Fault-Tolerant Conflict-Free Colorings

Mohammad Ali Abam* Mark de Berg† Sheung-Hung Poon‡

1 Introduction

Background. Consider a cellular network consisting of a set of base stations, where the signal from a given base station can be received by clients within a certain distance from the base station. In general, these regions will overlap. For a client, this may lead to interference of the signals. Thus one would like to assign frequencies to the base stations such that for any client within reach of at least one base station, there is a base station within reach with a unique frequency (among all the ones within reach). The goal is to do this using only few distinct frequencies. Recently, Even *et al.* [5] introduced *conflict-free colorings*, as defined next, to model this problem.

Let $\mathcal{S}$ be a set of n objects, and let $\mathcal{R}$ be a, possibly infinite, family of ranges. In this paper, we only consider objects and ranges that are subsets of $\mathbb{R}^2$, or sometimes of $\mathbb{R}^1$. For a range $r \in \mathcal{R}$, let $\mathcal{S}(r)$ be the subset of objects from $\mathcal{S}$ intersecting the range r. A *conflict-free coloring (CF-coloring) of* $\mathcal{S}$ *with respect to* $\mathcal{R}$ is a coloring of $\mathcal{S}$ with the following property [5]: for any range $r \in \mathcal{R}$ for which $\mathcal{S}(r) \neq \emptyset$ there is an object $o \in \mathcal{S}(r)$ with a unique color in $\mathcal{S}(r)$, that is, with a color not used by any other object in $\mathcal{S}(r)$. Trivially, a conflict-free coloring always exists: just assign a different color to each object. However, one would like to find a coloring with only few colors. This is the conflict-free coloring problem. Note that if we take $\mathcal{S}$ to be a set of disks—namely, the regions within reach of each base station—and we take $\mathcal{R}$ to be the set of all points in $\mathbb{R}^2$, then we get exactly the range-assignment problem discussed earlier. However, other versions—for example the dual version, where $\mathcal{S}$ is a point set and $\mathcal{R}$ is the family of all disks—are interesting as well.

Related work. The CF-coloring problem for points with respect to disks was studied by Even *et al.* [5]. They showed that for this setting one can always find a CF-coloring using $O(\log n)$ colors, which is tight in the worst case. They also studies CF-colorings for points with respect to disks. Har-Peled and Smorodinsky [7] extended those results by considering other range spaces. In particular, they gave sufficient conditions for a range space to allow a CF-coloring with few colors. Recently, Smorodinsky [8] improved several results from [5] by providing deterministic coloring algorithms. Chen *et al.* [4] and Bar-Noy *et al.* [2] studied various CF-coloring problems in an on-line setting. Here the objects are given one by one, and each object has to be colored when it arrives, in such a way that the coloring remains conflict-free at all times.

Our results. Base stations in cellular networks are often not completely reliable: every now and then some base station may (temporarily or permanently) fail to function properly. This leads us to study fault-tolerant CF-colorings: colorings that remain conflict-free even after some objects are deleted from $\mathcal{S}$. More precisely, a *k-fault-tolerant CF-coloring (k-FTCF-coloring)* is a coloring that remains conflict-free after an arbitrary collection of k objects is deleted from $\mathcal{S}$. Thus a k-FTCF-coloring for $k = 0$ is simply a standard CF-coloring.

Such colorings for points with respect to disks were also studied by Abellanas *et al.* [1], who showed that any set of n points admits a k-FTCF coloring with respect to disks that uses $O(k \log n)$ colors—see Section 2. We show that this is tight, and we obtain upper and lower bounds on the worst-case number of colors needed in fault-tolerant colorings for various other types of range spaces. We also obtain results on *region-fault-tolerant CF-colorings (region-FTCF-colorings)*: colorings that remain conflict-free after the objects intersecting a geometric fault region are deleted from $\mathcal{S}$.

2 k-FTCF coloring of points with respect to disks

In this section we study conflict-free colorings of a point set $\mathcal{P} = \{p_1, \dots, p_n\}$ in the plane with respect to the family $\mathcal{D}$ of all disks in the plane.

The algorithm for $k = 0$ from Even et al. [5] for this case works as follows. Compute a maximal independent set of the Delaunay triangulation of $\mathcal{P}$; assign all points in the independent set color 1; recursively assign colors to the rest of the points, not using color 1 anymore. (Thus in the i-th recursive call, the color i is assigned to the points in the independent set.)

