Non-Monochromatic and Conflict-Free Colorings in Tree Spaces*

B. Aronov[1], M. de Berg[2], A. Markovic[2], and G. Woeginger[3]

1 **Department of Computer Science and Engineering, New York University, USA.**
2 **TU Eindhoven, the Netherlands.**
3 **RWTH Aachen, Germany.**

Abstract

We study non-monochromatic and conflict-free colorings on tree spaces, that is, one-dimensional spaces with a tree topology. More specifically, we analyze the number of colors needed to color a set $\mathcal{A}$ of n objects in a tree space $\mathcal{T}$ with k leaves, with each object being a connected subset of $\mathcal{T}$, in a non-monochromatic or conflict-free fashion. We prove that there exists a non-monochromatic coloring with $O(\min(\ell, \sqrt{k}))$ colors, where ℓ denotes the maximum number of leaves of any object in $\mathcal{A}$. This bound is tight in the worst case. This result implies that there exists a conflict-free coloring with $O(\ell \log k)$ colors.

1 Introduction

Conflict-free colorings, or CF-colorings for short, were introduced by Even *et al.* [4] and Smorodinsky [8] to model frequency assignment to base stations in wireless networks. In the basic setting one is given a set S of objects in the plane—often disks are considered—and the goal is to assign a color to each object such that the following holds: for any point p in the plane such that the set $S_p := \{D \in S \mid p \in D\}$ of objects containing p is non-empty, S_p must contain an object whose color is different from the colors of the other objects in S_p. Even *et al.* proved, among other things, that any set of disks admits a CF-coloring with $O(\log n)$ colors. Since then many different geometric variants of CF-colorings have been studied. For example, Har-Peled and Smorodinsky [5] generalized the result to objects with near-linear union complexity, while Even *et al.* [4] considered the dual version of the problem. See the survey by Smorodinsky [10] for an overview. A restricted type of CF-colorings are *unique-maximum colorings* (UM-colorings), in which the colors are identified with integers, and the maximum color in the set S_p is required to be unique. Another type of coloring, often used as an intermediate step to obtain a CF-coloring, is *non-monochromatic* (NM). In an NM-coloring—sometimes called *a proper coloring*—we only require that, for any point p in the plane, if the set S_p contains at least two elements, not all of them have the same color. Smorodinsky [9] showed that if an NM-coloring on n elements using $\beta(n)$ colors is given, one can create a CF-coloring using $O(\beta(n) \log n)$ colors.

CF-colorings can also be defined in a more abstract setting. Here one is given a hypergraph $\mathcal{H} = (V, E)$ and the goal is to color V such that for every (non-empty) hyperedge $e \in E$, there is a vertex in e whose color is different from that of the other vertices in e. Ashok *et al.* [2] showed that deciding whether a given hypergraph can be CF-colored using k colors is fixed-parameter tractable. Note that the basic geometric version mentioned above—coloring objects in $\mathbb{R}^2$ with respect to points—can be phrased in terms of hypergraphs by letting the objects be the vertex set V and for each point p in the plane creating a hyperedge $e := S_p$.

* MdB and AM are supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 024.002.003. BA has been supported by NSF Grants CCF-11-17336, CCF-12-18791, and CCF-15-40656, and by BSF grant 2014/170.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

Another avenue for constructing a hypergraph $\mathcal{H}$ to be colored is to start with a graph $\mathcal{G}$, let the vertices of $\mathcal{H}$ be the vertices of $\mathcal{G}$ and create hyperedges for (the sets of vertices of) certain subgraphs of $\mathcal{G}$. For example, Pach and Tardos [7] considered the case where hyperedges are all the vertex neighborhoods. For this case, Abel *et al.* [1] recently showed that a planar graph can always be colored with only three colors, if we allow some vertices to be uncolored. (Otherwise, we can use a dummy color, increasing the number of colors to four.) As another example, we let the hyperedges be induced by all the paths. This setting is equivalent to an older notion of *vertex ranking* [3], also known as *ordered coloring* [6].

