Online conflict-free coloring for geometric hypergraphs

Amotz Bar-Noy* Panagiotis Cheilaris† Svetlana Olonetsky‡ Shakhar Smorodinsky§

Abstract

(i) We provide a framework for online conflict-free coloring (CF-coloring) any hypergraph. We use this framework to obtain an efficient randomized online algorithm for CF-coloring any k-degenerate hypergraph. Our algorithm uses $O(k \log n)$ colors with high probability and this bound is asymptotically optimal for any constant k. Moreover, our algorithm uses $O(k \log k \log n)$ random bits with high probability. We obtain asymptotically optimal randomized algorithms for online CF-coloring some hypergraphs that arise in geometry and model an important version of the frequency assignment task for cellular networks. Our algorithms use exponentially fewer random bits compared to previous results for these special cases ($O(\log n)$ bits instead of $\Theta(n \log n)$ bits).
(ii) We initiate the study of deterministic online CF-coloring with recoloring. The goal is to use few colors, but also resort to recoloring as little as possible. We provide an algorithm for CF-coloring with respect to halfplanes using $O(\log n)$ colors and $O(n)$ recolorings.

1 Introduction

A *hypergraph* is a pair $(V, \mathcal{E})$, where V is a finite set and $\mathcal{E} \subset 2^V$. The set V is called the *ground set* or the *vertex set* and the elements of $\mathcal{E}$ are called *hyperedges*. A *proper* k-coloring of a hypergraph $H = (V, \mathcal{E})$, for some positive integer k, is a function $C : V \to \{1, 2, \ldots, k\}$ such that no $S \in \mathcal{E}$ with $|S| \geq 2$ is monochromatic. A *conflict-free* coloring (CF-coloring) of H is a coloring of V with the further restriction that for any hyperedge $S \in \mathcal{E}$ there exists a vertex $v \in S$ with a unique color (i.e., no other vertex of S has the same color as v).

The study of conflict-free colorings was originated in the work of Even et al. [5] and Smorodinsky [9] who were motivated by the problem of frequency assignment in cellular networks. Specifically, cellular networks are heterogeneous networks with two different types of nodes: *base stations* (that act as servers) and *clients*. Base stations are interconnected by an external fixed backbone network whereas clients are connected only to base stations. Connections between clients and base stations are implemented by radio links. Fixed frequencies are assigned to base stations to enable links to clients. Clients continuously scan frequencies in search of a base station with good reception. The fundamental problem of frequency assignment in such cellular networks is to assign frequencies to base stations so that every client, located within the receiving range of at least one station, can be served by some base station, in the sense that the client is located within the range of the station and no other station within its reception range has the same frequency (such a station would be in "conflict" with the given station due to mutual interference). The goal is to minimize the number of assigned frequencies ("colors") since the frequency spectrum is limited and costly. In addition to the practical motivation, this new coloring model has drawn much attention of researchers through its own theoretical interest and such colorings have been the focus of several recent papers (see, e.g., [2, 3, 4, 7, 8, 10]). To capture a dynamic scenario where antennas can be added to the network, Fiat et al. [4] initiated the study of online CF-coloring of hypergraphs.

In this paper, we study the most general form of online CF-coloring applied to arbitrary hypergraphs. Suppose the vertices of an underlying hypergraph $H = (V, \mathcal{E})$ are given online by an adversary. Namely, at every given time step t, a new vertex $v_t \in V$ is given and the algorithm must assign v_t a color such that the coloring is a valid conflict-free coloring of the hypergraph that is induced by the vertices $V_t = \{v_1, \ldots, v_t\}$ (see the exact definition in section 2). Once v_t is assigned a color, that color cannot be changed in the future. The goal is to find an algorithm that minimizes the maximum total number of colors used (where the maximum is taken over all permutations of the set V).

We present a general framework for online CF-coloring any hypergraph. Interestingly, this framework is a generalization of some known coloring algorithms. For example the Unique-Max Algorithm of [4] can be described as a special case of our framework. Also, when the underlying hypergraph is a

*Brooklyn College, City University of New York, amotz@sci.brooklyn.cuny.edu. Supported by the CUNY Collaborative Incentive Research Grants Program Round 11 (2004–2006).

