AN IMPROVED BOUND OF ACYCLIC VERTEX-COLORING

LEFTERIS KIROUSIS AND JOHN LIVIERATOS

Abstract. The acyclic chromatic number of a graph is the least number of colors needed to properly color its vertices so that none of its cycles has only two colors. We show that for all \( \alpha > \frac{2}{3} \) there exists an integer \( \Delta_\alpha \) such that if the maximum degree \( \Delta \) of a graph is at least \( \Delta_\alpha \), then the acyclic chromatic number of the graph is at most \( \lceil \alpha \Delta^{4/3} \rceil + \Delta + 1 \). The previous best bound, due to Gonçalves et al (2020), was \((\frac{3}{2})\Delta^{4/3} + O(\Delta)\).

1. Introduction

The acyclic chromatic number of a graph \( G \), a notion introduced back in 1973 by Grünbaum [5] and denoted here by \( \chi_a(G) \), is the least number of colors needed to properly color the vertices of \( G \) — i.e. color them in a way that no adjacent vertices are homochromatic (of the same color) — so that no cycle of even length is bichromatic (has only two colors). Notice that in any properly colored graph, no cycle of odd length can be bichromatic.

The literature on the acyclic chromatic number for general graphs with unbounded maximum degree \( \Delta \) includes:

- Alon et al. [1] proved that \( \chi_a(G) \leq \lceil 50\Delta^{4/3} \rceil \). They remarked that the constant in this bound can be easily improved, but that they make no attempt to optimize constants. They also showed that there are graphs for which \( \chi_a(G) = \Omega\left( \frac{\Delta^{4/3}}{\log \Delta} \right) \).
- Ndreca et al. [9] proved that for \( \chi_a(G) \leq \lceil 6.59\Delta^{4/3} + 3.3\Delta \rceil \).
- Sereni and Volec [10] proved that \( \chi_a(G) \leq \frac{9}{5}\Delta^{4/3} + \Delta < 2.835\Delta^{4/3} + \Delta \).
- Finally, Gonçalves et al. [4] proved the best previous bound, namely that for \( \Delta \geq 24 \),
  \[
  \chi_a(G) \leq (3/2)\Delta^{4/3} + \min\left( 5\Delta - 14, \Delta + \frac{8\Delta^{4/3}}{\Delta^{2/3} - 4} + 1 \right)
  .
  \]

All the above results make use of some variant of the Lóvasz Local Lemma (LLL). The last two are based on the algorithmic proofs of LLL by Moser [7] and Moser and Tardos [8], which use an approach that has been known as the entropy compression method. The main difficulty in this approach is to prove the eventual halting of a randomized algorithm that successively and randomly assigns colors to the vertices, unassigning some colors when a violation of the desired properties arises.

1Department of Mathematics, National and Kapodistrian University of Athens
2Corresponding author
E-mail addresses: {lkirousis, jlivier89}@math.uoa.gr.
Towards proving the eventual halting (actually proving that the expected time of duration of the process is constant), a structure called witness forest is associated with the process, so that at every step, the history of the random choices made can be reconstructed from the current witness forest and the current coloring; the key observation is that the number of such forests (entropy) is not compatible, probabilistically, with the number of random choices made if the process lasted for too long. For nice expositions, see Tao [12] and Spencer [11]. It should be kept in mind that as the algorithm develops, dependencies are introduced between the colors of the vertices.

Very roughly and in qualitative terms, to get our result, namely that for every \( \alpha > 2^{-1/3} \), and for any graph \( G \), \( \chi_a(G) \leq \lceil \alpha \Delta^{4/3} \rceil + \Delta + 1 \), given that the maximum degree of \( G \) is large enough (how large depending on \( \alpha \) but not on \( G \)—see Theorem 1), we proceed as follows: we first prove that choosing a color for each vertex independently and uniformly at random from the palette gives, with positive probability, a proper coloring where the color of any vertex \( u \) is different from the color of all vertices \( v \) that are at distance two from \( u \) and have many common neighbors with \( u \); such vertices are said to form a special pair with \( u \)—formal definitions will be given in the next section. We call such colorings, specially proper.

Next we give a Moser-type algorithm that although may produce a coloring that is not specially proper (not even proper), it succeeds nevertheless, when it halts, in producing a coloring where for every vertex \( u \), the 4-cycles through \( u \) such that the opposing to \( u \) vertex does not form a special pair with \( u \) (these are the commoner 4-cycles), and also all cycles of even length at least six, do not have all their equal parity vertices homochromatic (equal parity vertices are the vertices separated by an odd number of vertices in a traversal of the cycle). We show that the probability this algorithm to last for at least \( n \) steps is exponentially small in \( n \). Towards proving this we do not use the entropy compression method. Instead we define a second algorithm which we call the validation algorithm that takes as input a witness structure and halts, always, outputting success or failure. The probability that the Moser-type algorithm lasts for at least \( n \) steps is bounded from above by the probability that the validation algorithm is successful on input at least one witness structure of size \( n \). It turns out that the latter probability, and therefore also the former, are exponentially small in \( n \). Furthermore, the validation algorithm always assigns to the vertices colors in an independent fashion. From independence it follows that with positive probability, the color obtained from the Moser-type algorithm is specially proper. We then repeat, as a Bernoulli trial, this whole process anew until we succeed to get from the Moser-type algorithm a coloring that is specially proper. We thus succeed in fully proving our result.

