# The local nature of list colorings for graphs of high girth*

Flavio Chierichetti† Andrea Vattani‡

**Abstract**

We consider list coloring problems for graphs $\mathcal{G}$ of girth larger than $c \log_{\Delta-1} n$, where $n$ and $\Delta \geq 3$ are, respectively, the order and the maximum degree of $\mathcal{G}$, and $c$ is a suitable constant. First, we determine that the edge and total list chromatic numbers of these graphs are $\chi'_l(\mathcal{G}) = \Delta$ and $\chi''_l(\mathcal{G}) = \Delta+1$. This proves that the general conjectures of Bollobás and Harris (1985), Behzad and Vizing (1969) and Juvan, Mohar and Škrekovski (1998) hold for this particular class of graphs.

Moreover, our proofs exhibit a certain degree of "locality", which we exploit to obtain an efficient distributed algorithm able to compute both kinds of optimal list colorings.

Also, using an argument similar to one of Erdös, we show that our algorithm can compute $k$-list vertex colorings of graphs having girth larger than $c \log_{k-1} n$.

## 1 Introduction

Graph coloring is a fundamental problem in computer science and combinatorics. Applications arise in many different areas, such as networks, resource allocations and VLSI design. For many coloring problems, though, no efficient algorithms are known to exist. This led to considering restrictions of coloring problems to special classes of graphs.

In this paper we consider (list) edge, total and vertex colorings for graphs of high girth (that is, for graphs having no small cycles), with

*A preliminary version of this paper appeared in [4]. This work was partially supported by a grant of Yahoo! Research and by the MIUR PRIN Project "Web Ram: web retrieval and mining". The main part of this work was carried out at the Computer Science Department of Sapienza University of Rome.

†Dipartimento di Informatica, Sapienza University of Rome. Email: chierichetti@di.uniroma1.it

‡CSE Department, University of California, San Diego. Email: avattani@ucsd.edu

special emphasis on distributed algorithms. In the list edge coloring problem, we have a graph $\mathcal{G}$ and, for each of its edges $e$, a list of colors available for $e$. The goal is to compute a proper coloring using colors from the lists, where a coloring is proper if adjacent edges have different colors. The list edge chromatic number (or list chromatic index) of $\mathcal{G}$, denoted $\chi'_l(\mathcal{G})$, equals the minimum integer $t$ such that, for each possible assignment of lists of $t$ colors to the edges of $\mathcal{G}$, a proper coloring of $\mathcal{G}$ exists. The edge chromatic number (or chromatic index) $\chi'(\mathcal{G})$ of $\mathcal{G}$ is the minimum integer $t$ such that, if all the edges of $\mathcal{G}$ are assigned the same list of $t$ colors, a proper coloring of $\mathcal{G}$ exists.

The list vertex coloring problem, and the list and non-list vertex chromatic numbers $\chi_l(\mathcal{G}), \chi(\mathcal{G})$, are analogous, except that in this case the vertices are the elements of the graph to be properly colored. In the list total coloring problem both vertices and edges have lists, and the problem is to color the graph in such a way that any two adjacent or incident objects, whether edges or vertices, have different colors. We use $\chi''(\mathcal{G})$ and $\chi''_l(\mathcal{G})$ to denote, respectively, the total chromatic number and the list total chromatic number of $\mathcal{G}$.

In this paper we show that, if $\mathcal{G}$ is a graph with $n$ vertices, maximum degree $\Delta \notin \{1, 2\}$, and girth at least $c \log_{\Delta-1} n$ (for a suitable constant $c$), then $\chi'(\mathcal{G}) = \chi'_l(\mathcal{G}) = \Delta$ and $\chi''(\mathcal{G}) = \chi''_l(\mathcal{G}) = \Delta + 1$.

Our constructive proofs highlight a local property of the coloring operations, that we use to obtain such optimal colorings efficiently in a distributed setting.

From an existential point of view, our results settle, for the case of high girth graphs, a conjecture of Bollobás and Harris [2] (that states $\chi'(G) = \chi'_l(G)$), a conjecture of Behzad and Vizing [1] ($\chi''(G) \leq \Delta+2$) and a related conjecture of Juvan, Mohar, Škrekovski [14] ($\chi''(G) = \chi''_l(G)$), all of which were formulated for general graphs. The first two conjectures date back to more than two decades ago, while the third one is more recent.

Also, by extending an argument of Erdös [8], we show that list vertex colorings can be obtained efficiently by a distributed algorithm. In this case, though, the number of colors used may not be optimal.

## 1.1 Main results

We now state more precisely our results. Recall that no chromatic number is greater than its list counterpart.

**Theorem 1.** *If $\mathcal{G}$ is a graph with $n$ nodes, $\Delta(\mathcal{G}) \neq 2$, having girth $g(\mathcal{G}) > 4\lceil \log_{\Delta-1} n \rceil + 1$, then $\chi'_l(\mathcal{G}) = \Delta(\mathcal{G})$.*

**Theorem 2.** *Let $k \geq 3$. If $\mathcal{G}$ is a graph with $n$ nodes, having girth $g(\mathcal{G}) > 2\lceil \log_{k-1} n \rceil$, then $\chi_l(\mathcal{G}) \leq k$.*

**Theorem 3.** *If $\mathcal{G}$ is a graph with $n$ nodes, $\Delta(\mathcal{G}) \neq 1, 2$, having girth $g(\mathcal{G}) > 4\lceil \log_{\lceil \frac{\Delta}{2} \rceil} n \rceil + 3$, then $\chi_l''(\mathcal{G}) = \Delta + 1$.*

These theorems imply that, if the girth of a graph is high enough and the graph is not a collection of paths and cycles, then that graph is both Class-1 and Type-1 (a graph $\mathcal{G}$ is Class-1 if $\chi'(\mathcal{G}) = \Delta$ and it is Type-1 if $\chi''(\mathcal{G}) = \Delta + 1$).

The constraint $\Delta \neq 2$ is necessary for trivial reasons, since any cycle $C_n$ has $g(C_n) = n$, $\Delta(C_n) = 2$, and (a) if $n \bmod 2 \neq 0$ then $\chi'(C_n) = \Delta + 1$, while (b) if $n \bmod 3 \neq 0$ then $\chi''(C_n) = \Delta + 2$. For the total chromatic number even the requirement $\Delta \neq 1$ is needed as any matching is Type-2 (a graph $\mathcal{G}$ is Type-2 if $\chi''(\mathcal{G}) \geq \Delta + 2$).

As for the girth requirement, we show the existence of an infinite family of Class-2 graphs (a graph $\mathcal{G}$ is Class-2 if $\chi'(\mathcal{G}) = \Delta + 1$) having a girth three times smaller than the one required in Thm. 1.

**Proposition 4.** *For each element of an infinite sequence of increasing degrees $\{\Delta_i\}_{i=1}^{\infty}$, there exists an infinite family of graphs $\{\mathcal{G}_j(\Delta_i)\}_{j=1}^{\infty}$ of maximum degree $\Delta_i$ and increasing order $n_j$, such that $g(\mathcal{G}_j(\Delta_i)) \geq \frac{4}{3} \log_{\Delta_i - 1} n_j - O(1)$ and $\chi'(\mathcal{G}_j(\Delta_i)) = \Delta_i + 1$.*

On the other hand, we were not able to obtain an infinite family of Type-2 graphs having reasonably large girth.

This introduces our main open question: what is the minimum constant $c$ such that, for each $\Delta \geq 3$, each graph of order $n$ of maximum degree $\Delta$ is Class-1 if its girth is at least $(c + o(1)) \log_{\Delta-1} n$? Proposition 4 shows that $c \geq \frac{4}{3}$. Its proof requires regular graphs of high girth — in fact, the bound is obtained using the family of graphs of Lubotzky et al. [19] (having girth asymptotic to $\frac{4}{3} \log_{\Delta-1} n$).

An easy argument shows that the girth of a regular graph cannot be more than $(2 \pm o(1)) \log_{\Delta-1} n$. If graphs of such girth exist then we would have $c \geq 2$.

Similar questions can be asked for the other two kinds of coloring.

Another interesting question, raised by an anonymous referee, considers graphs whose girth exceeds the lower bound of Thm. 1, 2, 3 *only* after the removal of few edges (where "few" is to be determined). Under which conditions does the original graph have the same coloring numbers predicted by Thm. 1, 2, 3?

## 1.2 Algorithmic consequences

An interesting feature of our method is that it illustrates a certain local nature of list colorings for high girth graphs, in a way related to the result of [22] on $\Delta$-vertex colorings of general graphs. For instance, for list edge colorings we show the following. Assume that the whole graph $\mathcal{G}$ is colored except for one last edge $e$. Then, to color $e$ it is enough to re-color a neighborhood of $e$ of radius $O(\log n)$. Analogous properties hold for list total and list vertex colorings. These properties lead to efficient distributed implementations of our algorithms.

**Theorem 5.** *The list colorings of Theorems 1, 2, 3 can be computed in $O(\log^3 n)$-many communication rounds, in the synchronous, message passing model of computation.*

All the three kinds of colorings will be computed by the same algorithm.

We remark that this meta-algorithm can be simulated sequentially in polynomial-time, as every node in the network only performs polynomially many operations in every round of the protocol.

## 2 Related work

A well-known result of Vizing [25] shows that the chromatic index $\chi'(\mathcal{G})$ of any graph $\mathcal{G}$ of maximum degree $\Delta$ is either $\Delta$ or $\Delta + 1$. A conjecture of Bollobás and Harris [2] states that the list chromatic index $\chi'_l(\mathcal{G})$ is equal to $\chi'(\mathcal{G})$. For total coloring, a conjecture independently suggested by Behzad [1] and Vizing states that $\chi''(\mathcal{G}) \leq \Delta+2$. A more recent conjecture by Juvan, Mohar, Škrekovski [14] states that the list total chromatic number $\chi''_l(\mathcal{G})$ equals $\chi''(\mathcal{G})$.

On the other hand, it is known [9] that the gap between the chromatic number $\chi(\mathcal{G})$ and the list chromatic number $\chi_l(\mathcal{G})$ of a graph can be logarithmic in its order.

There has been a lot of work in trying to prove the first two conjectures. For arbitrary graphs $\mathcal{G}$, Kahn [15] proved that the list chromatic index is $\chi'_l(\mathcal{G}) \leq (1 + o(1))\Delta$, while Molloy and Reed [21] proved that the the total chromatic number is $\chi''(\mathcal{G}) \leq \Delta + c$, for some (rather large) constant $c$. Exact values are known only for special classes of graphs; we now comment on these kinds of results as they are more directly related to ours.

