On the Algorithmic Lovász Local Lemma and Acyclic Edge Coloring[*]

Ioannis Giotis[†] Lefteris Kirousis[‡] Kostas I. Psaromiligkos[§] Dimitrios M. Thilikos[¶]

Abstract

The algorithm for Lovász Local Lemma by Moser and Tardos gives a constructive way to prove the existence of combinatorial objects that satisfy a system of constraints. We present an alternative probabilistic analysis of the algorithm that does not involve reconstructing the history of the algorithm from the witness tree. We apply our technique to improve the best known upper bound to acyclic chromatic index. Specifically we show that a graph with maximum degree Δ has an acyclic proper edge coloring with at most $\lceil 3.74(\Delta - 1)\rceil + 1$ colors, whereas the previously known best bound was $4(\Delta - 1)$. The same technique is also applied to improve corresponding bounds for graphs with bounded girth. An interesting aspect of this application is that the probability of the "undesirable" events do not have a uniform upper bound, i.e. it constitutes a case of the *asymmetric* Lovász Local Lemma.

1 Introduction

The Lovász Local Lemma (LLL) first appeared in 1975 in a paper by Erdős and Lovász [6].

> "The proof is so elementary that it could, and I think it should, be taught in a first course in probability. It has had and continues to have a profound effect on probabilistic method."

J. Spencer wrote about it in the book on his Durango lectures [17]. This elegant proof however was non-constructive. Up until the seminal work by Moser [13] and Moser and Tardos [14], and despite important work on the question of supplying a constructive proof by Beck [4], Alon [2], Srinivasan [19] and others, the question and its answer succinctly given in the book on the Durango lectures [17]: "Algorithm? Sometimes!" remained valid.

The constructive proof of Moser [13] and Moser and Tardos [14] essentially showed how to algorithmically produce a satisfying assignment to a Constraint Satisfaction Problem (CSP), given that the variables are independent. Their randomized algorithm roughly did nothing more than locating unsatisfied constraints and resampling their variables. It continues doing so until no constraint is left unsatisfied. They proved that the expected time of steps for this algorithm to halt is linear in the number of variables, given that the degree of the dependency graph of the constraints does not exceed a certain constant fraction of the inverse of the probability of a constraint not to be satisfied. Their proof depended on supplying a witness structure which encodes the history of the random choices of the algorithm. The average size of this structure cannot be less than the entropy of the sequence of random choices made by the algorithm. See [20] for a presentation of this information theoretic argument known as "*entropy compression argument*". Another elegant presentation of this approach is given by Spencer in [18].

In this work, we first analyze Moser's algorithm directly, using probabilistic arguments to estimate its number of steps. The essence of our approach is the observation (based on the Principle of Deferred Decisions of Knuth [12]) that each resampling renders the current assignment with the same as the original distribution, conditional on the previous event being satisfied. Thus we manage to express the probability that at least n steps are needed until the algorithm halts by a simple recurrence relation. This recurrence relation is asymptotically analyzed by classical tools of the Analysis of Algorithms. It turns out that given that the degree of the dependency graph does not exceed a certain constant fraction of the inverse probability of an undesirable event to occur, the probability that

[*]Research co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: ARISTéA II.

[†]Department of Mathematics, National Kapodistrian University of Athens, Greece and Computer Technology Institute Press "Diophantus", Patras, Greece, igiotis@cs.upc.edu

[‡]Department of Mathematics, National Kapodistrian University of Athens, Greece and Computer Technology Institute Press "Diophantus", Patras, Greece, lkirousis@math.uoa.gr

[§]Department of Mathematics, National Kapodistrian University of Athens, Greece, kostaspsa@gmail.com

[¶]Department of Mathematics, National Kapodistrian University of Athens, Greece, Computer Technology Institute Press "Diophantus", Patras, Greece, and AlGCo project-team, CNRS, LIRMM, France, sedthilk@thilikos.info

the algorithm lasts for n steps is exponentially small in n, after a cutoff point (a result made known to us by Achlioptas and Iliopoulos [1], but through a proof based on the entropic method). We believe that this direct and completely elementary probabilistic proof avoids some of the intricacies of the entropy compression argument. It also completely unveils, we think, the extreme elegancy of Moser's algorithm (we adopted the original version of the algorithm in [13]).

We used our alternative approach to a particular coloring problem, namely the acyclic edge coloring. The entropy compression method had been used by Esperet and Parreau [7] to find bounds for the acyclic chromatic index, the least number of colors needed to properly color the edges of a graph so that no bichromatic cycle exists. Our approach leads to a direct and simple probabilistic analysis that yields the upper bound of $\lceil 3.74(\Delta - 1)\rceil + 1$ for the acyclic chromatic index, improving the bound of $4(\Delta - 1)$ given by Esperet and Parreau. We also improve their bounds for graphs with bounded girth.

2 Algorithmic Lovász Local Lemma

Let $X_i, i = 1, \dots, l$ be mutually independent random variables on a common probability space, taking values in the sets $D_i, i = 1, \dots, l$, respectively.

Let $E_j, j = 1, \dots, m$ be a sequence of events, each depending on a sequence of the random variables X_i. The sequence of variables that an event E_j depends on is called the *scope* of E_j and is denoted by e^j. The events E_j are considered "undesirable", i.e., the objective is to design a randomized algorithm that will return an assignment α to the variables X_i for which none of the events E_j hold.

We say that two events overlap, and write $E_i \sim E_j$, if $e^i \cap e^j \neq \emptyset$. The binary, reflexive and symmetric binary relation $\sim$ defines a graph with all vertices looped (but no multiple edges) called the *dependency graph* of the events.