As observed by Abellanas *et al.* [1], generalizing this

*MADALGO Center, Department of Computer Science, Arhus University, abam@madalgo.au.dk. MADALGO is a Center of the Danish National Research Foundation.

†Department of Computer Science, TU Eindhoven, mdberg@win.tue.nl. Supported by the Netherlands' Organisation for Scientific Research under project no. 639.023.301.

‡Department of Computer Science, National Tsing Hua University, Hsin-Chu, Taiwan, spoon@cs.nthu.edu.tw.

algorithm to $k > 1$ is rather easy: we only need to replace the Delaunay triangulation by the set of all k-order Delaunay edges [6]. (Two points $p, q \in \mathcal{P}$ form a k-order Delaunay edge if and only if there is a disk containing p and q and at most k other points.) This results in the following theorem.

Theorem 1 *For any set $\mathcal{P}$ of n points in the plane, there is k-FTCF coloring with respect to disks that uses $O(k \log n)$ colors, and this is tight in the worst case.*

Proof. The correctness proof and analysis of the number of colors[1] of the method described above were already given by Aballenas *et al.* For completeness, we summarize their argument.

To show that the algorithm produces a k-FTCF coloring, let D be a disk containing at least one point. If every point inside D has a unique color, we are done. Otherwise, let i be the maximum color appearing in the disk at least twice. We shrink the disk until it contains exactly two points p and q of color i. Since p and q are in the independent set in the i-th step of the algorithm, there must be at least $k+1$ points in the shrunk disk. By the choice of i, all these points have different colors. Hence, even after deleting k of those points we still have a unique color inside D. To prove the bound on the number of colors, we use that there are $O(kn)$ k-order Delaunay edges [6], which implies there is an independent set of size $\Omega(n/k)$. If $C(n)$ denotes the number of colors used by the algorithm, we therefore have $C(n) = 1 + C(n - \Omega(n/k))$. Hence, $C(n) = O(k \log n)$.

Next we prove the lower bound (which was not given in [1]). Consider a set $\mathcal{P}$ of n points on the x-axis. Obviously, there must be $k+1$ points with a unique color. We split the points into two roughly equal size subsets, A and B, such that all points in A are to the left to all points in B. One subset, say A, must contain at least $\lceil (k+1)/2 \rceil$ points with a unique color. In particular, those colors are not used in B and we can recurse on B. If $C(n)$ denotes the minimum number of colors needed for $\mathcal{P}$, we thus have $C(n) \geq (k+1)/2 + C(n/2)$. Hence, $C(n) = \Omega(k \log n)$. □

3 k-FTCF coloring of disks with respect to points

We now turn our attention to the case where we want to color a set of disks with respect to points. We start with the 1-dimensional version of this problem, where the set of objects is a set $\mathcal{I} = \{I_1, \ldots, I_n\}$ of n intervals on the real line. Our algorithm consists of two phases.

Phase 1: We process the intervals one by one, as follows. To process I_j, we check if every point in I_j is contained in at least $k+1$ intervals from the current set $\mathcal{I}$. If this is the case, we assign color 0 to I_j and remove I_j from $\mathcal{I}$, otherwise I_j stays in $\mathcal{I}$ and does not get a color yet.

We claim that after Phase 1, every point q is contained in at most $2k+2$ intervals. Indeed, the $k+1$ intervals containing q and extending the farthest to the left, and the $k+1$ intervals containing q and extending the farthest to the right, must cover every other interval containing q. Hence, any such other interval will be removed in Phase 1.

Phase 2: Now we color the remaining intervals, only using colors from the set $S = \{1, \ldots, \lceil (3k+3)/2 \rceil + 1\}$. To this end, we sweep from left to right. When the sweep arrives at the left endpoint of an interval I, we assign a color to I, as follows. Let S_I be a set of *forbidden colors* for I in the sense that if we assign one of them to I, then the collection of intervals colored so far is not a k-FTCF anymore. We take an arbitrary color from $S \backslash S_I$ and assign it to I.