The relationship between girth and list edge chromatic number has been studied in [16], where it is shown that, if $g(\mathcal{G}) \geq 8\Delta(\ln \Delta + 1.1)$, then $\chi'_l(\mathcal{G}) \leq \Delta + 1$ (and thus $\chi''_l(\mathcal{G}) \leq \Delta + 3$). Our requirement on the girth is less stringent for large enough $\Delta$ (e.g. already for $\Delta$ logarithmic in $n$). Note that, even for smaller $\Delta$, we establish a better bound of $\Delta$ (resp. $\Delta + 1$) for the list edge (resp., total) chromatic number.

Another approach uses the concept of *degeneracy* of a graph, i.e. the maximum smallest degree of its subgraphs. Vizing showed (see for example [13]) that any graph $\mathcal{G}$ with degeneracy $\leq \Delta/2$ is of Class-1. One can show that for high enough $\Delta(\mathcal{G})$, the degeneracy of our graphs (that is, graphs with girth as high as we need) is small enough for Vizing's result to hold. On the other hand, for small maximum degrees, there are graphs that satisfy our requirement and not Vizing's (for instance, take two cycles intersecting on a single edge).

In [12, 26], the authors give parallel algorithms for computing $\Delta$ and $\Delta+1$ edge and total colorings of graphs of small degeneracy. Their results are not directly comparable to ours. While their requirement is weaker than ours for large enough $\Delta$, there exist graphs of small degree that satisfy our requirement and not theirs. The main difference, however, is that our results hold for the more general case of *list* colorings, as opposed to non-list ones. Also, our method leads to efficient distributed algorithms, while their parallel algorithms do not seem to be efficiently distributable.

Borodin et al. attacked the problem from another point of view. The maximum average degree (MAD) of a graph is the maximum of the average degrees of its subgraphs. In [3], they show that, if $\Delta(\mathcal{G}) \geq 4$ and the MAD is "small enough", then $\chi_l'(\mathcal{G}) = \Delta(\mathcal{G})$ and $\chi_l''(\mathcal{G}) = \Delta(\mathcal{G}) + 1$. The result extends to list chromatic index for $\Delta(\mathcal{G}) = 3$. The relationship between this result and that in the present paper is unclear and intriguing. Let $m(g, n)$ be the maximum number of edges of graphs having girth $g$ and order $n$, and let $M(g, n) := n^{1+1/\lfloor (g-1)/2 \rfloor}$. A well-known bound states that $m(g, n) \leq M(g, n)$, and improving this is known to be a challenging open problem (see for instance [20]). It can be shown that if our result is subsumed by that of [3] then a sharper bound holds for $m(g, n)$ at least for the girths we require. More precisely, for these girths, the bound would have to be improved non trivially, by at least a $\frac{1}{\sqrt{2}}$ factor.

Be as it may, our proof is conceptually different and highlights an interesting local property of list colorings that leads directly to efficient distributed algorithms. Our result holds even for $\Delta(\mathcal{G}) = 3$ in the case of list total coloring.

As for vertex colorings, it was shown by Erdös [8] that graphs of high enough girth have small chromatic number — his proof can be modified to give an upper bound on their *list* chromatic number; our distributed algorithm can color the vertices of these graphs using a number of colors equal to that upper bound.

The distributed (non list) edge coloring problem has been the object of a lot of study (see [5, 6, 7, 10, 24] and references therein). All the previous works we are aware of considered the edge coloring problem for general graphs, obtaining suboptimal colorings.

## 3 List edge coloring

In this section we prove Theorem 1. Here we are interested in the existential result deferring the algorithmic discussion to a later section. The idea of the proof is as follows. Suppose by induction that we have list-colored the entire graph except for one last edge $e = uv$. The following local property holds. No matter how $\mathcal{G} - e$ is colored, it is always possible to reassign the colors inside a neighborhood of $e$ of radius $O(\log n)$ in such a way that there will be a free color for $e$, drawn from $e$'s list. More precisely, the neighborhood to be re-colored consists of two disjoint BFS trees, each of which is rooted at one of the two endpoints of $e$. The basis of the induction is trivial, since we can start with any edge and assign it any color from its list.

The local nature of the re-coloring operation will be later exploited to show that such list-colorings can be obtained by means of efficient distributed algorithms.

The above discussion motivates the following definitions. From now on, let $\mathcal{G}$ be the graph we are list coloring and let $\Delta = \Delta(\mathcal{G})$ denote its maximum degree. For the rest of the section, we will use the term coloring to mean list edge coloring.

**Definition 6.** *A $\Delta$-tree is a rooted tree of maximum degree at most $\Delta$ whose leaves are all at the same distance from the root. Furthermore, the degree of the root is $< \Delta$.*

The intuition that drives this and the following definitions is that, after removing $e = uv$ from $\mathcal{G}$, we want to consider two BFS trees $T(u)$ and $T(v)$ starting from $e$'s endpoints and show that they can always be re-colored in such a way that there will be a free color for $e$, regardless of how $\mathcal{G} - e$ is colored initially. The degree of the roots is $< \Delta$ by the removal of $e$. Intuitively, we do not consider leaves above the lowest level as they are not affected by the rest of the graph's coloring.

The next definition captures the idea of a tree $T$ whose set of possible colorings is constrained by the coloring of $\mathcal{G} - e$. Henceforth, we will denote by $T(u)$ a tree that is rooted at $u$.

**Definition 7.** *Let $T(r)$ be a $\Delta$-tree. $G$ is an* augmentation *of $T(r)$ if it is obtained from $T(r)$ by adding edges and paths of length two connecting only leaves of $T(r)$. Furthermore, it must be that $\Delta(G) \leq \Delta$ and $\deg_G(r) < \Delta$.*

The constraints on a $\Delta$-tree $T$ given by the list coloring of $\mathcal{G} - e$ can be succinctly expressed by coloring the edges of an augmentation $G$ of $T$.

**Definition 8.** *Let $T$ be a $\Delta$-tree. Then $(\mathcal{L}, G, \gamma)$ is a* legal triple *for $T$ if $\mathcal{L}$ is a $\Delta$-list assignment to $E(T)$, $G$ is an augmentation of $T$, and $\gamma$ is a proper coloring of $G$ that agrees with $\mathcal{L}$.*

Note that every $\Delta$-tree has at least a legal triple, say, the identical list-assignment, the trivial augmentation $G = T$, and any of its proper colorings. We now define a notion of "freedom" of $\Delta$-trees. Intuitively, a tree $T$ is $t$-free if, regardless of how $\mathcal{G} - e$ is colored, we can always re-color it in such a way that $t$ colors become available at the root.

**Definition 9.** *Let $T(r)$ be a $\Delta$-tree. $T(r)$ is* at least $t$-free *if, for each list $L$ of $\Delta$ colors and each legal triple $(\mathcal{L}, G, \gamma)$, there exists some set $C \subseteq L$ of $t$ colors such that for all $c \in C$, there exists a proper coloring $\gamma_c$ of $G$ such that*

1. *$\gamma_c(e) = \gamma(e)$, for all $e \in E(G) - E(T(r))$,*
2. *$\gamma_c(e) \in \mathcal{L}(e)$, for all $e \in E(T(r))$, and*
3. *$\gamma_c$ do not assign the color $c$ to any edge incident to $r$.*

We will occasionally say that $\gamma_c$ *agrees* with a given triple $(\mathcal{L}, G, \gamma)$, if conditions 1-2 of definition 9 are satisfied. Also, we will say that $\gamma_c$ leaves $c$ available at the root, if condition 3 also holds.

We now show some basic properties of $\Delta$-trees that will be used later in the proofs.

- Each $\Delta$-tree is at least 1-free, as for each of its legal triples, using the coloring of the triple, at most $\Delta - 1$ colors will be unavailable at the root, thus at least a color will remain in any list $L$ of cardinality $\Delta$.
- No $\Delta$-tree is at least $(\Delta + 1)$-free (again by $|L| = \Delta$).

A tree is exactly $t$-free (or, simply, $t$-free) if it is at least $t$-free, but not at least $(t + 1)$-free. If a tree is $t$-free we say that it has $t$ degrees of freedom.

**Lemma 10.** *Let $T$ be a $\Delta$-tree that is exactly $t$-free, let $L$ be a set of $\Delta$ colors, and $C \in \binom{L}{t}$. Then, there exists a legal triple $(\mathcal{L}, G, \gamma)$ for $T$ such that, for all and only $c \in C$, there exists a proper coloring $\gamma_c$ of $G$ that satisfies*

- *$\gamma_c(e) = \gamma(e)$, for all $e \in E(G) - E(T)$,*
- *$\gamma_c(e) \in \mathcal{L}(e)$, for all $e \in E(T)$, and*
- *$\gamma_c$ do not assign the color $c$ to any edge incident to $r$.*

*Proof.* Since $T$ is not at least $(t + 1)$-free, there exists a legal triple $R$ which does not leave $t + 1$ colors available at the root. But since $T$ is at least $t$-free, all legal triples allow the choice of $t$ colors at the root. So $R$ allows exactly $t$ colors. We can obtain all possible sets $C \subseteq L$ of $t$ colors from $R$ just by renaming the colors of the coloring of $R$. $\square$

**Observation 11.** *If the root of a $\Delta$-tree $T$ has exactly $k$ children and the subtree rooted at one of them is at least $(k+1)$-free, the degree of freedom of $T$ does not change if that subtree is deleted.*

*Proof.* Let $u$ be a child of $T$ such that $T(u)$ is at least $(k+1)$-free. Also, let $T'$ be the tree obtained by removing $T(u)$ from $T$. It is clear that the degree of freedom of $T'$ is not less than that of $T$.

Now we prove that is not more either. Suppose $T'$ is $t$-free, and let $R' = (\mathcal{L}', G', \gamma_0')$ any legal triple for $T'$. Consider any proper coloring $\gamma'$ of $G'$ that agrees with $R'$, and let $c$ any color available at the root of $T'$. Consider any legal triple $R = (\mathcal{L}, G, \gamma_0)$ for $T$ that can be obtained by "extending[1]" $R'$.

We claim that we can find a proper coloring $\gamma$ of $E(T)$ that agrees with $R$ and leaves $c$ available at the root of $T$. Given the claim, the statement follows by observing that every legal triple for $T$ can be obtained as an extension of some legal triple for $T'$.

The coloring $\gamma$ will agree with $\gamma'$ on all the edges in $E(G')$. As for the edge $e$ connecting the root of $T$ with $u$, we have at least two colors, for $T(u)$ is at least $(k+1)$-free and we used at most $k-1$ colors for the other edges incident to the root. Let $c' \neq c$ be one of these two colors. We can color $T(u)$ so that $c'$ is available at the root of $T(u)$, while having $\gamma(e) = c'$. Therefore $\gamma$ is a coloring satisfying all the conditions of the claim. □

**Observation 12.** *Any smallest $t$-free tree $T(r)$ has $\Delta - t$ children.*

*Proof.* If $T = T(r)$ has less than $\Delta - t$ children, then it is necessarily more than $t$-free, for less than $\Delta - t$ colors are blocked at its root.

