A Pseudo-Random Method for List Colouring

Michael Molloy*
Department of Computer Science
University of Toronto
Toronto, Canada

Bruce Reed*
CNRS IME-USP
Paris Sao Paulo
France Brasil

December 8, 1998

Abstract We show that a simple variant of a naive colouring algorithm can be used to list colour the edges of a k-uniform linear hypergraph of maximum degree Δ provided every vertex has a list of at least $\Delta + c(log\Delta)^6\Delta^{1-\frac{1}{k}}$ available colours (where c is a constant which depends on k). We can extend this to colour hypergraphs of maximum codegree $o(\Delta)$ with $\Delta + o(\Delta)$ colours. This improves earlier results of Kahn and our techniques are quite similar. We also develop efficent algorithms to obtain such colourings.

1 A Colouring Problem and A Colouring Procedure

A *hypergraph* H consists of a set $V(H)$ (or simply V) of vertices and a set $E(H)$ (or simply E) of edges, each of which is a subset of $V(H)$. The chromatic index of a hypergraph is the minimum number of colours needed to colour its edges so that no two edges which share an endpoint receive the same colour. Given a list of colours for each edge of H, we say that an edge colouring is *acceptable* if every edge is coloured with a colour on its list. The list chromatic index of a hypergraph H is the minimum r which satisfies: if every list has at least r members then there is an acceptable colouring. Obviously, both of these numbers can be no smaller than the maximum degree of H, denoted $\Delta(H)$.

In a similar manner, we define the *chromatic number* of a graph G, denoted $\chi(G)$ to be the minimum number of colours needed to colour its vertices so that no two adjacent vertices receive the same colour. Given a list of colours for each vertex of G, we say that a vertex colouring is *acceptable* if every vertex is

*This work was supported by NATO Collaborative Research Grant #CRG950235. The work of the first author was supported by an NSERC Research Grant. The work of the second author was supported by a FAPESP Grant

coloured with a colour on its list. The list chromatic number of a graph is the minimum r which satisfies: if every list has at least r members then there is an acceptable colouring. Obviously, both of these numbers can be no smaller than the size of the largest clique (set of pairwise adjacent vertices) of G.

A hypergraph is *k-uniform* if every edge contains k vertices. It is *k-bounded* if every edge contains at most k vertices. We shall consider list colouring the edges of k-bounded hypergraphs. For ease of exposition we consider k-uniform hypergraphs, the results extend easily (we simply add dummy vertices to each edge).

In order to reformulate our problem in terms of colouring the vertices of a graph we need to consider line graphs of hypergraphs. The line graph $L(H)$ of a hypergraph H has a vertex corresponding to each edge of H. Two vertices of $L(H)$ are adjacent if and only if the corresponding two edges of H share an endpoint. We want to calculate the list chromatic number of line graphs of certain k-uniform hypergraphs.

Note first that the maximum degree of a line graph of a k-uniform hypergraph of maximum degree Δ may be as large as $k(\Delta-1)$. Furthermore, the chromatic index of H may be as large as $k(\Delta-1)$. To see this, consider $k=p+1$ for some prime p and let H be a hypergraph whose edges and vertices correspond to the lines and points of a projective plane generated by Z_p. That is we have a set of k^2-k+1 edges, every pair of which meet in exactly one vertex. Now, we let $\Delta=kd$ for some integer d and take d copies of each edge of H to obtain a new hypergraph H' whose line graph is a clique.

Note that in these examples, if two vertices are contained in an edge of the hypergraph then they are contained in many edges. The *codegree* of a pair $\{x,y\}$ of vertices in a hypergraph H, denoted $d_H(x,y)$ or simply $d_(x,y)$, is the number of edges that contain both the vertices.

We can show that the list chromatic index of a k-uniform hypergraph is at most $\Delta+o(\Delta)$ provided the maximum codegree is $o(\Delta)$. To begin, we consider linear hypergraphs: that is hypergraphs such that no two edges intersect in more than one vertex. Kahn[8] proved that if a k-uniform linear hypergraph has maximum degree Δ then it has list chromatic index at most $\Delta+o(\Delta)$. We show that there is a constant c (depending on k) such that any such hypergraph has list chromatic index at most $\Delta+O(\Delta^{1-\frac{1}{k}}(log\Delta)^6)$. This also improves a result of Haggkvist and Janssen[7] who proved that when $k=2$, i.e. for the line graphs of graphs, the list chromatic index is bounded by $\Delta+O(\Delta^{\frac{2}{3}})$. We remark that it has been conjectured(see [3]) that the list chromatic index of a graph H equals its chromatic index and hence is at most $\Delta+1$ (We also remark that we have made no attempt to optimize the exponent in the $log\Delta$ term. Certainly we could replace the six by a four, and probably by a two. Possibly with extreme care we could obtain $1+\epsilon$). We leave the case of linear hypergraphs of larger maximum codegree to a fuller version of this paper.

In this paper we use the naive colouring algorithm which consists of choosing a colour for each vertex, uncolouring those involved in a conflict, and recursively repeating this process on the uncoloured vertices. Suprisingly, analyzing this procedure using simple but powerful probabilistic tools allows us to find optimal or at least asymptopically optimal bounds on the chromatic number and/or list chromatic number of various classes of graphs. See [11] for a survey of the applications of the technique.

