Entropy compression method applied to graph colorings

D. Gonçalves, M. Montassier, and A. Pinlou
LIRMM - Univ. Montpellier 2, CNRS

May 28, 2014

Abstract

We propose a framework based on the entropy compression method, inspired by the one of Esperet and Parreau [L. Esperet and A. Parreau, Acyclic edge-coloring using entropy compression, *European J. Combin.*, 34(6):1019–1027, 2013], to prove upper bounds for some chromatic numbers. From this method, in particular, we derive that every graph with maximum degree Δ has an acyclic vertex-coloring using at most $\frac{3}{2}\Delta^{\frac{4}{3}} + O(\Delta)$ colors, and a non-repetitive vertex-coloring using at most $\Delta^2 + 1.89\Delta^{\frac{5}{3}} + O(\Delta^{\frac{4}{3}})$ colors.

1 Introduction

In the 70's Lovász introduced the celebrated *Lovász Local Lemma* (LLL for short) to prove results on 3-chromatic hypergraphs [6]. It is a powerful probabilistic method to prove the existence of combinatorial objects satisfying a set of constraints. Since then, this lemma has been used in many occasions. In particular, it is a very efficient tool in graph coloring to provide upper bounds on several chromatic numbers [1, 3, 7, 9, 10, 11, 12, 13]. Recently Moser and Tardos [14] designed an algorithmic version of LLL by means of the so-called *Entropy Compression Method* (ECM for short). This method seems to be applicable whenever LLL is, with the benefits of providing tighter bounds. For example, the ECM has been used in graph coloring to get bounds on non-repetitive coloring [4] that improve previous results using LLL (see e.g. [1]) and on acyclic-edge coloring [5]. In this latter paper, Esperet and Parreau provide a framework applicable to many graph colorings. Inspired by this work, we provide a slightly different framework and give new tools to improve the analysis.

This extended abstract is organized as follows. In Section 2, we apply the ECM to acyclic vertex coloring. This serves as an introductory example of our framework, which is then roughly described in Section 3. We also provide tools for its analysis and apply it to non-repetitive coloring.

2 Acyclic coloring of graphs

A *proper coloring* of a graph is an assignment of colors to the vertices of the graph such that two adjacent vertices do not use the same color. An *acyclic coloring* of a graph G is a proper coloring of G such that G contains no bicolored cycles; in other words, the graph induced by every two color classes is a forest. Let $\chi_a(G)$, called the *acyclic chromatic number*, be the smallest integer k such that the graph G admits an acyclic k-coloring.

We consider the family $\mathcal{F}_\gamma$ of graphs having no copy of $K_{2,\gamma+1}$ (the complete bipartite graph with partite sets of size 2 and $\gamma+1$) in which the two vertices in the first class are non-adjacent. Alon, McDiarmid, and Reed [2] considered graphs $G \in \mathcal{F}_\gamma$ and using LLL, they proved that these graphs satisfy $\chi_a(G) \leq \lceil 32\sqrt{\gamma}\Delta \rceil$ (Δ denotes the maximum degree). We improve this bound by a constant factor:

Theorem 1 *For some $\gamma \geq 1$, let $G \in \mathcal{F}_\gamma$ with maximum degree Δ. We have $\chi_a(G) < 1 + \Delta\left(1 + \sqrt{2\gamma+4}\right)$.*

We prove Theorem 1 by contradiction. Suppose there exists $G \in \mathcal{F}_\gamma$ such that $\chi_a(G) \geq 1 + \Delta(1 + \sqrt{2\gamma+4})$. Let κ be the unique integer such that $\Delta(1 + \sqrt{2\gamma+4}) \leq \kappa < 1 + \Delta(1 + \sqrt{2\gamma+4})$. We define an algorithm that "tries" to acyclically color G with κ colors; by hypothesis, it should fail. Define a total order $\prec$ on the vertices of G.

2.1 The algorithm

Let $V = \{1, 2, \ldots, \kappa\}^t$ be a vector of length t, for some arbitrarily large $t \gg n = |V(G)|$. Algorithm AcyclicColoringGamma_G takes the vector V as input and returns a partial acyclic coloring $\varphi : V(G) \to \{\bullet, 1, 2, \ldots, \kappa\}$ of G ($\bullet$ means that the vertex is uncolored) and a text file R that is called a *record*. The acyclic coloring φ is necessarily partial since we try to color G with a number of colors less than its acyclic chromatic number. For a given vertex v of G, we denote by $N(v)$ the set of neighbors of v.

Algorithm 1: ACYCLICCOLORINGGAMMA_G

Input : V (vector of length t).
Output: (φ, R).