To show that $\Delta - t$ is also an upper bound, let $r$ (the root of $T$) have $k \geq \Delta - t$ children. Let $T_1, \ldots, T_k$ be their corresponding subtrees.

By observation 11 and the minimality of $T$, it follows that each tree $T_i$ $(1 \leq i \leq k)$ is at most $k$-free.

We claim that $T$ is at most $(\Delta - k)$-free. We observe that since $k \geq \Delta - t$, the claim implies that it has to be $k = \Delta - t$, which concludes the proof.

Observe that it is sufficient to prove this claim in the case where every tree $T_i$ is exactly $k$-free. Indeed the degree of freedom of $T$ cannot increase if the degree of freedom of its children decreases. By lemma 10, choosing $L = \{c_1, c_2, \ldots, c_\Delta\}$, given any set $C \subseteq L$ of $k$ colors (say $C = \{c_1, c_2, \ldots, c_k\}$), for all $1 \leq i \leq k$, there exists a legal triple $R_i = (\mathcal{L}_i, G_i, \gamma_i)$ for $T_i$ such that, in every proper coloring of $T_i$ that agrees with $R_i$, the colors available at the root of $T_i$ are a subset

[1] More precisely, $\mathcal{L}$ and $\mathcal{L}'$ assign same lists of colors to edges in $E(T')$, $E(G') \subseteq E(G)$, and $\gamma_0(e) = \gamma_0'(e)$ for each $e \in E(G')$.

of $C$. Again by lemma 10, there exists a coloring $\gamma_i'$ of $T_i$ that leaves $c_i$ available at its root and agrees with $R_i$.

We observe that these augmentations $G_i$ of the $T_i$'s, taken together, constitute an augmentation for $T$ that forces every edge incident to $r$ to take a color from $C$. Formally, consider the triple $R = (\mathcal{L}, G, \gamma)$, where (a) $\mathcal{L}(e) = \mathcal{L}_i(e)$ if $e \in E(T_i)$; otherwise $\mathcal{L}(e) = L$; (b) $E(G) = E(T) \cup \bigcup_{i=1}^{k} E(G_i)$; (c) $\gamma(e) = \gamma_i'(e)$ if $e \in E(T_i)$, and $\gamma(e) = c_i$ if $e$ is the edge connecting the root $r$ to its $i$-th child. Note that $R$ is a legal triple for $T$. Moreover, in any proper coloring that agrees with $R$, every edge incident to $r$ takes a color from $C$. Since there are $k$ such edges and $|C| = k$, the set of colors of those edges must be $C$. So exactly $\Delta - k$ colors are available at the root of $T$. That is, $T$ is at most $(\Delta - k)$-free. □

The next definition is pivotal.

**Definition 13.** *Let $\mathcal{T}_t^h$ be the set of $t$-free $\Delta$-trees of height $h$. Also, let $n_t^h$ be the order of any smallest $t$-free tree in $\mathcal{T}_t^h$ (or $\infty$ if that set is empty).*

Recall our goal: we start with a list coloring of $\mathcal{G} - e$, $e = uv$, and grow two BFS trees $T(u)$ and $T(v)$ with the aim of showing that they can be re-colored in such a way that (a) the coloring in $\mathcal{G} - (T(u) \cup T(v))$ remains unchanged and (b) there is an available color for $e$. We will do this by showing that if the height of $T(u)$ and $T(v)$ is large enough then they both are at least $\left(\left\lceil \frac{\Delta}{2} \right\rceil + 1\right)$-free and hence there is at least one spare color for $e$ to complete the coloring. In what follows we will characterize precisely the minimum order of a tree of height $h$ that is $t$-free, i.e. $n_t^h$. We will then show that if we grow $T(u)$ and $T(v)$ at sufficient depth $\hat{h}$, their size will be less than the minimal size $n_t^{\hat{h}}$, for $t = 1, \ldots, \left\lceil \frac{\Delta}{2} \right\rceil$, and therefore their degree of freedom must be at least $\left\lceil \frac{\Delta}{2} \right\rceil + 1$, and this ensures the existence of an available color for $e$. The orders $n_t^h$ are pinned down in the next couple of lemmas by a double induction.

**Lemma 14.** *The following holds:*

*(i)* $n_1^0 = 1$ *and* $n_t^0 = \infty$ *for* $t \geq 2$*;*

*(ii)* $n_t^1 = \Delta - t + 1$*, for* $t \geq 1$*;*

*(iii)* $n_t^h = 1 + (\Delta - t) \min_{1 \leq i \leq \Delta - t} n_i^{h-1}$*, for* $t \geq 1, h \geq 2$*.*

*Proof.* For (i) it is sufficient to note that $\mathcal{T}_1^0$ contains only the tree composed of a single node. We can augment it with $\Delta - 1$ edges properly colored with $1, \ldots, \Delta - 1$; with $L = \{1, 2, \ldots, \Delta\}$ we obtain the 1-freedom of the tree. Also (ii) is trivial, if we observe that, for $t \geq 1$, $\mathcal{T}_t^1$ contains only one tree, the star with $\Delta - t$ edges. The endpoints of the star are "roots" of tree of height 0. At least $t$ colors are available at the root, regardless of how a legal triple for the star is chosen. Also, in the worst case, no more than $t$ colors can be available at the root because its list contains just $\Delta$ colors and, by lemma 10, the set of colors of the edges can be forced to be any set of $\Delta - t$ colors.

Observations 11-12 imply that every smallest tree in $\mathcal{T}_t^h$ (that is, one having order $n_t^h$) has to have a root with $\Delta - t$ children, each of which is at most $(\Delta - t)$-free. Thus, this tree consists of a root connected to $\Delta - t$ smallest trees in $\bigcup_{i=1}^{\Delta-t} \mathcal{T}_i^{h-1}$. Now (iii) follows. □

**Lemma 15.** *The following properties hold:*

- $\mathcal{O}$: *For odd* $h \geq 1$, $n_1^h = \frac{\Delta}{2} n_{\Delta-1}^h$ *and* $n_t^h = \left(n_{\Delta-1}^h - 1\right)(\Delta - t) + 1$, *for* $2 \leq t \leq \Delta - 1$;
- $\mathcal{O}'$: *For odd* $h \geq 3$, $n_2^h \geq n_3^h \geq \cdots \geq n_{\lceil \frac{\Delta}{2} \rceil - 1}^h \geq n_1^h \geq n_{\lceil \frac{\Delta}{2} \rceil}^h \geq \cdots \geq n_{\Delta-1}^h$;
- $\mathcal{E}$: *For even* $h \geq 2$, $n_{\Delta-1}^h \leq n_1^h \leq n_t^h$, *for* $2 \leq t \leq \Delta - 2$.

*Furthermore, for* $h \geq 1$, $n_{\Delta-1}^h = \min_{1 \leq t \leq \Delta-1} n_t^h$.

*Proof.* The three properties imply the minimality of $n_{\Delta-1}^h$, for $h \geq 1$. We show them by induction on $h$, starting with the base cases. For $h = 1$, the value of $n_{\Delta-1}^h$ and $\mathcal{O}$ follow from (ii) of lemma 14. For $h = 2$, (iii) and $\mathcal{O}$ imply that $n_t^2 = (\Delta - t) n_{\Delta-t}^1 + 1 = (\Delta - t)(t+1) + 1$. So the sequence $\{n_t^2\}_{t=1}^{\Delta-1}$ is bitonic: it starts by increasing and then decreases until the end. Thus to obtain $\mathcal{E}$ (which in turn implies the lemma for $h = 2$) it is sufficient to verify that $n_{\Delta-2}^2 \geq n_1^2 \geq n_{\Delta-1}^2$.

Now, assuming that for even $h - 1 \geq 2$ property $\mathcal{E}$ holds, we prove that property $\mathcal{O}$ holds for $h$. By (iii) and $\mathcal{E}$ we have that

$$n_t^h = 1 + (\Delta - t) \min_{1 \leq i \leq \Delta - t} n_i^{h-1} = \begin{cases} 1 + (\Delta - 1) n_{\Delta-1}^{h-1} & t = 1 \\ 1 + (\Delta - t) n_1^{h-1} & 2 \leq t \leq \Delta - 1 \end{cases}$$

This proves property $\mathcal{O}$ for $2 \leq t \leq \Delta - 1$. We consider $t = 1$ separately. By the equation above for $t = 1$ and (iii), we have that $n_1^h = 1 + (\Delta - 1) n_{\Delta-1}^{h-1} = 1 + (\Delta - 1)(n_1^{h-2} + 1)$.
Also, respectively by (iii), $\mathcal{E}$, $\mathcal{O}$ (that hold inductively), we get

$$n_{\Delta-1}^h = 1 + n_1^{h-1} = 2 + (\Delta - 1) \min_{1 \leq i \leq \Delta-1} n_i^{h-2} = 2 + (\Delta - 1) \frac{2}{\Delta} n_1^{h-2}$$

which is equivalent to $n_1^{h-2} = (n_{\Delta-1}^h - 2) \frac{\Delta}{2(\Delta-1)}$. By substituting this term in the previous equation we obtain $n_1^h = \frac{\Delta}{2} n_{\Delta-1}^h$. Thus property $\mathcal{O}$ is proved.

To prove $\mathcal{O}'$, we first note that the sequence $\{n_t^h\}_{t=2}^{\Delta-1}$ is decreasing by property $\mathcal{O}$. So it is sufficient to prove that $n^h_{\lceil\frac{\Delta}{2}\rceil} \leq n_1^h \leq n^h_{\lceil\frac{\Delta}{2}\rceil-1}$.

To prove $n_1^h \geq n^h_{\lceil\frac{\Delta}{2}\rceil}$ we apply $\mathcal{O}$ equations on both terms, to get the following equivalent inequality:

$$\frac{\Delta}{2} n^h_{\Delta-1} \geq (n^h_{\Delta-1}-1)\left\lfloor\frac{\Delta}{2}\right\rfloor + 1 \iff n^h_{\Delta-1}\left\{\frac{\Delta}{2}\right\} \geq 1 - \left\lfloor\frac{\Delta}{2}\right\rfloor$$

where $\{x\} = x - \lfloor x \rfloor$ is the fractional part of $x$. The LHS of the last inequality is always non-negative, while the RHS is non-positive for $\Delta \geq 3$. This proves $n^h_{\lceil\frac{\Delta}{2}\rceil} \leq n_1^h$.