The approach to prove the algorithmic version of LLL by the introduction of a validation algorithm was first presented by Giotis et al. in [3]. An approach of temporarily ignoring properness was first used by Kirousis and Livieratos in [6] in order to provide an improved bound for edge-colorings. We should also note that in the present work, we strongly use graph-theoretic notions and results introduced by Gonçalves et al. [4].

2. Preliminaries, oriented paths and special pairs

2.1. Preliminaries. Let \( G = (V, E) \) be a (simple) graph with \( l \) vertices and \( m \) edges (both \( l \) and \( m \) are considered constants). The maximum degree of \( G \) is
denoted by $\Delta$ and we assume, to avoid trivialities, that it is $> 1$. A (simple) $k$-path is a succession $u_1, \ldots, u_k, u_{k+1}$ of $k+1 \geq 2$ distinct vertices any two consecutive of which are connected by an edge. A $k$-cycle is a succession of $k \geq 3$ distinct vertices $u_1, \ldots, u_k$ any two consecutive of which, as well as the the pair $u_1, u_k$, are connected by an edge. A path (respectively, cycle) is a $k$-path (respectively, $k$-cycle) for some $k$. Vertices of a cycle or a path separated by an odd number of other vertices are said to have equal parity.

A vertex-coloring (or simply, a coloring) of $G$ is an assignment of colors to its vertices selected from a given palette of colors. A coloring is proper if no neighboring vertices have the same color. A path or a cycle of a properly colored graph is called bichromatic if the vertices of the path or the cycle are colored by only two colors. A proper coloring is $k$-acyclic for some $k \geq 3$, if there are no bichromatic $l$-cycles, for any $l \geq k$. A proper coloring is called acyclic if there are no bichromatic cycles of any length. Note that for a cycle to be bichromatic in a proper coloring, its length must be even. The acyclic chromatic number of $G$, denoted by $\chi_a(G)$, is the least number of colors needed to produce a proper, acyclic coloring of $G$.

In the algorithms of this paper not necessarily proper colorings are constructed by independently selecting one color for each vertex from a palette of $K$ colors, for suitable values of $K$, uniformly at random (u.a.r.). Thus, for any vertex $v \in V$ and any color $i \in \{1, \ldots, K\},$

\begin{equation}
\Pr[v \text{ receives color } i] = \frac{1}{K}.
\end{equation}

We will prove:

**Theorem 1.** For every $\alpha > 0$ there exists an integer $\Delta_\alpha$ such that if the maximum degree $\Delta$ of a graph is at least $\Delta_\alpha$, a palette with $\lceil \alpha \Delta^{4/3} \rceil + \Delta + 1$ colors suffices to properly and acyclically color the vertices of the graph.

The rest of the paper is devoted to proving the above theorem.

In all that follows, we assume the existence of some arbitrary (total, strict) ordering among all vertices, paths and cycles of the given graph to be denoted by $\prec$. Also, given a cycle, among the two possible consecutive traversals of its vertices we choose one and name it positive.

2.2. Special pairs. We define the notion of special pairs originally introduced by Alon et al. [1]. Gonçalves et al. [4] generalized this notion and proved about it results of which we make strong use. The reason that the notion of special pairs is useful for us is on one hand that 4-cycles through a given vertex that forms a special pair with its opposing vertex can be handled with respect to bichromaticity directly and on the other that 4-cycles through a given vertex that does not form a special pair with its opposing vertex, although commoner, their number (see Lemma 1) allows them to be handled with respect to bichromaticity with a Moser-type algorithm.

Below, we follow the notation and terminology of Gonçalves et al. slightly adjusted to our needs. We give in detail the relevant definitions and proofs.

Given a vertex $u$, let $N(u)$ and $N^2(u)$, respectively, denote the set of vertices at distance one and two, respectively, from $u$. Among the vertices in $N^2(u)$ define a strict total order $\prec_u$ as follows: $v_1 \prec_u v_2$ if either $|N(u) \cap N(v_1)| < |N(u) \cap N(v_2)|$ or alternatively $|N(u) \cap N(v_1)| = |N(u) \cap N(v_2)|$ and $v_1$ precedes $v_2$ in the ordering $\prec$ between vertices we assumed to exist.
Definition 1 (Gonçalves et al. [4]). A pair \((u, v)\) of vertices such that \(v \in N^2(u)\) is called an \(\alpha\)-special pair if it belongs to the at most \([\alpha \Delta^{4/3}]\) highest, in the sense of \(< u\), elements of \(N^2(u)\). The set of vertices \(v\) for which \((u, v)\) form a special pair is denoted by \(S_\alpha(u)\). Also \(N^2(u) \setminus S_\alpha(u)\) is denoted by \(\overline{S_\alpha(u)}\).