Theorem 2 *For any set of n intervals on the real line, there is a k-FTCF coloring with respect to points that uses $\lceil (3k+3)/2 \rceil + 2$ colors. Moreover, for any $n \geq 2k+2$, there is a set of n intervals such that any k-FTCF coloring needs at least $\lceil (3k+3)/2 \rceil$ colors.*

Proof. To prove the upper bound, consider the algorithm described above. The intervals with color 0 can be ignored: any point contained in such an interval is contained in at least $k+1$ intervals from the set $\mathcal{I}$ processed in Phase 2, and we will show that Phase 2 produces a k-FTCF coloring. To show this, it suffices to argue that the algorithm does not get stuck. Thus we must prove that $S \backslash S_I$ is not empty when some interval I is processed. Let p be the left endpoint of I. If p is contained in at most k other intervals, we are done, since $|S_I| \leq k$ in this case. Otherwise, let $q \in I$ be the leftmost point that is contained in *exactly* $k+1$ intervals with unique colors. Such a point exists, because (i) p is contained in at least $k+1$ uniquely colored intervals, (ii) there is a point on I contained in at most k intervals due to Phase 1, and (iii) the current k-FTCF coloring is valid. Since at most $2k+2$ intervals contain q and exactly $k+1$ of these intervals have unique colors, the number of colors which is used to color intervals containing q is at most $k+1+\lceil (k+1)/2 \rceil = \lceil (3k+3)/2 \rceil$. Hence, there is at least one color, j, not used by an interval containing q. We claim that j is not forbidden. Indeed, points in I to the left of q have at least $k+2$ unique colors by the choice of q, so we do not have to worry about them. Moreover, an interval that already has a color and contains a point to the right of q also contains q, so such an interval will not have color j. Hence, j is not forbidden.

[1] Aballenas *et al.* give a bound of $\log n / \log(24k/(24k-1))$ on the number of colors, but it is easy to see that this is $O(k \log n)$.

To prove the lower bound, it suffices to look at the case $n = 2k+2$. Let A be a set of $k+1$ intervals starting at $x = 1$ and ending at $x = 2$, and let B be a set of $k+1$ intervals starting at $x = 2$ and ending $x = 3$. Consider a k-FTCF coloring of $A \cup B$. At point $x = 2$, there must at least $k+1$ unique colors. Therefore, one of the sets, say A, contains at least $\lceil (k+1)/2 \rceil$ unique colors. Consider the point $x = 3$. Since exactly $k+1$ intervals contain p, then all of them must have different colors. Therefore, there are at least $k+1+\lceil (k+1)/2 \rceil$ different colors. □

Now we go back to the 2D problem, namely k-FTCF coloring of a set $\mathcal{D} = \{D_1, \ldots, D_n\}$ of n disks with respect to points. Our algorithm is based on a generalization of admissible subsets [7], defined as follows. We say that $\hat{\mathcal{D}}$ is an *admissible subset* of $\mathcal{D}$ if for every point $p \in \mathbb{R}^2$ at least one of the following conditions holds:

1. $p \notin \bigcup \hat{\mathcal{D}}$.
2. $p \in \bigcup \hat{\mathcal{D}}$, but only one of the disks in $\hat{\mathcal{D}}$ contains p.
3. $p \in \bigcup \hat{\mathcal{D}}$, and there are k disks in $\mathcal{D} \backslash \hat{\mathcal{D}}$ containing p.

We will show that for a set $\mathcal{D}$ there is an admissible subset of size $\Omega(n/k)$. Based on this fact, our algorithm is as follows. Compute an admissible subset, assign to all disks in the admissible subset the color 1, and color the remaining disks recursively (where in the i-th step we assign the color i). If $C(n)$ is the number of colors used by the algorithm, we have $C(n) = 1 + C(n - \Omega(n/k))$ which gives us $C(n) = O(k \log n)$. The following example shows that our algorithm is tight. Consider n disks with radius 1 whose centers are on a line and that have a common point on the line. It is easy to see that for any m $(1 \leq m \leq n)$ consecutive disks there is a point just contained in those disks. Therefore, the same lower-bound proof given in Section 2 can be applied here.

Theorem 3 *For any set of n disks in the plane, there is a k-FTCF coloring with respect to points that uses $O(k \log n)$ colors, and this bound is tight in the worst case.*

Proof. *(Sketch.)* The lower bound has been described above. To prove the upper bound, we need to prove our claim that we can always find an admissible subset of size $\Omega(n/k)$. The proof of this fact is similar to the proof of Har-Peled and Smorodinsky [7] for the non-fault-tolerant case. Here we sketch the basic steps in our proof.

