with k new colors. Assign to all vertices in H that also belong to H' the new color thus chosen while all vertices not in H' get the color that they are given by Reduce (different from the k new colors). This coloring is a legal coloring of the vertices in H with at most $k + n/t$ colors. □

The maximum degree of a hypergraph, usually denoted by Δ, is the maximum number of edges containing the same vertex. Suppose that we have a polynomial time algorithm that colors any 2-colorable hypergraph H of dimension d with maximum degree Δ with Δ^α colors, then we also have a polynomial time algorithm for coloring any such hypergraph on n vertices with $\Theta(n^{\frac{(d-1)\alpha}{1+\alpha}})$ colors.
Proof. Applying algorithm Reduce to H we obtain H' with maximum degree at most $n^{d-2}t$ (a more precise bound is $\binom{n-1}{d-2} \cdot t$). Thus, we can color H' with at most $(n^{d-2}t)^\alpha$ colors. The total number of colors used is then (by lemma ??) $n/t + (n^{d-2}t)^\alpha$. With $t = n^{\frac{1-\alpha(d-2)}{1+\alpha}}$, we obtain the asymptotically minimum number of colors, namely $\Theta(n^{\frac{(d-1)\alpha}{1+\alpha}})$ colors. □

3 A Simple Randomized Algorithm

Let $H = (V, E)$ be a hypergraph of dimension $d \geq 3$ that is 2-colorable. Let Δ_k denote the maximum number of edges in H of size k that contain the same vertex and let $\Delta'_k = \Delta_k^{\frac{1}{k-1}}$. Finally, let $\Delta' = max\{\Delta'_k : 2 \leq k \leq d\}$. The following simple randomized algorithm computes an approximate coloring of H.

Algorithm Rancolor.

(1) While H is non-empty do:

(1.1) Each vertex chooses uniformly at random a color from the same set of size $\lceil 4\Delta' \rceil$.

(1.2) A vertex is *good* if it is not contained in any monochromatic edge. Remove all good vertices as well as the edges containing them from H. Do not reuse these colors in later rounds (i.e., remove them from the ground set).

Algorithm Rancolor does indeed produce a valid coloring of H since among all the good vertices eliminated in one round there is no monochromatic edge and each color is used in at most one round. The expected total number of colors used is $O(\Delta' \log n)$. *Proof.* Fix a round of algorithm Rancolor. The probability that an edge of size k is monochromatic is at most $\frac{1}{4^{k-1}} \frac{1}{\Delta'^{k-1}} \leq \frac{1}{4^{k-1}} (1/\Delta'_k)^{k-1} = \frac{1}{4^{k-1}\Delta_k}$. Thus, a vertex is good with probability at least $1/2$. The expected number of good vertices removed in one round is thus at least $n/2$. We conclude that the expected total number of rounds is at most $O(\log n)$. This implies the claim of the theorem. □
We note that the claim about the expected number of colors can be transformed into a high probability claim using standard techniques

This bound by itself is not useful since Δ' can be as large as $\Theta(n)$ making the number of colors used by Rancolor larger than n. Together with the degree reduction technique from the last section we get, however, an interesting result: A 2-colorable hypergraph of dimension d can be colored in polynomial time with $O(n^{1-1/d}(\log n)^{1-1/d})$ colors. *Proof.* Applying lemma **??** we see that after running algorithm Reduce we have $\Delta_k \leq n^{k-2}t$ for all k (a more precise bound would be $\binom{n-1}{k-2}t$) and hence

$$\Delta' \leq max\{(n^{k-2}t)^{\frac{1}{k-1}} : 2 \leq k \leq d\}.$$

Fix k such that $(n^{k-2}t)^{\frac{1}{k-1}}$ is maximum (for $2 \leq k \leq d$). Combining this bound with the previous theorem (and applying lemma **??** once more) we see that we can color the hypergraph with $\log n (n^{k-2}t)^{\frac{1}{k-1}} + n/t$ colors. For $t = n^{\frac{1}{k}}/(\log n)^{1-\frac{1}{k}}$ we obtain the (asymptotically) minimum number of colors, namely $n^{1-1/k} \cdot (\log n)^{1-1/k}$. □

4 Hypergraph Coloring via Semidefinite Programming

In this section we describe a better coloring algorithm for the special case where each edge of H has size at most 3. We use semidefinite programming in conjunc-

tion with our degree reduction technique to achieve this result. We interweave our exposition with a description of the semidefinite programming paradigm.

