Journal of Combinatorial Theory, Series B 134 (2019) 264–284

Contents lists available at ScienceDirect

Journal of Combinatorial Theory, Series B

www.elsevier.com/locate/jctb

The list chromatic number of graphs with small clique number

Michael Molloy [1]

Dept of Computer Science, University of Toronto, Canada

ARTICLE INFO

Article history:
Received 31 January 2017
Available online 13 July 2018

Keywords:
List colouring
Local lemma
Entropy compression

ABSTRACT

We prove that every triangle-free graph with maximum degree Δ has list chromatic number at most $(1+o(1))\frac{\Delta}{\ln\Delta}$. This matches the best-known upper bound for graphs of girth at least 5. We also provide a new proof that for any $r \geq 4$ every K_r-free graph has list-chromatic number at most $200r\frac{\Delta\ln\ln\Delta}{\ln\Delta}$.

1. Introduction

We provide new proofs of two results of Johansson. The proofs are much shorter and simpler, and obtain an improvement in the constant of the first result. We use entropy compression, a powerful new take on the Lovász Local Lemma.

The first result bounds the list chromatic number of a triangle-free graph. The list chromatic number of a graph G is the smallest q such that: for any assignment of colour-lists of size q to each vertex, it is possible to give each vertex a colour from its list and obtain a proper colouring. Johansson [17] proved that every triangle-free graph has list-

E-mail address: molloy@cs.toronto.edu.

[1] Research supported by an NSERC Discovery Grant.

https://doi.org/10.1016/j.jctb.2018.06.007

chromatic number at most $9\Delta/\ln\Delta$ where Δ is the maximum degree of the graph. The leading constant was improved to 4 in [25]. Here we obtain $1+o(1)$:

Theorem 1. *For every $\epsilon > 0$ there exists Δ_ϵ such that every triangle-free graph G with maximum degree $\Delta \geq \Delta_\epsilon$ has $\chi_\ell(G) \leq (1+\epsilon)\Delta/\ln\Delta$.*

In other words: every triangle-free graph with maximum degree Δ has list chromatic number at most $(1+o(1))\frac{\Delta}{\ln\Delta}$.

The bound in Theorem 1 matches the best known upper bound for graphs of girth five [19], and indeed for any constant girth. The best known lower bound is $\frac{1}{2}\frac{\Delta}{\ln\Delta}$ and comes from random Δ-regular graphs. For constant Δ, random Δ-regular graphs are essentially high girth graphs: For any constant K, we expect $O(1)$ cycles of length greater than K, and so we can form a high-girth graph by removing a relatively small number of edges; furthermore, those edges form a matching, and so this changes the chromatic number by at most one.

This bound matches what is called the *shattering threshold* for colouring random regular graphs [32], which is often referred to as the "algorithmic barrier" [1,32]. This threshold arises in a wide class of problems on random graphs, and finding an efficient algorithm to solve any of these problems for edge-densities beyond the algorithmic barrier is a major open challenge (see e.g. [1]); for colourings of random regular graphs, this means finding an efficient algorithm using $(1-\epsilon)\frac{\Delta}{\ln\Delta}$ colours for some $\epsilon > 0$. Our proof of Theorem 1 yields an efficient randomized algorithm to find a colouring for maximum degree up to the algorithmic barrier, not just for random regular graphs (where such algorithms are previously known [4]), but for every triangle-free graph.

In a followup paper, Johansson [17] proved that for any constant $r \geq 4$, every K_r-free graph has list-chromatic number at most $O(\Delta\ln\ln\Delta/\ln\Delta)$. Here we match his bound, even when r grows with Δ.

Theorem 2. *For any $r \geq 4$, every K_r-free graph G with maximum degree Δ has $\chi_\ell(G) \leq 200r\frac{\Delta\ln\ln\Delta}{\ln\Delta}$.*

Theorem 2 holds for any r but it is trivial unless $r < \ln\Delta/200\ln\ln\Delta$. Note also that this implies a bound on the chromatic number of H-free graphs for every fixed subgraph H, as an H-free graph is also $K_{|H|}$-free. (Here H-free means that there is no subgraph isomorphic to H; the subgraph is not necessarily induced.)

These two results of Johansson were never published. His proof for triangle-free graphs was presented in [21] and his proof for K_r-free graphs was presented in [25].

It is a longstanding conjecture [7] that for constant r, every K_r-free graph has chromatic number $O(\Delta/\ln\Delta)$. So we make no attempt to optimize the constant in Theorem 2. Thus far, we do not even know whether the independence number is large enough to support this conjecture. Prior to Johansson's work, Shearer [29,30] proved that every triangle-free graph on n vertices has independence number at least

$(1 - o(1))n \ln \Delta/\Delta$ (see also [5]) and that every K_r-free graph has independence number at least $\Omega(n \ln \Delta/\Delta \ln \ln \Delta)$. His latter bound plays an important role in our proof of Theorem 2. Ajtai et al. conjectured that the $\ln \ln \Delta$ term can be removed here [5].

Previous proofs of these, and similar results, used an iterative colouring procedure. In each iteration, one would colour some subset of the vertices, where each vertex received a random colour from its list. Every vertex that received the same colour as a neighbour would be uncoloured. (See [21] for a presentation of this technique.) One of the reasons for doing this is that the Local Lemma is much easier to apply when vertices are assigned colours independently. Entropy compression allows us to use Local Lemma like calculations for random colouring procedures where, roughly speaking, vertices are coloured one-at-a-time with colours not appearing on any neighbours.

This technique began with Moser's algorithm [22] which generated solutions to k-SAT whose existence was guaranteed by the Local Lemma; this was then extended by Moser and Tardos [23] to a very wide range of applications of the Local Lemma. (See [31,14] for good expositions of the technique.) Subsequently, Grytczuk, Kozik and Micek [16] and Achlioptas and Iliopoulos [2] noted that this algorithm in fact can be applied to yield new existence results. Previous applications to graph colouring (e.g. [13,26,3,27,9,11,15]) involved situations where, throughout the algorithm, each vertex is guaranteed to have a large number of available colours to choose from. That is not true in this paper since the degree of a vertex can be much higher than its list-size. The novelty we use here is to treat a vertex having a small number of available colours as a bad event.

2. Preliminary tools

We begin with a common version of the Local Lemma; see e.g. Chapter 19 of [21].

The Lovász Local Lemma. *[12] Let $A_1, ..., A_n$ be a set of random events, each with probability at most $\frac{1}{4}$. Suppose that for each $1 \le i \le n$ we have a subset $\mathcal{D}_i$ of the events such that A_i is mutually independent of all other events outside of $\mathcal{D}_i$. If for each $1 \le i \le n$ we have*

$$\sum_{j \in \mathcal{D}_i} \mathbf{Pr}(A_j) < \frac{1}{4},$$

then $\mathbf{Pr}(\overline{A_1} \cap ... \cap \overline{A_n}) > 0$.

We say that boolean variables $X_1, ..., X_m$ are *negatively correlated* if

$$\text{for all } I \subseteq \{1, ..., m\} : \qquad \mathbf{Pr}\left(\wedge_{i \in I} X_i\right) \le \prod_{i \in I} \mathbf{Pr}(X_i).$$

Panconesi and Srinivasan [24] noted that many Chernoff-type bounds on independent variables also hold on negatively correlated variables. We will use the following:

Lemma 3. *Suppose $X_1, ..., X_m$ are boolean variables, and set $Y_i = 1 - X_i$. Set $X = \sum_{i=1}^m X_i$. Then for any $0 < t \leq \boldsymbol{E}(X)$:*

(a) If $X_1, ..., X_m$ are negatively correlated then $\mathbf{Pr}(X > \boldsymbol{E}(X) + t) < e^{-t^2/3E(X)}$.
(b) If $Y_1, ..., Y_m$ are negatively correlated then $\mathbf{Pr}(X < \boldsymbol{E}(X) - t) < e^{-t^2/2E(X)}$.

In this paper, we only require part (b).