It is possible that \(v \in S_\alpha(u)\) but \(u \in \overline{S_\alpha(v)}\). Also by definition,

\[
|S_\alpha(u)| = \min([\alpha \Delta^{4/3}], |N^2(u)|).
\]

We now give the proof of the following, that is essentially the proof presented by Gonçalves et al. [4].

Lemma 1 (Gonçalves et al. [4, Claim 11]). For all vertices \(u\), there are at most \(\frac{\Delta^{8/3}}{8\alpha}\) 4-cycles that contain \(u\) but contain no vertex \(v \in S_\alpha(u)\).

Proof. Let \(d\) be an integer such that

\[
\begin{align*}
&\text{if } v \in S_\alpha(u) \text{ then } |N(u) \cap N(v)| \geq d \quad \text{and} \\
&\text{if } v \in \overline{S_\alpha(u)} \text{ then } |N(u) \cap N(v)| \leq d.
\end{align*}
\]

Now, because cycles that contain \(u\) and a given \(v \notin S_\alpha(u)\) are in one to one correspondence with a subset of the at most \(\binom{|N(u) \cap N(v)|}{2}\) pairs of distinct edges from \(u\) to \(N(u) \cap N(v)\), and because of Equation (4), we conclude that the 4-cycles through \(u\) whose opposing vertex is not in \(S_\alpha(u)\) are at most \(\sum_{v \notin S_\alpha(u)} \binom{|N(u) \cap N(v)|}{2} \leq (1/2)d \sum_{v \in S_\alpha(u)} |N(u) \cap N(v)|\).

Assume now that \(\alpha \Delta^{4/3} \leq |N^2(u)|\), and therefore by Equation (2) that \(|S_\alpha(u)| = \alpha \Delta^{4/3}\) (otherwise all vertices in \(N^2(u)\) are special and so there is nothing to prove). Observe that because there at most \(\Delta^2\) edges between \(N(u) \cap N(v)\) and \(N^2(u)\), and because of Equation (3) above,

\[
\sum_{v \in S_\alpha(u)} |N(u) \cap N(v)| \leq \Delta^2 - dS_\alpha(u) \leq \Delta^2 - d\alpha \Delta^{4/3}
\]

and therefore the number of 4-cycles through \(u\) whose opposing vertex \(v \notin S_\alpha(u)\) is at most \((1/2)d(\Delta^2 - d\alpha \Delta^{4/3})\), a binomial in \(d\) whose maximum is \(\frac{\Delta^{8/3}}{8\alpha}\). \(\Box\)

To facilitate the notation, we now give the following technical definition.

Definition 2. We call a coloring \(\alpha\)-specially proper, if for any two vertices \(u, v\) such that \(v\) is a neighbor of \(u\) or \(v \in S_\alpha(u)\), \(u\) and \(v\) are differently colored.

3. Some properties of random colorings

For simplicity, we call colorings obtained by choosing for each vertex a color from the palette at our disposal uniformly at random in an independent fashion for each vertex just “random colorings”. We claim:

Lemma 2. For any graph with maximum degree \(\Delta\) and any positive \(\alpha\), a random coloring from a palette with \([\alpha \Delta^{4/3}] + \Delta + 1\) colors is \(\alpha\)-specially proper with positive probability.
Proof. Given a vertex \( u \), there are at most \( \lceil \alpha \Delta^{4/3} \rceil \) vertices forming a \( \alpha \)-special pair with \( u \) and also at most \( \Delta \) neighbors of \( u \). Therefore a palette with at least \( \lceil \alpha \Delta^{4/3} \rceil + \Delta + 1 \) colors suffices for a random coloring to have positive probability to avoid for \( u \) all the colors of its neighbors and of vertices that form a special pair with \( u \). Since vertices are assigned their colors independently, positive is the probability for a random coloring to be \( \alpha \)-specially proper (recall at this point that all parameters except the number of steps of the algorithms are considered constant). □

4. The Moser part of the proof

In this section we will show that, for any \( \alpha > 2^{-1/3} \), \( \lceil \alpha \Delta^{4/3} \rceil + \Delta + 1 \) colors suffice to color the vertices of a graph in a way that although may produce a not \( \alpha \)-specially proper (or not even proper) coloring, it succeeds nevertheless in producing a coloring where for every vertex \( u \), for all 4-cycles that contain \( u \) and the opposing to \( u \) vertex does not form a \( \alpha \)-special pair with \( u \) (these are the commoner 4-cycles), as well as for all cycles of length at least 6 that contain \( u \), not both equal parity sets are monochromatic, i.e. not all equal parity pairs of vertices are homochromatic. For this we will need to assume that the maximum degree of the graph is at least as small as an integer depending on \( \alpha \) (but not depending on the graph).