Let $H = (V, E)$ be a 2-colorable hypergraph of dimension 3. W.l.o.g., we may assume that all edges of H have size exactly three. (This assumption is not essential but simplifies the exposition.) Indeed, if there are edges of size two we can proceed as follows: in a first phase 2-color H, taking only the edges of size 2 into account. In a second phase color H as described in this section, only considering edges of size 3. Now assign to each vertex as color the pair of colors it receives in these two phases. This is clearly a legal coloring and the total number of colors is at most twice that used in the second phase.

Let ϕ denote a 2-coloring of H. For convenience we shall assume that the range of ϕ is $\{-1, 1\}$. We can express the fact that ϕ is a 2-coloring by the following set of equalities: for every edge $e = (x_1, x_2, x_3)$ we have

$$\phi(x_1)\phi(x_2) + \phi(x_2)\phi(x_3) + \phi(x_1)\phi(x_3) = -1. \tag{1}$$

Note that the validity of this equation follows from the fact that in any edge (x_1, x_2, x_3), the coloring ϕ maps exactly two x_i to one value (e.g., -1) and the remaining vertex to the other value (e.g., $+1$).

Write the vertex set of H as $V = \{u_1, \ldots, u_n\}$. Consider the following semidefinite optimization problem:

minimize α (*)
where $n \times n$-matrix (a_{ij}) is positive semidefinite $(1 \leq i, j \leq n)$
subject to
$a_{ij} + a_{jk} + a_{ik} \leq \alpha$ if $(u_i, u_j, u_k) \in E$
$a_{ij} = a_{ji}$
$a_{ii} = 1$.

The semidefinite optimization problem (*) above has a solution $\alpha \leq -1$. *Proof.* Recall that ϕ denotes a 2-coloring of H (with color set $\{-1, 1\}$). Fix two unit vectors u and v in R^n with[1] $u \cdot v = -1$, e.g., $u = (-1, 0, \ldots, 0)$ and $v = (1, 0, \ldots, 0)$. Define vectors $z_1, \ldots, z_n$ by $z_i = u$ if $\phi(u_i) = -1$ and $z_i = v$ if $\phi(u_i) = +1$. Since $u \cdot v = -1$, we have $z_i \cdot z_j + z_j \cdot z_k + z_i \cdot z_k = -1$ if $(u_i, u_j, u_k) \in E$. We also trivially have $z_i \cdot z_j = z_j \cdot z_i$ because of symmetry of the dot product. Finally $z_i \cdot z_i = 1$ because both u and v are unit vectors.

Define $a_{ij} = z_i \cdot z_j$ for $1 \leq i, j \leq n$. The matrix $A = (a_{ij})$ is symmetric positive definite since it is of the form $A = BB^T$. The claim of the lemma follows. □

We can use the ellipsoid method or other interior-point methods ([?],[?]) to find a solution where $\alpha \leq -1 + \delta$ in time polynomial in n and $\log(1/\delta)$.

Lemma ?? tells us that we can go from the 2-coloring of H to a good solution to the semidefinite optimization problem (*). Conversely, we shall now show

[1] We use $u \cdot v$ to denote the dot product of the two vectors u and v.

how to get a good coloring from a good solution to the semidefinite optimization problem. The method we describe in this section is based on separating the vectors z_i using random hyperplanes. A more sophisticated technique yielding stronger results will be developed in the next section. Both techniques were originally used in [?] in the context of graph coloring. The present paper presents the first application of these techniques to hypergraphs.

For the following discussion we fix a solution to (*) with $\alpha = -1 + \delta$. Using the notation from the proof of lemma ?? we denote the corresponding (unit) vectors by $z_1, \ldots, z_n$. Thus we have

$$z_i \cdot z_j + z_j \cdot z_k + z_i \cdot z_k \leq -1 + \delta \text{ if } (u_i, u_j, u_k) \in E. \tag{2}$$

Let r be a unit vector which is uniformly distributed on the unit sphere. r defines uniquely a hyperplane which is normal to it. We need the following technical lemma. For three vectors x_1, x_2 and x_3, the probability that they do not all lie on the same side of the hyperplane (or equivalently, $sgn(x_1 \cdot r)$,$sgn(x_2 \cdot r)$ and $sgn(x_3 \cdot r)$ are not all equal) is

$$\frac{1}{2\pi}(\alpha + \beta + \gamma)$$

where $\alpha = arccos(x_1 \cdot x_2)$, $\beta = arccos(x_2 \cdot x_3)$ and $\gamma = arccos(x_1 \cdot x_3)$. Thus, if $x_1 \cdot x_2 + x_2 \cdot x_3 + x_1 \cdot x_3 \leq -1 + \delta$, then this probability is at least $\frac{3}{2\pi} arccos(\frac{-1+\delta}{3})$.