Part (a) follows from Corollary 3.3 of [24]. The proof of part (b) is very similar and we sketch it here.

For independent variables, the bound follows from standard Chernoff-type bounds; e.g. we refer to Theorem 2.3(c) in [20]. To adapt the proof so that it holds when $Y_1, ..., Y_m$ are negatively correlated, we only need one change. Set $Y = \sum_{i=1}^m Y_i = m - X$. The proof for independent variables uses that for any $h > 0$:

$$\mathbf{E}(e^{hY}) = \mathbf{E}(\prod_{i=1}^m e^{hY_i}) = \prod_{i=1}^m \mathbf{E}(e^{hY_i}).$$

We replace this with

$$\mathbf{E}(e^{hY}) = \mathbf{E}(\prod_{i=1}^m e^{hY_i}) \leq \prod_{i=1}^m \mathbf{E}(e^{hY_i}). \tag{1}$$

The highlights of the proof from [20] are: Set $p_i = \mathbf{Pr}(Y_i)$ for each i and set $p = \sum p_i/m = \mathbf{E}(Y)/m$. For any $h > 0$ we have $\mathbf{E}(e^{hY_i}) = 1 - p_i + p_i e^h$ and so (1) and the arithmetic mean-geometric mean inequality yield

$$\mathbf{E}(e^{hY}) \leq \prod_{i=1}^m (1 - p_i + p_i e^h) \leq (1 - p + pe^h)^m.$$

Thus $\mathbf{Pr}(Y \geq s) \leq e^{-hs}(1 - p + pe^h)^m$. A good choice of h (see the proof of Lemma 2.2 in [20]) yields that for any $0 \leq z \leq 1$,

$$\mathbf{Pr}(X \leq E(X) - mz) = \mathbf{Pr}(Y \geq E(Y) + mz) \leq \left(\left(\frac{p}{p+z}\right)^{p+z}\left(\frac{1-p}{1-p-z}\right)^{1-p-z}\right)^m.$$

Now set $t = mz$ and apply some calculus (see the proof of Lemma 2.3(c) in [20]) to obtain the bound for Lemma 3(b).

Remark. Intuitively, it seems that when $X_1, ..., X_m$ are negatively correlated then typically $Y_1, ..., Y_m$ would also be negatively correlated. Indeed that is the case in the application of Lemma 3 in this paper. However, it is not always the case. Choose a string from an urn containing two copies of the strings $\{000, 011, 101, 110\}$ and one copy of each of the other boolean strings of length three. Let X_i be the event that the ith digit is 1. Then X_1, X_2, X_3 are negatively correlated but $\mathbf{Pr}(Y_1 \wedge Y_2 \wedge Y_3) = \frac{1}{6} > \frac{1}{8} = \mathbf{Pr}(Y_1)\mathbf{Pr}(Y_2)\mathbf{Pr}(Y_3)$.

3. Triangle-free graphs

Each vertex v has a list $\mathcal{C}_v$ of colours that may be assigned to v of size

$$|\mathcal{C}_v| = q := (1+\epsilon)\frac{\Delta}{\ln \Delta}.$$

It suffices to prove Theorem 1 for small ϵ; in particular we will assume $\epsilon < 1$.

A partial list colouring σ is a colour assignment to a subset of the vertices, where the colours are drawn from their lists. Given a partial colouring, it is helpful if each vertex has many colours which do not appear on its neighbourhood. To this end, we set

$$L = \Delta^{\epsilon/2}.$$

Note that if Δ neighbours of v are each independently given a uniformly random colour from their lists, then the expected number of colours from $\mathcal{C}_v$ that are not chosen for any neighbour of v is at least $q\,(1-1/q)^{\Delta} \approx (1+\epsilon)\Delta^{\frac{\epsilon}{1+\epsilon}} / \ln \Delta > L$. So it is plausible that we can obtain a colouring in which every vertex has at least L colours which do not appear on its neighbourhood. In fact we will prove that we can obtain such a *partial* colouring with a substantial number of vertices coloured. From this, it will be straightforward to complete the colouring.

It will be convenient to treat Blank as a colour, and the uncoloured vertices are viewed as having been assigned this colour. Blank is the only colour that can be assigned to two neighbours. Most of our work goes towards finding a partial list colouring with certain properties that make it easy to complete to a full colouring.

We use N_v to denote the open neighbourhood of v (to be clear: $v \notin N_v$). Given a partial colouring σ, we define for each vertex v and colour $c \neq \mathsf{Blank}$:

L_v is the set of colours in $\mathcal{C}_v$ not appearing on N_v, along with Blank;

$T_{v,c}$ is the set of vertices $u \in N_v$ such that $\sigma(u) = \mathsf{Blank}$ and $c \in L_u$.

Note that the preceding definition does not apply to $T_{v,\mathsf{Blank}}$; it will be convenient to set $T_{v,\mathsf{Blank}} = \emptyset$ for all v.

Given a partial colouring, we define the following two *flaws* for any vertex v:

$$B_v \equiv |L_v| < L$$

$$Z_v \equiv \sum_{c \in L_v} |T_{v,c}| > \frac{1}{10} L \times |L_v|$$

We say v is the *vertex* of flaw $f = B_v$ or Z_v, and we denote $v(f) := v$.

Observation 4. *B_v is determined by the colours of the vertices in $N(v)$ and Z_v is determined by the colours of the vertices within distance two of v.*

Remark. If we were content with proving the weaker bound of $\chi_\ell(G) < (2 + o(1))\frac{\Delta}{\ln \Delta}$ colours, then we could have defined Z_v to be a much simpler flaw, namely that v has at least L blank neighbours. We use that flaw in Section 4.

Our main goal is to find a partial colouring which has no flaws. The following proof that such a colouring can be completed to a proper colouring with no blank vertices is essentially the proof of the main result in [28].

Lemma 5. *Suppose we have a partial list colouring σ such that for every vertex v, neither B_v nor Z_v hold. Then we can colour the blank vertices to obtain a full list colouring.*

Proof. We give each blank vertex v a uniformly chosen colour from $L_v \backslash \mathsf{Blank}$. For any edge uv and colour $c \in L_u \cap L_v$, $c \neq \mathsf{Blank}$ we define $A_{uv,c}$ to be the event that u, v both receive c. Then $\mathbf{Pr}(A_{uv,c}) = 1/(|L_u| - 1)(|L_v| - 1)$. Furthermore, $A_{uv,c}$ shares a vertex with at most $\sum_{c' \in L_v} |T_{v,c'}| + \sum_{c' \in L_u} |T_{u,c'}|$ other events. The number of such events is at most $\frac{1}{10} L(|L_v| + |L_u|)$ since Z_u, Z_v do not hold. It is straightforward to check that $A_{uv,c}$ is mutually independent of all events with which it does not share a vertex (see e.g. the Mutual Independence Principle in Chapter 4 of [21]). So our lemma follows from the Local Lemma as B_u, B_v do not hold and so

$$\begin{aligned}
&\frac{1}{(|L_u| - 1)(|L_v| - 1)} \times \frac{L(|L_v| + |L_u|)}{10} \\
&\quad \leq \frac{L}{10(|L_u| - 1)} \times \frac{|L_v|}{|L_v| - 1} + \frac{L}{10(|L_v| - 1)} \times \frac{|L_u|}{|L_u| - 1} \\
&\quad < \frac{1}{9} + \frac{1}{9} < \frac{1}{4},
\end{aligned}$$

for $\Delta > 20^{2/\epsilon}$; i.e. $L > 20$. $\square$

In the next section, we will present an algorithm to find a flaw-free colouring.

3.1. Our colouring algorithm

Consider a partial colouring σ and any flaw f of σ. We will use a recursive algorithm to correct f. Recall that every neighbourhood is an independent set, and so we recolour the vertices in a neighbourhood independently.

We use the following ordering on the flaws: every B_v comes before every Z_u, and the B_v's and Z_u's are each ordered according to the labels of v, u. We use $\mathrm{dist}(w, v)$ to denote the distance from w to v; i.e. the number of edges in a shortest w, v-path.

