The Local Nature of List Colorings for Graphs of High Girth*

Flavio Chierichetti[1] and Andrea Vattani[2]

[1] CS Department, Sapienza University of Rome
chierichetti@di.uniroma1.it
[2] CSE Department, University of California San Diego
avattani@ucsd.edu

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 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.

* 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.

L. Aceto et al. (Eds.): ICALP 2008, Part I, LNCS 5125, pp. 320–332, 2008.

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 [3] (that states $\chi'(G) = \chi_l'(G)$), a conjecture of Behzad and Vizing [2] ($\chi''(G) \leq \Delta + 2$) and a related conjecture of Juvan, Mohar, Škrekovski [11] ($\chi''(G) = \chi_l''(G)$), 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 [7], 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. *If $\mathcal{G}$ is a graph with n nodes, $\Delta(\mathcal{G}) \notin \{1,2\}$, 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 1. *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 Δ_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.

Erdös [7] gave an upper bound on the vertex chromatic number of graphs of high girth. His argument can be extended to show the following:

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

The proofs of theorems 2 and 3 are omitted from this extended abstract for lack of space.

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.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 4. *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 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 [19] shows that the chromatic index $\chi'(\mathcal{G})$ of any graph $\mathcal{G}$ of maximum degree Δ is either Δ or $\Delta + 1$. A conjecture of Bollobás and Harris [3] 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 [2] and Vizing states that $\chi''(\mathcal{G}) \leq \Delta + 2$. A more recent conjecture by Juvan, Mohar, Škrekovski [11] states that the list total chromatic number $\chi''_l(\mathcal{G})$ equals $\chi''(\mathcal{G})$.

On the other hand, it is known [8] 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 [12] proved that the list chromatic index is $\chi'_l(\mathcal{G}) \leq (1 + o(1))\Delta$, while Molloy and Reed [17] 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 [13], 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

Δ (e.g. already for Δ logarithmic in n). Note that, even for smaller Δ, we establish a better bound of Δ (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 instance [10]) that any graph $\mathcal{G}$ with degeneracy $\leq \Delta/2$ belongs to 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 [9,20], the authors give parallel algorithms for computing Δ 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 Δ, 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 [4], 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 [16]). It can be shown that if our result is subsumed by that of [4] 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 it 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 [7] 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 [5,6,18] 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. Henceforth, we will use the term coloring to mean list coloring.

Definition 1. *A Δ-tree is a rooted tree of maximum degree at most Δ 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 2. *Let $T(r)$ be a Δ-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 $\Delta(G) = \Delta$ and $\deg_G(r) < \Delta$.*

The constraints on a Δ-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 3. *Let T be a Δ-tree. Given a Δ-list assignment $\mathcal{L}$ to $E(T)$, an augmentation G of T and a coloring γ of G, we say that the triple $(\mathcal{L}, G, \gamma)$ is* legal *if γ is a proper coloring of G that agrees with $\mathcal{L}$.*

Note that every Δ-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 Δ-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 4. *Let $T(r)$ be a Δ-tree. $T(r)$ is* at least t-free *if, for each list L of Δ 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 γ_c of G such that*

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

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

- Each Δ-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 Δ.
- No Δ-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 1. *Let T be a Δ-tree T that is exactly t-free, let L be a set of Δ colors, and $C \in \binom{L}{t}$. Then, there exists a legal triple $(\mathcal{L}, G, \gamma)$ such that, for all and only $c \in C$, there exists a proper coloring γ_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*
- γ_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. □

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

Proof. Let u be that child and let $T(u)$ denote its subtree. By deleting u the degree of freedom does not decrease. To see that it does not increase either, let T' be the tree obtained by removing $T(u)$ from T. Take any color c available at the root of T'. We show that c is also available at the root of T. Color T' as to have c available at the root; also color all the edges of T, excluding those of $T(u)$, in the same manner. Now the only uncolored edge incident to the root has at least 2 colors available, 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. Thus, we can choose a color other than c to color the edge connecting u to the root to complete the coloring, that is c is available for T. □

Observation 2. *Each minimum 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 the minimality of T, observation 1 cannot be applied to it. That is, each tree T_i $(1 \leq i \leq k)$ is at most k-free.

By lemma 1, given any set C of k colors, for all $1 \leq i \leq k$, there exists an augmentation of T_i such that in every proper coloring of T_i the colors available at its root are a subset of C.