For the other inequality, $n_1^h \leq n^h_{\lceil\frac{\Delta}{2}\rceil-1}$, we proceed in an analogous way

$$(n^h_{\Delta-1}-1)\left(\left\lfloor\frac{\Delta}{2}\right\rfloor+1\right)+1 \geq \frac{\Delta}{2}n^h_{\Delta-1} \Leftrightarrow n^h_{\Delta-1}\left(1-\left\{\frac{\Delta}{2}\right\}\right) \geq \left\lfloor\frac{\Delta}{2}\right\rfloor$$

The last inequality is implied by $\frac{1}{2}n^h_{\Delta-1} \geq \lfloor\frac{\Delta}{2}\rfloor$ which is in turn implied by $n^h_{\Delta-1} \geq \Delta$, which is true for any $h \geq 2$ just because $n_1^1 = \Delta$ by (ii) and $n_t^h > n_{t'}^{h'}$ for each $t, t'$ as long as $h > h'$, as it can be inferred from (iii).

It remains to prove that for all even $h \geq 4$ property $\mathcal{E}$ holds. Again, we assume that properties $\mathcal{O}$ and $\mathcal{O}'$ hold for $h-1$. For $1 \leq t \leq \lceil\frac{\Delta}{2}\rceil$, using respectively (iii), $\mathcal{O}'$ and $\mathcal{O}$ we obtain

$$n_t^h = 1 + (\Delta - t)\min_{1\leq i\leq \Delta-t} n_i^{h-1} = 1 + (\Delta-t)\left((n^{h-1}_{\Delta-1}-1)t+1\right).$$

The RHS is non-decreasing for $2t \leq \Delta - (n^{h-1}_{\Delta-1}-1)^{-1}$. This is implied by $t < \Delta/2$ assuming that $n^{h-1}_{\Delta-1} \geq 2$ (which holds for $h \geq 3$ as previously proved). Analogously, for $\lceil\frac{\Delta}{2}\rceil + 1 \leq t \leq \Delta - 1$, we obtain

$$n_t^h = 1 + (\Delta-t)\min_{1\leq i\leq\Delta-t} n_i^{h-1} = 1 + (\Delta-t)n_1^{h-1} = 1 + (\Delta-t)\frac{\Delta}{2}n^{h-1}_{\Delta-1}$$

which decreases in its range of $t$. To complete the proof of property $\mathcal{E}$, it remains to verify that $n^h_{\lceil\frac{\Delta}{2}\rceil} \geq n^h_{\lceil\frac{\Delta}{2}\rceil+1}$ and $n_1^h \geq n^h_{\Delta-1}$. Using the previous expressions, the former inequality is equivalent to

$$\left(\Delta - \left\lceil\frac{\Delta}{2}\right\rceil\right)\left((n^{h-1}_{\Delta-1}-1)\left\lceil\frac{\Delta}{2}\right\rceil+1\right) \geq \left(\Delta - \left(\left\lceil\frac{\Delta}{2}\right\rceil+1\right)\right)\frac{\Delta}{2}n^{h-1}_{\Delta-1}$$

which is implied by $n^{h-1}_{\Delta-1} \geq \Delta$ (already proven for $h \geq 3$). Again by the previous expressions, it is straightforward to check that $n_1^h \geq n^h_{\Delta-1}$ holds when $\Delta \geq 3$. □

**Lemma 16.** *A $\Delta$-tree $T$ of order $n$, $\Delta \geq 3$ and height $h \geq 2\log_{\Delta-1} n$ is at least $(\lceil \frac{\Delta}{2} \rceil + 1)$-free.*

*Proof.* The number $n$ of nodes of $T$ is at most $(\Delta - 1)^{\frac{h}{2}}$. Lemma 14 and 15 imply that

$$\min_{1 \leq i \leq \Delta-1} n_i^h = n_{\Delta-1}^h = \begin{cases} 2\frac{(\Delta-1)^{\lceil \frac{h}{2} \rceil}-1}{\Delta-2} + (\Delta-1)^{\lceil \frac{h}{2} \rceil} & h \text{ even} \\ 2\frac{(\Delta-1)^{\lceil \frac{h}{2} \rceil}-1}{\Delta-2} & h \text{ odd} \end{cases}$$

If $h$ is even then $n_{\Delta-1}^h > (\Delta - 1)^{\frac{h}{2}}$. For $h$ odd we have $n_{\Delta-1}^h > (\Delta - 1)^{\frac{h-1}{2}}$; thus, by $\mathcal{O}$,

$$n_{\lceil \frac{\Delta}{2} \rceil}^h > (\Delta - 1)^{\frac{h-1}{2}} \left\lfloor \frac{\Delta}{2} \right\rfloor + 1 > (\Delta - 1)^{\frac{h}{2}},$$

where the last inequality holds for $\Delta \geq 3$. Therefore, since $n \leq (\Delta - 1)^{\frac{h}{2}}$, if $h$ is even we have $n < n_{\Delta-1}^h = \min_{1 \leq i \leq \Delta-1} n_i^h$. It follows that $T$ has to be $\Delta$-free.

Analogously, if $h \geq 3$ is odd, $n < n_{\lceil \frac{\Delta}{2} \rceil}^h \leq n_1^h \leq n_{\lceil \frac{\Delta}{2} \rceil - 1}^h \leq \cdots \leq n_3^h \leq n_2^h$ by $\mathcal{O}'$, so $T$ is at least $(\lceil \frac{\Delta}{2} \rceil + 1)$-free. For the case $h = 1$, a similar argument holds by (ii). □

*Proof of Thm. 1.* We are given a graph $\mathcal{G}$ of maximum degree $\Delta \geq 3$ with girth $> 1 + 2R$ where $R = \lceil 2\log_{\Delta-1} n \rceil$ is the lower bound on the height of a tree stated in lemma 16. We are also given a $\Delta$-list assignment $\mathcal{L}$ for the edges of $\mathcal{G}$. We color the edges one at a time, starting with any edge and assigning it a color from its list. Assume by induction that $\mathcal{G} - e$ is colored, where $e = \{u, v\}$. Grow two BFS trees $T(u)$ and $T(v)$, respectively rooted at nodes $u$ and $v$, up to distance $R$ from their roots. From the girth assumption, these BFS' will be composed of disjoint trees. Consider the subtrees $T'(u)$ and $T'(v)$, respectively of $T(u)$ and $T(v)$, induced by the root-leaf paths of length exactly $R$. Both $T'(u)$ and $T'(v)$ are $\Delta$-trees of two possible heights: either $R$ or 0. Any tree having height $R$ is $(\lceil \frac{\Delta}{2} \rceil + 1)$-free by lemma 16, and any tree of height 0 has no edge to color. That is, there must exist a color $c \in \mathcal{L}(e)$, unused at both endpoints of $e$ that can be used to color $e$. After having colored the edges in $T'(u) \cup T'(v) \cup \{e\}$, re-coloring the rest of the edges in $T(u) \cup T(v) - (T'(u) \cup T'(v))$ is an easy matter (as they induce a forest of rooted trees having no constraints on their leaves). □

We conclude by explicitly stating that the re-coloring operations of the previous theorem can be performed in a local manner.

**Lemma 17.** *Let $\mathcal{G}$ be a graph of order $n$, maximum degree $\Delta \geq 3$ and girth greater than $2R + 1$ where $R = \lceil 2 \log_{\Delta-1} n \rceil$. Given any $\Delta$-list assignment for the edges of $\mathcal{G}$, any proper partial coloring of $\mathcal{G}$, and any uncolored edge $e = \{u, v\}$ of $\mathcal{G}$, it is possible to properly color $e$, by changing the colors of the already colored edges of the trees rooted at $u$ and $v$ having height $\leq R$.*

### 3.1 Class-2 graphs of high girth

We now sketch the proof of Prop. 4 which basically states that the girth requirement of Thm. 1 is within a factor of 3 of the optimal one. To this aim we give an infinite family of Class-2 graphs having girth $\geq \frac{4}{3} \log_{\Delta-1} n - O(1)$.

These graphs can be obtained by manipulating the graphs of high girth of [19]. We will give a function that maps regular graphs of order $n$ and girth $g$ to Class-2 graphs of odd order $2n - 1$ having girth $\geq g$. This will prove the proposition.

Let $\mathcal{G}$ be a $\Delta$-regular graph of order $n$ and let $\mathcal{G}'$ be the graph that $\mathcal{G}$ is mapped into; $\mathcal{G}'$ will either be a $\Delta$-regular graph, or a graph having $2n - 2$ nodes of degree $\Delta$ and a single node of degree $\Delta - 1$ — this implies that $\mathcal{G}'$ is "overfull" and, thus, belongs to Class-2 (a graph having $m$ edges, $n$ nodes and maximum degree $\Delta$ belongs to Class-2 if it is "overfull" — that is, if $m > \lfloor \frac{n}{2} \rfloor \Delta$).

To obtain $\mathcal{G}'$, delete any node $v$ from $\mathcal{G}$. The graph $\mathcal{G}'$ will consist of two disjoint copies of $\mathcal{G} - v$ and of a new node $v'$. The new edges are as follows. Take any subset of $\lceil \Delta/2 \rceil$ nodes of degree $\Delta - 1$ in a copy of $\mathcal{G} - v$ and connect each of them to its respective node in the other copy. Finally, connect node $v'$ to each of the remaining $2\lfloor \Delta/2 \rfloor$ nodes of degree $\Delta - 1$.

## 4 List vertex coloring

In this section we extend to list vertex colorings a result of Erdös [8] for vertex colorings. Our proof is quite similar to that in [8], even though algorithmic aspects are made more explicit. The main motivation to have it here, however, is the fact that it introduces a few ideas that will be used to extend to list total coloring the method we saw for list edge coloring.

Recall that our aim is to locally $k$ list-vertex-color a graph of order $n$, having girth $\geq 2\lceil \log_{k-1} n \rceil + 1$.

*Proof of Thm. 2.* Take a tree $T$ and assign to each vertex in $T$ a list of no less than $k \geq 3$ colors; also, consider any partial proper (w.r.t. the lists and adjacencies) coloring $\gamma$ of all the vertices of $T$ but the root. If there are at least two proper colorings of $T$ that 1) agree with $\gamma$ on the lowermost leaves and 2) give different colors to the root, we say that $T$ is lucky.

If $T$ has height 0, it is necessarily unlucky. If $T$ has height $h \geq 1$ then, to be unlucky, its root must have at least $k-1$ unlucky children. To see this, assume the contrary and start by coloring the unlucky children of the root. After that, at least two colors will be available at the root. Choose either one of them. All the remaining children of the root are lucky, thus at least a color in their lists will be different from the one of the root.

Therefore, if we denote by $n_h$ the minimum order of an unlucky tree of height $h$, we have that $n_0 = 1$, $n_i \geq (k-1)n_{i-1} + 1$. Thus, for $i \geq 1$, $n_i > (k-1)^i$. Now, if $T$ has $m \geq 1$ vertices and height $h \geq \lceil \log_{k-1} m \rceil$, then $T$ is necessarily lucky.