We believe the results in this paper to be of interest for a number of reasons. The first is that determining the list chromatic number of linegraphs is an important open problem and any improvement on the upper bound is of interest. The second is that this paper, which provides a slightly different analysis of Kahn's original colouring method, shows more clearly how his procedure fits into the general schema given in [11]. Furthermore, our treatment avoids the introduction of new concentration results and (via inclusion -exclusion) simplifies some expected value computations. This may make it more accessible than Kahn's treatment to some readers. Finally, as discussed in Section 6, we can use recent results of Molloy and Reed on algorithmic versions of the Local Lemma to obtain efficent algorithms for finding acceptable colourings for sets of lists which satisfy our bounds

Before presenting the results, we show that we can restrict our attention to k-uniform hypergraphs in which every vertex has degree Δ, i.e. Δ-regular k-uniform hypergraphs. We do so by applying the following:

Observation: Every k-uniform hypergraph of maximum degree Δ can be embedded in a Δ-regular k-uniform hypergraph of the same codegree.

Proof: We repeatedly apply the following operation: take k copies of the current hypergraph and for each vertex v of degree less than Δ, add an edge through the k copies of v. We note that each time we perform this operation, if the original hypergraph was not regular, then the minimum degree in the new hypergraph is one greater than the minimum degree in the old graph. Thus, after at most Δ iterations of this operation, we obtain a Δ-regular hypergraph.

We shall also need:

The Lovasz Local Lemma[5](Symmetric Version): *Let $\mathcal{E}$ be a set of events in a probability space. Suppose that each $A \in \mathcal{E}$ is mutually independent of all but at most d other events and occurs with probability at most p. If $epd < 1$ then with positive probability, none of the events in $\mathcal{E}$ occur.*

A Simple Observation[4]: *The vertices of any graph G of maximum degree d can be coloured using $d+1$ colours using a greedy procedure which takes $O(d|V(G)|)$ time. We simply consider the vertices in any order, colouring each vertex with a colour that does not appear on its coloured neighbours.*

2 The Chromatic Index of Linear Hypergraphs

To begin, we consider the chromatic number of the line graphs of linear hypergraphs. We will then generalize to list colouring such linegraphs. hypergraphs of bounded codegree.

So, consider a graph G which is the line graph of a k-uniform Δ-regular linear hypergraph. This implies that there is a set $\mathcal{C}$ of cliques of G each containing Δ vertices such that every vertex is in exactly k elements of $\mathcal{C}$ and each pair of vertices is contained in at most one element of $\mathcal{C}$.

As remarked earlier, we are going to colour the vertices by repeatedly assigning each uncoloured vertex a random colour which appears on none of its coloured neighours and then colouring a vertex with the colour it is assigned if this colour is not assigned to any of its (uncoloured) neighbours. So, for each vertex v, at any time in the algorithm we will have a list L_v of colours which do not appear on any coloured neighbour of v and hence can be assigned to v. Similarily for each clique $C \in \mathcal{C}$, we will have a list L_C of colours which appear on no vertex of C and hence can still be used to colour the uncoloured vertices of C. Note that L_v is simply the interesection of $\{L_C | C \in \mathcal{C}, v \in C\}$. Finally, for each $C \in \mathcal{C}$ and colour i in L_C, we have a subset $S_{C,i}$ of $V(C)$ consisting of those vertices of C for which i is available, i.e. $S_{C,i} = \{v | v \in C, i \in L_v\}$. Of course, whether v is in $S_{C,i}$ or not depends on the colours of the vertices in the other cliques containing v.

To begin, we attempt to colour G with Δ colours. Thus, initially $|L_v| = \Delta$ for all $v \in V(G)$, $|L_C| = \Delta$ for all $C \in \mathcal{C}$, and $|S_{C,i}| = \Delta$ for all $C \in \mathcal{C}$ and each colour i.

Consider what happens after one iteration of the naive colouring procedure in which each uncoloured vertex v chooses a colour uniformily from those on L_v with these choices mutually independent. Each vertex v has $k\Delta - k$ neighbours each of which is assigned the same colour as v with probability $\frac{1}{\Delta}$. Thus the probability that v is coloured is $(1 - \frac{1}{\Delta})^{k\Delta - k}$ which for Δ large is close to e^{-k}. So, for each $C \in \mathcal{C}$, we have that $Ex(|L_C|)$ is about $(1 - \frac{1}{e^k})\Delta$.

Now, if the choices of the new L_C were all independent, then for each v which remains uncloured we would have $Ex(|L_v|)$ is about $(1 - \frac{1}{e^k})^k \Delta$. Similarily, for a colour i not used in C the probability that an uncoloured vertex in $S_{C,i}$ does not have a neighbour coloured i in any of the other $k-1$ cliques containing it would be about $(1 - \frac{1}{e^k})^{k-1}$ and since we expect a vertex to be uncoloured with probability about $(1 - \frac{1}{e^k})$, we have that for each C and i remaining in L_C the expected number of vertices of $S_{C,i}$ is about $(1 - \frac{1}{e^k})^k \Delta$.

Of course, the choices made on the various cliques are not independent, the cliques intersect. However, the linearity of the hypergraph can be used to show that, for the random variables under consideration, the expected values computed above are never far wrong. We can then show (using Azuma's inequality, introduced below), that the probability that a particular such variable is far from its expected value is quite small. Finally, an application of the Local

Lemma allows us to prove the existence of an assignment of colours where the size of every L_v,L_C, and $S_{C,i}$ is close to its expected value.