These augmentations 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. Since there are k such edges and $|C| = k$, the set of colors of those edges must be C in any proper coloring. So exactly $\Delta - k$ colors are available at the root of T. That is, T is at most $(\Delta - k)$-free. Since $k \geq \Delta - t$, this can only be true for $k = \Delta - t$. □

The next definition is pivotal.

Definition 5. *Let $\mathcal{T}_t^h$ be the set of t-free Δ-trees of height h. Also, let n_t^h be the order of any smallest t-free tree in $\mathcal{T}_t^h$ (or ∞ 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 $\left(\lceil \frac{\Delta}{2} \rceil + 1\right)$-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 $t = 1, \ldots, \lceil \frac{\Delta}{2} \rceil$, and therefore their degree of freedom must be at least $\lceil \frac{\Delta}{2} \rceil + 1$, 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 2. *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 Δ colors and, by lemma 1, the set of colors of the edges can be forced to be any set of $\Delta - t$ colors.

Observations 1-2 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 3. *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 2. 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 $n_{\lceil\frac{\Delta}{2}\rceil}^h \leq n_1^h \leq n_{\lceil\frac{\Delta}{2}\rceil-1}^h$.

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

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

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

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

The last inequality is implied by $\frac{1}{2}n_{\Delta-1}^h \geq \lfloor\frac{\Delta}{2}\rfloor$ which is in turn implied by $n_{\Delta-1}^h \geq \Delta$, 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

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

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

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

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

$$\left(\Delta - \left\lceil \frac{\Delta}{2} \right\rceil\right)\left((n_{\Delta-1}^{h-1} - 1)\left\lceil \frac{\Delta}{2} \right\rceil + 1\right) \ge \left(\Delta - \left(\left\lceil \frac{\Delta}{2} \right\rceil + 1\right)\right)\frac{\Delta}{2} n_{\Delta-1}^{h-1}$$

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

Lemma 4. *A Δ-tree T of order n, $\Delta \ge 3$ and height $h \ge 2\log_{\Delta-1} n$ 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 2 and 3 imply that

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

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 $\mathcal{O}$,

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

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

Analogously, if $h \ge 3$ is odd, $n < n^h_{\lceil \frac{\Delta}{2} \rceil} \le n_1^h \le n^h_{\lceil \frac{\Delta}{2} \rceil - 1} \le \cdots \le n_3^h \le n_2^h$ 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 $\Delta \ge 3$ 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 4. We are also given a Δ-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 Δ-trees of two possible heights: either R or 0. Any tree having height R is $(\lceil \frac{\Delta}{2} \rceil + 1)$-free by lemma 4, 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 5. *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 Δ-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. 1 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 [15]. 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 Δ-regular graph of order n and let $\mathcal{G}'$ be the graph that $\mathcal{G}$ is mapped into; $\mathcal{G}'$ will either be a Δ-regular graph, or a graph having $2n-2$ nodes of degree Δ 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 Δ 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 Algorithms

In this section we sketch some algorithmic consequences of our theorems for list colorings, with special emphasis on distributed algorithms that are our main focus. In particular, we will sketch the proof of theorem 4.

We consider the classic model suggested by Linial [14] of a synchronous, message-passing distributed network. The running time of an algorithm is given by the number communication rounds. 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 only a polynomial amount of computation. Thus it could 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 vertex coloring 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 (α, β)-decomposition of a graph $\mathcal{G}$ is a partition of the nodes of $\mathcal{G}$ into β-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 in $\mathcal{G}$ hits both of them). A β-weakly-connected component of $\mathcal{G}$ is a subset of the nodes of $\mathcal{G}$ such that any two nodes of the subset are at distance at most β in $\mathcal{G}$. For the sake of brevity, we use the word *cluster* to refer to a β-weakly-connected component. In [1] the authors give a randomized distributed algorithm that obtains $(O(\log n), O(\log n))$-decompositions in $O(\log n)$ many rounds.

Lemma 5 gives a logarithmic bound on the depth of the trees to be considered for the edge re-coloring operations. The following lemmas, that we state without proof for lack of space, give analogous bounds for the vertex and total re-coloring operations.