In this paper we study CF- and NM-colorings in a setting that is closely related to both the geometric and the graph-based setting. More precisely, the spaces that we consider are *tree spaces*—that is, one-dimensional spaces with a tree topology—and the objects that we want to color are connected subsets (in other words, subtrees) of the given tree space. In this setting, we are interested in how the complexity of the given tree space and of the objects to be colored influence the chromatic number. Note that, if the given tree space is a single curve, the problem reduces to coloring intervals on the real line.

Our contributions. Let $\mathcal{T}$ be the given tree space. It may be convenient to visualize $\mathcal{T}$ as being embedded in $\mathbb{R}^2$, although the embedding is actually immaterial. We assume without loss of generality that $\mathcal{T}$ is bounded—it does not have infinitely long branches—and define the *vertices* of $\mathcal{T}$ in the natural manner. Any vertex of $\mathcal{T}$ is either an *internal vertex* (a branching point of degree at least three) or a *leaf*. The curves connecting the vertices, whose union is $\mathcal{T}$, are called the *edges* of the tree space. We denote the number of leaves of $\mathcal{T}$ by k.

Let $\mathcal{A}$ be the set of n objects that we wish to color, where each object $T \in \mathcal{A}$ is a connected subset of $\mathcal{T}$. Thus each object itself is also a tree. From now on, we will refer to the objects in $\mathcal{A}$ as “trees”, and always use “tree space” when talking about $\mathcal{T}$. We denote the maximum number of leaves of any tree in $\mathcal{A}$ by ℓ. Note that internal vertices of a tree are necessarily internal vertices of $\mathcal{T}$, but leaves of a tree may also lie in the interior of an edge of $\mathcal{T}$. CF-colorings of such a set $\mathcal{A}$ are now defined as above: for any point $p \in \mathcal{T}$, the set $S_p := \{T \in \mathcal{A} \mid p \in T\}$ (if non-empty) should have a tree with a unique color. We now define the CF-chromatic number $X^{\text{tree,tree}}_{\text{cf}}(k, \ell; n)$ as the minimum number of colors sufficient to CF-color any set $\mathcal{A}$ of n trees of at most ℓ leaves each in a tree space of at most k leaves. The NM-chromatic number $X^{\text{tree,tree}}_{\text{nm}}(k, \ell; n)$ is defined similarly. We will show that[1]

▶ **Theorem 1.1** (Main result).
(i) $X^{\text{tree,tree}}_{\text{nm}}(k, \ell; n) \leqslant \min(\ell + 3, 2\sqrt{6k} + 2)$*, and (ii)* $X^{\text{tree,tree}}_{\text{cf}}(k, \ell; n) = O(\ell \log k)$.

In the full version we also (a) show how to use two fewer colors in part (i) of the theorem and (b) provide two lower bounds for NM-colorings, namely $X^{\text{tree,tree}}_{\text{nm}}(k, \ell; n) \geqslant \min\left(\ell + 1, \left\lfloor \frac{\sqrt{1+8k}}{2} \right\rfloor, n\right)$, which clearly also apply to CF-colorings, and $X^{\text{tree,tree}}_{\text{cf}}(k, \ell; n) \geqslant \lfloor \log_2 \min(k, n) \rfloor$; and (c) study other variants, for example by considering more general network spaces (rather than tree spaces) and other types of objects to be colored.

2 The coloring algorithms

Preliminaries: The chain method. We start by describing a folklore technique, called the *chain method*, to color intervals in $\mathbb{R}^1$ in a non-monochromatic fashion using at most two

[1] Obviously the number of trees, n, is an upper bound as well. To avoid cluttering the bounds, we usually omit this trivial bound.