At this point we have:

1 *Each* L_C *has about* $(1-\frac{1}{e^k})\Delta$ *elements,*

2 *Each* L_v *has about* $(1-\frac{1}{e^k})^k\Delta$ *elements,*

3 *Each* $S_{C,i}$ *has about* $(1-\frac{1}{e^k})^k\Delta$ *elements.*

Of course $|L_C|$ is simply the number of uncoloured vertices in C, so we have essentially reduced our problem to a list colouring problem on a subhypergraph whose degree is about $(1-\frac{1}{e^k})\Delta$. Unfortunately, the lists are no longer identical. Nevertheless conditions 2 and 3 are enough to allow us to iterate our analysis, obtaining a sequence of iteration so that after the jth iteration, we have:

4 *Each* L_C *has about* $(1-\frac{1}{e^k})^j\Delta$ *elements,*

5 *Each* L_v *has about* $(1-\frac{1}{e^k})^{kj}\Delta$ *elements,*

6 *Each* $S_{C,i}$ *has about* $(1-\frac{1}{e^k})^{kj}\Delta$ *elements.*

Unfortunately, as the number of iterations increases, "about" becomes a looser and looser term and we eventually lose control of the whole process. Actually, with care, we can continue until the L_v are almost empty, at which point (by 4 and 5) each L_C has around $\Delta^{1-\frac{1}{k}}$ elements. In fact, we stop when the maximum order of an L_C is $O((log\Delta)^6\Delta^{1-\frac{1}{k}})$. Since the degree of any vertex in the uncoloured graph is at most k times this maximum, we can complete the colouring using $O(k(log\Delta)^6\Delta^{1-\frac{1}{k}})$ new colours (by our simple observation). Forthwith the details.

We consider each iteration separately. To begin, we assume that we have a line graph G of a hypergraph H, an integer α, and a list L_v of colours on each vertex v of G such that:

7 *Each* L_v *contains* α *integers, and*

8 *For each clique* $C \in \mathcal{C}$ *and colour* i*, if there is some* $v \in C$ *such that* L_v *contains* i *then there are exactly* α *such* L_v*. I.e. carrying on with the notation we have been using: for all* $C \in \mathcal{C}$ *and* $i \in L_C$*,* $|S_{C,i}| = \alpha$*.*

We now assign each vertex v a random colour from the uniform distribution on L_v where these choices are made independently. We then colour a vertex with the colour it has been assigned unless one of its neighbours has been assigned the same colour.

In what follows, we say $a \approx b$ if $\frac{a}{b} = 1+\epsilon$ where $\epsilon = 0(\frac{f(k)}{\alpha})$ for some function f.

If v remains uncoloured, we let L'_v be the subset of L_v not used to colour any neighbour of v during this iteration. For each $C \in \mathcal{C}$, L'_C is the set of colours in L_C not used to colour any vertex of C in this iteration. For each $i \in L'_C$, $S'_{C,i}$ is the subset $\{v | v \in C, i \in L'_v\}$ of $S_{C,i}$. Also, we denote by U_C the set of vertices of C which remain uncoloured at the end of the iteration. More generally, for any set S of vertices, U_S is the subset of S which remain uncoloured at the end of the iteration.

We want to compute: (i) $Ex(|L'_V|)$ for each $v \in V(G)$ which remains uncoloured, (ii) $Ex(|S'_{C,i}|)$ for each C and every i in L'_C, and (iii) $Ex(|U_C)|)$ for each C in $\mathcal{C}$.

To begin, we compute the probability $p_{col}(v,i)$ that a particular vertex v is coloured with a particular element i of L_v. Of course, by (7), the probability that v is assigned the colour i is $\frac{1}{\alpha}$. As H is a linear hypergraph, (8) implies that there are $k(\alpha-1)$ neighbours of v for which i is an available colour. By (7), each of these vertices is assigned the colour i with probability $\frac{1}{\alpha}$. Thus, $p_{col}(v,i) = \frac{1}{\alpha}(1-\frac{1}{\alpha})^{k\alpha-k}$. So, $p_{col}(v,i) \approx \frac{1}{e^k\alpha}$.

Since L_v has α elements, it follows that v is coloured with probability $\approx \frac{1}{e^k}$ and uncoloured with probability $\approx (1-\frac{1}{e^k})$. Hence, for all $S \subseteq V(G)$ we have $Ex(|U_S|) \approx (1-\frac{1}{e^k})|S|$. Furthermore, since there are α vertices in $S_{C,i}$ for each $i \in L_C$, the probability a colour $i \in L_C$ is used on some vertex of C is $\approx \frac{1}{e^k}$ and hence $Ex(|L'_C|) \approx (1-\frac{1}{e^k})|L_C|$.

Now, we consider two adjacent vertices v and w, and two colours i and j with $i \in L_v \cap L_w$ and $j \in L_v$. We compute the probability $p_c(w,v,i,j)$ that w is coloured with i and v is coloured with j. Let l be the number of vertices which are adjacent to both v and w and for which both i and j are available. If $j \notin L_w$ then:

$$p_c(w,v,i,j) = \frac{1}{\alpha}^2(1-\frac{1}{\alpha})^{k\alpha-k-l}(1-\frac{1}{\alpha})^{k\alpha-k-l-1}(1-\frac{2}{\alpha})^l$$

.

If $j \in L_w$ then:

$$p_c(w,v,i,j) = \frac{1}{\alpha}^2(1-\frac{1}{\alpha})^{k\alpha-k-l-1}(1-\frac{1}{\alpha})^{k\alpha-k-l-1}(1-\frac{2}{\alpha})^l$$

.

In either case since $l \leq k\alpha$, we have: $p_c(w,v,i,j) \approx \frac{1}{e^k\alpha}^2$.

Now, fix a clique $C \in \mathcal{C}$, a vertex $v \in C$, and a colour $i \in L_v$. Consider a vertex w in $S_{C,i} - v$. Since $L_v - i$ has $\alpha - 1$ elements, it follows that the probability that w is coloured i and v is coloured is $\approx \frac{1}{e^{2k}\alpha}$. Furthermore, as there are $\alpha - 1$ elements of $S_{C,i} - v$ it follows that the probability that some vertex in C is coloured i and v is coloured is $\approx \frac{1}{e^{2k}}$. Since the probability that v is coloured is $\frac{1}{e^k}$, it follows that the probability that v is coloured and i is not used on C is $\approx (1-\frac{1}{e^k})\frac{1}{e^k}$. Since the probability that i is not used on C is

$\approx 1 - \frac{1}{e^k}$, it follows that the probability that v is uncoloured and no element of C receives colour i is $\approx (1 - \frac{1}{e^k})^2$.