In what follows, assume we have a palette of \( \lceil \alpha \Delta^{4/3} \rceil + \Delta + 1 \) colors, \( \alpha > 2^{-1/3} \). Let also \( \mathcal{B} \) be the set comprised (i) of all 4-cycles whose opposing vertices do not form \( \alpha \)-special pairs and (ii) of all 5-paths, that is paths containing five edges and six vertices. Recall that the elements of \( \mathcal{B} \) are ordered according to \( \prec \). Given a set \( B \in \mathcal{B} \), a pivot vertex \( u \) of \( B \) is any vertex in \( B \) if \( B \) is a 4-cycle, or any of \( B \)'s endpoints if \( B \) is a 5-path. In the former case, let \( B(u) := \{ u = u_1^B, \ldots, u_4^B \} \) be the set of consecutive vertices of \( B \) in its positive traversal beginning from \( u \), while in the latter case let \( B(u) := \{ u = u_1^B, \ldots, u_6^B \} \) be the set of consecutive vertices of \( B \) starting from \( u \). Given a pivot vertex \( u \) of a set \( B \in \mathcal{B} \), we define the scope of \( B(u) \) to be the set \( sc(B(u)) := \{ u_1^B, \ldots, u_{k-2}^B \} \), where \( k = 4 \) or 6. In the sequel, we call badly colored the sets in \( \mathcal{B} \) whose both equal parity sets are monochromatic. Note that for a \( B \in \mathcal{B} \) to be badly colored, two distinguished vertices of different parity can have any color and the rest must agree with the color of their respective equal parity distinguished vertex. Consider now MOSER\textsc{Alg}, Algorithm 1 defined below.

MOSER\textsc{Alg} chooses independently for each vertex a color u.a.r. from the palette and then checks whether any vertex \( u \) is a pivot vertex of a badly colored \( B \in \mathcal{B} \). If it finds one, it chooses the least such \( u \) and the least such \( B \) and recolors its vertices in its scope, again by choosing colors independently and u.a.r from the palette. Finally, MOSER\textsc{Alg} checks if either of the recolored vertices is in a (possibly recolored) badly colored \( B \in \mathcal{B} \) and loops in this fashion.

Remark 1. Parenthetically, note that this process introduces dependencies between the colors, since choosing the least vertex \( u \) and set \( B \) means that all previous, with respect to the assumed ordering, vertices are not pivot vertices in badly colored sets of \( \mathcal{B} \).

It is immediate by line 4 of MOSER\textsc{Alg} that, if and when Algorithm 1 terminates, the produced coloring has no badly colored 4-cycles whose opposing vertices do not form special pairs, or badly colored cycles of length at least 6. On the other hand,
Algorithm 1 MoserAlg

\begin{algorithm}
\begin{algorithmic}[1]
\State \textbf{for} each \( u \in V \) \textbf{do}
\State \hspace{1em} Choose a color from the palette, independently for each \( u \), and \text{u.a.r.}
\State \textbf{end for}
\State \textbf{while} there is a pivot vertex of a badly colored set in \( B \), let \( u \) be the least such vertex and \( B \) be the least such set and \textbf{do}
\State \hspace{1em} \text{RECOLOR}(u, B)
\State \textbf{end while}
\State \textbf{return} the current coloring
\end{algorithmic}
\end{algorithm}

\text{RECOLOR}(u, B), \quad B(u) = \{u = u^B_1, \ldots, u^B_k\}, \quad k = 4 \text{ or } k = 6.

\begin{algorithm}
\begin{algorithmic}[1]
\State \text{Choose a color independently for each } v \in \text{sc}(B(u)), \text{ and } \text{u.a.r.}
\State \textbf{while} there is vertex in \( \text{sc}(B(u)) \) which is a pivot vertex of a badly colored set in \( B \), let \( u' \) be the least such vertex and \( B' \) be the least such set and \textbf{do}
\State \hspace{1em} \text{RECOLOR}(u', B')
\State \textbf{end while}
\end{algorithmic}
\end{algorithm}

as mentioned earlier, the coloring might not be \( \alpha \)-specially proper (or not even proper).

A step of MoserAlg is any (re-)coloring of a single vertex. A RECOLOR procedure made from line 5 is a root call, while one made from line 3 of the RECOLOR procedure is a recursive call. A phase is the collection of steps of a call of the RECOLOR procedure, not including the steps contained in recursive calls of RECOLOR within that call. Phases are nested and for our complexity analysis, we count the number of phases in an execution of MoserAlg, rather than the number of steps. Note that because phases are comprised of at most 4 steps (recoloring of a 5-path), this does not make any difference in matters of computational complexity.