We turn our attention to the graph $\mathcal{G}$, a graph of order $n$, with girth $g(\mathcal{G}) \geq 2\lceil \log_{k-1} n \rceil + 1$, for $k \geq 3$. We will color the nodes of $\mathcal{G}$ inductively. We start from the empty coloring. The inductive step is to consider (a) any vertex $v$ and (b) the subgraph $\mathcal{G}'$ induced by $v$ and by the colored vertices at distance at most $\lceil \log_{k-1} n \rceil$ from $v$. $\mathcal{G}'$ is a forest. If $v$ is a singleton in $\mathcal{G}'$, it is trivial to color $v$. Otherwise take the component of $\mathcal{G}'$ containing $v$. By the previous claim, this component is a lucky tree — so the whole component, including $v$, can be properly re-colored. □

The following corollary can be inferred from the previous proof.

**Lemma 18.** *Let $\mathcal{G}$ be a graph of order $n$ and girth greater than $2R+1$, where $R = \lceil \log_{k-1} n \rceil$, with $k \geq 3$. Given any $k$-list assignment for the nodes of $\mathcal{G}$, any proper partial coloring of $\mathcal{G}$, and any uncolored node $v$ of $\mathcal{G}$, it is possible to properly color $v$ by changing the colors of the already colored nodes of the tree rooted at $v$ having height $\leq R$.*

# 5 List total coloring

In this section we prove Thm. 3 along the lines of the proof of Thm. 1. In this case we are not able to pin down the exact orders of the minimum non-re-colorable trees; instead, we have to resort to lower bounds. In the proof of (list) vertex coloring, we saw that the degree of freedom of a tree is not affected by a subtree whose root can be colored with at least two colors. This fact turns out to be useful for list total coloring. Given an edge $e = uv$ in a rooted tree, where $v$ is the child of $u$ in the tree, we will distinguish between colors for $e$ allowing just one color for $v$ (called single colors) and those allowing at least two colors for $v$ (double colors).

**Definition 19.** *Let $T(r)$ be a $\Delta$-tree. $T(r)$ is* at least $(s,d)$-free *if, for each list $L$ of $\Delta+1$ colors and each legal triple $(\mathcal{L}, G, \gamma)$, there exist a set $S \subseteq L$ of $s$ colors and a set $D \subseteq L - S$ of $d$ colors such that for each color $a \in S$ and for each color $b \in D$, there exist one coloring $\gamma_a$ and two colorings $\gamma_b, \gamma'_b$, such that*

- *$\gamma_a(q) = \gamma_b(q) = \gamma'_b(q) = \gamma(q)$, for all $q \in (E(G) \cup V(G)) - (E(T(r)) \cup V(T(r)))$,*
- *$\gamma_a(q), \gamma_b(q), \gamma'_b(q) \in \mathcal{L}(q)$, for all $q \in E(T(r)) \cup V(T(r))$,*
- *$\gamma_a$ does not use the color $a$ for any of the edges incident to $r$, and*
- *$\gamma_b, \gamma'_b$ do not use the color $b$ for any of the edges incident to $r$; also, $\gamma_b(r) \neq \gamma'_b(r)$.*

*Colors in $S$ and $D$ are said* single *and* double *colors, respectively.*

Now we prove the analogous of lemma 16 for list total coloring. This will imply Thm. 3.

**Lemma 20.** *Let $\Delta \neq 1, 2$. Then, if the height $h$ of a $\Delta$-tree $T$ of order $n$ is $h > 2\log_{\lceil \frac{\Delta}{2} \rceil} n$, then $T$ is $(0, \Delta+1)$-free.*

We start by proving the assertion for $\Delta \geq 4$.

*Proof ($\Delta \geq 4$).* We define a partial order between the degrees of freedom of the trees. The degrees of freedoms are pairs $(s,d)$ such that $1 \leq s+d \leq \Delta+1$. We define $(s,d) \preceq (s+t,d)$ for any $t \geq 0$ such that $s+t+d \leq \Delta+1$. Also, $(s,d) \preceq (s-t, d+t)$, for any $t \geq 0$ such that $s-t \geq 0$ and $d+t \leq \Delta+1$.

Take any $\Delta$-tree $T$ with freedom $(s,d)$, and let $T'$ be any one of its subtrees, whose degree of freedom we denote by $(t,e)$. Take any $T''$ tree with degree of freedom $(t',e')$ such that $(t',e') \succ (t,e)$. Then, substituting $T'$ with $T''$ in $T$ will change the degree of freedom of $T$ to $(s',d')$ with $(s',d') \succeq (s,d)$.

Let $\mathcal{T}_h$ be the set of $\Delta$-trees of height $h$. We define a tripartition $(F_h, S_h, C_h)$ of that set. $F_h$ contains just the $(0, \Delta+1)$-free trees; $S_h \subseteq \mathcal{T}_h - F_h$ contains trees of freedom greater than or equal $\left(2, \lfloor \frac{\Delta}{2} \rfloor\right)$. Finally $C_h = \mathcal{T}_h - (F_h \cup S_h)$. We note that a lower bound on the freedom of the trees in $C_h$ is $(1,0)$.

Let us define $n_h^f = \min_{T \in F_h} |T|, n_h^s = \min_{T \in S_h} |T|$ and $n_h^c = \min_{T \in C_h} |T|$. Let $k = \lceil \frac{\Delta}{2} \rceil$.

We start by showing that

$$n_{h+1}^c \geq k \min\{n_h^c, n_h^s\} \tag{1}$$

If no more than $k-1$ children in $S_h \cup C_h$ are connected to a root of some tree $T$, then at least 2 colors, $x, y$, will be available at the root of $T$ (as at most $2\,(k-1)$ would be blocked by its children from the $\Delta+1$ available in its list). Disregarding the color of the root node, the number of colors available in the list of a hypothetical edge added to the root would be at least equal to $\lfloor \frac{\Delta}{2} \rfloor + 2$ (as at most $k-1$ colors will be blocked by the other edges incident to the root). If we exclude from this palette the colors $x, y$, we have that all remaining colors will be double (as choosing any of them for the edge, still leave $x$ and $y$ for coloring the node). Any color out of $\{x, y\}$ would still be usable for the edge (at least) as a single color. Thus, under the assumption that the root of $T$ has less than $k-1$ children out of $S_h \cup C_h$, the freedom of $T$ is at least $\left(2, \lfloor \frac{\Delta}{2} \rfloor\right)$; that is $T \notin C_{h+1}$.

No minimum-order $\Delta$-tree has some $(0, \Delta+1)$ subtree. Therefore, each tree in $C_h$ must have an order satisfying eqn. 1, otherwise it would not be able to contain a sufficient number of subtrees in $S_h \cup C_h$.

We now show that

$$n^s_{h+1} \geq \min\{n^c_h, k\; n^s_h\} \tag{2}$$

Suppose that a tree $T$ in $S_h$ has an order strictly less than the latter. Then the root of $T$ would have, as children, no subtrees in $C_h$ and at most $k-1$ subtrees out of $S_h$. We will show that $T \in F_h$, thus obtaining a contradiction.

Add a hypothetical edge to the root of $T$ and color it arbitrarily. At most 1 color will be deleted by the lists of the root and of its other incident edges. Each of these at most $k-1$ edges starts with a freedom of at least $\left(2, \lfloor \frac{\Delta}{2} \rfloor\right)$. Thus, after coloring the new edge, at least $\lfloor \frac{\Delta}{2} \rfloor - 1$ different double-colors will be available at each edge. Thus at least so many edges can be double-colored.

If $\Delta$ is even, all the other edges can be double colored. Thus the number of colors blocked at the root is at most $k = \frac{\Delta}{2}$; that is, the available to the root are at least $\frac{\Delta}{2} + 1 \geq 2$; so the color chosen for the new edge is a double color for the root — given the arbitrariness of that color, we get the contradiction $T \in F_h$.

If $\Delta$ is odd, at most one of the other edges will not be doubly colorable. Assign it a single color. The number of colors blocked at the root will thus be $k+1$ (the color of the new edge, the colors of the $k-2$ double-colorable siblings, and the possible two colors blocked by the last edge and its vertex). The number of colors available for the root will be at least $(\Delta+1) - \left(\lceil \frac{\Delta}{2} \rceil + 1\right) = \lfloor \frac{\Delta}{2} \rfloor$. If $\Delta \geq 4$, the latter will be at least two. So, again, we get $T \in F_h$ that contradicts the hypothesis.

For $h = 0$ we have that $n^c_0 = 1$ and $n^s_0 = \infty$ (each $\Delta$-tree of zero height is $(1, 0)$-free by definition). Let us define $m^c_0 = n^c_0$ and $m^s_0 = n^s_0$. For $h \geq 1$, let $m^c_h = k \min\{m^c_{h-1}, m^s_{h-1}\}$ and $m^s_h =$

| | $C_h$ | $S'_h$ | $A_h$ | $F_h$ |
|---|---|---|---|---|
| $C_h$ | $C_{h+1}$ | $C_{h+1}$ | $C_{h+1}$ | $S'_{h+1}$ |
| $S'_h$ | $C_{h+1}$ | $C_{h+1}$ | $A_{h+1}$ | $A_{h+1}$ |
| $A_h$ | $C_{h+1}$ | $A_{h+1}$ | $F_{h+1}$ | $F_{h+1}$ |
| $F_h$ | $S'_{h+1}$ | $A_{h+1}$ | $F_{h+1}$ | $F_{h+1}$ |

Table 1: The degree of freedom of a tree, given the degrees of freedom of its children.

$\min\{m^c_{h-1}, k\ m^s_{h-1}\}$; we have that $n^c_h \geq m^c_h$ and $n^s_h \geq m^s_h$, for all $h$.

It follows by induction that, for odd $h$, we have that $m^c_h = km^s_h$ while, for even $h \geq 2$, we have that $m^c_h = m^s_h$. These give, for each $h \geq 1$, $m^c_h \geq m^s_h = k^{\lfloor \frac{h}{2} \rfloor} = \lceil \frac{\Delta}{2} \rceil^{\lfloor \frac{h}{2} \rfloor}$.

Now, take a $\Delta$-tree $T$ of order $n$ and height $h$. For $T$ not to be in $F_h$, we must have

$$n \geq \left\lceil \frac{\Delta}{2} \right\rceil^{\lfloor \frac{h}{2} \rfloor} \qquad \Longleftrightarrow \qquad \left\lfloor \frac{h}{2} \right\rfloor \leq \log_{\lceil \frac{\Delta}{2} \rceil} n$$

which is impossible under the assumptions of the lemma. □

We now prove theorem 20 for $\Delta = 3$.