For $j = 1, \dots, m$, let N_j be the neighborhood of the event E_j in the dependency graph, i.e. $N_j = \{E_i \mid E_i \sim E_j\}$ (observe that $E_j \in N_j$).

Let Δ be the maximum of the cardinalities $|N_j|$ (i.e. the max degree of the dependency graph counting the loop as contributing 1 to the degree) and let p be the max of the probabilities $\Pr[E_j]$.

THEOREM 2.1. (LOVÁSZ LOCAL LEMMA) *If $ep\Delta \leq 1$, then $Pr[\overline{E_1} \wedge \overline{E_2} \wedge \cdots \wedge \overline{E_m}] > 0$, i.e. there exists an assignment to the variables X_i for which none of the events E_i hold.*

The original proof of Theorem 2.1, first presented essentially in this form in [16], was non-constructive, but was given for arbitrary events, i.e. without the assumption that the events depended on independent random variables. Below, we will give an algorithmic proof Theorem 2.1 within the framework already described, i.e. assuming the dependence of the events on independent variables.

```
ALGORITHM
1: Sample the variables $X_i$ (independently) and
   let $\alpha$ be the resulting assignment of values to
   them.
2: while there exist an event that occurs under the
   current assignment, let $E_i$ be the least indexed
   such event do
3:     RESAMPLE($E_i$)
4: end while
5: Output current assignment $\alpha$

RESAMPLE($E_i$)
1: Resample the variables in the scope $e^i$ (inde-
   pendently).
2: while some $E_j \in N_i$ occurs under the current
   assignment $\alpha$, let $E_j$ be the least indexed such
   event do
3:     RESAMPLE($E_j$)
4: end while
```

Figure 1: ***Randomized sampling algorithm***

We first present our algorithm in Figure 1, which is a direct derivation of the one given by Moser in [13][1].

On a particular execution of our algorithm, let us call a *phase* the execution period within a *root* call of RESAMPLE, more specifically the period spent in an execution of RESAMPLE from line 3 of ALGORITHM. For clarity, we will refer to calls of RESAMPLE from within another RESAMPLE execution as *recursive* calls.

Our goal is to upper bound the probability that ALGORITHM makes at least n RESAMPLE calls. We first show that the number of phases in any execution is bounded. Then, we argue that the probability of an event occurring at a given step of the algorithm can still be bounded by p conditional on the various resamplings performed so far. These will allow us to bound the total

[1]The algorithm in [13] was presented and analyzed only for the SAT problem and an alternative algorithm for a collection of arbitrary events determined by independent variables was subsequently presented and analyzed by Moser and Tardos in [14]; the generalization of the original algorithm in [13] for arbitrary events is straightforward, however the analysis in [13] does not immediately generalize.

number of steps by a function of p.

LEMMA 2.1. *Consider an arbitrary call of* RESAMPLE(E_i). *Let $\mathcal{E}$ be the set of events that do not occur at the beginning of this call. Then, if the call terminates, events in $\mathcal{E}$ will also not occur at the end of the call.*

Proof. Suppose some event E_j in $\mathcal{E}$ occurs during the call. But this means that some variable in its scope changed value. This implies that E_j is in the neighborhood of some other event E'_j and a call RESAMPLE(E'_j) was made, potentially a recursive call within our original call RESAMPLE(E_i). But because of the "while" loop of line 2 of RESAMPLE, the call RESAMPLE(E'_j) will not terminate until all the events in the neighborhood $N_{j'}$ of E'_j, and therefore E_j itself, do not occur. Assuming RESAMPLE(E_i) terminates, RESAMPLE(E'_j) must have also terminated and therefore at the end of RESAMPLE(E'_j), E_j does not occur. The same argument can be reapplied every time some event in $\mathcal{E}$ occurs, or re-occurs, during the call. ∎

By Lemma 2.1, we know that the set of events that do not occur at the start of a phase cannot be smaller at the end of the phase. Moreover, it will strictly increase because the event for which the root call RESAMPLE occured also has to not occur at the end of the call. Therefore, a root call of RESAMPLE can only occur at most once for each event.

COROLLARY 2.1. *There are at most m phases in any execution of* ALGORITHM.

Let us now examine the probability distribution of the variables after a resampling, caused by a call of RESAMPLE(E_i).

LEMMA 2.2. (RANDOMNESS LEMMA) *Let α be a random assignment sampled independently from the probability distribution of the variables $X_1, \dots, X_l$ and E_i an event. Let α' be the assignment obtained from α by resampling (independently) the variables in e^i if E_i occurs under α, and let α' be α otherwise. Then, conditional that E_i occurs under α, the distribution of α' is the distribution of assignments sampled independently from all variables i.e. it is the same as the distribution of α. Therefore the probability that any event E occurs under α' is equal to the probability that E occurs under α.*

Proof. This immediately follows from the principle of deferred decisions. One only needs to consider the values of the variables in e^i after the resampling and since these are sampled from the same distribution, α' can be seen as sampled from the same distribution as α. Notice that without the conditional that E_i occurs under α, the lemma is not, in general, correct as then the resampling does not necessarily take place. ∎

DEFINITION 2.1. *A sequence of events $\mathcal{E}_1, \dots, \mathcal{E}_k$ is called a* witness sequence *if the first k* RESAMPLE *calls (recursive or root) of* ALGORITHM *are applied to $\mathcal{E}_1, \dots, \mathcal{E}_k$, respectively.*

Notice that being a witness sequence is a random property, as whether a sequence of events $\mathcal{E}_1, \dots, \mathcal{E}_k$ is a witness sequence or not depends on the initial random sampling of the variables $X_1, \dots, X_l$ and the subsequent resamplings performed by the algorithm. In particular, if $\alpha_1, \dots, \alpha_k$ is the sequence of the first k assignments that the algorithm outputs and $\mathcal{E}_1, \dots, \mathcal{E}_k$ is the corresponding witness sequence, then besides α_1 being a sampling of $X_1, \dots, X_l$ and α_{i+1} being obtained by resampling the variables in e^i, $i = 1, \dots, k-1$, we necessarily (but not sufficiently) have that $\mathcal{E}_i$ occurs under $\alpha_i, i = 1, \dots, k$. Notice however, that the events in a witness sequence in addition to occurring under the respective assignments, must additionally satisfy conditions that correspond to characteristics of ALGORITHM like e.g. that each time the least indexed event that occurs is chosen and that the while loop in a call RESAMPLE$(\mathcal{E})$ lasts until no neighbor of $\mathcal{E}$ occurs under the current assignment.