†City University of New York and National Technical University of Athens, philaris@sci.brooklyn.cuny.edu. Supported by the European Social Fund (75%) and National Resources (25%), under the program EPEAEK II, 'Heraclitus'.

‡Tel-Aviv University, olonetsk@post.tau.ac.il

§Courant Institute for Mathematical Sciences, New York University, shakhar@cims.nyu.edu. Supported by the NSF Mathematical Sciences Postdoctoral Fellowship award 0402492.

simple graph then the First-Fit online algorithm is another special case of our framework.

Based on this framework, we introduce a randomized algorithm and show that the maximum number of colors used is a function of the 'degeneracy' of the hypergraph. We define the notion of a k-degenerate hypergraph as a generalization of the same notion for simple graphs. Specifically we show that if the hypergraph is k-degenerate, then our algorithm uses $O(k \log n)$ colors with high probability. This is asymptotically tight for any constant k.

As demonstrated in [4], the problem of online CF-coloring the very special hypergraph induced by points on the real line with respect to intervals is highly non-trivial. Kaplan and Sharir [7] studied the special hypergraph induced by points in the plane with respect to halfplanes and unit discs and obtained a randomized online CF-coloring with $O(\log^3 n)$ colors with high probability. Recently, the bound $\Theta(\log n)$ just for these two special cases was obtained independently by Chen [3]. Our algorithm is more general and uses only $\Theta(\log n)$ colors; an interesting evidence to our algorithm being fundamentally different from the ones in [3, 4, 7], when used for the special case of hypergraphs that arise in geometry, is that it uses exponentially fewer random bits. The algorithms of [3, 7] use $\Theta(n \log n)$ random coin flips and our algorithm uses $O(\log n)$ random coin flips. Another interesting corollary of our result is that we obtain a randomized online coloring for k-inductive graphs with $O(k \log n)$ colors with high probability. This case was studied by Irani [6] who showed that the first-fit greedy algorithm achieves the same bound deterministically.

Deterministic online CF-coloring with recoloring: We initiate the study of online CF-coloring where at each step, in addition to the assignment of a color to the newly inserted point, we allow some recoloring of other points. The bi-criteria goal is to minimize the total number of recoloring done by the algorithm and the total number of colors used by the algorithm. We provide an algorithm for CF-coloring with respect to halfplanes using $O(\log n)$ colors and $O(n)$ recolorings.

2 Preliminaries

Definition 1 *Let $H = (V, \mathcal{E})$ be a hypergraph. For a subset $V' \subset V$ let $H(V')$ be the hypergraph $(V', \mathcal{E}')$ where $\mathcal{E}' = \{e \cap V' | e \in \mathcal{E}\}$. $H(V')$ is called the* induced hypergraph *on V'.*

Definition 2 *For a hypergraph $H = (V, \mathcal{E})$, the* Delaunay graph *$G(H)$ is the simple graph $G = (V, E)$ where the edge set E is defined as $E = \{(x, y) \mid \{x, y\} \in \mathcal{E}\}$ (i.e., G is the graph on the vertex set V whose edges consist of all hyperedges in H of cardinality two).*

Definition 3 *A simple graph $G = (V, E)$ is called k-degenerate (or k-inductive) for some positive integer k, if every (vertex-induced) subgraph of G has a vertex of degree at most k.*

Definition 4 *Let $k > 0$ be a fixed integer and let $H = (V, \mathcal{E})$ be a hypergraph on n vertices. Fix a subset $V' \subset V$. For a permutation π of V' such that $V' = \{v_1, ..., v_i\}$ (where $i = |V'|$) let $C_\pi(V') = \sum_{j=1}^{i} d(v_j)$, where $d(v_j) = |\{l < j | (v_j, v_l) \in G(H(\{v_1, ..., v_j\}))\}|$, that is, $d(v_j)$ is the number of neighbors of v_j in the Delaunay graph of the hypergraph induced by $\{v_1, ..., v_j\}$. Assume that $\forall V' \subset V$ and for all permutations $\pi \in S_{|V'|}$ we have $C_\pi(V') \leq k\,|V'|$. Then we say that H is* k-degenerate.