Our first objective is to show that MoserAlg actually progresses during each phase. Specifically we show that at the end of each phase, the class of sets \( B \in B \) that are badly colored is a strict subclass of that class at the beginning of that phase.

Lemma 3. Let \( V \) be the set of vertices which at the beginning of some RECOLOR\((u, B)\) procedure are not pivot vertices in a badly colored set \( B \). Then, if and when that call terminates, no such vertex in \( V \cup \{u\} \) exists.

Proof. The result is obvious for \( u \), by line 2 of the RECOLOR\((u, B)\) procedure. Assume now that there is some \( v \in V \) that at the end of RECOLOR\((u, B)\) is a pivot vertex in a badly colored element of \( B \). Since \( v \in V \), there must be some point during RECOLOR\((u, B)\) where some other set of \( B \), that has \( v \) as pivot, was recolored and, as a result, is now badly colored. Let RECOLOR\((v', B')\) be the last time this happened and \( B^* \) be the badly colored set. Since \( B^* \) had its vertices recolored by RECOLOR\((v', B')\), sc\((B^*)\) and sc\((B')\) share at least one common vertex. Then, by line 2 of RECOLOR\((v', B')\), this call could not have terminated, and thus neither could RECOLOR\((u, B)\). Contradiction.

By Lemma 3, we get:

Lemma 4. There are at most \( l \), the number of vertices of \( G \), i.e. a constant, repetitions of the while-loop of line 4 of the main part of MoserAlg.
We will now show that for any $\alpha > 2^{-1/3}$, there is an integer $\Delta_\alpha$ such that for any graph whose maximum degree is at least $\Delta_\alpha$, the following two facts hold:

**Fact 1.** The probability $P_n$ that MOSERALG lasts at least $n$ phases is inverse exponential in $n$.

**Fact 2.** The probability $Q$ that MOSERALG halts, but the produced coloring is not $\alpha$-specially proper, is bounded away from 1.

The rest of this section is devoted into proving the above facts. As a first step towards the proof, we depict the phases of an execution of MOSERALG with a $(2, 4)$-full labeled rooted forest, the witness structure. A rooted tree is an acyclic and connected graph with a distinguished vertex, the root. We say that it is $(2, 4)$-full, if all its internal nodes (i.e. nodes with at least one child) have either two or four children. Nodes with no children are called leaves. A $(2, 4)$-full labeled rooted forest is a graph whose connected components are $(2, 4)$-full rooted trees, whose internal nodes are labeled with pairs $(u, B)$, where $u$ is a pivot vertex of $B \in B$ and whose leaves are labeled with pairs $(u, \emptyset)$, where $u$ is a vertex of $G$. We call $u$ the vertex-label and $B$ (or the $\emptyset$) the set-label of the node of the tree.

**Definition 3.** A labeled rooted forest $F$ is called feasible, if the following conditions hold:

i. Let $u$ and $v$ be the vertex-labels of two distinct nodes $x$ and $y$ of $F$. If $x$ and $y$ are both either roots of $F$ or siblings in $F$, then $u$ and $v$ are distinct.

ii. If $(u, B)$ is the label of an internal node $x$ of the forest, the vertex-labels of the children of $x$ comprise the set $sc(B(u))$.

Notice that because the vertex-labels of the roots of the trees are distinct, a feasible forest has at most as many trees as the number $l$ of vertices. Also, by condition (ii), if a set-label $B$ of a node $x$ is a 4-cycle, $x$ has two children and if it is a 5-path, four.

Let now $F$ be a feasible forest. We order the internal nodes of $F$ in the following way: (i) two internal nodes of the same tree that are siblings are ordered according to their vertex-label, (ii) the internal nodes of each tree are ordered in a depth-first fashion, respecting the order of siblings and (iii) internal nodes of different trees are ordered according to the vertex-labels of their corresponding roots. Taking the internal nodes of $F$ in this order, we obtain the label-sequence of $F$, denoted as:

$$\mathcal{L}(F) := ((u_1, B_1), \ldots, (u_{|F|}, B_{|F|})),$$

where $|F|$ is the number of internal nodes of $F$.