Now let $\hat{P}_n$ be the probability that ALGORITHM performs at least n RESAMPLE calls. Obviously,

$$\hat{P}_n = \Pr\left[\begin{array}{c}\text{there is some } \textit{witness} \\ \text{sequence of length } \ n\end{array}\right]. \tag{2.1}$$

To bound the probability in (2.1), we will relax the definition of a witness sequence. Towards giving a weaker definition, we start by describing in the next paragraph how the events in a witness sequence are structured. All trees to be considered next (and throughout the paper) are ordered, i.e. the children of any node are ordered, and labeled, i.e. their nodes have labels from a given set. Also forests are comprised of (ordered) sequences of ordered labeled trees.

Consider first the ordering of events in a witness sequence that appear within a root call RESAMPLE$(\mathcal{E})$, i.e. within a phase. Each such event, apart from the root event $\mathcal{E}$, is a neighbor, in the dependency graph, of the event of the corresponding previous call. Therefore, by following the order that the events appear in the recursion stack of RESAMPLE$(\mathcal{E})$, we can construct a labeled rooted tree of out-degree at most Δ whose root is labeled with $\mathcal{E}$ whereas its non-root nodes are labeled with the events that appear on the recursion stack on top of $\mathcal{E}$ (possibly $\mathcal{E}$ again). The preorder of the labels of this tree is the order they appear on the stack. Similarly, a witness sequence corresponds to the preorder of the labels of an ordered forest of ordered rooted trees, each of out-degree at most Δ. Notice that by Lemma 2.1, the indices of the successive root labels are strictly increasing and the same is true for the labels of the successive children of a node.

We now define:

DEFINITION 2.2. *A sequence of events $\mathcal{E}_1, \dots, \mathcal{E}_k$ is called a* valid sequence *if*

- *there is a rooted forest with at most m trees labeled with the events in the sequence so that the order of the events in the sequence coincides with the preorder of the labels of the forest (the same label may appear more than once), and*
- *the label of a non-root node v in the forest is a neighbor of the label of the parent of v, and*
- *the indices of the labels of the successive children of any node are strictly increasing (hence the out-degree of any node is at most Δ); the same is true for the indices of the successive root labels of the forest (hence there are at most m roots), and*
- *$\mathcal{E}_i$ occurs under the assignment $\alpha_i, i = 1, \dots, k$, where α_1 is obtained by sampling $X_1, \dots, X_l$ and $\alpha_{i+1}, i = 1, \dots, k-1$ is obtained by resampling the variables in e^i.*

Being a valid sequence is also a random property, however the property of a sequence being a witness sequence is stronger than the property of it being a valid sequence. Intuitively the property of being valid is more general than the property of being witness in the sense that (a) the label of a child of a node v is not the least indexed event that occurs under the current assignment; it is any event that holds under the current assignment, as long as its index is larger than the index of the preceding child, (b) the same is true for the root labels and (c) it is not necessary that all neighbors of the label of a node v do not occur under the assignment that immediately follows the assignment corresponding to the label of the last child of v. However, the (forest-) tree-like structure that characterizes witness sequences has been retained in the definition of a valid sequence, because this tree-like structure is necessary to express by a recurrence relation in n the probability that a valid sequence of at least n terms exists. Also exactly as in witness sequences, in a valid sequence, the indices of the successive root labels are strictly increasing; the same is true for the indices of the successive children of a node.

If we now define:

$$P_n = \Pr\left[\begin{array}{c}\text{there is some } \textit{valid} \\ \text{sequence of length } n\end{array}\right]. \tag{2.2}$$

Assuming $P_0 = 1$ and using (2.1) and (2.2), we have

$$\hat{P}_n \le P_n. \tag{2.3}$$

Below, we will bound P_n. We first notice that by repeated applications of Lemma 2.2, any assignment $\alpha_{i+1}, i = 1, \dots, k-1$ in the sequence $\alpha_1, \dots, \alpha_k$ of assignments that correspond to a valid sequence $\mathcal{E}_1, \dots, \mathcal{E}_k$ is distributed randomly in the sense that follows the distribution of α_1, i.e. the distribution obtained by sampling the variables $X_1, \dots, X_l$, independently.

A *valid phase-sequence* is defined as a valid sequence with the restriction that the corresponding forest is a tree. Let $Q_n, n \ge 1$ be the maximum probability over all events $\mathcal{E}$ that a valid phase-sequence whose first event is $\mathcal{E}$ and has length n exists. Let also also $Q_0 = 1$. Now since the root labels in a valid sequence are strictly increasing, since the events have cardinality m and since we have assumed that $Q_0 = 1$ we immediately have that:

$$P_n = \sum_{\substack{n_1+\cdots+n_m=n \\ n_1,\dots,n_m \ge 0}} Q_{n_1} \cdots Q_{n_m}. \tag{2.4}$$

Also,

$$Q_n \le p \sum_{\substack{n_1+\cdots+n_\Delta=n-1 \\ n_1,\dots,n_\Delta \ge 0}} Q_{n_1} \cdots Q_{n_\Delta}. \tag{2.5}$$

Above, the factor p appears because in order for a root recursive call to execute at least one step, the event of the root call should occur under the corresponding assignment (because of the condition in the while loop of step 2 of ALGORITHM).