FIX(f, σ)
Set $v = v(f)$ and assign each $u \in N_v$ a uniformly selected colour from L_u.
While there are any flaws B_w with $\text{dist}(w, v) \leq 2$ or Z_w with $\text{dist}(w, v) \leq 3$:
  Let g be the least such flaw and call FIX (g, σ') where σ' is the current colouring.
Return the current colouring.

Remark. It is possible that f still holds after recolouring the neighbourhood of f, but then f itself would count as a flaw within distance 2 or 3 in the next line (but is not necessarily the least of those flaws). Note further that even if f does not hold after the recolouring, it is possible for future recolourings to bring f back and so FIX may be called again on f further down in the recursive calls.

Next we note that if FIX terminates, then we have made progress in correcting the flaws.

Observation 6. *In the colouring returned by FIX(f, σ):*

(a) f does not hold; and
(b) there are no flaws that did not hold in σ.

Proof. Part (a) is true because we cannot exit the while loop if f holds. Part (b) is true because any new flaw f' must have arisen during a call of FIX on some f'' whose vertex is within distance two or three of $v(f')$ (depending on whether f' is a B-flaw or a Z-flaw), as these are the only calls in which a vertex within distance one or two of $v(f')$ can be recoloured (see Observation 4). But we would not have exited the while loop of that call if f' still held. □

So we can obtain a flaw-free colouring by beginning with any partial colouring, e.g. the all-blank colouring, and then calling FIX at most once for each of the at most $2n$ flaws of that colouring. Thus it suffices to prove that FIX terminates with positive probability; in fact, we will show that with high probability it terminates quickly (see the remark at the end of Subsection 3.3).

In the next subsection we prove that the proportion of colourings of $N(v)$ for which f holds is at most Δ^{-4}. In Subsection 3.3 we use that to show FIX terminates. Note that there are at most $2\Delta^3$ flaws g which could appear in the while loop in FIX (f, σ). Since $2\Delta^3 \times \Delta^{-4} < \frac{1}{4}$ (for large Δ) this feels like a Local Lemma computation. Entropy compression allows us to use such a computation in a procedure like FIX, which is more complicated than what we would typically apply the Local Lemma to; in particular note how quickly dependency spreads amongst the various flaws while running FIX.

3.2. Probability bounds

In this section, we prove the key bounds on the probability of our flaws.

Setup for Lemma 7: Each vertex $u \in N_v$ has a list L_u containing Blank and perhaps other colours. We give each $u \in N_v$ a random colour from L_u, where the choices are made independently and uniformly. This assignment determines $L_v, T_{v,c}$.

Lemma 7.

(a) $\mathbf{Pr}(|L_v| < L) < \Delta^{-4}$.
(b) $\mathbf{Pr}(\sum_{c \in L_v} |T_{v,c}| > \frac{1}{10} L \times |L_v|) < \Delta^{-4}$.

Remarks. (1) This looks like an analysis of the probability that the recolouring in the first line of FIX produces another flaw on N_v. But we will actually apply it to count the number of choices for the flawed colouring that was on N_v before the recolouring. This subtlety is important if one attempts to adapt this proof by using a different recolouring procedure designed to have a low probability of producing a flaw.

(2) Kim's proof [19] for graphs of girth five was much simpler than Johansson's proof [17] for triangle-free graphs. The main reason was that if G has girth five then the neighbours of v have disjoint neighbourhoods (other than v) which resulted in their lists being, in some sense, independent of each other. In a triangle-free graph with many 4-cycles, we could have two neighbours u_1, u_2 of v whose neighbourhoods overlap a great deal and thus their lists would be highly dependent. Intuitively, it was clear that this should be helpful: if L_{u_1} and L_{u_2} are very similar then u_1, u_2 would tend to get the same colour which would tend to increase the size of L_v. But, frustratingly, we did not know how to take advantage of this. In the current paper, the fact that dependencies between L_{u_1}, L_{u_2} do not hurt is captured by the stronger fact that Lemma 7 holds for any set of lists on the neighbours of v, even lists produced by an adversary.

Proof. For each colour $c \in \mathcal{C}_v \backslash \{\mathsf{Blank}\}$ we define:

$$\rho(c) = \sum_{u \in N_v : c \in L_u} \frac{1}{|L_u| - 1}.$$

Thus, since each L_u has $|L_u| - 1$ non-Blank colours,

$$\sum_{c \in \mathcal{C}_v \backslash \{\mathsf{Blank}\}} \rho(c) \leq \sum_{u \in N_v} \sum_{c \in L_u \backslash \{\mathsf{Blank}\}} \frac{1}{|L_u| - 1} \leq \Delta. \tag{2}$$

Part (a): If $c \in L_u$ then $|L_u| \geq 2$ and so we have $1 - \frac{1}{|L_u|} > e^{-1/(|L_u|-1)}$. We apply this inequality to obtain:

$$\mathbf{E}(|L_v|) = 1 + \sum_{c \in \mathcal{C}_v \backslash \{\mathsf{Blank}\}} \prod_{u \in N_v : c \in L_u} \left(1 - \frac{1}{|L_u|}\right) > \sum_{c \in \mathcal{C}_v \backslash \{\mathsf{Blank}\}} e^{-\rho(c)}. \tag{3}$$

By convexity of e^{-x}, (2) and recalling that $|\mathcal{C}_v| = q = (1 + \epsilon)\Delta / \ln \Delta$ we have

$$\mathbf{E}(|L_v|) > qe^{-\Delta/q} = \frac{(1+\epsilon)\Delta}{\ln \Delta} \times \Delta^{-\frac{1}{1+\epsilon}} > 2\Delta^{\epsilon/2} = 2L,$$

for $\epsilon < 1$.

To prove concentration, we set X_c to be the indicator variable that $c \in L_v$; thus $|L_v| = 1 + \sum_{c \in \mathcal{C}_v \setminus \{\mathsf{Blank}\}} X_c$. We wish to apply Lemma 3(b) to bound the probability that $|L_v|$ is too small, and so we set $Y_c = 1 - X_c$ and argue that the variables $\{Y_c\}$ are negatively correlated.

Claim. *For any $I \subseteq \mathcal{C}_v \setminus \{\mathsf{Blank}\}$, $\mathbf{Pr}(\wedge_{c \in I} Y_c) \leq \prod_{c \in I} \mathbf{Pr}(Y_c)$.*

Proof. Consider any $I \subseteq \mathcal{C}_v \setminus \{\mathsf{Blank}\}$ and $c' \notin I$. We will first argue that

$$\mathbf{Pr}(\wedge_{c \in I} Y_c | X_{c'}) \geq \mathbf{Pr}(\wedge_{c \in I} Y_c). \tag{4}$$

To sample a colour assignment conditional on $X_{c'}$ we simply choose for each $u \in N_v$, a uniform colour from $L_u \setminus \{c'\}$. Since $c' \notin I$, it is clear that this does not decrease the probability that every colour in I is selected at least once, i.e. $\mathbf{Pr}(\wedge_{c \in I} Y_c)$. This establishes (4). This is equivalent to $\mathbf{Pr}(\wedge_{c \in I} Y_c | Y_{c'}) \leq \mathbf{Pr}(\wedge_{c \in I} Y_c)$, which is equivalent to

$$\mathbf{Pr}(Y_{c'} | \wedge_{c \in I} Y_c) \leq \mathbf{Pr}(Y_{c'}). \tag{5}$$

Applying (5) inductively yields the claim. □

Now Lemma 3(b) yields:

$$\mathbf{Pr}(|L_v| < \frac{1}{2}\mathbf{E}(|L_v|)) < e^{-\frac{1}{8}\mathbf{E}(|L_v|)} < e^{-\frac{1}{4}\Delta^{\epsilon/2}} < \Delta^{-4},$$

for Δ sufficiently large in terms of ϵ. This proves part (a).