Following Har-Peled and Smorodinsky, we randomly and independently color each disk in $\mathcal{D}$ black or white, each with probability 1/2. We say a point p is *unsafe* if it is contained in at least two black disks and at most k white disks. We construct a graph G over the black disks, connecting two black disks if there is an unsafe point in their intersection. Note that any independent set of G is an admissible subset. As [7] showed, G has at least $n/3$ vertices with high probability. We show that G has $O(kn)$ edges, which implies there is an admissible subset of size $\Omega(n/k)$. To show that G has $O(kn)$ edges, we proceed as follows. For any point p in the plane, let $d(p)$ denote the number of disks containing p. The probability that p is unsafe (for points p with $d(p) > 1$) is $(1/2^{d(p)}) \sum_{i=0}^{k} \binom{d(p)}{i}$. We can use this to bound the expected number of edges created by unsafe points, which we then use to bound the overall number of edges in G by $O(kn)$. □

4 Region-FTCF coloring

Let $\mathcal{F}$ be a family of regions, which we call the *fault regions*. For a fault region $F \in \mathcal{F}$ and a point set $\mathcal{P}$, we define $\mathcal{P} \ominus F$ to be the set of points that remains after the points inside F have been removed from $\mathcal{P}$. An $\mathcal{F}$-FTCF coloring of $\mathcal{P}$ with respect to disks is a CF-coloring of $\mathcal{P}$ with respect to disks such that for any $F \in \mathcal{F}$, the coloring is a CF-coloring of $\mathcal{P} \ominus F$ with respect to disks. Unfortunately, when $\mathcal{F}$ is the set of convex regions, some sets $\mathcal{P}$ require n colors in any $\mathcal{F}$-FTCF coloring with respect to disks. To see this, consider a set $\mathcal{P}$ of n points on a line. For any two points $p, q \in \mathcal{P}$, there is a convex region $\mathcal{F}$ (which is an interval) containing all points between p and q. Points p and q are adjacent in $\mathcal{P} \ominus F$, and so they must have different colors.

The concept of region-FTCF coloring can also be defined for disks with respect to points. Let $\mathcal{D}$ be a set of n disks, and define $\mathcal{D} \ominus F$ to be set of disks that remains after the disks from $\mathcal{D}$ whose centers are inside F have been removed from $\mathcal{D}$. An $\mathcal{F}$-FTCF-coloring of $\mathcal{D}$ with respect to points is a CF-coloring of $\mathcal{D}$ with respect to points such that for any $F \in \mathcal{F}$, the coloring is a CF-coloring of $\mathcal{D} \ominus F$ with respect to points.

Unfortunately this variant does not allow a good solution either: there are sets of disks that do not admit a $\mathcal{F}$-FTCF coloring with few colors, even if $\mathcal{F}$ is the set of half-planes and the radii of the disks are equal. Consider n disks with radius 1 whose centers lie on a parabola very close to each other. Let $p_1, \ldots, p_n$ be the centers of the disks. For any i and j, there is a half-plane just containing centers $p_{i+1}, \ldots, p_{j-1}$. After removing the corresponding disks, there is a point in the plane just contained in the disks whose centers are p_i and p_j, which implies those disks must have different colors. Therefore, every two disks must have different colors, which means we need n colors.

However, we can get a coloring with few colors for the 1-dimensional version of this problem. Here we have a set $\mathcal{I}$ of n intervals on the real line, and the goal is to find an $\mathcal{F}$-FTCF coloring with respect to points, where