*Proof ($\Delta = 3$).* We partition $\mathcal{T}_h$ into four set $F_h, A_h, S'_h, C_h$, each representing different degrees of freedom. As in the proof of $\Delta \geq 4$, $F_h$ will contain just the $(0, \Delta+1)$-free trees. The new set $A_h \subseteq \mathcal{T}_h - F_h$ will contain the trees of freedom $(1, \Delta)$. Finally, $S'_h \subseteq \mathcal{T}_h - (F_h \cup A_h)$ will contain the trees of freedom at least $(2, \lfloor \frac{\Delta}{2} \rfloor)$ and $C_h = \mathcal{T}_h - (F_h \cup A_h \cup S'_h)$.

Table 1 determines which set a tree $T$ in $\mathcal{T}_{h+1}$ falls into, given the set memberships of its two subtrees $T'$ and $T''$ in $\mathcal{T}_h$ (a missing subtree is considered to have freedom $F_h$). We start by showing the validity of table 1.

Let $u, u', u''$ be the roots of $T, T', T''$ respectively, and let $e' = (u, u'), e'' = (u, u'')$. Also, let $e = (\cdot, u)$ be the hypothetical edge above $T$. In our notation, $c \to c'$ denote that a single color $c$ for an edge forces the color $c'$ on the lower node of that edge.

The following three properties bound from above the degree of freedom of $T$. For these properties, we assume that the lists of $e', e'', e$ and $u, u', u''$ are the set of colors $\{1, 2, 3, 4\}$.

**There exist two trees $T', T'' \in S'_h$ such that $T$ is in $C_{h+1}$.** Let 1 be the double color of both $e'$ and $e''$, and let $2 \to 3, 3 \to 2$ be their single colors. It is straightforward to verify the $T$ is $(4,0)$-free and thus $T \in C_{h+1}$.

**There exist a tree $T' \in C_h$ and a tree $T'' \in A_h$ such that $T$ is in $C_{h+1}$.** Let $e'$ be $(1,0)$-free with $1 \to 2$ as its single color. Also, let $2 \to 4$ be the single color of $e''$. The edge $e$ cannot be colored of 1. We prove by contradiction that 3 cannot be used on $e$. Suppose that $e$ is colored of 3; then, since the colors of $e'$ and $u'$ are respectively 1 and 2, the only available color for $u$ would be 4. This implies that no colors are available for $e''$. Thus, $T$ is at most $(0,2)$-free.

**There exist a tree $T' \in S'_h$ and a tree $T'' \in F_h$ such that $T$ is in $A_h$.** Let $T$ have 1 as its double color and $2 \to 3, 3 \to 2$ as its single ones. The color 1 is not a double color for $T$ as, if $e$ was colored of 1, then each proper coloring of $e'$ and $v'$ will leave just the color 4 for the node $v$. Thus the assertion is proved.

The remaining three properties give lower bounds on the degree of freedom of $T$.

**For any $T', T'' \in A_h$, it holds that $T$ is in $F_{h+1}$.** Choose any color $c \in \mathcal{L}(e)$ for $e$ and let $c'$ and $c''$ be the single colors of $T'$ and $T''$ respectively.

Suppose first that $c' = c''$. Then, if $c = c'$, we can choose for $u$ any color $c_u \in \mathcal{L}(u) - \{c\}$ leaving at least two double colors for both $e'$ and $e''$. Now assume $c \neq c'$. If each proper combination of double colors for $e'$ and $e''$ leaves the same color $c_u$ to $u$, then necessarily (1) the double colors for $e'$ and $e''$ are the same, (2) they are all in the list of $u$, and (3) $c$ is one of them. In such a case, there are two colors $c_{d1}, c_{d2} \in \mathcal{L}(u)$ that re double for $e'$ and different from $c$: now, if $c' \to c'_s$ is the single color of $e'$, choose in $\{c_{d1}, c_{d2}\}$ a color for $u$ different from $c'_s$, color $e'$ by $c'$ and use a double color for $e''$. This gives one possible color to $u$. To obtain another, give to $e', e''$ the two colors $c_{d1}, c_{d2}$, and select the remaining color in $\mathcal{L}(u)$ for $u$.

Consider now $c' \neq c''$. Then, for the case $c = c'$, we can choose for $u$ any color $c_u \in \mathcal{L}(u) - \{c'\}$, for $e''$ any double color $c_d \in \mathcal{L}(e'') - \{c', c'', c_u\}$ and for $e'$ any double color in $\mathcal{L}(e') - \{c', c_d, c_u\}$ (recall that $c'$ is single for $e'$). The case $c = c''$ is analogous. If $c \notin \{c', c''\}$, start by considering the case that the double colors $\mathcal{L}(e') - \{c, c'\}$ for $e'$ are different from the double colors $\mathcal{L}(e'') - \{c, c''\}$ for $e''$; then, after choosing any color $c_u \in \mathcal{L}(u) - \{c\}$ for $u$, at least two double colors can be chosen for $e'$ and $e''$.

The remaining case is $c' \neq c''$ and $c \notin \{c', c''\}$ with $\mathcal{L}(e') - \{c, c'\} = \mathcal{L}(e'') - \{c, c''\} = \{c_1, c_2\}$; coloring $e', e''$ of $c_1, c_2$ leaves for $u$ at least one color $\mathcal{L}(u) - \{c, c_1, c_2\}$. Now, if there is exactly one such color, say $c^*$, we have $\mathcal{L}(u) = \{c, c_1, c_2, c^*\}$. We want to find another coloring that allows a color different from $c^*$ for $u$. Since $c' \neq c''$ we have that at least one among $c^* \neq c'$ and $c^* \neq c''$ holds. Assuming w.l.o.g $c^* \neq c'$, we can color $e'$ by the single color $c' \to c_s$, leaving for $u$ the colors $\{c_1, c_2, c^*\} - \{c_s\}$. Now, coloring $e''$ by either $c_1$ or $c_2$ leaves to $u$ a color different from $c^*$.

**For any $T' \in C_h$ and $T'' \in F_h$, it holds that $T$ is in $S'_h$.** Let $c' \to c'_s$ be a single color of $e'$. We start by assigning $c'$ to $e'$.

Notice that any color $c \in \mathcal{L}(e) - \{c'\}$ is at least a single color for $e$. That is because a partial coloring assigning $c$ to $e$ can be completed by choosing a color $c_u \in \mathcal{L}(u) - \{c', c, c'_s\}$ for $u$ and a double color in $\mathcal{L}(e'') - \{c', c, c_u\}$ for $e''$.

Also notice that any proper coloring of $u, e, e'$ leaves at least a (double) color for $e''$. Thus we can ignore the coloring of $e''$ for the rest of the proof.

Now, if $c'_s \in \mathcal{L}(e)$, $c'_s$ is a double color for $e$ since all the colors in $\mathcal{L}(u) - \{c'_s, c'\}$ would be available for $u$ after having colored $e$ of $c'_s$. Otherwise, $c'_s \notin \mathcal{L}(e)$. First, consider the case $c'_s \in \mathcal{L}(u)$: take the set $(\mathcal{L}(e) - \{c'\}) - \mathcal{L}(u)$. If it is empty then necessarily $c' \notin \mathcal{L}(u)$ and we can choose the color $c' \to c'_s$ for $e'$, a color $c^*$ different from $c'$ for $e$, thus leaving at least two colors in $\mathcal{L}(u) - \{c'_s, c^*\}$. If the set is not empty, choosing a color $c^* \in (\mathcal{L}(e) - \{c'\}) - \mathcal{L}(u)$ for $e$ allows any color in $\mathcal{L}(u) - \{c', c'_s\}$ for $u$ — thus, $c^*$ is a double color for $e$. The remaining case is $c'_s \notin \mathcal{L}(e) \cup \mathcal{L}(u)$. Here, any color $c \in \mathcal{L}(e) - \{c'\}$ is a double color for $e$, as it leaves all colors in $\mathcal{L}(u) - \{c, c'\}$ available for $u$.

**For any $T' \in S_h$ and $T'' \in A_h$, it holds that $T$ is in $A_h \cup F_h$.** Let $c'_d$ and $c'_1 \to c'_{s1}$, $c'_2 \to c'_{s2}$ be, respectively, the double color and the single colors of $T'$. Let $c'' \to c''_s$ be the single color of $T''$.

Take any color $c \in \mathcal{L}(e)$. We show that if $e$ is colored by $c$ then the coloring of $e, e', e'', u$ can be completed, that is $c$ is either a single or a double color of $e$.

Suppose that $c \neq c'_d$, then color $e'$ by $c'_d$. At least two colors will be available at $u$. For at least one of this colors there will exist a double color for $e''$ that will allow the completion of the coloring, as just one of the colors in $\mathcal{L}(e'')$ is single.

Otherwise, color $e$ with $c = c'_d$. Choose the single color $c'_1$ for $e'$ and one of the remaining colors for $u$. In parallel, choose $c'_2$ for $e'$ and one of the remaining colors for $u$. If the sets of colors of $e, e', u$ are different in these two partial colorings, then a double color for $e''$ can be used in at least one of the two cases; otherwise, if the color of $u$ is the same in both cases then $\{c'_1, c'_{s1}\} = \{c'_2, c'_{s2}\}$ and, for at least one color $c'_i$ for $e'$, at least a double color will be available for $e''$. If the color of $u$ changes, then either a double color of $e''$ can be used to complete both colorings, or the its single color can be used as $c''_s$ won't conflict with the color of $u$ in at least one of the cases.

If $c'_d \in \mathcal{L}(e)$ and $c'_d$ is the only single color of $e$ we are done. Otherwise take any color $\hat{c} \in \mathcal{L}(e) - \{c'_d\}$; we show that if $\hat{c}$ is a single color then all the others are double.

Color $e$ with $\hat{c}$. If we color $e'$ of $c'_d$, there must exists a color $c_u \in \mathcal{L}(u) - \{\hat{c}, c'_d\}$ such that a color in $\mathcal{L}(e'') - \{\hat{c}, c'_d, c_u\}$ is double for $e''$ and we can complete the coloring. If just one such color exists then necessarily $\mathcal{L}(u) = \{\hat{c}, c'_d, c_u, \bar{c}_u\}$, for some $\bar{c}_u$ (that is it contains only one available color $\bar{c}_u$ other than $c_u$). Try to color $u$ of $\bar{c}_u$, and if the resulting partial coloring can not be extended to $e''$, not even with its single color $c''$, then necessarily $c''_s = \bar{c}_u$ and $\mathcal{L}(e'') = \{\hat{c}, c'_d, c''_s, c''\}$. Let $\mathcal{A}$ be the condition that $\mathcal{L}(e'') = \{\hat{c}, c'_d, c''_s, c''\}$, $\mathcal{L}(u) = \{\hat{c}, c'_d, c_u, \bar{c}_u\}$ and $c''_s = \bar{c}_u$.