Given an execution of MOSERALG with at least $n$ phases, we construct a feasible forest associated with the first $n$ such phases as follows: the feasible forest has $n$ internal nodes, one for each phase, where a node corresponding to a call of RECOLOR$(u, B)$ is labeled by $(u, B)$. We structure these nodes according to the order their labels appear in the recursive stack implementing MOSERALG: the children of a node $x$ labeled by $(u, B)$ correspond to the recursive calls of RECOLOR made by line 3 of RECOLOR$(u, B)$, with the leftmost child corresponding to the first such call and so on. Finally, to every node $x$ labeled by $(u, B)$, we add leaves labeled accordingly, such that the set of vertex-labels of the children of $x$ is $sc(B)$ (set-labels of the added leaves are $\emptyset$). We call the feasible forest $F$ created this way the witness forest of the first $n$ phases of MOSERALG’s execution.
For a feasible forest $F$ with $n$ internal nodes, let $W_F$ denote the event that $F$ is the witness forest of an execution of MoserAlg that lasts for at least $n$ phases.

It is immediate now to see that the probability $P_n$ of MoserAlg lasting for at least $n$ phases satisfies:

$$(5) \quad P_n = \Pr \left[ \bigcup_{F : |F| = n} W_F \right] = \sum_{F : |F| = n} \Pr[W_F],$$

where the second equality comes from the fact that different witness forests with $n$ internal nodes are necessarily associated with different executions of MoserAlg.

To bound the sum of Eq. (5), we use ValidationAlg, algorithm 2 below.

**Algorithm 2 ValidationAlg($F$)**

Input: Feasible forest $F$, where $L(F) = (u_1, B_1), \ldots, (u_{|F|}, B_{|F|})$.

1. Color the edges of $G$, independently and selecting for each a color u.a.r. from the palette.
2. for $i = 1, \ldots, |F|$ do
3. \hspace{1em} if $B_i$ is badly colored then
4. \hspace{2em} recolor the vertices in $sc(B_i(u_i))$ independently by selecting for each a color u.a.r. from the palette.
5. \hspace{1em} else
6. \hspace{2em} return failure and exit
7. \hspace{1em} end if
8. end for
9. return success

ValidationAlg takes as input a feasible forest and reports success if it manages to verify that all set-labels of internal nodes of the forest are badly colored. If at any point of its execution it examines a not badly colored $B \in B$, it terminates its procedure and reports failure. A phase of ValidationAlg is any iteration of the steps of lines 2–8.

**Lemma 5.** At the end of each phase that ValidationAlg did not report failure, each vertex of $G$ is as if it was independently assigned a color u.a.r. from the palette.

**Proof.** Consider the end of phase $i$, when ValidationAlg examined the set $B_i \in B$ with pivot $u_i$. Since ValidationAlg did not report failure, before the recoloring of line 4, the colors of $B_i$ were distributed as if chosen independently and u.a.r. from the palette, conditional that $B_i$ is badly colored. But since $sc(B_i(u_i))$ was subsequently recolored, the conditional is lifted (notice that in general recoloring only one of a given pair of vertices lifts the conditional that the vertices are homochromatic).

A direct consequence of Lemma 5 is that, on input a feasible forest $F$ with $n$ internal nodes and label sequence $L(F) = (u_1, B_1), \ldots, (u_n, B_n)$, the probability that ValidationAlg reports success in phase $i$, given that it did not fail in any previous phase, is $\Pr[B_i$ is badly colored for a random coloring], $i = 1, \ldots, n$. Let now $V_F$ denote the event that ValidationAlg is successful on input $F$ and set:

$$(6) \quad \hat{P}_n := \Pr \left[ \bigcup_{F : |F| = n} V_F \right] \leq \sum_{F : |F| = n} \Pr[V_F],$$
where the inequality comes from the union-bound. Let also \( \hat{Q} \) be the probability that \( \text{VALIDATIONALG} \), on input a forest \( F \), (i) reports \textit{success} and (ii) the produced coloring after its termination is not \( \alpha \)-specially proper. We now prove the following result.

\textbf{Lemma 6.} For every \( n \), it holds that \( P_n \leq \hat{P}_n \) and \( Q \leq \hat{Q} \).

\textit{Proof.} Let \( F \) be a witness forest of an execution of \( \text{MOSERALG} \) with \( n \) nodes. Then if \( \text{VALIDATIONALG} \), on input \( F \), makes the exact same random choices as \( \text{MOSERALG} \), it will report \textit{success} on input \( F \). Furthermore, if \( \text{MOSERALG} \) terminated in its \( n \)-th phase and produced a coloring that is not \( \alpha \)-specially proper, then the same holds for the produced coloring after the termination of \( \text{VALIDATIONALG} \). \( \square \)

We will bound \( \hat{P}_n \) using purely combinatorial arguments. Let:

\begin{equation}
q := \frac{1}{\alpha \Delta^{4/3}} > \frac{1}{[\alpha \Delta^{4/3}] + \Delta + 1}.
\end{equation}

We define the weight \( ||F|| \) of a feasible forest \( F \) by taking the product of weights assigned to its nodes as follows: for each node with set-label \( B \), if \( B \) is a 4-cycle, assign weight \( q^2 \); if \( B \) is a 5-path, assign weight \( q^4 \) and if it is \( \emptyset \), assign weight 1. It is not difficult to observe that for any feasible forest \( F \), \( \Pr[V_F] \leq ||F|| \) and thus, by Eq. (6):

\begin{equation}
\hat{P}_n \leq \sum_{F:||F||=n} ||F||.
\end{equation}

Recall that in each feasible forest we have \( l \), the number of vertices of \( G \), trees. Let \( T_j \) be the set of all possible feasible trees, whose root has as a vertex-label the vertex \( u_j \). Let also \( T \) be the collection of all \( l \)-ary sequences \((T_1, \ldots, T_l)\) with \( T_j \in T_j \).