We next turn to the much more complicated task of computing the probability that v is uncoloured and no neighbour of v recieves the colour i where i is some colour on L_v. To do so we make use of the inclusion-exclusion principle. A simple but powerful counting tool whoses proof is so short we include it:

Theorem 9 *Let X be a set and let $\mathcal{S}$ be a set of (possibly empty) subsets of X. Then the number of non-empty elements of $\mathcal{S}$ equals*

$$\sum_{S\in\mathcal{S}}\sum_{j=1}^{|S|}(-1)^{j+1}|\{U|U\subseteq S, |U|=j\}|$$

Proof: We know that $(1-x)^k = \sum_{i=0}^k \binom{k}{i}x^i$. So, in particular, $(1-1)^k = \sum_{i=0}^k \binom{k}{i}(-1)^i$. Thus, for any nonempty set S with k elements we have: $0 = \sum_{i=0}^k |\{U|U \subseteq S, |U| = i\}|(-1)^i$. Hence, $-1 = \sum_{j=1}^k |\{U|U \subseteq S, |U| = j\}|(-1)^j$, that is $1 = \sum_{j=1}^k |\{U|U \subseteq S, |U| = j\}|(-1)^{j+1}$. The desired result follows immediately.

Corollary: 10 *Let S be a random set chosen from a finite ground set X. Then, $Prob(S \neq \emptyset) = \sum_{j=1}^{|X|}(-1)^{j+1}Ex(|\{U|U \subseteq S, |U| = j\}|)$.*

Proof: To begin, we assume that for each subset Y of X, $Prob(S = Y)$ is rational (as is the case in all our applications of the corollary), the general result follows by successive approximations. So, by taking multiple copies of each subset of Y we can generate a multiset $\mathcal{S}$ of subsets of X where S is equally likely to be any element of $\mathcal{S}$. Suppose $\mathcal{S}$ has m elements. Now, $Prob(S \neq \emptyset) = \frac{|\{S\in\mathcal{S}, S\neq\emptyset\}|}{m}$. Furthermore, $Ex(|\{U|U \subseteq S, |U| = j\}|) = \frac{\sum_{S\in\mathcal{S}}|\{U|U\subseteq S,|U|=j\}|}{m}$. The result follows immediately.

We now return to computing the probability that v is uncoloured and no neighbour of v has colour i. To begin, we apply inclusion-exclusion to calculate the probability that some neighbour of v has colour i (where $i \in L_v$ and hence for all C with $v \in C : i \in L_C$). We let S be the (random) set of neighbours of v of colour i and calculate: $Prob(S \neq \emptyset) = \sum_{j=1}^k (-1)^{j+1}Ex(|\{U|U \subseteq S, |U| = j\}|)$.

We first calculate for each j, the number of stable sets of size j in $N(v)$. To begin we fix the $\binom{k}{j}$ cliques containing v which contain a member of the stable set. Having done so we arbitrarily order these cliques, and pick the vertices of the stable set. In the first clique C, we are free to pick any vertex of $S_{C,i}$. In any later clique C', we must avoid picking a neighbour of any vertex already picked. However, by the linearity of the hypergraph, each vertex not in C' has at most k neighbours in C', so in total at most $k(k-1)$ vertices are forbidden, and we have between α and $\alpha - k(k-1)$ choices for the next vertex of the stable set. Thus, the number of such stable sets is $\approx \binom{k}{j}\alpha^k$.

Now, for any such stable set W, the probability that every vertex in this stable set is assigned the colour i whilst no neighbour of any element of W is assigned i is: $\frac{1}{\alpha}^j(1-\frac{1}{\alpha})^N$ where N is the size of the union of the neighbourhoods of the elements in S. However, by the linearity of H, any two vertices of W have at most k^2 common neighbours. Thus, N is between $kj\alpha$ and $kj\alpha - j^2k^2$. Hence, the probability that all the vertices of W are coloured i is $\approx \frac{1}{\alpha}^j \frac{1}{e^k}^j$. So, $Prob(S \neq \emptyset) \approx \sum_{j=1}^{k} \binom{k}{j}(-1)^{j+1}\frac{1}{e^k}^j$. But the right hand side of this inequality is simply $1-(1-\frac{1}{e^k})^k$ (to see this we use the expansion of $(1-x)^k$ multiplied through by -1). So $Prob(v \textit{ has no neighbour coloured } i) \approx (1-\frac{1}{e^k})^k$.

Next, we apply inclusion-exclusion to calculate the probability $p^*(v,l,i)$ that v receives some colour $l \in L_v - i$ and some neighbour of v has colour i (where $i \in L_v$ and hence for all C with $v \in C : i \in L_C$). We let S be the (random) set of neighbours which is empty if v does not recive colour l and is the set of neighbours of colour i if v does recieve colour j. We calculate the probability that S is non-empty.

Recall that the number of stable sets of size j in N_v is $\approx \binom{k}{j}\alpha^j$.

Now, for any such stable set W, we let N be the number of neighbours of elements of W on which i is available, we let b be the number of neighbours of v also adjacent to a vertex of W for which l and i are both available, and we let d be the number of vertices of W for which l is available. Then, the probability that v is assigned the colour l and every vertex in W is assigned the colour i whilst no neighbour of v is coloured l and no neighbour of any element of W is assigned i is:

$$\frac{1}{\alpha}^{j+1}(1-\frac{1}{\alpha})^{(N-b-1)+(k\alpha-k-b-d)}(1-\frac{2}{\alpha})^b$$

. However, N differs from $kj\alpha$ by at most k^3, b is at most $k\alpha$, and d is at most j and hence no greater than k. So, $Prob(S \neq \emptyset) \approx \sum_{j=1}^{k} \frac{1}{e^k}\frac{1}{\alpha}\binom{k}{j}(-1)^{j+1}(\frac{1}{e^k})^j$. But the right hand side of this inequality is simply $\frac{1}{e^k}\frac{1}{\alpha}(1-(1-\frac{1}{e^k})^k)$. Thus, $p^*(v,l,i) \approx \frac{1}{e^k}\frac{1}{\alpha}(1-(1-\frac{1}{e^k}))^k$.