1 **for** $i \leftarrow 1$ **to** t **do**
2   Let v be the smallest (w.r.t $\prec$) uncolored vertex of G
3     $\varphi(v) \leftarrow V[i]$
4   Write "`Color \n`" in R
5   **if** $\varphi(v) = \varphi(u)$ for $u \in N(v)$ **then**
6         $\varphi(v) \leftarrow \bullet$
7   Write "`Uncolor, neighbor` u `\n`" in R
8   **else if** *v belongs to a bicolored cycle of length $2k$ ($k \geq 2$), say ($v = u_1, \ldots, u_{2k}$)* **then**
9   **for** $j \leftarrow 1$ **to** $2k - 2$ **do**
10             $\varphi(u_j) \leftarrow \bullet$
11   Write "`Uncolor, cycle` $(v = u_1, \ldots, u_{2k})$ `\n`" in R
12 **return** (φ, R)

Algorithm ACYCLICCOLORINGGAMMA_G runs as follows. Let φ_i be the partial coloring of G after i steps (at the end of the i^{th} loop). At Step i, we first consider φ_{i-1} and we color the smallest uncolored vertex v with $V[i]$ (lines 2 and 3 of Algorithm 1). We then verify whether one of the two following events happens: (1) the graph G contains a monochromatic edge vu for some u (line 5 of Algorithm 1) or (2) the graph G contains a bicolored cycle of length $2k$ ($v = u_1, u_2, \ldots, u_{2k}$) (line 8 of Algorithm 1). If such events happen, then we uncolor some vertices (including v) in order that none of the two previous events remains. Clearly, φ_i is a partial acyclic coloring of G.

Proof of Theorem 1. Let us first note that the function defined by Algorithm ACYCLICCOLORINGGAMMA_G is injective. This comes from the fact that from each output of the algorithm, one can determine the corresponding input (by Lemma 2). Now we obtain a contradiction by showing that the number of possible outputs is strictly smaller than the number of possible inputs when t is chosen large enough compared to n. The number of possible inputs is exactly κ^t while the number of possible outputs is $o(\kappa^t)$, as it is at most $(1+\kappa)^n \times o(\kappa^t)$. Indeed, there are at most $(1+\kappa)^n$ possible partial κ-colorings of G and there are at most $o(\kappa^t)$ possible records (by Lemma 3). This concludes the proof of Theorem 1. □

2.2 Algorithm analysis

Recall that φ_i denotes the partial acyclic coloring obtained after i steps. Let us denote by $\overline{\varphi}_i \subset V(G)$ the set of vertices that are colored in φ_i. Let also v_i, R_i and V_i respectively denote the current vertex v of the i^{th} step, the record R after i steps, and the input vector V restricted to its i first elements. Observe that as φ_i is a partial acyclic κ-coloring of G, and as G is not acyclically κ-colorable, we have that $\overline{\varphi}_i \subsetneq V(G)$, and thus v_{i+1} is well defined. This also implies that R has t "`Color`" lines.

Lemma 2 *One can recover V_i from (φ_i, R_i).*

We omit here its proof due to lack of space. By Lemma 2, Algorithm ACYCLICCOLORINGGAMMA_G defines an injective mapping. Let us now bound the number of possible records.

Lemma 3 *Algorithm* ACYCLICCOLORINGGAMMA_G *produces at most $o(\kappa^t)$ distinct records R.*

Sketch of proof. Since Algorithm ACYCLICCOLORINGGAMMA_G fails to color G, the record R has exactly t "`Color`" lines. It contains also "`Uncolor`" lines of two types: "`neighbor`" and "`cycle`". Let t_1 be the number of "`Uncolor, neighbor`" lines, and let t_k be the number of "`Uncolor, cycle`" lines, where the cycle has length $2k$ ($2 \leq k \leq \lfloor n/2 \rfloor$). Observe now that for every "`Uncolor, neighbor`" step (resp. "`Uncolor, cycle`" step), the algorithm uncolors 1 (resp. $2k-2$) previously colored vertex. It follows that $t_1 + \sum_{2 \leq k \leq \lfloor n/2 \rfloor}(2k-2)t_k \leq t$. By the previous equation, let us define the non-negative integer $t_0 = t - \sum_{1 \leq k \leq \lfloor n/2 \rfloor} t_k$. Let us bound the number $\#Seq(t_1, t_2, \ldots t_{\lfloor n/2 \rfloor})$ of possible sequences of "`Color`" | "`Uncolor, neighbor`" | "`Uncolor, cycle`" lines in the record, for fixed $t_1, t_2, \ldots, t_{\lfloor n/2 \rfloor}$:

$$\#Seq(t_1, t_2, \ldots, t_{\lfloor n/2 \rfloor}) \leq \binom{t}{t_0} \times \binom{t - t_0}{t_1} \times \binom{t - t_0 - t_1}{t_2} \times \cdots \times \binom{t - \sum_{0 \leq i < \lfloor n/2 \rfloor} t_i}{t_{\lfloor n/2 \rfloor}} = \binom{t}{t_0, t_1, t_2, \ldots, t_{\lfloor n/2 \rfloor}}$$