Part (b): Let Ψ be the set of colours $c \in L_v \setminus \{\mathsf{Blank}\}$ with $\rho(c) > \Delta^{\epsilon/4}$. Using the same calculations as those for (3), but this time applying $1 - \frac{1}{|L_u|} < e^{-1/|L_u|} < e^{-1/2(|L_u|-1)}$ for $|L_u| \geq 2$, the probability that L_v contains at least one colour from Ψ is at most

$$\mathbf{E}(|L_v \cap \Psi|) < \sum_{c \in \Psi} e^{-\frac{1}{2}\rho(c)} < qe^{-\frac{1}{2}\Delta^{\epsilon/4}} < \frac{1}{2}\Delta^{-4},$$

for Δ sufficiently large in terms of ϵ. For any $c \notin \Psi$:

$$\mathbf{E}(|T_{v,c}|) = \sum_{u: c \in L_u} \frac{1}{|L_u|} < \rho(c) \leq \Delta^{\epsilon/4}.$$

Since the choices of whether $u \in T_{v,c}$, i.e. whether u receives Blank, are made independently, standard concentration bounds apply. E.g. Theorem 2.3(b) of [20] says that for any $\epsilon > 0$,

$$\mathbf{Pr}(|T_{v,c}| > (1+\epsilon)\mathbf{E}(|T_{v,c}|) < e^{-\epsilon^2 \mathbf{E}(|T_{v,c}|)/2(1+\frac{\epsilon}{3})},$$

which yields $\mathbf{Pr}(|T_{v,c}| > \mathbf{E}(|T_{v,c}|) + \Delta^{\epsilon/4}) < e^{-\frac{3}{8}\Delta^{\epsilon/4}}$. So the probability that there is at least one $c \notin \Psi$ with $|T_{v,c}| > 2\Delta^{\epsilon/4}$ is at most

$$qe^{-\frac{3}{8}\Delta^{\epsilon/4}} < \frac{1}{2}\Delta^{-4},$$

for sufficiently large Δ. So with probability at least $1 - \Delta^{-5}$ we have

$$\sum_{c \in L_v \setminus \{\mathsf{Blank}\}} |T_{v,c}| = \sum_{c \in L_v \setminus \Psi} |T_{v,c}| \leq 2\Delta^{\epsilon/4}|L_v| < \frac{1}{10} L \times |L_v|. \qquad \square$$

3.3. The algorithm terminates

The basic idea behind entropy compression is that a string of random bits cannot be represented by a shorter string. We will consider the string of random bits used for the recolouring steps of FIX and show that as we run FIX we can record a file which allows us to recover those random bits. Each time we call FIX (g, σ), we record the name of g and the colours of the vertices that determine g. It is not hard to see that this, along with the current colouring, will allow us to reconstruct all of the preceding random colour choices. Because the colours which determine g indicate that something unlikely occurred (namely the flaw g), we can represent those colours in a very concise way. However, it may take a large amount of space to record the name of g. So instead, we use the degree bound in our graph to record a concise piece of information that will allow us to determine the name of g. This will lead to a compression of those random colour choices if the algorithm continues for too many steps.

First we describe these concise representations. Consider any vertex v. Let $N^3(v)$ denote the set of vertices within distance 3 of v (including v itself). For each $1 \leq \ell \leq |N^3(v)| < \Delta^3$ we let $\omega(\ell, v)$ denote the ℓth vertex of $N^3(v)$ when those vertices are listed in order of their labels. When we call, e.g. FIX (B_w, σ') while running FIX (Z_v, σ''), rather than recording the name "B_w" it will suffice to just record "(B, ℓ)" where $w = \omega(\ell, v)$. So despite the fact that the number of vertices, and hence the size of the label of w, is not bounded in terms of Δ, we are able to record w using only roughly $3\log_2 \Delta$ bits.

Suppose that we are given a collection of lists $\mathcal{L} = \{L_u : u \in N_v\}$ of available colours for the neighbours of v. Let $\mathcal{B}(\mathcal{L})$, resp. $\mathcal{Z}(\mathcal{L})$ be the set of all colour assignments from these lists such that B_v, resp. Z_v, holds. Lemma 7 implies that $|\mathcal{B}(\mathcal{L})|, |\mathcal{Z}(\mathcal{L})| < \Delta^{-4}\prod_{u \in N_v} |L_u|$. For each $1 \leq \ell \leq |\mathcal{B}(\mathcal{L})| + |\mathcal{Z}(\mathcal{L})|$, we let $\beta(\ell, \mathcal{L})$ denote the ℓth member of $\mathcal{B}(\mathcal{L}) \cup \mathcal{Z}(\mathcal{L})$ in some fixed ordering. When we run, e.g. FIX (B_v) we record the colours of N_v *before they get recoloured*; but instead of listing all the colours, we only need to record the value ℓ such that those colours are $\beta(\ell, \mathcal{L})$.

We add some write statements to FIX as follows.

FIX(f, σ)
Set $\mathcal{L} = \{L_u : u \in N_{v(f)}\}$.
Write "COLOURS $= \ell$" where $\beta(\ell, \mathcal{L})$ is the colouring of $N_{v(f)}$.
(*) Set $v = v(f)$ and assign each $u \in N_v$ a uniformly selected colour from L_u.
While there are any flaws B_w with $\text{dist}(w, v) \leq 2$ or Z_w with $\text{dist}(w, v) \leq 3$:
Let g be the least such flaw and call **FIX**(g, σ') where σ' is the current colouring.
Write "FIX (B,ℓ)" or "FIX (Z,ℓ)" (depending on whether g is a B-flaw or an Z-flaw) where $v(g) = \omega(v(f), \ell)$
Return the current colouring.
Write "Return"

Let σ_0 be any initial colouring and let f be any flaw of σ_0. We will analyze a run of FIX (σ_0, f). After t executions of the line (*) we set

σ_t is the current colouring

H_t is the file that we write to

R_t is the string of random bits that were used for all executions of (*)

In our formal proofs, we will not in fact make use of R_t; we only use it to give an intuitive picture of the compression of our random bits. Thus we are not careful about issues such as ensuring that each random choice uses an integer number of bits.

Lemma 8. *Given $\sigma_0, \sigma_t, f, H_t$ we can reconstruct the first t steps of FIX.*

Proof. Let f_i denote the flaw addressed during the ith execution of (*). First observe that $f_1, ..., f_t$ can be determined by σ_0, f, H_t. Indeed, proceed inductively: We know the sequence $f_1 = f, ..., f_{i-1}$. FIX (f_i, σ_{i-1}) was called while executing FIX (f_j, σ_{j-1}) for some $j < i$. The locations of the "Return" lines in H_t are enough to determine the value of j, and by induction we know f_j. So the ith "FIX (-,ℓ)" line tells us that $v(f_i) = \omega(v(f_j), \ell)$ and also tells us whether $f_i = B_{v(f_i)}$ or $Z_{v(f_i)}$.

Next observe that, having determined $f_1, ..., f_t$, we can reconstruct the colours assigned in each execution of (*) from H_t and σ_t. To see this, note that we can reconstruct σ_{t-1} from H_t, σ_t, f_t. We know that $\sigma_{t-1} = \sigma_t$ on all vertices other than $N_{v(f_t)}$. This and the fact that our graph is triangle-free imply that for every $u \in N_{v(f_t)}$, the list L_u does not change during step t. So the collection of lists $\mathcal{L} = \{L_u : u \in N(v(f_t))\}$ does not change during the tth recolouring and so σ_t and the tth "COLOURS$=\ell$" line allows us to recover $\sigma_{t-1}(N_{v(f_t)}) = \beta(\ell, \mathcal{L})$. Furthermore, $\mathcal{L}$ and $\sigma_t(N_v)$ tell us what colours were selected during the tth execution of (*). Working backwards, this determines $\sigma_t, \sigma_{t-1}, ..., \sigma_1$ and hence all of our random choices. $\square$

So R_t can be represented by $(\sigma_0, \sigma_t, f, H_t)$. The essence of the remainder of our argument is that if FIX (σ_0, f) continues for t steps, where t is large, then $(\sigma_0, \sigma_t, f, H_t)$ when expressed in binary will be much shorter than R_t. Any method to represent a random string of bits by a much shorter string must fail w.h.p. So this implies that w.h.p. we terminate before very many steps.