Summing over all the colours in $L_v - i$, we see that the probability that v is coloured and has a neighbour of colour i is $\approx \frac{1}{e^k}(1-(1-\frac{1}{e^k})^k)$. Now, the probability that v is coloured is $\approx \frac{1}{e^k}$. So the probability that v has no neighbour of colour i given that it is coloured is $\approx (1-\frac{1}{e^k})^k$. Furthermore, recall that we have shown that the probability that v has no neighbour of colour i is $\approx (1-\frac{1}{e^k})^k$. Thus, the probability that v has no neighbour of colour i given that it is uncoloured must also be $\approx (1-\frac{1}{e^k})^k$; that is: the probability $p^*(v,\emptyset)$ that v has no neighbour of colour i and is uncoloured must be $\approx (1-\frac{1}{e^k})^{k+1}$. Hence, the expected size of L'_v is $\approx (1-\frac{1}{e^k})^k\alpha$.

Furthermore, for any clique C containing v and any colour i in L_v, summing $p_{col}(w,i)$ over all α choices for w in $S_{C,i}$, we obtain that the probability that i

is in L'_C is $\approx (1-\frac{1}{e^k})$. So, since $p^*(v,\emptyset) = (1-\frac{1}{e^k})^{k+1}$, the probability that v is uncoloured and has no neighbour coloured i given that i is in L'_C is $\approx (1-\frac{1}{e^k})^k$. Thus, the expected size of $S'_{C,i}$ given $i \in L'_C$ is: $\approx \alpha(1-\frac{1}{e^k})^k$.

So, we have shown:

11 *For all v, $Ex(|L'_v|) \approx (1-\frac{1}{e^k})^k\alpha$.*

12 *For all $S \subseteq V(G)$, $Ex(|U_S|) \approx (1-\frac{1}{e^k})|S|$*

13 *For all $C \in \mathcal{C}$ and $i \in L'_C$, $Ex(|S'_{C,i}|) \approx (1-\frac{1}{e^k})^k\alpha$*

We still need to show that these random variables are concentrated around their expected values. We do this in Section 5. First, however, we generalize our computations so that we can apply them to arbitrary list colouring problems on linear hypergraphs (as well as later iterations of the simple colouring problem).

3 An Arbitrary Iteration

An arbitrary list colouring problem on a Δ-regular hypergraph in which every list has $\Delta + d$ colours obviously satisfies (7) with $\alpha = \Delta + d$. However, it may be very far from satisfying (8). For example, some colours may appear on only one vertex. So, we introduce the following weaker condition.

Definition: For any non-negative integer α, we denote by $d_\alpha(\Delta, k)$ or simply d_α, the value $\lfloor (log\Delta)^4\alpha^{1-\frac{1}{k}}\rfloor$.

14 *There is an integer α and an integer $d \leq d_\alpha$ such that for each vertex v, $|L_v| \geq \alpha$ and for each clique $C \in \mathcal{C}$ and colour i, if there is some L_v which contains i then there are at most $\alpha - d$ w in C such that $i \in L_w$, I.e. carrying on with the notation we have been using: for all $C \in \mathcal{C}$ and $i \in L_C$, $|S_{C,i}| \leq \alpha - d$.*

Now, in performing the next iteration, we would like to simulate the previous situation in which the size of $S_{C,i}$ is the same for every clique C and colour i in L_C. This could be achieved by adding $\alpha - d - |S_{C,i}|$ dummy vertices to a particular clique C for each $i \in L_C$, each of which is assigned colour i with probability $\frac{1}{\alpha}$. Furthermore, to simulate having neighbours in other cliques, the probability that a dummy vertex in C which is assigned the colour i would be coloured with colour i given that no other vertex of C is assigned colour i would be $(1-\frac{1}{\alpha})^{(k-1)(\alpha-d-1)}$.

Now, setting $r_{\alpha,d} = (1-\frac{1}{\alpha})^{k(\alpha-d-1)}$, essentially the same analysis as before yields:

15 *For all v and i, $Prob(v$ is coloured $i) \approx \frac{1}{\alpha}r_{\alpha,d}$.*

16 *For all v, $Prob(v$ is coloured$) \approx r_{\alpha,d}$.*

17 *For all* $C \in \mathcal{C}$ *and* $i \in L_C$, $Prob(i \in L'_C) \geq\approx 1 - \frac{\alpha-d}{\alpha} r_{\alpha,d}$.

18 *For all* v, $Ex(|L'_v|) \approx (1 - \frac{\alpha-d}{\alpha} r_{\alpha,d})^k \alpha$.

19 *For all* $S \subseteq V(G)$, $Ex(|U_S|) \approx (1 - r_{\alpha,d})|S|$

20 *For all* $C \in \mathcal{C}$ *and* $i \in L'_C$, $Ex(|S'_{C,i}|) \approx (1 - r_{\alpha,d})(1 - \frac{\alpha-d}{\alpha} r_{\alpha,d})^{k-1}(\alpha - d)$.