LEMMA 2.3. (PHASE BOUND) *Q_n is asymptotically bounded from above by*

$$\sqrt{1+\frac{1}{\Delta-1}}\left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^n < \sqrt{1+\frac{1}{\Delta-1}}\,(ep\Delta)^n.$$

Proof. Let Q_n^* be a sequence of numbers such that $Q_0^* = 1$ and for $n \ge 1$,

$$Q_n^* = p \sum_{\substack{n_1+\cdots+n_\Delta=n-1 \\ n_1,\dots,n_\Delta \ge 0}} Q_{n_1}^* \cdots Q_{n_\Delta}^*, \tag{2.6}$$

i.e. Q_n^* satisfies the recurrence obtained by putting the equality sign in (2.5). Obviously, $Q_n \le Q_n^*$. Let $Q(z)$

be the OGF of Q^*_n. Multiply both its sides of (2.6) with z^n and sum for $n \geq 1$ to get (since $Q^*_0 = 1$):

$$Q(z) - 1 = zpQ(z)^\Delta \tag{2.7}$$

Let now $W = W(z) = Q(z) - 1$. Then by (2.7), we have $W = zp(W+1)^\Delta$.

Let now $\phi(W) = p(W+1)^\Delta$. Then Equation (2.7) is equivalent to

$$W(z) = z\phi(W(z)). \tag{2.8}$$

Now apply Lagrange's Inversion Formula (see e.g. [9, Theorem A.2]) to get that for $n \geq 1$:

$$\begin{aligned} Q^*_n &= [z^n]W = (1/n)[u^{n-1}](\phi(u))^n = (1/n)p^n \binom{\Delta n}{n-1} \\ &= \frac{1}{(\Delta-1)n+1} p^n \binom{\Delta n}{n} \end{aligned}$$

Therefore by Stirling approximation there is a constant $c' > 0$ (depending on Δ) such that:

$$\begin{aligned} Q^*_n <& e^{c'/n} \frac{1}{(\Delta-1)n+1} \frac{1}{\sqrt{2\pi n}} \\ & \sqrt{1+\frac{1}{\Delta-1}} \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^n \\ <& e^{c'/n} \sqrt{1+\frac{1}{\Delta-1}} \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^n \\ <& e^{c'/n} \sqrt{1+\frac{1}{\Delta-1}} (ep\Delta)^n . \end{aligned}$$

■

Lemma 2.4. (Algorithm bound) *There is a constant $A > 1$ (depending on Δ) such that P_n is bounded from above by*

$$(An)^m \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^n < (An)^m (ep\Delta)^n .$$

Proof. By (2.4) and Lemma 2.3 we have that for some $A > 1$ (depending on Δ), P_n is bounded from above by

$$\begin{aligned} & \sum_{\substack{n_1+\cdots+n_m=n \\ n_1,\ldots,n_m \geq 0}} A \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^{n_1} \cdots \\ & \cdots A \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^{n_m} \\ & \leq (An)^m \left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right)^n . \end{aligned}$$

■

Now to have that P_n becomes exponentially small in n when $\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta < 1$, we must have that $m\log n + m\log A + n\log\left(\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta\right) < 0$.

Therefore by inequalities (2.1) and (2.3) we have:

Theorem 2.2. *Assuming p and Δ are constants such that $\left(1+\frac{1}{\Delta-1}\right)^{\Delta-1} p\Delta < 1$ (and therefore if $ep\Delta \leq 1$), there exists an integer N, which depends linearly on m, and a constant $c \in (0,1)$ (depending on p and Δ) such that if $n/\log n \geq N$ then the probability that* Algorithm *executes more than n calls of* Resample *is $< c^n$.*

The integer N in the above theorem is referred to as a *cut-off* point because it marks the onset of subexponential probability for the number of "steps" (calls of Resample) that Algorithm takes before it stops. Clearly, when the algorithm stops we have found an assignment such that none of the events occurs. And since this happens with probability close to 1 for large enough n, Theorem 2.1 easily follows. ■

3 The acyclic chromatic index

3.1 Preliminaries Let $G = (V, E)$ be (simple) graph with n vertices and m edges. The chromatic index of G is the least number of colors needed to *properly* color its edges, i.e. to color them so that no adjacent edges get the same color. If Δ is the maximum degree of G, it is known that its chromatic index is either Δ or $\Delta + 1$ (Vizing [21]).

A cycle of G of length s is a sequence $v_i, i = 0, \ldots, s-1$ of distinct vertices so that $\forall i = 0, \ldots s-1$, v_i and $v_{i+1 \pmod s}$ are connected by an edge. The least number of colors needed to properly color the edges of G so that no cycle is *bichromatic*, i.e. so that there is no cycle whose edges are colored with only two colors, is called the *acyclic* chromatic index of G. Notice that in any properly colored graph, any cycle of odd length is necessarily at least *trichromatic*, i.e. its edges have three or more colors. It has been conjectured (J. Fiamčik [8] and Alon et al. [3]) that the acyclic chromatic index of any graph with maximum degree Δ is at most $\Delta + 2$. A number of successively tighter upper bounds to the acyclic chromatic index has been provided in the literature. Most recently, Esperet and Parreau [7] proved that the acyclic chromatic index is at most $4(\Delta - 1)$. Their proof makes use of the technique of Moser and Tardos [14] that constructively proves the Lóvasz Local Lemma. Actually, they describe a randomized algorithm and they show that it produces an acyclic edge coloring with positive probability. An approach using the entropy compression method was also used for the vertex analogue of the edge chromatic number by Gonçalves et al. [11]

In this section, we modify the Esperet and Parreau [7] technique in two important aspects: (a) instead of the Moser and Tardos [14] algorithm, we use its antecedent version by Moser [13] and (b) we analyze it by using the approach described in the previous section, which avoids the counting of witness-trees and depends only on probability estimates. Also, our technique does not refer to partial colorings. Thus we get a direct probabilistic analysis that yields the upper bound of $\lceil 3.74(\Delta-1)\rceil+1$ for the acyclic chromatic index, improving over $4(\Delta-1)$ in [7]. Generalizing to graphs with bounded girth, we also get improved numerical results some specific values of which are sampled in Figure 2.