The rough idea is: During the ith execution of (*), recall that f_i is the flaw being addressed and define:

$$\Lambda_i = \prod_{u \in N_{v(f_i)}} |L_u| \text{ at the time of the } i\text{th execution of (*).}$$

The ith execution of (*) selects one of Λ_i possible colourings of $N_{v(f_i)}$ and so the total number of random bits used during the first t executions is $\sum_{i=1}^{t} \log_2 \Lambda_i$. Note that this number depends on the actual random choices that are made.

After t executions of (*) H_t consists of: (a) $t-1$ "FIX(-,ℓ)" lines in which $\ell < \Delta^3$; (b) t "COLOURS $= \ell$" lines in which the ith such line has $\ell \leq |\mathcal{B}(\mathcal{L})| + |\mathcal{Z}(\mathcal{L})| < 2\Delta^{-4}\Lambda_i$; (c) fewer than t "Return" lines. So the total number of bits required to record H_t is

$$\sum_{i=1}^{t} [3\log_2 \Delta + \log_2(2\Delta^{-4}\Lambda_i) + O(1)] = -t(\log_2 \Delta + O(1)) + \sum_{i=1}^{t} \log_2 \Lambda_i.$$

Thus in each execution of (*) writing to H_t requires roughly $\log_2 \Delta$ fewer bits than the number of random bits added to R_t.

Letting n be the number of vertices, the number of choices for each of the partial list colourings σ_0, σ_t is at most q^n and there are $2n$ choices for f. So to record (σ_0, σ_t, f) requires $2n \log_2 q + \log_2 n + 1 < 2n \log_2 \Delta$ bits (for sufficiently large n). The main point is that this does not change with t and so if t is large in terms of n then $|(\sigma_0, \sigma_t, f, H_t)| \leq |R_t|$, as required.

Annoying technical issues arise when Λ_i is not a power of 2, and so our formal proof will use direct probability bounds in which sizes of the bitstreams are only implicit.

Lemma 9. *For any partial colouring σ and any flaw f of σ, the probability that* FIX (f, σ) *continues for at least $2n$ executions of (*) is at most $\Delta^{-n/2}$, where n is the number of vertices.*

Proof. Set $T = 2n$ and run FIX (f, σ) until it either terminates or carries out T executions of (*).

Let $\mathcal{Q}$ be any possible run of FIX (f, σ) that lasts for at least T executions. At the ith execution, recall that $\Lambda_i = \prod_{u \in N_{v(f_i)}} |L_u|$ is the number of choices for the recolouring. We choose this recolouring by taking a uniform integer x_i from $\{1, ..., \Lambda_i\}$. Note that Λ_i is determined by f, σ and $x_1, ..., x_{i-1}$. Set $\Lambda = \Lambda(\mathcal{Q}) = \prod_{i=1}^{T} \Lambda_i$ and set $\lambda = \lambda(\mathcal{Q}) = \lfloor \log_2 \Lambda \rfloor$ (intuitively, λ can be thought of as the number of random bits generated). The probability that we carry out the run $\mathcal{Q}$ is $1/\Lambda \leq 2^{-\lambda}$.

Note that $\Lambda_i \leq (q+1)^\Delta$ for each i and so $\lambda < T\Delta \log_2(q+1) < T\Delta \log \Delta$.

Given $\sigma_0 = \sigma$ and f, Lemma 8 says that H_T, σ_T determine $\mathcal{Q}$. So we will enumerate the number of choices for $\mathcal{Q}$ by enumerating the number of choices for (H_T, σ_T). We will do this by considering the size of a string encoding (H_T, σ_T) in binary.

The number of choices for σ_T is $(q+1)^n$, so it can be recorded with $\lceil n\log_2(q+1)\rceil$ bits. The ith line of H_T consists of: (1) a FIX line containing a number of size at most $2\Delta^3$; it requires $3\log_2\Delta + O(1)$ bits; (2) we either do or do not write a "Return" line; this costs $O(1)$ bits; (3) a COLOURS line containing a number of size at most $2\Delta^{-4}\Lambda_i$; it requires $\log_2\Lambda_i - 4\log_2\Delta + O(1)$ bits. So the total size of the string recording (H_T, σ_T) and hence recording $\mathcal{Q}$ is at most

$$n\log_2(q+1) + \log_2\Lambda(\mathcal{Q}) - T(\log_2\Delta - O(1)) < \lambda(\mathcal{Q}) - \frac{2}{3}n\log_2\Delta,$$

for Δ sufficiently large and since $T = 2n$. So the total number of choices for a run $\mathcal{Q}$ of length T and with $\lambda(\mathcal{Q}) = \lambda$ is at most $2^{\lambda - \frac{2}{3}n\log_2\Delta} = 2^\lambda\Delta^{-2n/3}$. Thus the probability that we continue for $T = 2n$ steps is at most

$$\sum_{\lambda=1}^{T\Delta\log\Delta} 2^{-\lambda} \times 2^\lambda\Delta^{-2n/3} = 2n\Delta\log\Delta \times \Delta^{-2n/3} < \Delta^{-n/2}. \qquad \square$$

3.4. Proof of Theorem 1

As described above, the results of the preceding subsections provide a proof of Theorem 1:

Proof of Theorem 1. Consider any $\epsilon > 0$ and any assignment of lists of size $q = (1+\epsilon)\Delta/\ln\Delta$ colours to the vertices. We begin by assigning Blank to every vertex. Then we repeatedly call FIX to eliminate any remaining flaws. More formally: While there is any flaw f we call FIX (f, σ) where σ is the current partial colouring. By Lemma 9 each call terminates within $O(n)$ executions of (*) with probability at least $1 - \Delta^{-n/2}$. By Observation 6, the number of flaws decreases by at least one after each call. There are at most $2n$ initial flaws and so we obtain a flaw-free partial colouring σ^* after at most $2n$ calls of FIX (f, σ) with probability at least $1 - 2n\Delta^{-n/2} > 0$. Lemma 5 implies that the Blank vertices of σ^* can be recoloured to give the required proper list colouring. $\square$

Remark. This easily yields a polytime algorithm to produce the list colouring. Calling FIX at most $2n$ times w.h.p. produces σ^* in $O(n^2\Delta^2\ln\Delta)$ time; in fact, extending the definition of H_t, R_t, σ_t to cover the sequence of colourings/executions produced over the sequence of at most $2n$ calls of FIX can reduce this running time to $O(n\ln n\Delta^2\ln\Delta)$ (see e.g. the approach in [2]). The main result of [23] yields a polytime algorithm corresponding to Lemma 5, which we use to complete the colouring.

4. K_r-free graphs

With a more complicated recolouring step, the same proof can be adapted to K_r-free graphs. The setup is the same as in Section 3 except with a larger list size:

Each vertex v has a list of colours $\mathcal{C}_v$ that may be assigned to v of size

$$q := 200r\frac{\Delta \ln \ln \Delta}{\ln \Delta}.$$

A partial list colouring σ is an assignment to a subset of the vertices, where the colours are drawn from their lists. Given any partial colouring, L_v is defined to the set of colours in $\mathcal{C}_v$ not appearing on any neighbours of v along with Blank.

Because we are not trying for a good constant, we can afford to be a bit looser in our definition of L and our second flaw will be simpler than that in Section 3. We define

$$L = \Delta^{9/10}.$$

Given a partial colouring σ, we define the following two *flaws* for any vertex v:

$$B_v \equiv |L_v| < L$$
$$Z_v \equiv \text{ at least } L \text{ neighbours of } v \text{ are coloured } \mathsf{Blank}.$$

Observation 10. *B_v and Z_v are determined by the colours of the vertices in $N(v)$.*

It is trivial to see that any flaw-free partial colouring can be completed greedily to a full colouring of G, as the list of available colours for each vertex is greater than the number of uncoloured neighbours.