Now, obviously:

\begin{equation}
\sum_{F:||F||=n} ||F|| = \sum_{(T_1, \ldots, T_l) \in T} ||T_1|| \cdots ||T_l||
\end{equation}

\begin{equation}
= \sum_{\sum n_i = n, n_1, \ldots, n_l \geq 0} \left( \sum_{T_1 \in T_1: |T_1| = n_1} ||T_1|| \right) \cdots \left( \sum_{T_l \in T_l: |T_l| = n_l} ||T_l|| \right).
\end{equation}

We now obtain a recurrence for each factor of (9).

\textbf{Lemma 7.} Let \( T^u \) be anyone of the \( T_j \). Then:

\begin{equation}
\sum_{T \in T^u: |T| = n} ||T|| \leq R_n,
\end{equation}

\end{document}
where $R_n$ is defined as follows:

(11) for $n \geq 1$,

$$R_n := \frac{\Delta^{8/3}}{8\alpha^3} q^2 \sum_{n_1, n_2 \geq 0} \left( R_{n_1} R_{n_2} \right) + \frac{\Delta^5 q^4}{\Delta^{1/3} \alpha^4} \sum_{n_1, n_2, n_3, n_4 \geq 0} \left( R_{n_1} R_{n_2} R_{n_3} R_{n_4} \right)$$

$$= \frac{1}{8\alpha^3} \sum_{n_1+n_2=n-1 \atop n_1, n_2 \geq 0} \left( R_{n_1} R_{n_2} \right) + \frac{1}{\Delta^{1/3} \alpha^4} \sum_{n_1, n_2, n_3, n_4 \geq 0} \left( R_{n_1} R_{n_2} R_{n_3} R_{n_4} \right),$$

$R_0 = 1$.

**Proof.** The result is obvious for $n = 0$, because the tree without internal nodes (i.e. comprised of just one leaf) has weight 1. For $n > 0$, we have two cases for the set-label $B$ of the root of $T \in T^u$. If is one of the, by Lemma 1, $\Delta^{8/3}$-4-cycles whose opposing to $u$ vertices do not form special pairs, then the root $u$ has weight $q^2$ and two children. Otherwise, observe that there are at most $\Delta^3$ 5-paths beginning from $u$. In this case, $u$ has weight $q^4$ and four children.

To estimate the asymptotic behavior of the sequence $R_n$, we will follow well known methods presented in the book *Analytic Combinatorics* of Flajolet and Sedgewick, namely we will find the Ordinary Generating Function (OGF) of $R_n$ and apply [2, Proposition IV.5, p. 278]. For technical reasons, we find the OGF $R(z)$ for $R_n$, with $R_n, n \geq 0$ being the coefficient of $z^n$ instead of $z^n$, and the constant term being 0.

Multiply both sides of Eq. (11) by $z^{n+1}$ and sum for $n = 1, \ldots, +\infty$, to get:

$$R(z) - z R_0 = \frac{1}{8\alpha^3} z R(z)^2 + \frac{1}{\Delta^{1/3} \alpha^4} z R(z)^4 \Rightarrow$$

$$R(z) = z \left( \frac{1}{8\alpha^3} R(z)^2 + \frac{1}{\Delta^{1/3} \alpha^4} R(z)^4 \right) + z$$

Set $R := R(z)$ and observe that for:

$$\phi(x) = \frac{x^4}{\Delta^{1/3} \alpha^4} + \frac{x^2}{8\alpha^3} + 1,$$

we have that $R = z \phi(R)$.

Therefore following [2, Proposition IV.5, p. 278], we consider the characteristic equation:

$$x \phi'(x) - \phi(x) = 0 \iff \frac{3x^4}{\alpha^4 \Delta^{1/3}} + \frac{x^2}{8\alpha^3} - 1 = 0,$$

and we let $\tau$ be its unique positive solution. It only remains to find the range $\alpha$ for which $\phi'(\tau) < 1$. Towards this, we consider instead $\hat{\phi}(x) = \frac{x^2}{8\alpha^3} + 1$ whose characteristic equation is $\frac{x^2}{8\alpha^3} - 1 = 0$, which has unique positive solution $\sqrt[3]{8\alpha}$. We first observe that:

**Lemma 8.** The unique positive solution of the characteristic equation (14) for $\Delta \to +\infty$ is the same with that of the characteristic equation of $\hat{\phi}(x)$ that is, $\sqrt[3]{8\alpha}$. 