Lemma 6. *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 of height $\leq R$.*

Lemma 7. *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 $\leq R$.*

Note that for each kind of coloring, the depth of the trees to be re-colored is bounded by some $d \in O(\log n)$. To ensure non-interference between different re-coloring 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 such that $V(C^t) = V(C)$ and where two nodes are connected in C^t whenever 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. The meta-algorithm is as follows. For each label k, in parallel, each leader of a k-colored cluster,

1. gathers all the information about its cluster K and its neighbors;
2. colors, locally, the nodes of K with respect to the edges of C;
3. broadcasts the new colors to the nodes in K.

This algorithm runs for $O(T + \alpha\beta t)$ communication rounds, where T is the time needed to compute the (α, β)-decomposition of C^t. Using the randomized algorithm in [1], we have $\alpha = \beta = O(\log n)$ and $T = O(t \log n) = O(\log^2 n)$. This proves thm. 4.

So far, we have neglected to discuss how a leader can (locally) compute the re-coloring of a tree. 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, the re-colorings can be computed in polynomial time. First, note that to re-color a tree it is sufficient to compute the list of available colors of a generic object in the tree (i.e., edge and/or node depending on the type of coloring), given the lists of its children in the tree. This can be solved efficiently by means of a reduction to the maximum bipartite matching problem.

The proof of the following proposition is omitted for lack of space.

Proposition 2. *The recoloring 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.*

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. Anil Kumar, V.S., Marathe, M.V., Parthasarathy, S., Srinivasan, A.: End-to-end packet-scheduling in wireless ad-hoc networks. In: Proc. 15th ACM-SIAM Symp. on Discrete Alg (SODA 2004), pp. 1021–1030 (2004)
2. Behzad, M.: The total chromatic number. Comb. Math. and its Appl. (Proc. Conf., Oxford 1969). Academic Press, London (1971)
3. Bollobás, B., Harris, A.J.: List colorings of graphs. Graphs and Combinatorics 1, 115–127 (1985)
4. Borodin, O.V., Kostochka, A.V., Woodall, D.R.: List edge and list total colourings of multi-graphs. J. Comb. Theory, Series B 71 (1997)
5. Czygrinow, A., Handckowiak, M., Karonski, M.: Distributed $O(\Delta \log n)$-Edge-Coloring Algorithm. In: Proc. 9th Europ. Symp. on Alg. (2001)
6. Dubhashi, D., Grable, D., Panconesi, A.: Nearly-optimal, distributed edge-colouring via the nibble method. Theoretical Computer Science 203 (1998)
7. Erdös, P.: On circuits and subgraphs of chromatic graphs. Mathematika 9, 170–175 (1962)
8. Erdös, P., Rubin, A.L., Taylor, H.: Choosability in graphs. In: Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium XXVI, pp. 125–157 (1979)
9. Isobe, S., Zhou, X., Nishizeki, T.: Total colorings of degenerate graphs. Combinatorica 27, 167–182 (2007)
10. Jensen, T.R., Toft, B.: Graph Coloring Problems. John Wiley & Sons, New York (1995)
11. Juvan, M., Mohar, B., Škrekovski, R.: List Total Colourings of Graphs. Comb., Prob. and Comp. 7(2) (1998)

12. Kahn, J.: Asymptotics of the List Chromatic Index for Multigraph. Random Struct. & Alg. 17, 117–156 (2000)
13. Kostochka, A.V.: List edge chromatic number of graphs with large girth. Discrete Math. 101 (1992)
14. Linial, N.: Locality in distributed graph algorithms. SIAM J. on Comp. 21(1), 193–201 (1992)
15. Lubotzky, A., Phillips, R., Sarnak, P.: Ramanujan graphs. Combinatorica 8, 261–277 (1988)
16. Matoušek, J.: Lecture Notes in Discrete Geometry. Springer, New York (2002)
17. Molloy, M., Reed, B.: A bound on the Total Chromatic Number. Combinatorica 18, 241–280 (1998)
18. Panconesi, A., Srinivasan, A.: Fast randomized algorithms for distributed edge coloring. SIAM J. on Comp. 26(2) (1997)
19. Vizing, V.G.: On an estimate of the chromatic class of a p-graph. Diskret. Analiz. 3, 25–30 (1964)
20. Zhou, X., Nishizeki, T.: Edge-coloring and f-coloring for various classes of graphs. J. Graph Algorithms and Applications 3(1) (1999)