Again, we say v is the *vertex* of flaw $f = B_v$ or Z_v, and we denote $v(f) := v$. We use the same ordering on the flaws: Every B_v comes before every Z_u, and the B_v's and Z_u's are each ordered according to the labels of v, u.

We find a flaw-free partial colouring using essentially the same algorithm we used for triangle-free graphs, but we must be more careful about recolouring a neighbourhood. It will be useful to represent a partial colouring of a neighbourhood as a collection of disjoint independent sets.

We let $\mathcal{C} = \cup_{v\in G}\mathcal{C}_v$ denote the set of all colours that may appear in the graph, and define:

Definition 11. Given a vertex v and a fixed partial colouring of $V(G)\backslash N_v$, a *partial colour assignment* to N_v is a collection of disjoint independent sets $(\theta_1, ..., \theta_{|\mathcal{C}|})$, each a subset of N_v, such that for any $u \in \theta_i$ we have: $i \in \mathcal{C}_u$ and i does not appear on any neighbour of u outside of N_v.

It is possible that $\theta_i = \emptyset$, and we do not require that $\cup_{i=1}^{|\mathcal{C}|}\theta_i = N_v$. Any $u \in N_v$ that is not in any of the θ_i is considered to be coloured Blank.

To recolour N_v, we take a uniformly random partial colour assignment to N_v and then assign the colour i to every vertex in each θ_i. More specifically, given a colouring σ and a vertex v, we let Ω denote the set of all partial colour assignments to N_v and we choose a uniform member of Ω.

Note that if N_v contains no edges, then this recolouring is equivalent to giving each $u \in N_v$ a uniform colour from N_u, as we did in FIX.

We use the same flaw ordering as in Section 3; i.e. every B_v comes before every Z_u, and the B_v's and Z_u's are each ordered according to the labels of v, u.

The following procedure differs from FIX only in the distances: Observation 10 allows us to recurse on flaws Z_w within distance two rather than three. And we increase the distance for flaws B_w from two to three so that we get Observation 12 below, which will be very useful in our analysis.

FIX2(f, σ)
Set $v = v(f)$.
Choose a uniformly random partial colour assignment to N_v and then recolour N_v accordingly.
While there are any flaws B_w with $\mathrm{dist}(w, v) \leq 3$ or Z_w with $\mathrm{dist}(w, v) \leq 2$:
  Let g be the least such flaw and call **FIX**(g, σ') where σ' is the current colouring.
Return the current colouring.

Observation 12. *Whenever we call* FIX (Z_u, σ) *we have that* B_w *does not hold for any* $w \in N_u$.

This observation follows from our flaw ordering, and the fact that we call FIX on flaws B_w with w up to distance three from v rather than two.

The analog of Observation 6 holds again here, and so to prove Theorem 2 it suffices to prove that FIX2 terminates with positive probability.

We will assume throughout the remainder of this section that $\Delta \geq 2^{200r}$ as otherwise the bound of Theorem 2 is trivial.

4.1. More probability bounds

We begin with some key lemmas from Shearer's paper on the independence number of a K_r-free graph [30]. We rephrase the short proofs here for completeness and to extract a useful fact from them.

Given a graph H, we define:

$$I(H) \text{ is the number of independent sets of } H.$$

Lemma 13. *For any* $r \geq 2$, *if* H *is* K_r*-free then* $2^{|V(H)|} \geq I(H) \geq 2^{|V(H)|^{\frac{1}{r-1}}-1}$.

Proof. The upper bound is simply the number of subsets of $V(H)$. For the lower bound, we will prove that H has an independent set of size at least $|V(H)|^{1/r-1} - 1$; the bound follows by considering all subsets of that independent set.

We proceed by induction on r. The trivial base case is $r = 2$. For $r \geq 3$: If any vertex $u \in H$ has degree at least $d = |V(H)|^{\frac{r-2}{r-1}}$ then since the neighbourhood of u in

H is K_{r-1}-free, there is a sufficiently large independent set in that neighbourhood by induction. Otherwise, the maximum degree in H is less than d and so the straightforward greedy algorithm finds an independent set of size at least $|V(H)|/(d+1) > |V(H)|^{1/r-1} - 1$. □

Lemma 14. *If $H \neq \emptyset$ is K_r-free, $r \geq 4$, then half of the independent sets in H have size at least $\frac{1}{2r}\log_2 I(H)/\log_2\log_2 I(H)$.*

Proof. It suffices to show that at most $\frac{1}{2}I(H)$ subsets of $V(H)$ have size at most $\ell = \lfloor \frac{1}{2r}\log_2 I(H)/\log_2\log_2 I(H) \rfloor$; i.e.:

$$\sum_{i=0}^{\ell} \binom{|V(H)|}{i} \leq \frac{1}{2}I(H). \tag{6}$$

We can assume $\log_2 I(H) \geq 2$ as otherwise $\ell = 0$ and so the lemma is trivial (since $H \neq \emptyset$). We can also assume $r \leq \log_2 I(H)/2\log_2\log_2 I(H)$ else $\ell = 0$. We set $x = \log_2 I(H) \geq 2$. Rearranging the second inequality of Lemma 13 gives $|V(H)| \leq (1 + \log_2 I(H))^{r-1}$ and so we substitute $h = (1 + \log_2 I(H))^{r-1} \geq 27$ for $|V(H)|$ in (6). So $h = (1+x)^{r-1} < \frac{1}{4}x^{2r}$ for $x \geq 2$. Also, a simple induction on ℓ confirms that $\sum_{i=0}^{\ell}\binom{h}{i} \leq \sum_{i=0}^{\ell-1}\binom{h}{i} + \frac{h^\ell}{\ell!} \leq 2h^\ell$ for $\ell \geq 0, h \geq 2$. So the LHS of (6) is at most

$$2h^\ell < \frac{1}{2}x^{2r\ell} \leq \frac{1}{2}2^{\log_2 x \times \frac{x}{\log_2 x}} = \frac{1}{2}2^x = \frac{1}{2}I(H).$$

This proves (6). □

Remarks. (1) Lemma 13 is the only place where we use the fact that our graph is K_r-free. Our proof shows that the bound of Theorem 2 holds whenever every subgraph $H \subseteq G$ satisfies the implication of either Lemma 13 or Lemma 14. In fact, it is enough for this to hold for every v and $H \subseteq N(v)$.

(2) Note that the argument in Lemma 14 can in fact show that the average size of the independent sets of H is at least $\frac{1}{2r}\log_2 I(H)/\log_2\log_2 I(H)$, which is Lemma 1 of [30].

(3) Alon [6] proves that if G is locally r-colourable, meaning that every neighbourhood can be r-coloured, then for any v and $H \subseteq N_v$, the median size of the independent sets of H is at least $\frac{1}{10\log_2(r+1)}\log_2 I(H)$. Plugging this bound into the rest of our proof yields that $\chi_\ell \leq O(\ln r \frac{\Delta}{\ln \Delta})$ for such graphs, as shown in [18].

We use these to bound the probabilities of our flaws.

Setup for Lemma 15: Each vertex $u \in N_v$ has a list L_u^* containing Blank and perhaps other colours; specifically, L_u^* is the set of colours of $\mathcal{C}_u$ not appearing on any neighbour of u outside of N_v along with Blank. We give the vertices of N_v a random partial colour assignment consistent with these lists. This assignment determines L_v – the set of colours in $\mathcal{C}_v$ that do not appear in the partial colour assignment.

Lemma 15.

(a) $\mathbf{Pr}(|L_v| < L) < \Delta^{-4}$.
(b) *The probability that at least L neighbours of v are coloured* Blank *and* $|L_u| > L$ *for all* $u \in N_v$ *is at most* Δ^{-4}.

Proof. We begin with a method for sampling a partial colour assignment.