Proof. It is easy to check that the unique positive solution of Eq. (14) is:

\[
\left( \frac{n\Delta^{1/6}}{48} \left( \sqrt{\Delta^{1/3} + 768a^2} - \Delta^{1/6} \right) \right)^{1/2},
\]

which, by taking the limit for $\Delta$ going to infinity, is $8a^3$. □

So, the range of $\alpha$ for which $\phi'(\tau) < 1$ is computed as follows:

\[
4\frac{\sqrt{\tau}}{a^4 \Delta^{1/3}} + \frac{\sqrt{\tau}}{4a^3} < 1 \iff 0 + \frac{\sqrt{8a^3}}{4a^3} < 1 \iff 1 + \frac{1}{21^{1/2}a^{3/2}} < 1 \iff 2^{-1/3} < a.
\]

It follows that, for every $\alpha > 2^{-1/3}$, there is a $\Delta_\alpha$ (depending on $\alpha$), such that if the maximum degree $\Delta$ of the graph is at least $\Delta_\alpha$, then $\lceil \alpha \Delta^{4/3} \rceil + \Delta + 1$ colors suffice to color the graph such that there are no badly colored 4-cycles whose opposing vertices do not form special pairs, or badly colored cycles of length 6 or more.

We turn now to $Q$. By Lemma 5, the produced coloring at the end of any execution of VALIDATIONALG is random. Also, by Lemma 2, the probability that a random coloring is $\alpha$-specially proper is positive. This means that $Q$ and, by Lemma 6, $Q$ too, is less than 1. This completes the proof of Facts 1 and 2. □

5. The completion of the proof of Theorem 1

Algorithm 3 MAINALGORITHM

1: Execute MOSERALG and if it stops, let $C$ be the coloring it generates.
2: while $C$ is not $\alpha$-specially proper do
3: Execute MOSERALG anew and if it halts, set $C$ to be the newly generated coloring
4: end while

Obviously MAINALGORITHM, if and when it stops, generates a proper acyclic coloring. Now the probability that before MAINALGORITHM completed $n$ repetitions of the while loop of line 2, at least one of the loops lasted for at least $n$ steps is by Fact 1 inverse exponential in $n$. Also by Fact 2, the probability that the while loop of MAINALGORITHM is repeated at least $n$ times is inverse exponential in $n$, because a Bernoulli trial with constant positive probability of success has inverse exponential in $n$ probability to be repeated $n$ times before success is attained. Therefore the probability that MAINALGORITHM lasts for at least $n^2$ steps is inverse exponential in $n$. This completes the proof of Theorem 1. □

Remark 2. Our technique does not lead to the conclusion that for large enough maximum degree $\Delta$, the chromatic number is at most $\lceil 2^{-1/3} \Delta^{4/3} \rceil + \Delta + 1$, because we cannot exclude that $\Delta_\alpha$ approaches $+\infty$ as $\alpha$ approaches $2^{-1/3}$. 

\[ 2^{-1/3} < \alpha < 2^{1/3}. \]
References

[1] Noga Alon, Colin McDiarmid, and Bruce Reed. Acyclic coloring of graphs. *Random Structures & Algorithms*, 2(3):277–288, 1991.

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

[3] Ioannis Giotis, Lefteris Kirousis, Kostas I. Psaromiligkos, and Dimitrios M. Thilikos. On the algorithmic Lovász local lemma and acyclic edge coloring. In *Proceedings of the 12th workshop on analytic algorithms and combinatorics*. Society for Industrial and Applied Mathematics, 2015. Available: http://epubs.siam.org/doi/pdf/10.1137/1.9781611973761.2.

[4] Daniel Gonçalves, Michael Montassier, and Alexandre Pinlou. Acyclic coloring of graphs and entropy compression method. *Discrete Mathematics*, 343(2):1–13, 2020.

[5] Branko Grünbaum. Acyclic colorings of planar graphs. *Israel journal of mathematics*, 14(4):390–408, 1973.

[6] Lefteris Kirousis and John Livieratos. The acyclic chromatic index is less than the double of the max degree. 2020. arXiv:1901.07856.

[7] 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.

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

[9] Sokol Ndreca, Aldo Procacci, and Benedetto Scoppola. Improved bounds on coloring of graphs. *European Journal of Combinatorics*, 33(4):592–609, 2012.

[10] Jean-Sébastien Sereni and Jan Volec. A note on acyclic vertex-colorings. *Journal of Combinatorics*, 7(4):725–737, 2016.

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

[12] Terence Tao. Moser’s entropy compression argument, 2009. Available: https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/.