Proof. The first claim of the lemma is the same as lemma 2.2 in [?]. For the second part it suffices to show that

$$\frac{1}{2\pi}(\alpha + \beta + \gamma) \geq \frac{3}{2\pi} arccos(\frac{-1+\delta}{3})$$

Elementary differential calculus shows that the expression on the left is minimized for $\alpha = \beta = \gamma$. Since $x_1 \cdot x_2 + x_2 \cdot x_3 + x_1 \cdot x_3 = -1 + \delta$ we get for $\alpha = \beta = \gamma$ that $cos(\alpha)(= x_1 \cdot x_2) = \frac{-1+\delta}{3}$ or $\alpha = arccos(\frac{-1+\delta}{3})$. The claim follows. □

Let us fix a small delta such that $\frac{-1+\delta}{3} < -.33$. Let p denote the probability that the vectors corresponding to an edge of H do not lie on the same side of a hyperplane. With lemma ?? we get $p > \frac{3}{2\pi} arccos(-0.33) > 0.91$.

Let us choose k random hyperplanes $h_1, \ldots, h_k$ with normal vectors $r_1, \ldots, r_k$ independently. Map each vertex u_i with corresponding vector z_i to the vector $y_i = (sgn(z_i \cdot r_1), sgn(z_i \cdot r_2), \ldots, sgn(z_i \cdot r_k))$.

Fix an edge e in E. Let us say that this edge is *bad* (for the given hypergraph) if the vertices in e are mapped to the same vector under the mapping just described. The probability that an edge is bad is at most $(1-p)^k < .09^k$ (since the hypergraphs are chosen independently at random).

We construct a coloring of H as follows: let $V_b \subseteq V$ denote the vertices in H that are contained in bad edges. Assign to all vertices in $V - V_b$ as colors the corresponding y vectors. Recursively color the hypergraph $H' = (V_b, E')$, where E' is the set of edges in E all of whose vertices belong to V_b, with a set

of new colors. (To make sure no colors are reused, we may think of the color of a vertex as a pair (y, i) where i is the recursion depth at which a vertex was colored successfully and y is the vector that was assigned to it at that time.)

Let the random variable Z denote the number of edges that are bad w.r.t. k random hyperplanes. Note that $E[Z] < 0.09^k m$ where m denotes the number of edges in H. Let Δ denote the maximum degree of H. Thus $m \leq n\Delta/3$ and $E[Z] < 0.09^k n\Delta/3$. By choosing $0.09^k < \frac{3n}{4n\Delta} = \frac{3}{4\Delta}$ or $k > .416 \ln \Delta$ we get $E[Z] < n/4$ and hence $E[|V_b|] < 3n/4$. Thus for these values of k a constant fraction of the vertices of H are colored in the current stage. The recursion will only be performed for $O(\log n)$ levels and the total number of colors is $O(2^{.416 \ln \Delta} \log n) = O(\Delta^{.289} \log n)$.

We thus have the following result: A 2-colorable hypergraph of dimension 3 can be colored in polynomial time with $O(\Delta^{.289} \log n)$ colors.

By applying the degree reduction technique (theorem ??) to theorem ?? we obtain the first improvement on the result from theorem ?? (which yields $n^{.66}$ for dimension 3). There is a polynomial time algorithm that colors any 2-colorable hypergraph of dimension 3 with fewer than $n^{.45}$ colors. □

Although the algorithm is randomized, we can use the recently developed technique of Mahajan and Ramesh ([?]) to derandomize this algorithm into a deterministic polynomial time algorithm. We briefly explain how this is done. A random spherically symmetric vector is generated by choosing each of its coordinates independently from the normal distribution with mean 0 and variance 1. We need to choose the "random vector" deterministically. We use the method of conditional expectations and fix each coordinate in turn. The problem then is to compute conditional expectations and to find minima over a continuous range. This is described in [?].