It is not difficult to see that our definition of a k-degenerate hypergraph is a generalization of that of a k-degenerate graph.

3 An online CF-coloring framework

Let $H = (V, E)$ be any hypergraph. Our goal is to define a framework that colors the vertices V in an online fashion. That is, the vertices of V are revealed by an adversary one at a time. At each time step t, the algorithm must assign a color to the newly revealed vertex v_t. This color cannot be changed in the future. The coloring has to be conflict-free for all the induced hypergraphs $H(V_t)$ $t = 1, \ldots, n$, where $V_t \subset V$ is the set of vertices revealed by time t.

For a fixed positive integer h, let $A = \{a_1, \ldots, a_h\}$ be a set of h auxiliary colors (not to be confused with the set of 'real' colors used for the CF-coloring: {1, 2, ...}). Let $f : \mathbb{N} \to A$ be some fixed function. We now define the framework that depends on the choice of the function f and the parameter h.

A table (to be updated online) is maintained where each entry i at time t is associated with a subset $V_t^i \subset V_t$ in addition to an auxiliary proper coloring of $H(V_t^i)$ with at most h colors. We say that $f(i)$ is the color that represents entry i in the table. At the beginning all entries of the table are empty. Suppose all entries of the table are updated until time $t-1$ and let v_t be the vertex revealed by the adversary at time t. The framework first checks if an auxiliary color can be assigned to v_t such that the auxiliary coloring of V_{t-1}^1 together with the color of v_t is a proper coloring of $H(V_{t-1}^1 \cup \{v_t\})$. Any (proper) coloring procedure can be used by the framework. For example a first-fit greedy one in which all colors in the order $a_1, \ldots, a_h$ are checked until one is found. If such a color cannot be found for v_t, then entry 1 is left with no changes and the process continues to the next entry. If however, such a color can be assigned, then v_t is added to the set V_{t-1}^1. Let c denote such an auxiliary color assigned to v_t. If this color is the same as $f(1)$

(the auxiliary color that is associated with entry 1), then the final color in the online CF-coloring of v_t is 1 and the updating process for the t-th vertex stops. Otherwise, if an auxiliary color cannot be found or if the assigned auxiliary color is not the same as the color associated with this entry, the updating process continues to the next entry. The updating process stops at the first entry i for which v_t is both added to V_t^i and the auxiliary color assigned to v_t is the same as $f(i)$. The color of v_t in the final conflict-free coloring is then set to i.

It is possible that v_t never gets a final color. In this case we say that the framework does not halt. However, termination can be guaranteed by imposing some restrictions on the auxiliary coloring method and the choice of the function f. For example, if first-fit is used for the auxiliary colorings at any entry and if f is the constant function $f(i) = a_1$, for all i, then the framework is guaranteed to halt for any time t. In section 4 we derive a randomized online algorithm based on this framework. It is not difficult to prove that the algorithm halts after a "small" number of entries with high probability (w.h.p.).

Lemma 1 *If the above framework halts for any vertex v_t then it produces a valid online CF-coloring.*

4 An online randomized CF-coloring algorithm

There is a randomized online CF-coloring in the oblivious adversary model that always produces a valid coloring and the number of colors used is related to the degeneracy of the underlying hypergraph in a manner described in theorem 2. Proofs will be included in a longer version of this paper.