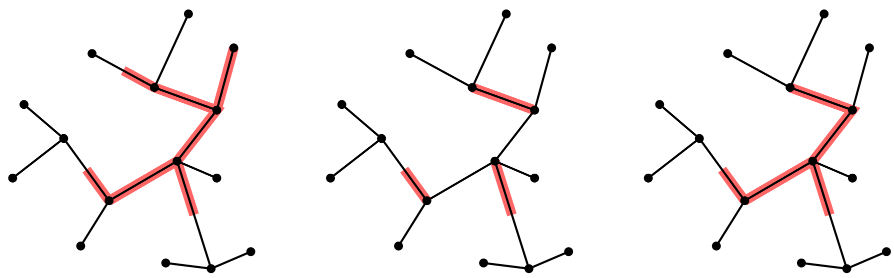

Figure 1 The original tree T (left), the set $\bigcup_{e \in E(T)} e \cap T$ (middle), and the new tree T' (right).

colors. We order the intervals left-to-right by their left endpoints (in case of ties, we take the longest interval first) and color them in this order using the so-called *active color* which is defined as follows. We start with blue as the active color. We color the first interval, then change the active color to red. We then use the following procedure: we color the next interval I in the ordering using the active color, then if the right endpoint of I is not contained in any other colored interval, we change the active color from red to blue or blue to red. It is easy to show the resulting coloring is non-monochromatic.

Overview of the coloring procedure. Let $\mathcal{T}$ be a tree space and let $\mathcal{A}$ be a set of n trees on $\mathcal{T}$, each with at most ℓ leaves. We will NM-color $\mathcal{A}$ in two phases: first, we select a subset $\mathcal{C} \subseteq \mathcal{A}$ of size at most $6k - 12$ and color it with at most $\min(\ell + 1, 2\sqrt{6k})$ colors. In the second phase we extend this coloring to the whole set $\mathcal{A}$ using at most two extra colors.

An edge e of $\mathcal{T}$ is a *leaf edge* if it is incident to a leaf; the remaining edges are *internal*. We define $\mathcal{C} \subseteq \mathcal{A}$ as the set of at most $6k - 12$ trees selected as follows. For every pair (e, v), where e is an edge of $\mathcal{T}$ and v is an endpoint of e that is not a leaf of $\mathcal{T}$, we choose two trees containing v and extending the furthest into e (if they exist), that is, trees T of $\mathcal{A}$ containing v for which $\text{length}(T \cap e)$ is maximal, and place them in $\mathcal{A}(e, v)$. Note that if two or more trees of $\mathcal{A}$ fully contain e, then $\mathcal{A}(e, v)$ contains two of them, chosen arbitrarily. Note also that, if a tree contains an internal edge e fully, it may be chosen by both endpoints. We now define $\mathcal{A}(e) := \mathcal{A}(e, u) \cup \mathcal{A}(e, v)$ for each internal edge $e = \{u, v\}$, define $\mathcal{A}(e) := \mathcal{A}(e, v)$ for each leaf edge $e = \{u, v\}$ with v being its non-leaf endpoint. Finally, we define $\mathcal{C} := \bigcup \mathcal{A}(e)$, with the union taken over all edges e of $\mathcal{T}$. Since $\mathcal{A}(e)$ contains at most four trees for any internal edge e and at most two trees for any leaf edge e, and since the number of internal edges of $\mathcal{T}$ is at most $k - 3$ and the number of leaf edges is at most k, where k is the number of leaves of $\mathcal{T}$ (which, as a topological tree, does not have degree-two vertices), $|\mathcal{C}| \leqslant 6k - 12$, as claimed. We first explain how to color $\mathcal{C}$.

Coloring $\mathcal{C}$. We color $\mathcal{C}$ in two steps. Let $E(T)$ be the set of edges e of $\mathcal{T}$ with $T \in \mathcal{A}(e)$. Firstly, if $\ell > 2\sqrt{6k}$ we select all subtrees T with $|E(T)| \geqslant \sqrt{6k}$, and give each of them a unique color. Since $\sum_e |\mathcal{A}(e)| \leqslant 6k - 12$ there are at most $\sqrt{6k} - 1$ such trees, so we use at most $\sqrt{6k} - 1$ colors. Then for each uncolored $T \in \mathcal{C}$ we create a new tree T', defined as the smallest tree containing $\bigcup_{e \in E(T)} e \cap T$; see Fig. 1. Note that T' has at most $\ell' := \min(\ell, \sqrt{6k})$ leaves because $|E(T)| < \sqrt{6k}$. Define $\mathcal{C}' := \{T' \mid T \in \mathcal{C}\}$. The second step is to color $\mathcal{C}'$. We need the following lemma, which shows that an NM-coloring of $\mathcal{C}'$ carries over to $\mathcal{C}$.