Remark: *(15) is straightforward. (16) follows because each* L_v *has* α *elements. Similarily, (17) follows because no* $S_{C,i}$ *has more than* $\alpha - d$ *elements. the difference between (16) and (17) explains the difference between (18) and (20).*

We shall show in Section 4 that if we are considering a subhypergraph of a Δ-regular hypergraph then, provided Δ is sufficently large in terms of k (i.e. $\Delta \geq \Delta_k$ defined later), a colouring exists such that:

21 *For all* v, $|L'_v| \geq (1 - \frac{\alpha-d}{\alpha} r_{\alpha,d})^k \alpha - (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.

22 *For all* $C \in \mathcal{C}$, $|U_C| \leq (1 - r_{\alpha,d})|C| + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{|C|}$.

23 *For all* $C \in \mathcal{C}$ *and* $i \in L'_C$, $|S'_{C,i}| \leq (1 - r_{\alpha,d})(1 - \frac{\alpha-d}{\alpha} r_{\alpha,d})^{k-1}(\alpha - d) + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.

We next randomly delete colours from each vertices list so that:

24 *For all* v, $|L'_v| = \lceil (1 - \frac{\alpha-d}{\alpha} r_{\alpha,d})^k \alpha - (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}\rceil$.

Note that (22) and (23) still hold. Now, we want to examine the relationship between the size of the L'_v and the $S'_{C,i}$ more carefully. To this end, we let $r' = 1 - \frac{\alpha-d}{\alpha} r_{\alpha,d}$ and let $\alpha' = \alpha(r')^k$.

Now, $|L'_v| = \lceil \alpha' - (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}\rceil$, whilst for all choices of C and i:
$|S'_{C,i}| \leq (1 - r_{\alpha,d})(r')^{k-1}(\alpha - d) + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.
Since $r' = 1 - r_{\alpha,d} + \frac{d}{\alpha} r_{\alpha,d}$, we obtain:
$|S'_{C,i}| \leq (r')^k\alpha - \frac{d}{\alpha} r_{\alpha,d}(r')^{k-1}\alpha - (1 - r_{\alpha,d})(r')^{k-1}d + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.
Simplifying, we have:
$|S'_{C,i}| \leq (r')^k\alpha - (r')^{k-1}d + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.
If we write d as $c\alpha^{1-\frac{1}{k}}$ then we have:
$|S'_{C,i}| \leq \alpha' - c(\alpha')^{1-\frac{1}{k}} + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.
Setting, $\alpha^* = \lceil \alpha' - (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}\rceil$, we obtain:

25 *For all* v, $L'_v = \alpha^*$

and

26 *For all $C \in \mathcal{C}$ and $i \in L'_C$, $|S'_{C,i}| \leq \alpha^* - c(\alpha^*)^{1-\frac{1}{k}} + 2(log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}$.*

It is these last two inequalities that allow us to iterate this procedure. I.e, we begin with $\alpha_0 = \lceil \Delta + (log\Delta)^3\Delta^{1-\frac{1}{k}} \rceil$. We set $L_v = \{1, ..., \alpha\}$ and for each $c \in \mathcal{C}$ we set $L_C = \{1, ..., \alpha\}$ and $S_{C,i} = C$ for all $i \in L_C$. We set $c_0 = (log\Delta)^3$ and set $d_0 = \lfloor c_o\Delta^{1-\frac{1}{k}} \rfloor$. Having obtained α_{j-1}, we apply our procedure again with $c_j = (log\Delta)^3 - 5j((log\Delta)^{\frac{3}{2}}(loglog\Delta)^2)$ and $d_j = \lfloor c_j\alpha^{1-\frac{1}{k}} \rfloor$. Recursively considering (25) and (26) ensures that this choice of d_i is possible. We stop when α_j first falls below $(log\Delta)^{4k}$. At this point, repeated applications of (22) (we omit the details) ensure that for every clique C, $|U_C| \leq 4\Delta\frac{\alpha_j}{\alpha_0}^{\frac{1}{k}} \leq 4(log\Delta)^4\Delta^{1-\frac{1}{k}}$. Thus, the uncoloured vertices form a hypergraph of maximum degree $4(log\Delta)^4\Delta^{1-\frac{1}{k}}$ which can be coloured using $4k(log\Delta)^4\Delta^{1-\frac{1}{k}}$ new colours.

4 List Colouring Linear Hypergraphs

We turn now to the list colouring problem. We can again assume that Δ is at least Δ_k because any grapher of smaller degree can be coloured with $k\Delta_k$ colours which we can add to the constant c.

Our procedure for list edge colouring linear hypergraphs is similar to that given above. Of course, the initial L_v are given as input to the problem, we only insist that each have at least $\Delta + \lceil 3k(log\Delta)^6 Delta^{1-\frac{1}{k}} \rceil$ elements and then truncate them so they all have exactly this many elements. The only complication arises at the end of the procedure, as we cannot simply introduce a new set of colours to finish things off. Instead, we shall choose a set of colours at the beginning of the procedure which will be used to colour the vertices still uncoloured after our general iterative procedure has run out of steam.

The preprocessing procedure takes as input a hypergraph with $\Delta + \lceil 3(log^{\Delta})^6\Delta^{1-\frac{1}{k}} \rceil$ colours on each list. It chooses a set $Save_C$ of colours for each clique C which cannot be used on C during the main colouring procedure but rather are saved for the cleanup. Thus, during the cleanup, a vertex v of the linegraph can only be coloured with a colour in $Sace_v = L_v \cap (\cap\{Save_C | v \in C\})$. Of course, during the main procedure, v can only be coloured with a colour in $Avail_v = L_v - (\cup\{Save_C | v \in C\})$.