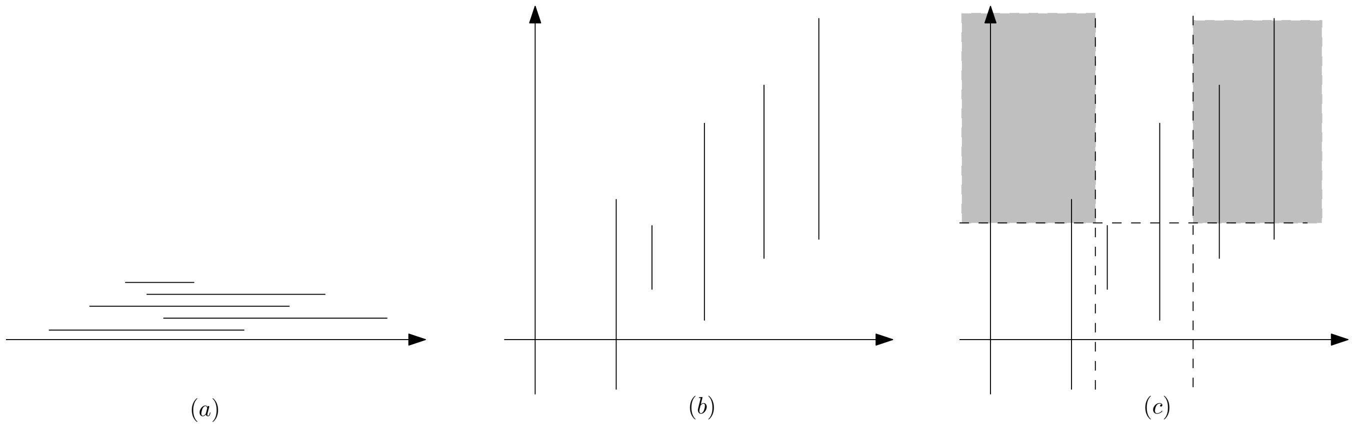

Figure 1: (a) A set of intervals (b) corresponding vertical segments in the xy-plane (c) cone regions.

$\mathcal{F}$ is the set of intervals. How to obtain such a coloring is explained next.

First we assume that all intervals have a common point (Fig. 1(a)). We map each interval $[p, q]$ with center c to the vertical segment in the xy-plane starting at (c, p) and ending at (c, q), as depicted in Fig. 1(b). Since the intervals have a common point, there is a horizontal line intersecting all vertical segments. A fault region $I = [a, b] \in \mathcal{F}$ is mapped to a vertical slab in the xy-plane, whose boundaries intersect the x-axis in $x = a$ and $x = b$. Every point $p \in \mathbb{R}^1$ is mapped to the horizontal line $y = p$ in the xy-plane. Now the problem reduces to the following. The goal is to color the vertical segments such that for every region depicted in Fig. 1(c) in gray—namely every region consisting of two axis-aligned cones whose apexes have the same y-coordinates—there is a segment with a unique color intersecting the region. To find such a CF-coloring, we apply the general approach: We construct a graph G over vertical segments, as follows. We connect two vertical segments if there is such a region intersecting just these two segments. It is easy to show that G has $O(n)$ edges. Therefore, it has an independent set of size $\Omega(n)$. We color all vertical segments in the independent set with color 1 and recursively color the rest of intervals. Since the size of independent set is $\Omega(n)$, the number of color used by the algorithm is $O(\log n)$.

If the intervals don't have a common point, we first construct an interval tree on the segments. Then we apply the above algorithm for the intervals stored in each internal node of the interval tree, where for each node we use the same set of colors. We then change the color assigned to each interval as follows: suppose it received color i and it is stored in a node in the interval tree at level ℓ. Then the new color of the interval is (i, ℓ). Since the tree has depth $O(\log n)$, the total number of colors used by the algorithm is $O(\log^2 n)$. Moreover, any two intervals stored at different nodes on the same level are disjoint, so this produces a valid coloring.

Theorem 4 *Let $\mathcal{F}$ be the family of all intervals on the real line. For any set of n intervals on the real line, there is an $\mathcal{F}$-FTCF coloring with respect to points that uses $O(\log^2 n)$ colors.*

Acknowledgement

We thank Bettina Speckmann for stimulating discussions during the early stages of our research.

References

[1] M. Abellanas, P. Bose, J. García, F. Hurtado, M. Nicolás, and P.A. Ramos. On properties of higher-order Delaunay graphs with applications. In *Abstr. 21st Europ. Workshop Comput. Geom.*, pages 119–122, 2005.

[2] A. Bar-Noy, P. Cheilaris and S. Smorodinsky. Conflict-free coloring for intervals: from offline to online. In *Proc. 18th Annu. ACM Sympos. Parallelism Algorithms Architectures (SPAA)*, pages 128–137, 2006.

[3] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[4] K. Chen, A. Fiat, H. Kaplan, M. Levy, J. Matousek, E. Mossel, J. Pach, M. Sharir, S. Smorodinsky, U. Wagner and E. Welzl. Online conflict-free coloring for intervals. *SIAM J. Comput.*, in press.

[5] G. Even, Z. Lotker, D. Ron and S. Smorodinsky. Conflict-freee colorings of simple geometric regions with applications to frequency assignment in cellular networks. *SIAM J. Comput.* 33:94–136, 2004.

[6] J. Gudmundsson, M.H. Hammar and M.J. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom. Theory Appl.* 23:85–98, 2002.

[7] S. Har-Peled and S. Smorodinsky. Conflict-Free coloring of points and simple regions in the plane. *Discr. Comput. Geom.* 34:47–70, 2005.

[8] S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *Proc. 17th Annu. ACM-SIAM Sympos. Discr. Algo. (SODA)*, pages 316–323, 2006.