Suppose that another single color $\tilde{c} \in \mathcal{L}(e) - \{c'_d\}$, $\tilde{c} \neq \hat{c}$, exists. Then a contradiction is implied by $\mathcal{L}(e'') = \{\hat{c}, c'_d, c''_s, c''\} = \{\tilde{c}, c'_d, c''_s, c''\}$ as $|\mathcal{L}(e'')| = 4$.

It remains to be proved that, if $c'_d \in \mathcal{L}(e)$ and $\hat{c}$ is single, then $c'_d$ is a double color for $e$. To do so, we first prove another condition implied by $\hat{c}$ being a single color.

Let $\mathcal{B}$ be the condition $\{c'_1, c'_{s1}\} = \{c'_d, c''_s\}$ or $\{c'_2, c'_{s2}\} = \{c'_d, c''_s\}$. We show that if $\hat{c}$ is single then $\mathcal{B}$ holds.

For suppose it does not. Recall that if $e$ is colored by $\hat{c}$ then there exists a proper coloring giving color $c_u$ to the node $u$. We want to find some other coloring that assigns a color different from $c_u$ to $u$ to show that $\hat{c}$ is not a single color.

Notice that either $\hat{c} \neq c'_1$ or $\hat{c} \neq c'_2$, as $c'_1 \neq c'_2$. Assume w.l.o.g. $\hat{c} \neq c'_1$.

Color $e'$ of $c'_1$. By $\mathcal{A}$ and not $\mathcal{B}$, we can choose for $u$ a color $c^* \in \mathcal{L}(u) - \{\hat{c}, c'_1, c'_{s1}\}$ different from $c_u$, since $\mathcal{L}(u) - \{\hat{c}, c'_1, c'_{s1}\} = \{c'_d, c''_s\} - \{c'_1, c'_{s1}\}$ is a non-empty set. Now, assume $c^* = c''_s$. Since $\mathcal{L}(e'') = \{\hat{c}, c'_d, c''_s, c''\}$ and $c'_1 \neq c'_d$, the double color $c'_d$ can be used on $e''$ to complete the coloring. Otherwise, $c^* = c'_d$. By $c'_d \neq c''_s$, we can use the single color $c''$ on $e''$ to complete the coloring. The same argument holds when $\hat{c} \neq c'_2$.

Thus, if $\hat{c}$ is single both $\mathcal{A}$ and $\mathcal{B}$ hold.

We now show that if $c'_d \in \mathcal{L}(e)$ and $\mathcal{A}, \mathcal{B}$ hold, then $c'_d$ is a double color for $e$. Color $e$ by $c'_d$. By $\mathcal{B}$, either $\{c'_1, c'_{s1}\} = \{c'_d, c''_s\}$ or $\{c'_2, c'_{s2}\} = \{c'_d, c''_s\}$; w.l.o.g. let the first equality be true. Then $c'_d = c'_{s1}$ and $c'_1 = c''_s$, by $c'_d \neq c'_1$.

Color $e'$ by $c'_1$. At least two colors will be available for $u$, and both will be different from $c''_s$. Thus the coloring can be completed.

The previous six properties, together with the ordering properties of the degrees of freedom, are enough to imply the table. By that, we get

- $n^c_{h+1} \geq 1 + \min\{2\min\{n^c_h, n^s_h\}, n^c_h + n^a_h\}$,

- $n^s_{h+1} \geq 1 + n^c_h$,
- $n^a_{h+1} \geq 1 + n^s_h$.

Let, respectively, $m^c_{h+1}, m^s_{h+1}, m^a_{h+1}$ be the RHS of these three inequalities. We have that $n^c_0 = 1$ (as in the proof for $\Delta \geq 4$) and $n^s_0 = n^a_0 = n^a_1 = \infty$ (all 0-height trees are in $C_0$ and no 1-height tree is in $A_1$). Let us set, conventionally, $m^c_0 = 1$ and $m^s_0 = m^a_0 = m^a_1 = \infty$. We define the rest of the recursion as

$$
\begin{aligned}
m^c_{h+1} &= 1 + \min\{2\min\{m^c_h, m^s_h\}, m^c_h + m^a_h\} \\
m^s_{h+1} &= 1 + m^c_h \\
m^a_{h+1} &= 1 + m^s_h
\end{aligned}
$$

Now, for all $h$, it holds that $m^c_h \leq n^c_h, m^s_h \leq n^s_h$ and $m^a_h \leq n^a_h$. Note that, $m^a_{h+2} = 2 + m^c_h$ and $m^c_{h+3} = \min\{2m^c_{h+2} + 1, 2m^c_{h+1} + 3, m^c_{h+2} + m^c_h + 3\}$.

We want to show that $m^c_h = \alpha_h 2^{\lfloor \frac{h}{2} \rfloor + 1} - 3$ where $\alpha_h = 2$ (resp., $\alpha_h = 3$) for even (odd) $h$. The base cases $h = 0, 1, 2$ are trivial ($m^c_0 = 1$ by definition, $m^c_1 = 2m^c_0 + 1$ and $m^c_2 = 2m^s_1 + 1$, where $m^s_1 = 1 + m^c_0$). Suppose $h \geq 3$, we have that

$$
\begin{aligned}
m^c_h &= \min\{2m^c_{h-1} + 1, 2m^c_{h-2} + 3, m^c_{h-1} + m^c_{h-3} + 3\} = \\
&= \min\{2\alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor + 1} - 5, 2\alpha_{h-2}2^{\lfloor \frac{h-2}{2} \rfloor + 1} - 3, \\
&\qquad \alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor + 1} + \alpha_{h-3}2^{\lfloor \frac{h-3}{2} \rfloor + 1} - 3\} = \\
&= \min\{4\alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor} - 5, 2\alpha_{h-2}2^{\lfloor \frac{h}{2} \rfloor} - 3, 3\alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor} - 3\} = \\
&= \min\{2\alpha_{h-2}2^{\lfloor \frac{h}{2} \rfloor} - 3, 3\alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor} - 3\}
\end{aligned}
$$

where the last equality comes from the first argument of the minimum function being greater than the last for $h \geq 3$, as $\alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor} \geq 4$.

Suppose $h$ is even. Then $m^c_h = \min\{2^{\frac{h}{2}+2} - 3, 9 \cdot 2^{\frac{h}{2}-1} - 3\} = 2^{\frac{h}{2}+2} - 3 = \alpha_h 2^{\lfloor \frac{h}{2} \rfloor + 1} - 3$. If $h$ is odd, the two arguments of the minimum function are equal and $m^c_h = 3 \cdot 2^{\frac{h+1}{2}} - 3 = \alpha_h 2^{\lfloor \frac{h}{2} \rfloor + 1} - 3$. Thus the induction is proved.

The proof can be concluded as the one for $\Delta \geq 4$, by showing that $m^s_h = \alpha_{h-1}2^{\lfloor \frac{h-1}{2} \rfloor + 1} - 2$, $m^a_h = \alpha_h 2^{\lfloor \frac{h-2}{2} \rfloor + 1} - 1$ and $m^a_h \leq m^s_h \leq m^c_h$.□

Even for total coloring, the re-coloring operations can be performed locally.

**Lemma 21.** *Let $\mathcal{G}$ be a graph of order $n$, maximum degree $\Delta \geq 3$ and girth greater than $2R+1$, where $R = \lceil 2\log_{\lceil \frac{\Delta}{2} \rceil} n \rceil + 1$. Given any $(\Delta+1)$-list assignment for edges and nodes of $\mathcal{G}$, any proper partial coloring of $\mathcal{G}$, and any uncolored edge $e = \{u, v\}$ of $\mathcal{G}$, it is possible to properly color $e$, $u$ and $v$, by changing the colors of the already colored nodes and edges of the trees rooted at $u$ and $v$ having height $\leq R$.*

# 6 Algorithms

In this section we sketch some algorithmic consequences of our theorems on list colorings, with special emphasis on distributed algorithms that are our main focus. In particular we will give the proof of Thm. 5.

We consider the classic model suggested by Linial [18] of a synchronous, message-passing distributed network. The running time of an algorithm is given by the number communication rounds it runs for. In each round a processor can broadcast a message to all of its neighbors, receive messages from all of them, and perform any amount of local computation. An algorithm is *efficient* if its running time is at most poly-logarithmic in the network size.

All list colorings can be computed efficiently in a distributed setting by means of some sort of meta-algorithm. We will see that this algorithm can be implemented in such a way that each processor actually performs locally only a polynomial amount of computation. Thus the distributed algorithm can also be simulated sequentially in polynomial time. The meta-algorithm operates on a conflict graph $C$ of the input graph $\mathcal{G}$. For vertex coloring $C = \mathcal{G}$; for edge coloring, $C$ is the line graph of $\mathcal{G}$; for total coloring, $C$ is the so-called total graph of $\mathcal{G}$, i.e. there is a node in $C$ for every edge or vertex in $\mathcal{G}$ and two nodes are adjacent in $C$ if their respective elements are adjacent, or incident, in $\mathcal{G}$. The meta-algorithm produces a node coloring[2] of $C$, regardless of the requested type of coloring.

The main idea is that the local nature of list colorings illustrated in the previous sections allows to color several nodes of $C$ in parallel. For these operations not to interfere, it is sufficient to ensure that the corresponding re-coloring trees do not overlap. This can be achieved by selecting, in each phase of the algorithm, the nodes to re-color according to a network decomposition of a power of $C$. Recall that an $(\alpha, \beta)$-decomposition of a graph is a partition of its nodes into $\beta$-weakly-connected components, each labeled with an integer in $\{1, \ldots, \alpha\}$, such that two adjacent components have different labels (two components are adjacent if at least one edge hits both of them). A $\beta$-weakly-connected component of a graph is a subset of its nodes such that any two nodes of the subset are at distance at most $\beta$ in the graph. For the sake of brevity, we use the word *cluster* to refer to a $\beta$-weakly-connected component.

It is known that, in the distributed model of computation we are considering, $(O(\log n), O(\log n))$-decompositions can be computed in $O(\log n)$ many rounds by a randomized distributed algorithm and in $2^{O(\sqrt{\log n})}$ many rounds by a deterministic one [17, 23].

[2]We will use the term *node coloring* to distinguish the coloring of the nodes of $C$ from the vertex (or edge, or total) coloring of $\mathcal{G}$.

Each of the lemmas 17, 18 and 21 gives a bound on the depth of the trees to be considered for the re-coloring operations. Let $d \in O(\log n)$ be this upper bound. To ensure non-interference between different operations, we compute a decomposition of $C^t$, for $t = 2d + O(1)$, where $C^t$, the $t$-th power of $C$, is the graph with $V(C^t) = V(C)$ and where two nodes are connected if they are at distance at most $t$ in $C$.