Girth	Number of colors	Previously known [7]
-	$3.731(\Delta-1)+1$	$4(\Delta-1)$
7	$3.326(\Delta-1)+1$	$3.737(\Delta-1)$
53	$2.494(\Delta-1)+1$	$3.135(\Delta-1)$
219	$2.323(\Delta-1)+1$	$3.043(\Delta-1)$

Figure 2: Our results

An interesting aspect of this application is that the probability of the "undesirable" events do not have a uniform upper bound, i.e. it constitutes a case of the asymmetric LLL.

To be more specific, the algorithm that we will give in the next subsection obviously produces an acyclic-edge coloring *if it ever stops*. We analyze it, along the same lines of the previous section, to prove that the probability of the algorithm taking more than n steps is exponentially small in n.

To facilitate notation, we call a proper edge-coloring s-*acyclic* if it contains no bichromatic cycle of length s or less (s is an even natural). We call the corresponding graph parameter the s-*acyclic chromatic index*.

We start by mentioning the following fact, proved in Esperet and Parreau [7]:

Fact 3.1. $2(\Delta-1)+1=2\Delta-1$ *colors suffice to produce a 4-acyclic edge coloring of G.*

Proof. [Sketch] Successively, in any order, color the edges using, at each step, a color that does not destroy 4-acyclicity (hence, by definition, neither properness). To show that $2(\Delta-1)+1$ colors suffice, notice that for each edge e, one has to avoid the colors of all edges adjacent to e, and also for each pair of homochromatic (of the same color) edges e_1, e_2 adjacent to e at different endpoints, one has to avoid the color of the at most one edge e_3 that together with e, e_1, e_2 define a cycle of length 4. So at each step, one has to avoid $2(\Delta-1)$ colors. ■

Assume now that we have $K=\lceil(2+\gamma)(\Delta-1)\rceil+1$ colors, where γ is a nonnegative constant to be computed, so that the randomized Moser-type algorithm that we will give will halt with positive probability. Actually for the value of γ to be computed, the probability of it making more than n steps will be shown to be exponentially small in n.

We assume below that the edges are ordered according to a fixed ordering.

Notice that the number of cycles of length $2k$ that contain a given edge e is at most $(\Delta-1)^{2k-2}$. Also, we assume that cycles that contain a given edge e are ordered according to a fixed ordering.

Below we will first give some results that refer to the distribution of the colorings obtained by certain randomized processes we describe below:

Sample

- Successively color the edges of G (in their ordering), choosing, at each step, among the colors that do not destroy 4-acyclicity (hence, by definition, neither properness), one uniformly at random.

End Sample

Fact 3.2. *Let $e, e_1, \ldots, e_n$ be edges and $c, c_1, \ldots, c_n$ be colors. Assume $e \neq e_i, i=1,\ldots,n$. Then the probability that* Sample *assigns the color c to e conditional it assigns the colors $c_1,\ldots,c_n$ to $e_1,\ldots,e_n$, respectively, is at most $\frac{1}{\gamma(\Delta-1)+1}$.*

Proof. [Sketch] If the probability of assigning c to e, conditional $c_1,\ldots,c_n$ are assigned to $e_1,\ldots,e_n$, respectively, is positive, then c is among the colors that can be assigned to e when Sample assigns a color to e. The result follows since by Fact 3.1 we have at each step of Sample at least $\lceil\gamma(\Delta-1)+1\rceil$ colors to choose from. ■

Given now cycles $C_1,\ldots,C_k$ of G consider the following randomized process:

Sample & Cycle-Recolor

1. Successively color the edges of G (in their ordering), choosing, at each step, among the colors that do not destroy 4-acyclicity, one uniformly at random.
2. For $i=1,$ to k, if C_i is bichromatic, do
   - recolor the edges of C_i (in their ordering) so that at each step the color of the current edge of C_i is chosen uniformly at random between the colors that do not destroy 4-acyclicity.

End Sample & Cycle-Recolor

Lemma 3.1. (Randomness lemma) *Let $e, e_1, \ldots, e_n$ be edges and $c, c_1, \ldots, c_n$ be colors. Assume $e \neq e_i, i = 1, \ldots, n$. Then the probability that* Sample & Cycle-Recolor *assigns the color c to e, conditional it assigns the colors $c_1, \ldots, c_n$ to $e_1, \ldots, e_n$, respectively, and also conditional that the output assignment is produced by going through phase (2) and without exiting the for loop is at most $\frac{1}{\gamma(\Delta-1)+1}$.*

Proof. [Sketch] Similarly as in Fact 3.2. The going through phase (2) without exiting the for loop is required to guarantee that whenever we expose information that might enlarge the upper bound $\frac{1}{\gamma(\Delta-1)+1}$ on an edge's e probability of getting a specified color, then e is recolored, to retain this upper bound. ■

Given a sequence $C_1, \ldots, C_k$ of cycles, for the rest of this paper, and perhaps by abuse of the standard terminology, we define:

Definition 3.1. *A 4-acyclic coloring produced by* Sample & Cycle-Recolor *as applied to $C_1, \ldots, C_k$ by going through phase (2) without exiting the for loop is called* random.