In the next section we will use a more sophisticated technique to achieve an even better bound, namely $O(n^{2/9})$.

5 The Center Method

In this method, we take t independent random vectors, called *centers* (t will turn out to be $O(\Delta^{\frac{1}{8}} \log^{9/8} n)$) and we assign to each vertex vector the center which has the largest projection on the vertex vector as its color. This method reduces the number of bad edges to $n/6$, and hence the number of bad vertices to $n/2$. Hence by iterating this method $\log n$ times, we can properly color a 2-colorable hypergraph with $O(\Delta^{\frac{1}{8}} \log^{\frac{17}{8}} n)$ colors. Ideally we would like each random center to be of unit length, but as Karger, Motwani and Sudan [?] argue, this leads to technical difficulties in the analysis. Instead we choose each coordinate of each center independently normally distributed with mean 0 and variance 1. Then each center is spherically symmetric. However the norm of such a center is now a random variable (although as the number of dimensions tends to infinity, this norm becomes almost constant). The main contribution of this section is the

analysis which requires more sophistication than the graph case.

$P(t)$ is the probability that c_1 gets assigned to all three of v_i, v_j, v_k. In this case, the probability that a hyperedge (i,j,k) is bad is the probability that the same center gets assigned to all three of v_i, v_j, v_k. Let the centers be indexed as $c_1, ...c_t$. Then the probability that an edge is bad is $tP(t)$. Our core theorem states that $P(t)$ is $O(\frac{\log^9 t}{t^9})$. Hence the probability that an edge is bad is $O(\frac{\log^9 t}{t^8})$. $P(t)$ is $O(\frac{\log^9 t}{t^9})$ Before we give the proof of the core theorem, we derive the final result. By using the center method $O(\log n)$ times each time reducing the number of bad vertices by half, we can properly color a 2-colorable hypergraph in $O(\Delta^{\frac{1}{8}} \log^{\frac{17}{8}} n)$ colors. *Proof of the Corollary.* We want the number of bad edges to be less than $n/6$. Let p denote the probability that an edge is bad. It follows from the core theorem that $p = O(\frac{\log^9 t}{t^8})$. The expected number of bad edges is at most $mp \leq n\Delta p/3$. This number is less than $n/6$ provided that $p < \frac{1}{2\Delta}$. Routine algebra shows that this is satisfied by $t = \log^{9/8} n\Delta^{1/8}$. In each stage we use this many colors for a total of $O(\log n)$ stages, yielding the desired result. □ We can color a 2-colorable hypergraph with $O(n^{2/9} \log^{\frac{17}{8}} n)$ colors in polynomial time. *Proof.* Straightforward application of theorem **??** in conjunction with the previous corollary. □

Proof of the Core Theorem. We have $P(t) = Pr(c_1 \cdot v_i > \max\{c_2 \cdot v_i, ..., c_t \cdot v_i\} \wedge c_1 \cdot v_j > \max\{c_2 \cdot v_j, ..., c_t \cdot v_j\} \wedge c_1 \cdot v_k > \max\{c_2 \cdot v_k, ..., c_t \cdot v_k\})$.

Now as all of c_i's are spherically symmetric, we can rotate the coordinate system so that all but the first 3 coordinates of v_i, v_j and v_k are 0, or so that essentially we are in a 3-dimensional coordinate system. So we can assume that each of the c_i's is a 3-dimensional vector with each coordinate distributed independently with distribution N(0,1). Now $c \cdot v$ (where c is a center and v a vertex vector) is $|c| \cos(< c, v >)$ (where $< c, v >$ is angle between c and v).