Theorem 2 *Let $H = (V, \mathcal{E})$ be a k-degenerate hypergraph on n vertices. Then there exists a randomized online CF-coloring for H which uses at most $O(\log_{1+\frac{1}{4k+1}} n) = O(k \log n)$ colors with high probability.*

The algorithm is based on the framework of section 3. In order to define the algorithm, we need to choose: (a) the set of auxiliary colors of each entry, (b) the algorithm we use for the auxiliary coloring at each entry, and (c) the function f. We use: (a) auxiliary colors in $A = \{a_1, \ldots, a_{2k+1}\}$, (b) a first-fit algorithm for the auxiliary coloring, and (c) for each entry i, the representing color $f(i)$ is chosen uniformly at random from A. Our assumption on the hypergraph H (being k-degenerate) implies that at least half of the vertices up to time t that 'reached' entry i (but not necessarily added to entry i), and we denote by $X_{\geq i}^t$, have been actually given some auxiliary color at entry i (that is, $\left|V_t^i\right| \geq \frac{1}{2}\left|X_{\geq i}^t\right|$). This is easily implied by the fact that at least half of those vertices v_t have at most $2k$ neighbors in the Delaunay graph of the hypergraph induced by $X_{\geq i}^{t-1}$ (since the sum of these quantities is at most $k\left|X_{\geq i}^t\right|$ and since $V_t^i \subset X_{\geq i}^t$). Therefore since we have $2k+1$ colors available, there is always a free color to assign to such a vertex. The following lemma shows that if we use one of these 'free' colors then the updated coloring is indeed a proper coloring of the corresponding induced hypergraph as well.

Lemma 3 *Let $H = (V, \mathcal{E})$ be a k-degenerate hypergraph and let V_t^j be the subset of V at time t and at level j as produced by the above algorithm. Then for any j and t if v_t is assigned a color distinct from all its neighbors in the Delaunay graph $G(H(V_t^j))$ then this color together with the colors assigned to the vertices V_{t-1}^j is also a proper coloring of the hypergraph $H(V_t^j)$.*

Lemma 4 *Let $H = (V, \mathcal{E})$ be a hypergraph and let C be a coloring produced by the above algorithm on an online input $V = \{v_t\}$ for $t = 1, \ldots, n$. Let X_i (respectively $X_{\geq i}$) denote the random variable counting the number of points of V that were assigned a final color at entry i (respectively a final color at some entry $\geq i$). Let $\mathbf{E}_i = \mathbf{E}[X_i]$ and $\mathbf{E}_{\geq i} = \mathbf{E}[X_{\geq i}]$ (note that $X_{\geq i+1} = X_{\geq i} - X_i$). Then:*

$$\mathbf{E}_{\geq i} \leq \left(\frac{4k+1}{4k+2}\right)^{i-1} n$$

Lemma 5 *The expected number of colors used by the above algorithm is at most $\log_{\frac{4k+2}{4k+1}} n + 1$.*

Remark: In the above description of the algorithm, all the random bits are chosen in advance (by deciding the values of the function f in advance). However, one can be more efficient and calculate the entry $f(i)$ only at the first time we need to update entry i, for any i. Since at each entry we need to use $O(\log k)$ random bits and we showed that the number of entries used is $O(k \log n)$ w.h.p then the total number of random bits used by our algorithm is $O(k \log k \log n)$ w.h.p.

5 Application to Geometry

Our randomized algorithm has applications to CF colorings of certain geometric hypergraphs studied in [3, 4, 7]. We obtain the same asymptotic result as in [3], but with better constants of proportionality and much fewer random bits. An algorithm for intervals is given in [1]. When the hypergraph H is induced by points in the plane intersected by halfplanes or unit disks, we obtain online randomized algorithms that use $O(\log n)$ colors w.h.p. We summarize it as follows:

Lemma 6 *Let V be a finite set of n points in the plane and let $\mathcal{E}$ be all subsets of V that can be obtained by intersecting V with a halfplane. Then the hypergraph $H = (V, \mathcal{E})$ is 4-degenerate.*

Proof. The proof uses a few geometric lemmas. Details are omitted. □

Corollary 7 *Let H be the hypergraph as in lemma 6. Then the expected number of colors used by our randomized online CF-coloring applied to H is at most $\log_{\frac{18}{17}} n + 1$. Also the actual number of colors used is $O(\log_{\frac{18}{17}} n)$ with high probability. The number of random bits is $O(\log n)$ with high probability*