A consequence of Lemma 3.1 is:

Lemma 3.2. *The probability for a random coloring that an edge e belongs in a cycle of length $2k$ is at most $\frac{1}{\gamma}\left(1 - e^{-\frac{1}{\gamma}}\right)^{2k-3}$.*

Proof. At first, we will prove that given an edge $e' = \{u, v\}$, the probability that some other edge stemming from v has a specific color (say white) is bounded by $1 - e^{-\frac{1}{\gamma}}$. Indeed, the probability that one edge $\neq e'$ stemming from v does *not* have a specified color is at least

$$1 - \frac{1}{\gamma(\Delta-1)+1}, \tag{3.9}$$

so the probability that none of them (at most $\Delta - 1$ in number) is at least

$$\left(1 - \frac{1}{\gamma(\Delta-1)+1}\right)^{\Delta-1}.$$

Now we use the inequality

$$1 - x > e^{-\frac{x}{1-x}}, \forall x < 1, x \neq 0$$

(see e.g. [5, inequality (4.5.7)]), for $x = \frac{1}{\gamma(\Delta-1)+1}$, which implies, after elementary operations, that the expression in (3.9) is at least $e^{-\frac{1}{\gamma}}$, so the probability that a white edge $\neq e'$ stemming from v does exist is at most $1 - e^{-\frac{1}{\gamma}}$.

Suppose now that e' is colored red. A bichromatic cycle of length $2k$ can start with $\Delta - 1$ ways, say with the color white, then for each of the $2k-3$ next steps there is probability at most $1 - e^{-\frac{1}{\gamma}}$ that the cycle will continue in the same pattern (red, white, red...), and finally for the last edge to be correctly colored the probabliity is at most

$$\frac{1}{\gamma(\Delta-1)+1} < \frac{1}{\gamma(\Delta-1)}$$

(the last edge, whose both endpoints are determined, must be white). So, overall the probability that a bichromatic cycle with $2k$ edges which contains e' exists, is at most

$$(\Delta-1)\left(1 - e^{-\frac{1}{\gamma}}\right)^{2k-3} \frac{1}{\gamma(\Delta-1)} = \frac{1}{\gamma}\left(1 - e^{-\frac{1}{\gamma}}\right)^{2k-3},$$

which completes the proof. ■

```
Coloring Algorithm
1: Color all edges following their ordering and
   choosing at each step a color uniformly at
   random among those that retain 4-acyclicity.
2: while some edge belongs to a bichromatic cycle
   let $e$ be the least such do
3:     Recolor($e$)
4: end while

Recolor($e$)
1: Let $C$ be the first bichromatic cycle that con-
   tains $e$. Recolor the edges of $C$ (following
   their ordering) choosing at each step uniformly
   at random a color among those that retain 4-
   acyclicity.
2: while some edge of $C$ belongs to a bichromatic
   cycle, let $e'$ be the least such do
3:     Recolor($e'$)
4: end while
```

Figure 3: ***Randomized coloring algorithm***

3.2 The algorithm

We consider the randomized algorithm in Figure 3, which is the algorithm we will analyze:

Obviously, if Coloring Algorithm ever stops it outputs an acyclic edge-coloring of G. In analogy to Lemma 2.1, we have:

Lemma 3.3. *Consider an arbitrary call of* Recolor(e). *Let $\mathcal{E}$ be the set of edges that do* *not belong to any bichromatic cycle at the beginning of this call. Then, if the call terminates, the edges in $\mathcal{E}$ will also not belong to a bichromatic cycle at the end of the call.*

Proof. Suppose some edge e' in $\mathcal{E}$ becomes part of a bichromatic cycle during the call. But this means that e' belonged to a cycle C that was recolored during a call Recolor(e'') which either coincides with or is a recursive call within our original call Recolor(e). But the call Recolor(e'') will not terminate until all edges in C do not belong to a bichromatic cycle. Assuming Recolor(e) terminates, Recolor(e'') must have also terminated. The same argument can be reapplied every time e' becomes part of bichromatic cycle during Recolor(e). ■

Let now a *phase* be the execution period within a *root* call of Recolor. Notice that at the end of each phase, the edge e to which the root call of Recolor that started the phase is applied does not belong to any bichromatic cycle. Therefore, using the previous lemma, we get:

Corollary 3.1. Coloring Algorithm *can have at most as many phases as the number of the edges of the graph.*

3.3 The recurrence relation For language convenience, and also to stress the analogy with the previous section, we will refer to an edge also as an event, which occurs when the edge belongs to a bichromatic cycle.

Given a random (in the sense of Definition 3.1) 4-acyclic coloring α and a cycle C which is bichromatic under α, the process of recoloring the edges of C by choosing at each step uniformly at random a color so that 4-acyclicity is retained is called a *random recoloring of the cycle C*. We thus obtain a random coloring. Define:

Definition 3.2. *A sequence of events $e_1, \dots, e_s$ is called a* witness sequence *if the first s* Recolor *calls (recursive or root) of* Coloring Algorithm *are applied to $e_1, \dots, e_s$, respectively.*

Now let $\hat{P}_n$ be the probability that Coloring Algorithm performs at least n Recolor calls. Obviously,

$$(3.10) \qquad \hat{P}_n = \Pr\left[\begin{array}{c}\text{there is some } witness \\ \text{sequence of length } n\end{array}\right].$$

Define:

Definition 3.3. *A sequence of events $e_1, \dots, e_s$ is called a* valid sequence *if*