Also note that as $v_i \cdot v_j + v_j \cdot v_k + v_k \cdot v_i = -1$ or that $|v_i + v_j + v_k| = 1$, this implies that for any unit vector l, one of $l \cdot v_i$, $l \cdot v_j$ and $l \cdot v_k$ is at most $1/3$. This is because $l \cdot (v_i + v_j + v_k)$ is at most 1. So for vector c_1, the angle between either c_1 and v_i, or c_1 and v_j or c_1 and v_k is at least $\arccos(1/3)$. Let us say that this happens for v_i. Then

$$P(t) \leq Pr(c_1 \cdot v_i > \max\{c_2 \cdot v_i, ..., c_t \cdot v_i\})$$

. or

$$P(t) \leq Pr(|c_1|/3 > \max\{|c_2| \cos(< c_2, v_i >), ..., |c_t| \cos(< c_t, v_i >)\}$$

or

$$Pr(|c_1|/3 > \max\{|c_2| \cos(< c_2, v_i >), ..., |c_t| \cos(< c_t, v_i >)\}$$

Now let the conditional probability that given that there are r centers which have angle at most ϵ with v_i, the event inside the above probability ($|c_1|^2/9 >$

...) holds, be denoted by $P(t|r)$. W.l.o.g. call these r centers $c_2, ..., c_{r+1}$. Then it is clear that $c_2 \cdot v_i, ..., c_{r+1} \cdot v_i$ are all non-negative for any $epsilon <= \pi/2$.

Hence $P(t|r)$ is bounded from the above by

$$Pr(|c_1|^2/9 > \max \|c_2|^2 \cos^2(< c_2, v_i >), ..., |c_{r+1}|^2 \cos^2(< c_t, v_i >)\}$$

or

$$Pr(|c_1|^2/9 > \max\{|c_2|^2 \cos^2 \epsilon, ..., |c_{r+1}|^2 \cos^2 \epsilon\}$$

For a fixed vector v, the probability that a random vector is within an ϵ angle from it is $\epsilon^2/4$ for small enough ϵ. This can be seen by looking at the volume of the cone(of the unit sphere) of angle ϵ centered around v. Therefore the probability that there are r centers which have an angle at most ϵ (for small enough ϵ) with v_i is $\binom{t}{r}(\frac{\epsilon^2}{4})^r(1-\frac{\epsilon^2}{4})^{t-r}$.

Hence $P(t) \leq \sum_{r=0}^{t} P(t|r)\binom{t}{r}(\frac{\epsilon^2}{4})^r(1-\frac{\epsilon^2}{4})^{t-r}$.

We now bound $P(t|r)$. We need to calculate $Pr(|c_1|^2 > q \max\{|c_2|^2, ..., |c_{r+1}|^2\})$ for $q = 9\cos^2(\epsilon)$.

Each of $|c_i|^2$ is a sum of squares of 3 independent normal random variables with mean 0 and variance 1. So we need to calculate $Pr(X_1 > q \max\{X_2, ..., X_{r+1}\})$ where each X_i is independently the sum of squares of 3 normals with mean 0 and variance 1. It is known that the probability density function $f(t)$ of each X_i is $c\sqrt{t}e^{-t/2}$ for some normalization constant c. The next theorem proves an appropriate upper bound on the above probability.

If $X_1 ..., X_{r+1}$ are independent with density function $f(t)$, then $Pr(X_1 > q \max\{X_2, ..., X_{r+1}\})$ is $O(\frac{1}{\binom{q+r}{r}})$, for $q \geq 1$.

Proof. We have $Pr(X_1 > q \max\{X_2, ..., X_{r+1}\}) = c\int_0^\infty (Pr(X < t/q))^r \sqrt{t}e^{-t/2}dt$ where X is a random variable with density function $f(t)$.

So $Pr(X < x) = 1 - g(x)e^{\frac{-x}{2}}$ where $g(x) \geq 1$ for all x. Hence

$$Pr(X_1 > q \max\{X_2, ..., X_{r+1}) \leq c\int_0^\infty (1 - g(t/q)e^{\frac{-t}{2q}})^r \sqrt{t}e^{-t/2}dt.$$

As $q \geq 1$ and $g(t/q) \geq 1$ for all t, the term on the right-hand side of the above inequality is at most

$$c\int_0^\infty (1 - g(t/q)e^{\frac{-t}{2q}})^r (g(t/q))^{q-1}\sqrt{t}e^{-t/2}dt$$

We now make a change of variables. Let $g(t/q)e^{\frac{-t}{2q}} = z$ Then we have

$$Pr(X_1 > q \max\{X_2, ..., X_{r+1}\}) \leq c'\int_0^1 (1-z)^r z^{q-1}dz$$

(where c' is some constant) because the derivative wrt t of $g(t/q)e^{\frac{-t}{2q}}$ is some constant times $\sqrt{t}e^{\frac{-t}{2q}}$. The term on the right-hand side of the above inequality is $\frac{c'}{q}/\binom{q+r}{r}$. Hence the required probability is $O(\frac{1}{\binom{q+r}{r}})$.