▶ **Lemma 2.1.** *Any NM-coloring of $\mathcal{C}'$ corresponds to an NM-coloring of $\mathcal{C}$, that is, if we give each tree $T \in \mathcal{C}$ the color of the corresponding tree $T' \in \mathcal{C}'$ then we obtain an NM-coloring.*

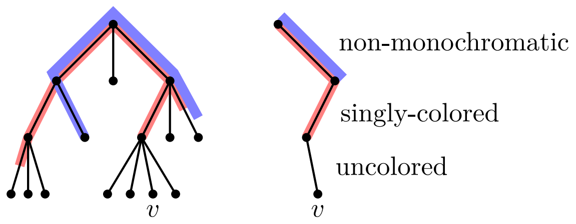

Figure 2 A coloring of trees (left) and an illustration of the invariant for v (right).

Proof. Let q be a point on an edge e of $\mathcal{T}$ contained in at least two trees of $\mathcal{C}$ (if no such trees exists, the coloring is trivially non-monochromatic at q). Since q is contained in at least two trees of $\mathcal{C}$, it is also contained in two trees of $\mathcal{A}(e)$. Call these trees T_1 and T_2. Note that T_1 either receives a color in the first coloring step—namely when $|E(T_1)| \geqslant 2\sqrt{6k}$—or $T_1' \in \mathcal{C}'$ contains q (since $e \in E(T_1)$). A similar statement holds for T_2. Since the colors used in the first step are unique and $\mathcal{C}'$ is NM-colored, this implies that T_1 and T_2 have different colors. Hence, $\mathcal{C}$ is NM-colored. ◀

Next we show how to NM-color $\mathcal{C}'$. Fix an arbitrary root r of the tree space $\mathcal{T}$. Our coloring procedure for $\mathcal{C}'$ maintains the following invariant: any path from r to a leaf v of $\mathcal{T}$ consists of three disjoint consecutive subpaths (some possibly empty), in this order, as illustrated in Fig. 2:

- a *non-monochromatic* subpath containing the root on which at least two trees are colored with at least two different colors,
- a *singly-colored* subpath containing exactly one colored tree, and
- an *uncolored* subpath containing the leaf on which no tree is colored.

▶ **Observation 2.2.** *Any set of trees containing r and satisfying the invariant described above is NM-colored if we disregard uncolored trees.*

We color the trees $T \in \mathcal{C}'$ that contain r in an arbitrary order, using $\ell' + 1$ colors, as follows: for each leaf v of T, we follow the path from v to the root r to find a singly-colored part. Note that if we find a singly-colored part—by the invariant there is at most one such part on the path from v to r—we cannot use that color for T. Since T has at most ℓ' leaves, this eliminates at most ℓ' colors. Hence, at least one color remains for T.

▶ **Lemma 2.3.** *The procedure described above maintains the invariant and colors all trees of $\mathcal{C}'$ containing r with at most $\ell' + 1$ colors.*

Proof. Suppose the invariant holds before the coloring of T. Then we need to make sure the invariant still holds after T has been colored. Let w be a leaf of $\mathcal{T}$ and π_w the path from w to the root. If π_w does not contain a leaf of T then the invariant obviously still holds on π_w. Now suppose π_w contains a leaf v of T, and let $\pi_v \subseteq \pi_w$ be the path from v to r. The part of π_v that was uncolored (if it was non-empty) now is singly-colored. The part that was singly-colored now becomes non-monochromatic, as we eliminated that color for T. And the part that was already non-monochromatic stays so. Therefore the invariant is indeed maintained for π_w, concluding the proof. ◀

Once all the trees containing r are colored we delete r from $\mathcal{T}$, that is, we consider the space $\mathcal{T} \setminus \{r\}$, and we take the closures of the resulting connected components. This creates a number of subspaces such that each uncolored tree in $\mathcal{C}'$ is contained in exactly one of