In each cluster, the re-coloring operations will all be handled by just one of its nodes. This node (or *leader*) can be chosen as the one having the smallest ID of the cluster. The re-coloring operations of equally-labeled clusters can be performed in parallel, as their distance in $C$ is greater than or equal $t$.

A description of the meta-algorithm for a graph $\mathcal{G}$ follows:

1. Compute an $(\alpha, \beta)$-decomposition of the $t$-th power $C^t$ of the conflict graph $C$ of $\mathcal{G}$, where $t = 2d + O(1)$.
2. For each label $k \in \{1, 2, \ldots, \alpha\}$:

   In each $k$-labeled cluster in parallel:

   (a) Elect a leader $u$ for the cluster (e.g. the node with the smallest ID in the cluster)

   (b) The leader $u$ gathers all the information (structure and coloring) of the subgraph of $C^t$ induced by the nodes in the cluster, and of its neighbors.

   (c) *Locally*, for each node in the cluster (corresponding to either a vertex or an edge in $\mathcal{G}$) the leader $u$ colors that node, possibly recoloring the other nodes in the cluster, while guaranteeing that no conflicts arise in the node coloring of $C$. That is, the cluster, and the nodes at distance one from it, must be properly node-colored after this step. (Observe that the nodes at distance one from the cluster are not re-colored by $u$.) This can be done for each of the three kinds of coloring of $\mathcal{G}$, edge, vertex and total, thanks to lemmas 17, 18 and 21.

   (d) The leader $u$ broadcasts the new colors to the nodes in the cluster.

We observe that, for a given cluster, steps (a),(b), and (d) can be performed in $O(\beta t)$ rounds, while (c) is performed locally (only one round needed). Therefore, this meta-algorithm runs for $O(T + \alpha\beta t)$ communication rounds, where $T$ is the time needed to compute the $(\alpha, \beta)$-decomposition of $C^t$. Using the randomized algorithm in [17], we have $\alpha = \beta = O(\log n)$ and $T = O(t \ \log n) = O(\log^2 n)$; in the deterministic case, via [23] we obtain $\alpha = \beta = O(\log n)$ and $T = O(t\ 2^{O(\sqrt{\log n})})$. Thus, Thm. 5 is proved.

So far, we have neglected to discuss how a leader can (locally) compute the re-coloring of a tree in the cluster. In Linial's model, each node of the network is allowed to perform an unlimited amount of computation, thus the tree re-colorings (the most complex operations performed by the leaders) could just be obtained by exhaustive search.

Nonetheless, as the following theorem states, tree re-colorings can be computed in polynomial time. The theorem directly implies that complete colorings can be computed in polynomial time in a sequential setting.

**Proposition 22.** *The re-coloring of a tree can be computed in time $O(n \cdot \Delta^{7/2})$ for list edge coloring, in time $O(n \cdot \Delta^{9/2})$ for list total coloring and in time $O(n \cdot k)$ for list vertex coloring.*

*Proof.* The re-coloring of a tree is performed in two phases: first the tree is visited bottom-up and then top-down.

For the case of list edge coloring, we say that a color $c$ is available for $e = uv$ ($u$ being the node nearer to the root) if (a) $c$ is in $e$'s list of colors and (b) the tree rooted at $v$ can be re-colored in such a way that $c$ is not used in any of the edges incident to $v$.

The algorithm $A$ we will describe decides, given a color $c$, an edge $e$ and its children's list of available colors, whether $c$ is available for $e$, and – in that case – obtains a good coloring of $e$'s children that allow $e$ to be $c$-colored. Let $T(A)$ be the maximum running time of $A$. Then one can re-color an entire tree in time $O(n\Delta\ T(A))$ in the following way: starting from the leaves, compute the list of available colors for all edges (that is, for each edge $e$ of the tree, execute $A$ once for each color in $e$'s list). Fix the color for the edge missing at the root, and re-color the tree using the partial colorings given by the various executions of $A$.

The description of $A$ follows. Consider a bipartite graph $B$ having on one side the union of the lists of colors of $e$ and its children, and on the other the children of $e$. A child is connected to a color iff the color is in that child's list.

Suppose that the algorithm were to decide whether $c$ is available to $e$. The algorithm would delete the node $c$ from $B$ and compute a maximum matching in $B - c$. The color $c$ would then be available to $e$ iff all its children were hit by the maximum matching. If that was the case, the maximum matching would also induce a coloring for the children leaving the color $c$ available to the edge $e$.

Using the maximum bipartite matching algorithm in [11], $c$'s availability for $e$ can be decided in time $T(A) \le O(\Delta^{5/2})$.

A similar algorithm works for list total coloring. Take an edge $e = uv$ ($u$ being the node nearer to the root of the tree). In this case, we need to obtain the lists of single and double colors of $e$ (see def. 19 and the proofs of lemma 20). Recall that a single color $s$ for $e$, if used, fixes uniquely the color $c_s$ of $v$ — we say that $s$ forces $c_s$ for $e$. Let us again construct the bipartite graph $B$ having the union of the lists of colors of $e$ and of its children on one side, and the set of $e$'s children on the other. Edges will connect children to available colors.

Given a color $c$ in the list of $e$, and a color $c' \neq c$ in the list of $v$, delete $c$ and $c'$ from $B$. Further, delete each edge corresponding to single colors that force $c'$. There will exist a maximum matching in the resulting graph iff $e$ and $v$ can be, respectively, $c$ and $c'$ colored.

Having fixed a color $c$ for $e$ the algorithm can count the number of colors for $v$ that can be extended to a proper coloring. If the number of these colors is 0, then $c$ is not available for $e$. Otherwise, $c$ is single or double depending on whether the number of the colors for $v$ is equal to, or greater than, 1.

The number of maximum bipartite matching to be computed per node is $O(\Delta^2)$, thus the complexity of the whole tree re-coloring operation follows.

As for list vertex coloring, note that a color $c$ in the list of a vertex $v$, is available to $v$ iff none of the children of $v$ has $c$ as the only available color. This observation directly leads to an algorithm having the stated time complexity. □

# Acknowledgments

We greatly thank Alessandro Panconesi for his guidance and for his constant encouragement, advice and support. We also thank Zoltán Füredi, Alexandr V. Kostochka, Ravi Kumar, Mike Molloy, Romeo Rizzi and Aravind Srinivasan for useful comments and tips.

# References

[1] M. Behzad, "The total chromatic number", Combinatorial Math. and its Applications (Proc. Conf., Oxford 1969), Academic Press, London, 1971.

[2] B. Bollobás, A. J. Harris, "List colorings of graphs", Graphs and Combinatorics 1 (1985), 115-127.

[3] O. V. Borodin, A. V. Kostochka, D. R. Woodall, "List edge and list total colourings of multigraphs", J. Combinatorial Theory, Series B 71 (1997), 184-204.

[4] F. Chierichetti, A. Vattani, "The Local Nature of List Colorings for Graphs of High Girth", Proceedings of ICALP 2008 (1), 320-332.

[5] A. Czygrinow, M. Handckowiak, M. Karonski, "Distributed $O(\Delta \log n)$-Edge-Coloring Algorithm", Proceedings of the 9th Annual European Symposium on Algorithms, 345-355, August 28-31, 2001.

[6] D. Dubhashi, D. Grable, A. Panconesi, "Nearly-optimal, distributed edge-colouring via the nibble method", Theoretical Computer Science 203 (1998) 225-251, a special issue for the 3rd European Symposium on Algorithms (ESA 95).

[7] D. Durand, R. Jain, D. Tseytlin, "Applying Randomized Edge Coloring Algorithms to Distributed Communications: An Experimental Study", Proc. 7th Symposium on Parallel Algorithms and Architecture, 264-274, 1995.

[8] P. Erdös, "On circuits and subgraphs of chromatic graphs", Mathematika 9 (1962), 170-175.

[9] P. Erdös, A.L. Rubin and H. Taylor, "Choosability in graphs", Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium XXVI (1979), 125-157.

[10] D. Grable, A. Panconesi, "Nearly optimal distributed edge coloring in log log rounds", Random Structures and Algorithms, Vol. 10, No. 3, 1997. Also in Proceedings of the 1997 ACM-SIAM symposium on Discrete Algorithms, 278-285.

[11] J.E. Hopcroft, R.M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs", SIAM Journal on Computing, vol.2, (no.4), Dec. 1973. 225-31.

[12] S. Isobe, X. Zhou, T. Nishizeki, "Total colorings of degenerate graphs", Combinatorica, 27, 167-182, 2007.

[13] T. R. Jensen, B. Toft, "Graph Coloring Problems", John Wiley & Sons, New York, 1995.

[14] M. Juvan, B. Mohar, R. Škrekovski, "List Total Colourings of Graphs", Combinatorics, Probability and Computing, Volume 7, Issue 2 (June 1998).

[15] J. Kahn, "Asymptotics of the List Chromatic Index for Multigraph", Random Structures & Algorithms 17:117-156, 2000.

[16] A. V. Kostochka, "List edge chromatic number of graphs with large girth", Discrete Math. 101, 1992.

[17] V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy, A. Srinivasan, "End-to-end packet-scheduling in wireless ad-hoc networks", Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, SODA 2004, 1021-1030.

[18] N. Linial, "Locality in distributed graph algorithms", SIAM Journal on Computing, 21(1):193-201, February 1992.

[19] A. Lubotzky, R. Phillips, P. Sarnak, "Ramanujan graphs", Combinatorica 8 (1988), 261-277.

[20] J. Matoušek, "Lecture Notes in Discrete Geometry", Springer-Verlag, New York, 2002.

[21] M. Molloy, B. Reed, "A bound on the Total Chromatic Number", Combinatorica 18:241-280, 1998.

[22] A. Panconesi, A. Srinivasan, "The local nature of $\Delta$-colorings and its algorithmic applications", Combinatorica 15 (2) 1995, 255-280.

[23] A. Panconesi, A. Srinivasan, "On the complexity of Distributed Network Decomposition", Journal of Algorithms 20, 356–374 (1996).

[24] A. Panconesi, A. Srinivasan, "Fast randomized algorithms for distributed edge coloring", SIAM Journal on Computing, Vol. 26, No. 2, 350-368, April 1997. Also in Proceedings of the ACM Symposium on Principles of Distributed Computing, 1992, 251-262.

[25] V.G. Vizing, "On an estimate of the chromatic class of a p-graph", Diskret. Analiz. 3, 25:30, 1964.

[26] X. Zhou, T. Nishizeki, "Edge-coloring and $f$-coloring for various classes of graphs", J. Graph Algorithms and Applications 3(1), 1999.