Hence $P(t|r)$ is bounded by $O(\frac{1}{\binom{q+r}{r}})$ for $q = 9\cos^2\epsilon$.

Therefore

$$P(t) \leq d\sum_{r=0}^{t}\binom{t}{r}(\frac{\epsilon^2}{4})^r(1-\frac{\epsilon^2}{4})^{t-r}\frac{1}{\binom{q+r}{r}}$$

for some constant d and $q = 9\cos^2\epsilon$. Manipulating algebraically the rhs, we get that $P(t)$ is $O(\epsilon^{2(\lceil q\rceil - q)}(\epsilon^2 t)^{-q})$ for $q = 9\cos^2\epsilon$. Now we set $\epsilon^2 = 1/\log t$. Then $\lceil q\rceil - q = 9 - 9\cos^2(\epsilon)$ which is $O(\epsilon^2)$. After doing some algebra, we see that $P(t) = O(\log^9 t/t^9)$. □

6 Conclusions

In this final section we address the following two questions: can the semidefinite approach be extended to hypergraphs of higher dimension? What about dimension 3: can the bound on the number of colors be improved significantly?

We have fairly strong evidence to believe that the answer to the first question is negative. Consider the example of a 2-colorable hypergraph of dimension 4. We can write a semidefinite program very similar to the program (*) of section ??: instead of taking the sum of the dot products of three vectors we take the sum of dot products of four vectors (remember that semidefiniteness implies that each a_{ij} in (*) can be written as a dot product), i.e.,

$$\begin{array}{l}\text{minimize } \alpha \qquad (**)\\ \text{where } n\times n\text{-matrix } (a_{ij}) \text{ is positive semidefinite } (1\leq i,j\leq n)\\ \text{subject to}\\ \qquad a_{ij}+a_{jk}+a_{kl}+a_{ik}+a_{jl}+a_{il}\leq\alpha \text{ if } (u_i,u_j,u_k,u_l)\in E\\ \qquad a_{ij}=a_{ji}\\ \qquad a_{ii}=1.\end{array}$$

Now consider a complete bipartite hypergraph (with n vertices on both sides) that is a maximal 2-colorable hypergraph with half the vertices of one color and half the other. As n tends to infinity, the semidefinite program is minimized for such a hypergraph at $\alpha = 0$ (intuitively, what is happening is that in a hyperedge you could have three reds and one blue, and that gives the sum of dot products to be 0, if red is +1 and blue -1). Now $\alpha = 0$ is a trivial solution, as we don't even need to solve the program to have all vectors orthogonal, and in fact for every hypergraph, $\alpha = 0$ is a feasible solution.

Let us now turn our attention to the second question. Let us say that a hypergraph H of dimension 3 is vector 2-colorable if there exist n unit vectors v_i such that for any hyperedge $\{i,j,k\}$, $v_i\cdot v_j + v_j\cdot v_k + v_k\cdot v_i \leq -1$. Notice

that all vector-2-colorable hypergraphs can be colored by the number of colors given by the center method, not just 2-colorable hypergraphs. We would like to construct a hypergraph that is vector-2-colorable but has a lower bound of N^α on its chromatic number ($\alpha > 0$), where N is the number of vertices in the hypergraph. This then shows the limitation of semidefinite programming in this context.

Construct a hypergraph as follows. The vertices of the hypergraph are all strings of length n over $\{0,1,2\}$. Three vertices s_1, s_2, s_3 define a hyperedge in the hypergraph if the sum of agreements between them is at most $n/3$. We want to show an upper bound of c^n, for some $c < 3$, on the size of its independent set. This will then show a lower bound on the chromatic number of the hypergraph of the kind d^n for some $d > 1$, or in terms of number of vertices $N = 3^n$ of the hypergraph, N^α.