If we chose each $Save_C$ randomly, putting each colour in $SaveC$ with probability $(log\Delta)^6\Delta^{\frac{-1}{k}}$, then we would expect $|Save_v|$ to be about $(log\Delta)^{6k}$ and $|Avail_v|$ to be about $\Delta + 2k(log\Delta)^6\Delta^{1-\frac{1}{k}}$. Furthermore, for each colour i in $Save_C$ we would expect $T_{C,i} = \{v \in C | i \in Save_v\}$ to contain at most $(log\Delta)^{6k-6}\Delta^{\frac{1}{k}}$ vertices. It follows that we can use a quasi-random procedure for choosing the $Save_C$ which ensures that (i) each $|Save_v|$ has size at least $\frac{(log\Delta)^{6k}}{2}$, (ii) each $Avail_v$ has size at least $\Delta + k(log\Delta)^6\Delta^{\frac{1-1}{k}}$, and (iii) for each $C \in \mathcal{C}$ and each $i \in Save_C$, the size of $T_{C,i}$ is at most $2(log\Delta)^{6k-6}\Delta^{\frac{1}{k}}$. We omit the details of this process.

We will now perform our standard iterative procedure, with one slight twist, we also need to keep track of the number of uncoloured vertices remaining in each $T_{C,i}$ after each iteration. Analagously to (22), we can choose our colouring so that letting $\mathcal{T}$ be the set of the all the $T_{C,i}$ (restricted to the set of uncoloured vertices in the given iteration):

27 *For all* $T \in \mathcal{T}$, $|U_T| \leq (1 - r_{\alpha,d})|T| + (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{|T|}$.

Using the same technique used for computing the final value of $|U_C|$, we can show that at the end of the procedure there are at most $|T_{C,i}|4\frac{\alpha_j}{\alpha_o}^{\frac{1}{k}} \leq 8(log\Delta)^{6k-2}$ uncoloured elements in each $T_{C,i}$. Because each vertex has $log\Delta^{6k}$ available colours in $Save_v$ and each of these colours is available on at most $4klog\Delta^{6k-2}$ of its neighbours, it follows that a quasi-random procedure can be used to finish off the colouring, we omit the details.

5 The Concentration Results

Consider an arbitrary iteration of our main procedure. That is we have the line graph G of a k-uniform hypergraph H and lists satisfying (i) $|L_v| = \alpha$ for all v, and (ii) $|S_{C,i}| \leq \alpha - d$ for all C and $i \in L_C$. We have shown

28 $\forall v, \quad Ex(|L'_v|) \approx (1 - (\frac{\alpha-d}{\alpha})r_{\alpha,d})^k\alpha$

29 $\forall C \in \mathcal{C}, \ \forall i \in L_C, \quad Ex(|S'_{C,i}|) \approx (1 - r_{\alpha,d})(1 - (\frac{\alpha-d}{\alpha}r_{\alpha,d})^{k-1}(\alpha - d)$

30 $\forall S \subseteq V(G), Ex(|U_S|) \approx (1 - r_{\alpha,d})|S|$

In this section, we show

31 $\forall v, Prob(||L'_v| - Ex(|L'_v|)| > (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}) \leq 2\Delta^{\frac{loglog\Delta}{2}}$

32 $\forall C \in \mathcal{C}, \ \forall i \in L_C, \quad Prob(||S'_{C,i}| - Ex(|S'_{C,i}|)| > (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{\alpha}) \leq 2\Delta^{\frac{loglog\Delta}{2}}$

33 $\forall C \in \mathcal{C}, \quad T \subseteq C, Prob(||U_T| - Ex(|U_T|)| > (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{|T|}) \leq 2\Delta^{\frac{loglog\Delta}{2}}$

From these facts, an easy application of the Local Lemma implies that we can choose a colouring satisfying (21), (22), (23),(27). To see this, for each vertex v, let A_v be the event that (21) fails for v. For each clique C, let A_C be the event that (22) fails for C. For each clique C and $i \in L_C$, let $A_{C,i}$ be the event that (23) fails for C. For each $T \in \mathcal{T}$, let A_T be the event that (27) fails for T. Let $\mathcal{A}$ be the set of all events we just defined. Then, (31),(32), and (33) ensure that each event in $\mathcal{A}$ holds with probability at most $2\Delta^{\frac{loglog\Delta}{2}}$. Further, each event in $\mathcal{A}$ depends only on the colour of vertices within distance three of some clique in $\mathcal{C}$ which contains the set of vertices by which it is indexed. Moreover, each clique contains the indexing set of at most $2k\Delta$ events. So, each event is mutually independent of a set of events containing all but at most Δ^9 of the other events. Thus, an application of the Local Lemma yields the desired result.

We turn now to the proof of the concentration results. We will need the following tool:

Azuma's Inequality [2] *Let $Y = Y_1, Y_2, \ldots, Y_n$ be a sequence of random events. Let $X = X(Y_1, Y_2, \ldots, Y_n)$ be a random variable defined by these Y_i. If for each i:*

$$\max \mid Ex(X \mid Y_1, Y_2, \ldots Y_{i+1}) - Ex(X \mid Y_1, Y_2, \ldots Y_i) \mid \leq c_i$$

then the probability that $\mid X - Ex(X) \mid > a$ is at most $2e^{-a^2/(2\sum c_i^2)}$.

With this tool in hand we can concentrate on constructing the proof.

Consider first the number of uncoloured vertices in some clique C of $\mathcal{C}$. This depends only on the colour of the vertices in C and the colour of the set of vertices not in C but adjacent to a vertex in C with which they share an available colour. There are at most $k\alpha^2|C|$ vertices in this second set, which we shall call W.