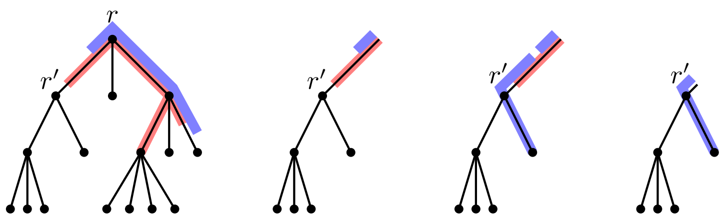

Figure 3 When recursing on the subspace rooted at r' (leftmost), the invariant does not hold anymore (middle left), as the parts are switched on the edge between r and r'. To remedy this, we first color the tree extending the furthest into that edge (middle right), starting from r'. We then trim the tree to fix the invariant (rightmost).

them. Consider such a subspace $\mathcal{T}'$ and let r' be the neighbor of r in $\mathcal{T}'$. We now want to recursively color the uncolored trees in $\mathcal{T}'$, taking r' as the root of $\mathcal{T}'$. However, the invariant might not hold on the edge e from r' to the old root r: Since now r is considered a child of r', the order of the three parts might switch on e—see Fig. 3. Suppose this is the case, and let c_e be the color of the singly-colored part on the edge e. Note that for the order to switch, the non-monochromatic part needs to end on e, and therefore the only color used in any singly-colored part of the tree rooted at r' is c_e. We overcome this problem by carefully choosing the order in which we color the trees containing r'. Namely, we first color the tree T extending the furthest in e. In this case, there is only one color forbidden, namely c_e. We can therefore easily color T. We can then trim the tree space $\mathcal{T}'$ to remove any non-monochromatic part and hence restore the invariant and continue with the coloring.

▶ **Lemma 2.4.** *$\mathcal{C}$ admits an NM-coloring with* $\min(\ell+1, 2\sqrt{6k})$ *colors.*

Proof. The fact that the procedure above produces an NM-coloring follows from Lemmas 2.1 and 2.3. When $\ell > 2\sqrt{6k}$ we use $\sqrt{6k}-1$ colors to deal with trees T with $|E(T)| \geqslant \sqrt{6k}$ and $\ell'+1 \leqslant \min(\ell, 2\sqrt{6k})+1 \leqslant \sqrt{6k}+1$ colors for the other trees, giving $2\sqrt{6k}$ colors in total. When $\ell \leqslant 2\sqrt{6k}$ we do not treat the trees with $|E(T)| \geqslant \sqrt{6k}$ separately, so we just use $\ell'+1 \leqslant \min(\ell, \sqrt{6k})+1 \leqslant \ell+1$ colors. ◀

Extending the coloring from $\mathcal{C}$ to $\mathcal{A}$. Let $c\colon \mathcal{C} \to \mathbb{N}$ be an NM-coloring on $\mathcal{C}$. We extend the coloring to $\mathcal{A}$ as follows. We start by coloring all trees containing an internal vertex of $\mathcal{T}$ using an arbitrary color already used. Then, for each edge $e = \{r, r'\}$ we color the set of uncolored trees contained in e using the chain method. For this we use two new colors, which are used for all chains—we can re-use the same two colors for the chains, since trivially the chains in any two edges e, e' do not interact. (In the full version we describe a more careful approach, which avoids using two new colors.) The following lemma proves the extended coloring is non-monochromatic.