Why is this hypergraph vector-2-colorable? Each string of length n over $\{0,1,2\}$ is considered a vector of dimension $2n$ where 0 is replaced by the vector $(1,0)/\sqrt{n}$, 1 by $(\frac{-1}{2}, \frac{\sqrt{3}}{2})/\sqrt{n}$ and 2 by $(\frac{-1}{2}, \frac{-\sqrt{3}}{2})/\sqrt{n}$. The $\sqrt{n}$ factor comes in because we wish to normalize each vector corresponding to an n-length string to a unit vector. It is now easily seen that if the sum of agreements between s_1, s_2, s_3 is atmost $n/3$, or equivalently if the sum of the hamming distances $d(s_1, s_2), d(s_2, s_3), d(s_3, s_1)$ is at least $8n/3$, then the sum of dot products of the corresponding vectors is at most -1.

In the conference version of this paper, we conjectured that this hypergraph has an upper bound of N^α ($\alpha < 1$) on the size of an independent set, yielding a lower bound of N^β ($\beta > 0$) on the chromatic number. We settle the conjecture here. Let S be a set of strings of length n over $\{0,1,2\}$ such that the following property holds. For any s_1, s_2, s_3 in S, $d(s_1, s_2) + d(s_2, s_3) + d(s_3, s_1) < 8n/3$. Then for some constant $c < 3$, $|S| \leq c^n$. To prove the theorem, we first establish the following simple lemma. Let T be a subset of $\{0,1,2\}^n$, such that $|T| > \frac{2}{3}3^n$. Then there exist three strings $s_1, s_2, s_3 \in T$ such that $d(s_1, s_2) = n, d(s_2, s_3) = n, d(s_3, s_1) = n$. *Proof:* Take s_1 to be a random string in $\{0,1,2\}^n$. Let $s_1 = a_1 a_2 ... a_n$. Let $s_2 = (a_1+1)(a_2+1)...(a_n+1)$ and $s_3 = (a_1+2)(a_2+2)...(a_n+2)$ (where $+$ is modulo 3) (so, for instance, if $s_1 = 1201$, then $s_2 = 2012$, and $s_3 = 0120$). Observe that $d(s_1, s_2) = d(s_2, s_3) = d(s_3, s_1) = n$. Then the probability that s_1 is in T is greater than $\frac{2}{3}$, and the same is true of s_2 and s_3. So the expected size of the intersection of $\{s_1, s_2, s_3\}$ and T is bigger than 2. Hence there exist s_1, s_2 and s_3 in T such that $d(s_1, s_2) = d(s_2, s_3) = d(s_3, s_1) = n$, and we are done. □

Let n be sufficiently large. Let W be a subset of $\{0,1,2\}^n$ such that $|W| > 3^n e^{\frac{-\delta'^2 n}{2}}$. Let $\delta > \delta'$ and $T = \{v \in \{0,1,2\}^n : d(v, W) \leq \delta n\}$. Here $d(v, W)$ denotes $\min_{w \in W} d(v, w)$. Then $|T| > \frac{2}{3}3^n$

Before we prove Theorem ??, we notice immediately that it and the lemma ?? imply Theorem ??. This is because, whenever W is a set of size bigger than $3^n e^{\frac{-\delta'^2 n}{2}}$ (that is some $d(\delta')^n$, where $d(\delta') < 3$ for $\delta' > 0$), the set $T = \{v \in$

$\{0,1,2\}^n : d(v,W) \leq \delta n\}$ has cardinality bigger than $\frac{2}{3}3^n$, and therefore contains strings $u_1, u_2, u_3 \in T$ such that $d(u_1,u_2) = d(u_2,u_3) = d(u_3,u_1) = n$. Therefore there exist strings s_1, s_2, s_3 in W such that $d(s_1,u_1) \leq \delta\sqrt{n}$, $d(s_2,u_2) \leq \delta\sqrt{n}$ and $d(s_3,u_3) \leq \delta\sqrt{n}$ for every $\delta > \delta'$. Then $d(s_1,s_2)+d(s_2,s_3)+d(s_3,s_1) \geq (3-6\delta)n$. Now taking δ such that $3-6\delta = 8/3$ and δ' anything less proves the result.

Proof of Theorem **??***:* We will actually show the following. Given S a subset of $\{0,1,2\}^n$ such that $|S| \geq \frac{1}{3}3^n$, $|v \in \{0,1,2\}^n : d(v,S) \geq (c+2)\sqrt{n}|$ is at most $3^n \mathrm{e}^{\frac{-c^2}{2}}$. Then taking $S = \{0,1,2\}^n - T$ and $c = \delta'\sqrt{n}$ establishes the result.