To compute the total number of possible records, let us compute how many different records a given "`Uncolor`" step can produce. Observe that an "`Uncolor, neighbor`" line can be of Δ different types according to the neighbor of v that shares the same color, and an "`Uncolor, cycle`" line involving a cycle of length $2k$ can be of at most $\frac{1}{2}\gamma\Delta^{2k-2}$ types (the number of types is equal to the number of $2k$-cycles going through v; this number is given by Lemma 3.2 in [2]). Consequently, the number of different records for fixed $t, t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor}$ is bounded by the following function B_t:

$$B_t(t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor}) \quad = \quad \binom{t}{t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor}} \times \Delta^{t_1} \times \prod_{2 \le k \le n/2} \left(\frac{1}{2}\gamma\Delta^{2k-2} \right)^{t_k}$$

Summing over all possible tuples $(t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor})$, the number of different records $\#Rec$ is bounded by:

$$\#Rec \le \sum_{(t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor})} B_t(t_0, t_1, \ldots, t_{\lfloor \frac{n}{2} \rfloor})$$

After some calculations, we can prove that, for a sufficiently large t and any real x such that $0 < x \le 1$,

$$\#Rec < t(t+1)^{\lfloor \frac{n}{2} \rfloor} \frac{1}{x^t} \left(1 + \Delta x + \frac{1}{2}\gamma \sum_{2 \le i \le \lfloor \frac{n}{2} \rfloor} (\Delta x)^{2i-2} \right)^t$$

Setting $x = \dfrac{1}{\Delta\sqrt{\frac{\gamma}{2}+1}}$, we obtain that $\frac{1}{x}\left(1 + \Delta x + \frac{1}{2}\gamma \sum_{2 \le i \le \lfloor \frac{n}{2} \rfloor} (\Delta x)^{2i-2}\right) < \left(\Delta\left(1 + \sqrt{2\gamma+4}\right)\right) \le \kappa$. Finally, this gives $\#Rec = o(\kappa^t)$. This completes the proof of Lemma 3. □

3 Our framework and applications

Let us now describe the framework. Given an arbitrarily chosen graph G, our method proves the existence of a particular coloring of G using κ colors, for some κ. The proof is done by contradiction: we consider that G needs more that κ colors.

Algorithm 2: COLORING_G

Input : $V = \{1, 2, \ldots, \kappa\}^t$ (vector of length t).
Output: (φ, R).

```
for i ← 1 to t do
    v ← NextUncolorVertex(φ̄)
    φ(v) ← V[i]
    Write "Color \n" in R
    for j ← 1 to p do
        if BadEvent_j(v, φ) then
            k ← BadEventClass_j(v, φ)
            for ∀u ∈ UncolorSetBadEvent_j(v, φ̄, k) do
                φ(u) ← •
            Write "Uncolor, Bad Event j, k \n" in R
return (φ, R)
```

Following the same scheme as in Section 2, we define an algorithm COLORING_G. To do so, we first define a set of forbidden partial colorings of G and call them bad events. We partition this set into p types (for example, we had 2 types of bad events in Section 2). Moreover, each type of bad events is divided into classes. Let C_j denote the number of type-j bad event classes. The algorithm COLORING_G needs the following functions:

- `NextUncolorVertex` outputs an uncolored vertex of G.
- $\texttt{BadEvent}_j$ tests whether the current partial coloring is a bad event of type j.
- Given a type-j bad event, $\texttt{BadEventClass}_j$ outputs the class of this bad event.
- Given a type-j bad event and its class, $\texttt{UncolorSetBadEvent}_j$ outputs a s_j-subset of $V(G)$ whose uncoloring leads to a valid partial coloring, for some $s_j \ge 1$.
- If at some step i of the main loop in COLORING_G a type-j bad event occurs, given the class of this bad event and the partial coloring φ_i obtained at the end of the step, then $\texttt{RecoverBadEvent}_j$ outputs the partial coloring φ_{i-1} of G.