Define Ω to be the set of all partial colour assignments to N_v, and let $W = (W_1, ..., W_{|\mathcal{C}|})$ be a uniform member of Ω. Define Q_1 to be the vertex set consisting of W_1 and all blank vertices which can be given the colour 1; i.e. all blank $u \in N_v$ with $1 \in L_u^*$. Select a uniformly random independent set W_1' of Q_1 and form W' by replacing W_1 with W_1'.

Claim 1. *W' is a uniform member of Ω.*

Proof of Claim 1. For any $|\mathcal{C}| - 1$ disjoint independent sets $S_2, ..., S_{|\mathcal{C}|} \subseteq N_v$ we define $\Omega_{S_2,...,S_{|\mathcal{C}|}} \subseteq \Omega$ to be the set of partial colour assignments $(\theta_1, ..., \theta_{|\mathcal{C}|})$ with $\theta_2 = S_2, ..., \theta_{|\mathcal{C}|} = S_{|\mathcal{C}|}$; so this yields a partition of Ω. Note that W' is a uniform member of $\Omega_{W_2,...,W_{|\mathcal{C}|}}$. Furthermore, because W is a uniform member of Ω, the part $\Omega_{W_2,...,W_{|\mathcal{C}|}}$ is selected with the correct distribution, i.e. with probability $|\Omega_{W_2,...,W_{|\mathcal{C}|}}|/|\Omega|$. So W' is a uniform member of Ω.

Repeating this argument, we can resample $W_2, ..., W_{|\mathcal{C}|}$ in the same manner. Specifically:

Let $W = (W_1, ..., W_{|\mathcal{C}|})$ be a uniform member of Ω.
For $i = 1$ to $|\mathcal{C}|$
  Define Q_i to be the subgraph induced by W_i and all vertices that are blank at this step and can be given the colour i.
  Let W_i' be a uniform independent set of Q_i
  Modify W by replacing W_i with W_i'.

To be clear: the blank vertices in the definition of Q_i are blank in the current partial colour assignment $W = (W_1', ..., W_{i-1}', W_i, ..., W_{|\mathcal{C}|})$. By repeating the argument from Claim 1, we see that the partial colour assignment produced by this procedure is a uniform member of Ω.

Part (a): Let A_1 be the set of colours $i \in \mathcal{C}_v$ such that $I(Q_i) \leq \Delta^{1/20}$, and set $A_2 := \mathcal{C}_v \backslash A_1$. Since the subgraph induced by N_v is K_{r-1}-free, Lemma 14 implies that for each $i \in A_2$ the median independent set of Q_i has size at least $\frac{1}{2(r-1)} \log_2 I(Q_i)/\log_2 \log_2 I(Q_i) > \frac{1}{40r} \log_2 \Delta/\log_2 \log_2 \Delta$. (When applying Lemma 14 note that if $Q_i = \emptyset$ then $i \in A_1$.)

At iteration i: If colour $i \in A_1$ then the probability that we choose $W_i' = \emptyset$ is $\frac{1}{I(Q_i)} \geq \Delta^{-1/20}$. Note that if $W_i' = \emptyset$ then i will be in L_v. If $i \in A_2$, then with probability at least $\frac{1}{2}$ we choose a W_i' with $|W_i'| \geq \frac{1}{40r} \log_2 \Delta/\log_2 \log_2 \Delta$. Since the total size of the sets W_i' is at most Δ, this cannot happen for more than $40r\frac{\Delta \log_2 \log_2 \Delta}{\log_2 \Delta}$ colours.

We consider two random binary strings, each of length $|\mathcal{C}_v|$. In the first, each bit is 1 with probability $\Delta^{-1/20}$, and 0 otherwise. In the second, the bits are uniform. By coupling the choice of W_i' with these bits, we ensure that: (a) for each $i \in A_1$, if the corresponding bit in the first stream is 1 then $W_i' = \emptyset$; (b) for each $i \in A_2$, if the corresponding bit in the second stream is 1 then $|W_i'| \geq \frac{1}{40r} \log_2 \Delta / \log_2 \log_2 \Delta$. For example, in iteration i if we have $I(Q_i) < \Delta^{1/20}$ and so $i \in A_1$ then we look at the next bit of the first string. If that bit is 1 then we set $W_i' = \emptyset$; otherwise we set $W_i' = \emptyset$ with probability $\frac{1}{I(Q_i)} - \Delta^{-1/20}$. Similarly when $i \in A_2$.

Set $\ell = \frac{1}{2}|\mathcal{C}_v| = 100r\Delta \log_2 \log_2 \Delta / \log_2 \Delta$, and so we must have either $A_1 \geq \ell$ or $|A_2| \geq \ell$.

Claim 2. *If the outcomes of this procedure yield $|L_v| < L$ then at least one of these two events must hold:*

- $E_1 =$ *at most L of the first ℓ bits of the first string are 1*
- $E_2 =$ *at most $40r\frac{\Delta \log_2 \log_2 \Delta}{\log_2 \Delta}$ of the first ℓ bits of the second stream are 1*

Proof. If $W_i' = \emptyset$ then $i \in L_v$. So $\overline{E_1}$ and the event $|A_1| \geq \ell$ imply that at least L colours in A_1 are in L_v. $\overline{E_2}$ and the event $|A_2| \geq \ell$ imply that for more than $40r\frac{\Delta \log_2 \log_2 \Delta}{\log_2 \Delta}$ colours $i \in A_2$ we have $|W_i'| \geq \frac{1}{40r} \log_2 \Delta / \log_2 \log_2 \Delta$, which contradicts the fact that the sets W_i' are disjoint and have total size at most $|N_v| \leq \Delta$. Since we must have either $|A_1| \geq \ell$ or $|A_2| \geq \ell$ then if $|L_v| < L$ we must have $E_1 \vee E_2$.

Claim 2 implies $\mathbf{Pr}(|L_v| < L) \leq \mathbf{Pr}(E_1) + \mathbf{Pr}(E_2)$. Note that the expected number of 1's in the first ℓ bits of the first string is $\ell \times \Delta^{-1/20} \gg L = \Delta^{9/10}$ and the expected number of 1's in the first ℓ bits of the second string is $\frac{1}{2}\ell = 50r\Delta \log \log \Delta / \log \Delta$. So the Chernoff Bounds (or Lemma 3) imply that each of E_1, E_2 occur with probability less than $\frac{1}{2}\Delta^{-4}$ for $r \geq 4$ and $\Delta \geq 2^{500r}$. This proves part (a).

Part (b): Consider any L neighbours $u_1, ..., u_L \in N_v$. We will prove the probability that each u_i is coloured blank and satisfies $|L_{u_i}| > L$ is at most $1/L!$. This proves part (b) as $\binom{\Delta}{L}/L! < \Delta^{-4}$ for $\Delta \geq 100$.

Fix a colouring of $V(G) \backslash N_v$ and let $\Omega_B \subset \Omega$ be the set of partial colour assignments in which every u_i is coloured Blank and satisfies $|L_{u_i}| > L$. (Note: a partial colour assignment in Ω_B may also have additional blank vertices.) Take any $W \in \Omega_B$ and extend it to a partial colour assignment W_2 in which each of $u_1, ..., u_t$ are not blank as follows:

begin with the colouring W
for $i = 1$ to L
    give u_i a colour from L_u^* which does not appear on any of its neighbours in N_v.

This yields a colouring W' of N_v which can be viewed as the partial colour assignment $(\theta_1, ..., \theta_{|\mathcal{C}|})$ where θ_j is the set of vertices with colour j in W'.

By definition of Ω_B, each u_i has at least L available colours in W. By the time we reach iteration i, at most $i-1$ of those colours have been assigned to a neighbour of u_i in $\{u_1, ..., u_{i-1}\}$. So there are always at least $L-i+1$ choices for a colour to assign to u_i and so the number of choices for W' is at least $L!$. Each partial colour assignment W' can arise from at most one $W \in \Omega_B$, namely the W obtained from W' by colouring $u_1, ..., u_L$ all Blank. So $|\Omega_B| \leq |\Omega|/L!$, which is what we need to establish part (b). □

4.2. FIX2 terminates

Now the same argument from Section 3.3 implies that FIX2 terminates with positive probability, and thus proves Theorem 2.