We consider assigning colours to the vertices in W first and then assigning colours to the vertices in C. Clearly, the colour assigned to any particular vertex can affect the expected number of coloured vertices in C by at most 2, for changing its colour will have no effect on any vertex not assigned either the old or new colour. Moreover, changing the colour of a vertex w in W will affect the expected number of uncoloured vertices in C only if one of the vertices of C that it is adjacent to w receives either the old or the new colour. Now, as the linearity of H implies that each vertex in W has at most k neighbours in C, the probability that this will occur is at most $\frac{2k}{\alpha}$ ($\frac{2}{\alpha}$ per neighbour in C). Thus, assigning a colour to a vertex in W affects $Ex(|U_C|)$ by at most $\frac{4k}{\alpha}$. Setting $l_C = \sum_{v \in W} (\frac{4k}{\alpha})^2 + \sum_{v \in C} 2^2$ and applying Azuma's inequality, we obtain:

$Prob(||U_C| - Ex(|U_C|)| > t) \leq e^{\frac{-t^2}{2l_C}}$

Since $l_C \leq 16k^3|C| + 4|C|$, setting $t = (log\Delta)^{\frac{3}{2}}(loglog\Delta)^2\sqrt{|C|}$ yields (33) when T is all of C.

The general concentration results follow in a similar manner. However, the details get quite hairy, and we omit them.

6 An Algorithm

Beck(see [1]) developed a polynomial time algorithm for one specific application of the Local Lemma. Alon(again see [1]) developed a paralllel and slightly faster version of the algorithm. In [10], Molloy and Reed gave a Local Lemma-like set of conditions which if satisified not only guarantee the existence of the desired structure but also ensure that a polynomial time (deterministic) algorithm exists for finding such a structure. This set of conditions seem to apply to most (if not all) of the known applications of the Local Lemma.

As a special case of this result, Molloy and Reed proved the following:

Lemma 34 (Local Lemma Like Lemma) *Let F be an l-bounded hypergraph of maximum degree d. For some integer n, we consider the set of n-colourings of $V(F)$. For each edge e, we are explicitly given a boolean function bad on the set of n-colourings of $V(e)$. Suppose that for each e, we have: $Prob(bad(e) = 1) < \frac{1}{4(dl)^6}$. Then there exists an n colouring such that every $bad(e)$ is 0, and we can find such a colouring in polynomial time (assuming l,d,n are fixed).*

We show below that this lemma implies that for fixed k and Δ, we can obtain colourings of k-uniform Δ regular linear hypergraphs with the number of colours gauranteed by our results. In fact, one can show that as long as Δ is $o(\frac{\log n}{loglog\Delta})$ then Molloy and Reed's nore general results can be used to generate the colourings of k-bounded linear hypergraphs of maximum degree Δ guaranteed to exist by our results. We save the messy details for a longer version of this paper.We also postpone the discussion of similar algorithms for colouring hypergraphs of bounded co-degree.

We turn now to proving that an algorithm exists for the special case of k-uniform Δ-regular linear hypergraphs. So, we let H be a such a hypergraph. We can assume that $\Delta(H)$ is at least Δ_k as otherwise we can simply colour $E(H)$ greedily. Now, consider an arbitrary iteration of our colouring procedure. That is, we have a list colouring problem in a subhypergraph of H satisfying (14) and we want to obtain a partial colouring satisfying (21),(22),(23), and (27). We let F be the hypergraph with the same vertex set as the line graph G of H such that there is a one to one correspondence between the edges of F and the uncoloured vertices of G and for each uncoloured vertex v of G, the corresponding edge e_v of F consists of those vertices within distance 4 of v in G. For each edge e_v of F, we define $bad(e_v)$ to be 1 if either A_v fails for v, or A_S fails for some S containing v. Otherwise, $bad(e_v) = 0$. It is clear that $bad(e_v)$ depends only on the colouring of the vertices of e_v. Furthermore, each edge of F contains at most $(k\Delta)^6$ vertices, and each vertex of F is in at most $k\Delta^6$ edges. The results (31),(32), and (33) ensure that for each v we have: $Prob(bad(e_v)) < p = (k\Delta + 1)(2\Delta^{\frac{loglog\Delta}{2}})$. So applying the Local Lemma like Lemma with $l = d = (k\Delta)^6$ we obtain the desired colouring. The preprocesssing and postprocessing steps can be handled similarily.

References

[1] N. Alon and J. Spencer, *The Probabilistic Method.* Wiley (1992).

[2] K. Azuma, *Weighted sums of certain dependent random variables.* Tokuku Math. Journal **19** (1967), 357 - 367.

[3] B. Bollobás and A. Harris, *List-colourings of graphs*, Graphs and Combinatorics **1** (1985), 115 - 127.

[4] R.L. Brooks, *On colouring the nodes of a network*, Proc. Cambridge Phil. Soc. **37** (1941), 194 - 197.

[5] P. Erdős and L. Lovász, *Problems and results on* 3*-chromatic hypergraphs and some related questions*, in: "Infinite and Finite Sets" (A. Hajnal et. al. Eds), Colloq. Math. Soc. J. Bolyai 11, North Holland, Amsterdam, 1975, 609-627.

[6] P. Erdős, A. Rubin and H. Taylor, *Choosability in graphs*, Congr. Num. **26** (1979), 125 - 157.

[7] J. Combinatorial Th. (A), **73** (1996), 1 - 59.

[8] J. Kahn, *Asymptotically good list-colorings*, J. Combinatorial Th. (A), **73** (1996), 1 - 59.

[9] C. McDiarmid, *On the method of bounded differences.* Surveys in Combinatorics, Proceedings of the Twelfth British Combinatorial Conference (1989), 148 - 188.

[10] M. Molloy and B. Reed, *Algorithms from the Local Lemma*, in preparation.

[11] M. Molloy and B. Reed, *Graph Colouring via the Probabilistic Method*, submitted.

[12] V. Vizing, *Some unsolved problems in graph theory*, Russian Math Surveys **23**, pp. 125 - 141 (1968).

[13] V. Vizing, *Colouring the vertices of a graph with prescribed colours*, Diskret. Analiz. **29** (1976), 3 - 10.