- *there is a sequence of cycles $C_1, \dots, C_s$ of length $2k_1, \dots 2k_s$, respectively, and*
- *there is a rooted forest, labelled with events from the sequence, so that the order of the events in the sequence coincides with the preorder of the labels of the forest, and*
- *if the parent of a node w in the forest is v, and their respective labels are e_i and e_j then e_i belongs to C_j, and*
- *the labels of the successive children of any node are strictly increasing (in the ordering of edges), and therefore a node labelled with e_i can have at most $2k_i$ children, and*
- *the successive root labels of the forest are increasing (and therefore there are at most m trees), and*
- *the cycles $C_1, \dots, C_s$ are bichromatic for the colorings $\alpha_i, i = 1, \dots, s$, respectively, where α_1 is a random coloring (in the sense of Definition 3.1) and $\alpha_{i+1}, i = 1, \dots, s-1$ is obtained from α_i by recoloring C_i.*

Notice that the random property of being a valid sequence is weaker than the random property of being a witness sequence. Besides the differences pointed out for the respective definitions of the previous section, here we have that for an event e in a witness sequence, because it is an event where a call of Recolor is applied, its corresponding cycle C is the *first* cycle that contains e and is bichromatic under the current coloring. No such demand on being the *first* is made on the cycles of a valid sequence.

If we now define:

$$(3.11) \qquad P_n = \Pr\left[\begin{array}{c}\text{there is some } valid \\ \text{sequence of length } n\end{array}\right].$$

Assuming $P_0 = 1$ and from (3.10) and (3.11), we have

$$\hat{P}_n \leq P_n.$$

Below, we will bound P_n. We first notice that any coloring $\alpha_i, i = 1, \dots, s$ in the sequence $\alpha_1, \dots, \alpha_s$ of colorings that correspond to a valid sequence $e_1, \dots, e_s$ is random in the sense of Definition 3.1.

Define a *valid phase-sequence* to be a valid sequence with the restriction that the corresponding forest is a tree. Let $Q_n, n \geq 1$ be the maximum probability over all events e that a valid phase-sequence whose first event is e and has length n exists. Let also also $Q_0 = 1$. Now since the root labels in a valid sequence are strictly increasing, since the events have cardinality m and since we have assumed that $Q_0 = 1$ we immediately have that:

$$P_n \le \sum_{\substack{n_1+\dots+n_m=n \\ n_1,\dots,n_m\ge 0}} Q_{n_1}\cdots Q_{n_m}. \tag{3.12}$$

Also, because by Lemma 3.2 the probability of an edge e belonging to a bichromatic cycle of length $2k$ for a random coloring is $\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3}$, we have:

$$\begin{aligned} Q_n \le \sum_{k\ge 3} \Bigg[&\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3} \\ &\cdot \sum_{\substack{n_1+\dots+n_{2k}=n-1 \\ n_1,\dots,n_{2k}\ge 0}} Q_{n_1}\cdots Q_{n_{2k}} \Bigg]. \end{aligned} \tag{3.13}$$

3.4 Analytic asymptotics Consider the recurrence relation:

$$\begin{aligned} Q^*_n = \sum_{k\ge 3} \Bigg[&\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3} \\ &\cdot \sum_{\substack{n_1+\dots+n_{2k}=n-1 \\ n_1,\dots,n_{2k}\ge 0}} Q^*_{n_1}\cdots Q^*_{n_{2k}} \Bigg], \end{aligned} \tag{3.14}$$

with $Q^*_0 = 1$. Obviously $Q_n \le Q^*_n$. We will asymptotically analyze the coefficients of the OGF $Q(z)$ of Q^*_n. Towards this end, multiply both sides of the first equality in (3.14) with z^n and sum for $n = 1,\dots,\infty$ to get

$$Q(z) - 1 = \sum_{k\ge 3} \left[\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3} zQ(z)^{2k} \right], \tag{3.15}$$

with $Q(0) = 1$. Setting $W(z) = Q(z) - 1$ we get

$$W(z) = \sum_{k\ge 3} \left[\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3} z(W(z)+1)^{2k} \right], \tag{3.16}$$

with $W(0) = 0$. For notational convenience, set $W = W(z)$. Then from (3.16) we get:

$$\begin{aligned} W =& z\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{3}(W+1)^6 \\ &\cdot \frac{1}{1-\left(1-e^{-\frac{1}{\gamma}}\right)^{2}(W+1)^2}. \end{aligned} \tag{3.17}$$

Set now

$$\begin{aligned} \phi(x) =& \frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{3}(x+1)^6 \\ &\cdot \frac{1}{1-\left(1-e^{-\frac{1}{\gamma}}\right)^{2}(x+1)^2}, \end{aligned} \tag{3.18}$$

to get from (3.17):

$$W = z\phi(W). \tag{3.19}$$

By [9, Proposition IV.5] (it is trivial to check that the hypotheses in that Theorem are satisfied for $\gamma > 0$), we obtain that , if we set

$$\rho = \frac{\phi(\tau)}{\tau}, \tag{3.20}$$

where τ is the (necessarily unique) solution in the interval $(0, R)$, where $R = \frac{1}{\left(1-e^{-\frac{1}{\gamma}}\right)} - 1$ is the radius of convergence of the series representing ϕ at 0, of the characteristic equation (in τ):

$$\phi(\tau) - \tau\phi'(\tau) = 0, \tag{3.21}$$

then $[z^n]u \bowtie \rho^n$ (i.e. $\limsup\ ([z^n])^{1/n} = \rho$, see [9, IV.3.2]).