The latter function ensures the injectivity of COLORING_G. Now let κ be the smallest integer such that $\kappa > \inf_{0<x\leq 1} Q(x)$, where

$$Q(x) = \frac{1}{x}\left(1 + \sum_{1\leq j\leq p} C_j x^{s_j}\right)$$

One can prove that Algorithm COLORING_G produces at most $o(\kappa^t)$ distinct outputs. This is less than the κ^t possible inputs, and thus contradicts the injectivity of COLORING_G.

Let us now apply this framework to non-repetitive coloring, that is a vertex coloring with no path of even length so that the sequence of colors of the first half equals the sequence of colors of the second half. In that case, let the bad events of type j be the colorings with a badly colored $2j$-path, $j \geq 1$. A given current vertex v may belong to at most $j\Delta^{2j-1}$ paths of length $2j$. Each type of bad event is partitioned into $C_j = j\Delta^{2j-1}$ classes. One can define the above-mentioned functions so that $s_j = j$, for all $j \geq 1$. As we have

$$Q\left(\frac{1}{\Delta^2} - \left(\frac{2}{\Delta^7}\right)^{\frac{1}{3}}\right) \leq \Delta^2 + 1.89\Delta^{\frac{5}{3}} + O(\Delta^{\frac{4}{3}})$$

this implies that G admits a non-repetitive coloring with $\Delta^2 + 1.89\Delta^{\frac{5}{3}} + O(\Delta^{\frac{4}{3}})$ colors. This slightly improves on previously known bounds. The framework also implies the following new results:

- Any graph G with maximum degree Δ has acyclic chromatic number at most $\frac{3}{2}\Delta^{\frac{4}{3}} + O(\Delta)$.
- Any graph G with maximum degree Δ has generalized r-acyclic chromatic number at most $\Delta^\ell + O(\Delta^{\frac{2}{3}(l+1)})$, where $\ell = \lfloor r/2 \rfloor$.
- Any graph G with maximum degree Δ has Thue choice index at most $\Delta^2 + 2^{\frac{4}{3}}\Delta^{\frac{5}{3}} + O(\Delta^{\frac{4}{3}})$.
- Any plane graph G with maximum degree $\Delta \geq 3$ has facial Thue choice number at most $\Delta + 4\sqrt{\Delta} + 4$.
- Any plane graph G has facial Thue choice index at most 10.

References

[1] N. Alon, J. Grytczuk, M. Hałuszczak, and O. Riordan. Nonrepetitive colorings of graphs. *Random Structures & Algorithms* 21(3-4):336-346, 2002.

[2] N. Alon, C. McDiarmid, and B. Reed. Acyclic coloring of graphs. *Random structures and algorithms*, 2(3):277–288, 1991.

[3] N. Alon, B. Sudakov, and A. Zaks. Acyclic edge colorings of graphs. *Journal of Graph Theory*, 37(3):157–167, 2001.

[4] V. Dujmović, G. Joret, J. Kozik, and D.R. Wood. Nonrepetitive colouring via entropy compression. *Combinatorica* to appear, 2014.

[5] L. Esperet and A. Parreau. Acyclic edge-coloring using entropy compression. *European J. Combin.*, 34(6):1019–1027, 2013.

[6] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal, R. Rado, and V. T. Sós, eds. *Infinite and Finite Sets (to Paul Erdős on his 60th birthday) II*. North-Holland. pp. 609–627.

[7] G. Fertin, A. Raspaud, and B. Reed. Star coloring of graphs. *Journal of Graph Theory*, 47(3):163–182, 2004.

[8] A. Fiedorowicz. Acyclic 6-colouring of graphs with maximum degree 5 and small maximum average degree. *Discussiones Mathematicae Graph Theory*, 33(1):91–99, 2013.

[9] C. Greenhill and O. Pikhurko. Bounds on the generalised acyclic chromatic numbers of bounded degree graphs. *Graphs and Combinatorics*, 21(4):407–419, 2005.

[10] H. Hatami. $\Delta + 300$ is a bound on the adjacent vertex distinguishing edge chromatic number. *J. Combin. Theory Ser. B*, 95(2):246–256, 2005.

[11] F. Havet, J. van den Heuvel, C. McDiarmid, and B. Reed. List colouring squares of planar graphs. Research Report RR-6586, INRIA, July 2008.

[12] M. Molloy and B. Reed. A bound on the strong chromatic index of a graph. *J. Combin. Theory Ser. B*, 69(2):103–109, 1997.

[13] M. Molloy and B. Reed. A bound on the total chromatic number. *Combinatorica*, 18(2):241–280, 1998.

[14] R. A. Moser G. Tardos. A constructive proof of the general lovasz local lemma. *Journal of ACM*, 57(2):1-15, 2010.