Each time we call FIX2 (v, σ) we let $\mathcal{L} = \{L_u^* : u \in N_v\}$ be the lists of available colours on the neighbours of v in the colouring obtained from σ by uncolouring N_v; i.e. L_u^* is the set of colours in $\mathcal{C}_u$ that do not appear on any neighbours of u outside of N_v, along with Blank. We let $\Omega(\mathcal{L})$ be the set of partial colour assignments to N_v consistent with $\mathcal{L}$. We let $\mathcal{B}(\mathcal{L}) \subset \Omega(\mathcal{L})$ be the set of partial colour assignments that have the flaw B_v. We let $\mathcal{Z}(\mathcal{L}) \subset \Omega(\mathcal{L})$ be the set of partial colour assignments which have the flaw Z_v.

We define H_t, R_t analogously to Section 3.3. At each step: If we are addressing the flaw B_v then Lemma 15(a) implies that the number of choices for the colouring of N_v before the recolour line is at most $|\mathcal{B}(\mathcal{L})| \leq \Delta^{-4}|\Omega(\mathcal{L})|$. If we are addressing the flaw Z_v then by Observation 12, each $u \in N_v$ has at least L available colours in σ and so must have $|L_u| \geq L$ before uncolouring N_v; thus $|L_u^*| \geq |L_u| \geq L$. So Lemma 15(b) implies that the number of choices for the colouring of N_v before the recolour line is at most $|\mathcal{Z}(\mathcal{L})| \leq \Delta^{-4}|\Omega(\mathcal{L})|$. This yields that the size of what is written to H_t is $3\log_2 \Delta + \log_2 |\Omega(\mathcal{L})| - 4\log_2 \Delta + O(1)$ whereas the number of random bits used is $\log_2 |\Omega(\mathcal{L})|$. This is enough for the analysis from Section 3.3, in particular the proof of Lemma 9 to carry through.

Remark. This time it is not clear how to obtain a polytime algorithm; the challenge is to select a uniform partial colour assignment efficiently. Johansson's proof yields a polytime algorithm (see [8]).

5. Lopsided local lemma

Bernshteyn notes that the proofs of Theorems 1 and 2 could have been carried out using the Lopsided Local Lemma rather than an entropy compression argument. One considers taking a uniformly random partial colouring of the entire graph. The bad events are: B_v and $Z_v \wedge \overline{B_v}$. By conditioning on the colours of all vertices at distance at least two or three from v, Lemmas 7 and 15 imply that the probability of the bad events is sufficiently small, even when conditioning on the outcomes of distant events. See [10] for more details and for an extension of these results to DP-colouring.

Acknowledgments

My thanks to Dimitris Achlioptas and Fotis Iliopoulos for some very helpful discussions. I am also grateful to two anonymous referees for many helpful comments and to Zdenek Dvorak for pointing out a problem with Lemma 3 in an earlier version. This research is supported by an NSERC Discovery grant.

References

[1] D. Achlioptas, A. Coja-Oghlan, Algorithmic barriers from phase transitions, in: Proceedings of FOCS, 2008, pp. 793–802. Longer version available at arXiv:0803.2122.
[2] D. Achlioptas, F. Iliopoulos, Random walks that find perfect objects and the Lovasz Local Lemma, J. ACM 63 (3) (2016) 22, https://doi.org/10.1145/2818352. Preliminary version in Proc. of FOCS.
[3] D. Achlioptas, F. Iliopoulos, Focused local search and the Lovasz Local Lemma, in: Proc. of SODA, 2016.
[4] D. Achlioptas, C. Moore, Random k-SAT: two moments suffice to cross a sharp threshold, SIAM J. Comput. 36 (2006) 740–762.
[5] M. Ajtai, P. Erdős, J. Komlós, E. Szemerédi, On Turan's theorem for sparse graphs, Combinatorica 1 (1981) 313–317.
[6] N. Alon, Independence numbers of locally sparse graphs and a ramsey type problem, Random Structures Algorithms 9 (1996) 271–278.
[7] N. Alon, M. Krivelevich, B. Sudakov, Coloring graphs with sparse neighborhoods, J. Combin. Theory Ser. B 77 (1999) 73–82.
[8] N. Bansal, A. Gupta, G. Guruganesh, On the Lovász Theta function for independent sets in sparse graphs, in: Proceedings of STOC, 2015.
[9] B. Bartlomiej, S. Czerwiński, J. Grytczuk, P. Rzażewski, Harmonious coloring of uniform hypergraphs, Appl. Anal. Discrete Math. 10 (2016) 73–87.
[10] A. Bernshteyn, The Johansson–Molloy theorem for DP-coloring, arXiv:1708.03843.
[11] V. Dujmović, G. Joret, J. Kozik, D.R. Wood, Nonrepetitive colouring via entropy compression, Combinatorica 36 (2016) 661–686.
[12] P. Erdős, L. Lovász, Problems and results on 3-chromatic hypergraphs and some related questions, in: A. Hajnal, et al. (Eds.), Infinite and Finite Sets, in: Colloq. Math. Soc. János Bolyai, vol. 11, North-Holland, Amsterdam, 1975, pp. 609–627.
[13] L. Esperet, A. Parreau, Acyclic edge-coloring using entropy compression, European J. Combin. 34 (2013) 1019–1027.
[14] L. Fortnow, A Kolmogorov complexity proof of the Lovász Local Lemma. Blog post. http://blog.computationalcomplexity.org/2009/06/kolmogorov-complexity-proof-of-lov.html.
[15] A. Gagol, G. Joret, J. Kozik, P. Micek, Pathwidth and nonrepetitive list coloring, Electron. J. Combin. 23 (4) (2016) 4.40.
[16] J. Grytczuk, J. Kozik, P. Micek, A new approach to nonrepetitive sequences, Random Structures Algorithms 42 (2013) 214–225.
[17] A. Johansson, Asymptotic choice number for triangle free graphs, Unpublished manuscript, 1996.
[18] A. Johansson, The choice number of sparse graphs, Unpublished manuscript, 1996.
[19] J.H. Kim, On Brooks' Theorem for sparse graphs, Combin. Probab. Comput. 4 (1995) 97–132.
[20] C. McDiarmid, Concentration, in: M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, B. Reed (Eds.), Probabilistic Methods for Algorithmic Discrete Mathematics, Springer, 1998, pp. 195–248.
[21] M. Molloy, B. Reed, Graph Colouring and the Probabilistic Method, Springer, 2002.
[22] R. Moser, A constructive proof of the Lovász Local Lemma, in: Proceedings of the 41st ACM Symposium on Theory of Computing, 2009.
[23] R. Moser, G. Tardos, A constructive proof of the general Lovász Local Lemma, J. ACM 57 (2) (2010).
[24] A. Panconesi, A. Srinivasan, Randomized distributed edge coloring via an extension of the Chernoff–Hoeffding Bounds, SIAM J. Comput. 26 (1997) 350–368.
[25] S. Pettie, H. Su, Distributed coloring algorithms for triangle-free graphs, Inform. and Comput. 243 (2015) 263–280.

[26] J. Przybyło, On the facial Thue choice index via entropy compression, J. Graph Theory 77 (2014) 180–189.
[27] J. Przybyło, J. Schreyer, E. Škrabuľáková, On the facial Thue choice number of plane graphs via entropy compression method, Graphs Combin. 32 (2016) 1137–1153.
[28] B. Reed, The list colouring constants, J. Graph Theory 31 (1999) 149–153.
[29] J. Shearer, A note on the independence number of triangle-free graphs, Discrete Math. 46 (1983) 83–87.
[30] J. Shearer, On the independence number of sparse graphs, Random Structures Algorithms 7 (1995) 269–271.
[31] T. Tao, Moser's entropy compression argument, Blog post. https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/.
[32] L. Zdeborová, F. Krzakala, Phase transitions in the colouring of random graphs, Phys. Rev. E 76 (2007) 031131.