Now by a simple search (through Maple, for the code see [10]) we found that for $\gamma = 1.73095$, the unique positive solution of (3.21) in the radius of convergence is $\tau = 0.1747094762$, and this value of τ gives to ρ in (3.19) $\rho = 0.9999789027 < 1$. Therefore by making use of (3.12) and working as in the previous section, we get:

THEOREM 3.1. *Assuming Δ, the maximum degree of the graph G, is constant, and given the availability of at least $3.74(\Delta - 1) + 1$ colors, there exists an integer N, which depends linearly on m, the number of edges of G, and a constant $\rho \in (0, 1)$ such that if $n/\log n \ge N$ then the probability that* COLORING ALGORITHM *executes more than n calls of* RECOLOR *is $< \rho^n$.*

Now if the graph has girth $2l - 1$ for $l \ge 4$, the previous arguments carry over with minimal changes. Namely, equation (3.14) becomes:

$$\begin{aligned} Q^*_n = \sum_{k\ge l} \Bigg[&\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2k-3} \\ &\sum_{\substack{n_1+\dots+n_{2k}=n-1 \\ n_1,\dots,n_{2k}\ge 0}} Q^*_{n_1}\cdots Q^*_{n_{2k}} \Bigg], \quad Q^*_0 = 1. \end{aligned} \tag{3.22}$$

Also in (3.15) and (3.16), the starting point of the summation is changed from 3 to l. Moreover, equation (3.17) becomes:

$$\begin{aligned} W = z&\frac{1}{\gamma}\left(1-e^{-\frac{1}{\gamma}}\right)^{2l-3}(W+1)^{2l} \\ &\cdot \frac{1}{1-\left(1-e^{-\frac{1}{\gamma}}\right)^{2}(W+1)^2} \end{aligned} \tag{3.23}$$

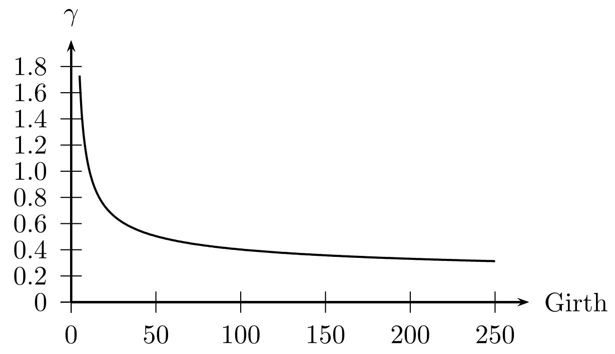

Figure 4: γ as a function of girth

and equation (3.18) becomes:

$$\phi(x) = \frac{1}{\gamma}\left(1 - e^{-\frac{1}{\gamma}}\right)^{2l-3}(x+1)^{2l} \cdot \frac{1}{1 - \left(1 - e^{-\frac{1}{\gamma}}\right)^2 (x+1)^2}. \tag{3.24}$$

Working as before, we get numerical results depicted in Figure 4 with sample specific values explicitly given in Figure 2.

4 Discussion

We plan to apply our technique to other problems, e.g. to the vertex coloring problem where results were very recently announced by Gonçalves et al. [11].

Acknowledgements

We are grateful to J. Rué for showing to us how to deal with the asymptotics of the coefficients of inverse generating functions. We are grateful to D. Mitsche on one hand and to D. Achlioptas and F. Iliopoulos, on the other, for pointing out errors in previous versions of this paper; the second and fourth authors are also indebted to the latter two for initiating them to this line of research. We are indebted to S. Messaris and Z. Terzopoulou, both undergraduate students, for their valuable comments during informal presentations of the results in this work.

References

[1] D. Achlioptas and F. Iliopoulos. Untitled notes. 2013. Unpublished private communication.

[2] Noga Alon. A parallel algorithmic version of the local lemma. *Random Structures & Algorithms*, 2(4):367–378, 1991.

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

[4] József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Structures & Algorithms*, 2(4):343–365, 1991.

[5] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/4.5.7, Release 1.0.9 of 2014-08-29. Online companion to [15].

[6] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 10:609–627, 1975.

[7] Louis Esperet and Aline Parreau. Acyclic edge-coloring using entropy compression. *Eur. J. Comb.*, 34(6):1019–1027, 2013.

[8] J. Fiamčik. The acyclic chromatic class of a graph (in Russian). *Math. Slovaca*, 28:139–145, 1978.

[9] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, New York, NY, USA, 1 edition, 2009.

[10] Ioannis Giotis, Lefteris Kirousis, Kostas I. Psaromiligkos, and Dimitrios M. Thilikos. Maple code. http://www.lsi.upc.edu/~igiotis/bichromatic_calc.zip.

[11] Daniel Gonçalves, Mickaël Montassier, and Alexandre Pinlou. Entropy compression method applied to graph colorings, 2014. arXiv:1406.4380.

[12] Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An introduction to the mathematical analysis of algorithms*, volume 10. American Mathematical Soc., 1997.

[13] Robin A Moser. A constructive proof of the Lovász Local Lemma. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 343–350. ACM, 2009.

[14] Robin A Moser and Gábor Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM (JACM)*, 57(2):11, 2010.

[15] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, 2010. Print companion to [5].

[16] Joel Spencer. Asymptotic lower bounds for Ramsey functions. *Discrete Mathematics*, 20(0):69 – 76, 1977.

[17] Joel Spencer. *Ten lectures on the probabilistic method*, volume 64. SIAM, 1994.

[18] Joel Spencer. Robin Moser makes Lovász Local Lemma algorithmic! 2010. http://cs.nyu.edu/spencer/moserlovasz1.pdf.

[19] Aravind Srinivasan. Improved algorithmic versions of the Lovász Local Lemma. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 611–620. Society for Industrial and Applied Mathematics, 2008.

[20] Terence Tao. Mosers entropy compression argument. 2009. http://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument.

[21] Vadim G Vizing. Critical graphs with a given chromatic class. *Diskret. Analiz*, 5(1):9–17, 1965.