Now given such an S, we define a function $f : \{0,1,2\}^n \to \mathcal{R}^+$ as $f(v) = d(v,S)$. Let u_j denote the jth coordinate of a given string u. Let u be a random string from $\{0,1,2\}^n$. We now define a sequence of random variables, $X_0, ..., X_n$ such that $X_i(v) = E(f(u)|\forall j \leq i, u_j = v_j)$. Notice that $X_0 = E(f)$ and $X_n = f(u)$. This sequence defines a martingale [**?**], and $|X_{i+1} - X_i| \leq 1$. That it is a martingale follows from the fact that each position of the string is 'exposed' one at a time. That $|X_{i+1} - X_i| \leq 1$ follows from the fact that $d(u,S)$ cannot change by more than 1, when exactly one more position of u is known at step i.

So we can apply Azuma's inequalities [**?**]. Let $X_0, ..., X_n$ be a martingale such that $|X_{i+1} - X_i| \leq 1$. Then $Pr(X_n - X_0 > l\sqrt{n}) \leq \mathrm{e}^{\frac{-l^2}{2}}$ and $Pr(X_n - X_0 < -l\sqrt{n}) \leq \mathrm{e}^{\frac{-l^2}{2}}$.

So in our case, $Pr(f(u) - E(f) > l\sqrt{n}) \leq \mathrm{e}^{\frac{-l^2}{2}}$ and $Pr(f(u) - E(f) < -l\sqrt{n}) \leq \mathrm{e}^{\frac{-l^2}{2}}$.

Now as $f(u) = d(u,S)$ and $|S| \geq \frac{1}{3}3^n$, $Pr(f(u) = 0) \geq \frac{1}{3}$. Putting $l = 2$ in the second inequality gives $Pr(f(u) - E(f) < -2\sqrt{n}) \leq e^{-2} < \frac{1}{3}$. Hence $E(f) \leq 2\sqrt{n}$. Now using the first inequality, we have

$$Pr(f(u) > (c+2)\sqrt{n}) \leq Pr(f(u) - E(f) \leq c\sqrt{n}) \leq e^{\frac{-c^2}{2}}$$

and so we are done.□

Acknowledgments. The authors would like to thank Magnus Haldorsson for helping to formulate the statement of Theorem 9 in the last section.

References

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

[2] N. Alon, N. Kahale, Approximating the Independence number via the Θ function, Manuscript, 1995.

[3] N. Alon, J. Spencer, The Probabilistic Method, Wiley Eastern Limited.

[4] A. Blum, New Approximation Algorithms for Graph Coloring, JACM, 41, pp. 470-516, 1994.

[5] H. Chen, A. Frieze, Coloring Bipartite Hypergraphs, 5th Integer Programming and Combinatorial Optimization Conference, 1996.

[6] A. Frieze, M.Jerrum, Improved Approximation Algorithms for Max k-Cut and Max Bisection, 4th Integer Programming and Combinatorial Optimization Conference, 1995.

[7] M.R. Garey, D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness,* Freeman, San Francisco, CA, 1979.

[8] M. Goemans, D. Williamson, 0.878 Approximation Algorithms for Max Cut and Max 2SAT, Proc. of 26th Annual Symposium on the Theory of Computing, pp. 422-431, 1993.

[9] M. Grötschel, L. Lovász, A Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1, pp. 169-197, 1981.

[10] D. Karger, R. Motwani, M. Sudan, Approximate Graph Coloring by Semidefinite Programming, Proc. of the 35th IEEE Symposium on Foundations of Computer Science, pp. 1-10, 1994.

[11] P. Kelsen, S. Mahajan, H. Ramesh, Approximate Hypergraph Coloring, 5th Scandinavian Workshop on Algorithms and Theory, pp. 41-52, 1996.

[12] L. Lovasz, Colorings and Coverings of Hypergraphs, Proc. 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, Winnipeg, pp. 3-12.

[13] S. Mahajan, H. Ramesh, Derandomizing Semidefinite Programming Based Approximation Algorithms, Proc. of the 36th IEEE Symposium on Foundations of Computer Science, 1995, pp. 162-169

[14] A. Wigderson, Improving the Performance Guarantee for Approximate Graph Coloring, JACM 30(4), pp. 729-735, 1983.