▶ **Lemma 2.5.** *Any NM-coloring c on $\mathcal{C}$ can be extended to $\mathcal{A}$ by using two extra colors.*

Proof. Let $\mathcal{A}_1$ be the subset of trees in $\mathcal{A} \setminus \mathcal{C}$ that contain an internal vertex of $\mathcal{T}$, and let $\mathcal{A}_2$ be the remaining trees in $\mathcal{A} \setminus \mathcal{C}$. By Lemma 2.4 we have an NM-coloring on $\mathcal{C}$, and the chain method gives us an NM-coloring for the trees in $\mathcal{A}_2$ using two additional colors. It is easy to see that together this gives us an NM-coloring on $\mathcal{C} \cup \mathcal{A}_2$. The trees in $\mathcal{A}_1$ received an arbitrary color already used. To prove that this gives an NM-coloring for $\mathcal{A} = \mathcal{C} \cup \mathcal{A}_1 \cup \mathcal{A}_2$, it suffices to prove that each tree $T \in \mathcal{A}_1$ is *doubly-covered* by $\mathcal{C}$, that is, any point $q \in T$ is contained in at least two trees in $\mathcal{C}$. To this end, let e be an edge such that $q \in e$. Then, since $T \notin \mathcal{C}$ and T contains an endpoint v of e, the two trees in $\mathcal{A}(e, v)$ contain q. Hence, T is doubly-covered by $\mathcal{C}$, as claimed. ◀

Proof of Theorem 1.1. For the NM-coloring part of the theorem, we use Lemmas 2.4 and 2.5. For the second part, if $\ell > 2\sqrt{6k}$ we again reduce $\mathcal{C}$ to $\mathcal{C}'$ using at most $\sqrt{6k} - 1$ colors. Then use the result by Smorodinsky [9] on the NM-coloring on $\mathcal{C}'$ provided by Lemma 2.3. Since this coloring uses at most $\ell' + 1$ colors and $|\mathcal{C}'| \leqslant 6k - 12$, the CF-coloring uses $O(\ell \log k)$ colors. We then extend the coloring to $\mathcal{A}$ using similar techniques as for the NM-coloring. This coloring uses $O(\sqrt{k} \log k)$ colors if $\ell > 2\sqrt{6k}$, which is in $O(\ell \log k)$, and directly $O(\ell \log k)$ colors otherwise. Note that a direct application of the result by Smorodinsty [9] would give a $O(\ell \log n)$ bound instead. ◀

3 Concluding remarks

We studied NM- and CF-colorings on tree spaces, where the objects to be colored are connected subsets of the tree space. We showed that the number of colors can be bounded as a function of the complexity (that is, number of leaves) of the tree space and the objects, rather than on the number of objects. In the full version we show that this is also the case for balls on network spaces. It would be interesting to find more settings where this is the case.

References

1 Z. Abel, V. Alvarez, E. D. Demaine, S. P. Fekete, A. Gour, A. Hesterberg, P. Keldenich, and C. Scheffer. Three colors suffice: Conflict-free coloring of planar graphs. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1951–1963, 2017.

2 P. Ashok, A. Dudeja, and S. Kolay. Exact and FPT algorithms for max-conflict free coloring in hypergraphs. In *Proceedings of the 26th International Symposium of Algorithms and Computation*, pages 271–282, 2015.

3 H. L. Bodlaender, J. S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, and Z. Tuza. Ranking of graphs. In *Proceedings of the 20th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 292–304, 1994.

4 G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM Journal on Computing*, 33(1):94–136, 2003.

5 S. Har-Peled and S. Smorodinsky. Conflict-free coloring of points and simple regions in the plane. *Discrete & Computational Geometry*, 34(1):47–70, 2005.

6 M. Katchalski, W. McCuaig, and S. M. Seager. Ordered colourings. *Discrete Mathematics*, 142(1-3):141–154, 1995.

7 J. Pach and G. Tardos. Conflict-free colourings of graphs and hypergraphs. *Combinatorics, Probability & Computing*, 18(5):819–834, 2009.

8 S. Smorodinsky. *Combinatorial Problems in Computational Geometry*. PhD thesis, Tel-Aviv University, 2003.

9 S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 316–323, 2006.

10 S. Smorodinsky. Conflict-free coloring and its applications. In I. Bárány, K. J. Böröczky, G. F. Tóth, and J. Pach, editors, *Geometry — Intuitive, Discrete, and Convex: A Tribute to László Fejes Tóth*, pages 331–389. Springer Berlin Heidelberg, 2013. See also arXiv:abs/1005.3616.