Proof. The proof follows immediately from lemma 6, lemma 5 and theorem 2. □

Proposition 8 *Let V be a finite set of n points in the plane and let $\mathcal{E}$ be all subsets of V that can be obtained by intersecting V with a unit disc. Then there exists a randomized online algorithm for CF-coloring H which uses $O(\log n)$ colors and $O(\log n)$ random bits with high probability.*

Proof. By a technique of Kaplan and Sharir [7] and Corollary 7. □

6 Deterministic online CF-coloring with recoloring

In this section we describe a deterministic algorithm for online CF-coloring points with respect to halfplanes that uses $O(\log n)$ colors and recolors $O(n)$ points. At every time instance t, the algorithm maintains the following invariant (V_t is the set of points that have appeared): All points (strictly) inside the convex hull of V_t are colored with the same special color, say '$\star$'. The set of points on the convex hull of V_t, denoted by $\mathrm{CH}(V_t)$, are colored with another set of colors, such that every set of consecutive points on the convex hull has a point with a unique color. The number of colors used in $\mathrm{CH}(V_t)$ must be logarithmic on t. It is not difficult to see that every subset of points of V_t induced by a halfplane contains a set of consecutive points, and thus the whole coloring is conflict-free. We describe how the algorithm maintains the above invariant. A new point v_{t+1} that appears at time $t+1$ is colored as follows: If it is inside the convex hull of V_t, then it gets color '$\star$'. If however it is in $\mathrm{CH}(V_{t+1})$, it might force some points that where in $\mathrm{CH}(V_t)$ to get inside the convex hull of V_{t+1}. In order to maintain the invariant, if there exist such points, they are recolored to '$\star$', and v_{t+1} is colored greedily, so that the coloring of $\mathrm{CH}(V_{t+1})$ is conflict-free (it can be proved that no new color is introduced). If, on the other hand, no points of $\mathrm{CH}(V_t)$ are forced into the convex hull, then $v_{t+1} \in \mathrm{CH}(V_{t+1})$ is colored with the algorithm that is used for intervals, given in [1], with a slight adaptation to address the closed curve nature of the convex hull. In that last case, in order to maintain logarithmic number of colors on t, one recoloring of a point in $\mathrm{CH}(V_{t+1})$ might be needed. The number of recolorings is guaranteed to be $O(n)$, because for every insertion, at most one recoloring happens on the new convex hull, and every point colored with '$\star$' stays with that color, because the convex hull never shrinks.

References

[1] A. Bar-Noy, P. Cheilaris, S. Olonetsky, and S. Smorodinsky. Weakening the online adversary just enough to get optimal conflict-free colorings for intervals. Manuscript, 2006.

[2] A. Bar-Noy, P. Cheilaris, and S. Smorodinsky. Conflict-free coloring for intervals: from offline to online. In *Proceedings of the 18th annual ACM symposium on Parallelism in algorithms and architectures (SPAA)*, pages 128–137, 2006.

[3] K. Chen. How to play a coloring game against a color-blind adversary. In *Proceedings of the 22nd Annual ACM Symposium on Computational Geometry (SoCG)*, pages 44–51, 2006.

[4] K. Chen, A. Fiat, H. Kaplan, M. Levy, J. Matoušek, E. Mossel, J. Pach, M. Sharir, S. Smorodinsky, U. Wagner, and E. Welzl. Online conflict-free coloring for intervals. *SIAM Journal on Computing*, 36(5):956–973, 2006.

[5] 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*, 33:94–136, 2003.

[6] S. Irani. Coloring inductive graphs on-line. *Algorithmica*, 11(1):53–72, 1994.

[7] H. Kaplan and M. Sharir. Online CF coloring for halfplanes, congruent disks, and axis-parallel rectangles. Manuscript, 2004.

[8] J. Pach and G. Tóth. Conflict free colorings. In *Discrete and Computational Geometry, The Goodman-Pollack Festschrift*, pages 665–671. Springer Verlag, 2003.

[9] S. Smorodinsky. *Combinatorial Problems in Computational Geometry*. PhD thesis, School of Computer Science, Tel-Aviv University, 2003.

[10] S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.