Strong conflict-free coloring for intervals

Panagiotis Cheilaris[1], Luisa Gargano[2],
Adele A. Rescigno[2], and Shakhar Smorodinsky[3]

[1] Faculty of Informatics, Università della Svizzera Italiana, Switzerland.
[2] Dipartimento di Informatica, University of Salerno, 84084 Fisciano (SA), Italy.
[3] Mathematics Department, Ben-Gurion University, Be'er Sheva 84105, Israel.

Abstract. We consider the k-strong conflict-free (k-SCF) coloring of a set of points on a line with respect to a family of intervals: Each point on the line must be assigned a color so that the coloring is conflict-free in the following sense: in every interval I of the family there are at least k colors each appearing exactly once in I.
We first present a polynomial time algorithm for the general problem; the algorithm has approximation ratio 2 when $k = 1$ and $5 - \frac{2}{k}$ when $k > 1$ (our analysis is tight). In the special case of a family that contains all possible intervals on the given set of points, we show that a 2-approximation algorithm exists, for any $k \geq 1$. We also show that the problem of deciding whether a given family of intervals can be 1-SCF colored with at most q colors has a quasipolynomial time algorithm.

1 Introduction

A coloring of the vertices of a hypergraph is said to be conflict-free if every hyperedge contains a vertex whose color is unique among those colors assigned to the vertices of the hyperedge. We denote by $\mathbb{Z}^+$ the set of positive integers and by $\mathbb{N}$ the set of non-negative integers.

Definition 1. ***(CF coloring)*** *A conflict-free vertex coloring of a hypergraph* $H = (V, \mathcal{E})$ *is a function* $C: V \to \mathbb{Z}^+$ *such that for each* $e \in \mathcal{E}$ *there exists a vertex* $v \in e$ *such that* $C(u) \neq C(v)$ *for any* $u \in e$ *with* $u \neq v$.

Conflict-free coloring was first considered in [8]. It was motivated by a frequency assignment problem in cellular networks. Such networks consist of fixed-position *base stations*, each assigned a fixed frequency, and roaming *clients*. Roaming clients have a range of communication and come under the influence of different subsets of base stations. This situation can be modeled by means of a hypergraph whose vertices correspond to the base stations and whose hyperedges correspond to the different subsets of base stations corresponding to ranges of roaming agents. A conflict-free coloring of such a hypergraph corresponds to an assignment of frequencies to the base stations, which enables any client to connect to one of the base stations (holding the unique frequency in the client's range) without interfering with the other base stations. The goal is to minimize the number of assigned frequencies. Due to both its practical motivations and its theoretical interest, conflict-free coloring has been the subject of several papers; a survey of results in the area is given in [14].

CF-coloring also finds application in RFID (Radio Frequency Identification) networks. RFID allows a reader device to sense the presence of a nearby object by reading a tag attached to the object itself. To improve coverage, multiple RFID readers can be deployed in an area. However, two readers trying to access a tagged device simultaneously might cause mutual interference. It can be shown that CF-coloring of the readers can be used to assure that every possible tag will have a time slot and a single reader trying to access it in that time slot [14].

The notion of *k-strong* CF coloring (k-SCF coloring), first introduced in [2], extends that of CF-coloring. A k-SCF coloring is a coloring that remains conflict-free after an arbitrary collection of $k-1$ vertices is deleted from the set [1]. In the context of cellular networks, a k-SCF coloring implies that for any client in an area covered by at least k base stations, there always exist at least k different frequencies the client can use to communicate without interference. Therefore, up to k clients can be served at the same location, or the system can deal with malfunctioning of at most $k-1$ base stations per location. Analogously, in the RFID networks context, a k-SCF coloring corresponds to a fault-tolerant activation protocol, i.e., every tag can be read as long as at most $k-1$ readers are broken. A CF-coloring is just a 1-SCF coloring.

We will allow the coloring function $C\colon V \to \mathbb{Z}^+$ to be a partial function (i.e., some vertices are not assigned a color). Alternatively, we can use a special color '0' given to vertices that are not assigned any positive color and obtain a total function $C\colon V \to \mathbb{N}$. Then, we arrive at the following definition.

Definition 2. ***(k-SCF coloring)*** *Let $H = (V, \mathcal{E})$ be a hypergraph and $k \in \mathbb{Z}^+$. A coloring $C\colon V \to \mathbb{N}$ is called a k-strong conflict-free coloring if for every $e \in \mathcal{E}$ at least $\min\{|e|, k\}$ positive colors are unique in e, namely there exist $c_1, \dots, c_{\min\{|e|,k\}} \in \mathbb{Z}^+$ such that $|\{v \mid v \in e,\ C(v) = c_i\}| = 1$, for $i = 1, \dots,$ $\min\{|e|, k\}$. The goal is to minimize the number of positive colors in the range of the k-SCF coloring function C. We denote by $\chi^*_k(H)$ the smallest number of positive colors in any possible k-SCF coloring of H.*

Remark 1. We claim that this variation of conflict-free coloring, with the partial coloring function or the placeholder color '0', is interesting from the point of view of applications. A vertex with no positive color assigned to it can model a situation where a base station is not activated at all, and therefore the base station does not consume energy. One can also think of a bi-criteria optimization problem where a conflict-free assignment of frequencies has to be found with small number of frequencies (in order to conserve the frequency spectrum) and few activated base stations (in order to conserve energy). It is not difficult to see that a partial SCF coloring with q positive colors implies always a total SCF coloring with $q+1$ positive colors.

SCF-coloring points with respect to intervals. Several authors recently focused on the special case of CF coloring n collinear points with respect to the family of all intervals. The problem can be modeled in the hypergraph

$$H_n = ([n], \mathcal{I}^{[n]}) \text{ with } [n] = \{1, \ldots, n\} \text{ and } \mathcal{I}^{[n]} = \{\{i, \ldots, j\} \mid 1 \le i \le j \le n\},$$

where each (discrete) interval is a set of consecutive points.

Conflict-free coloring for intervals models the assignment of frequencies in a chain of unit disks; this arises in approximately unidimensional networks as in the case of agents moving along a road. Moreover, it is important because it plays a role in the study of conflict-free coloring for more complicated cases, as for example in the general case of CF coloring of unit disks [8, 11].

While some papers require the conflict-free property for all possible intervals on the line, in many applications good reception is needed only at some locations, i.e., it is sufficient to supply only a given subset of the cells of the arrangement of the disks [10]. In the context of channel assignment for broadcasting in a wireless mesh network, it can occur that, at some step of the broadcasting process, sparse receivers of the broadcast message are within the transmission range of a linear sequence of transmitters. In this case only part of the cells of the linear arrangements of disks representing the transmitters are involved [12, 15]. In this work we consider the k-strong conflict-free coloring of points with respect to an arbitrary family of intervals. Hence, in the remainder, we consider subhypergraphs of H_n. We shall refer to these subhypergraphs of the form $H = ([n], \mathcal{I})$, where $\mathcal{I} \subseteq \mathcal{I}^{[n]}$, as *interval hypergraphs* and to H_n as the *complete interval hypergraph.*

Conflict-free coloring the complete interval hypergraph was first studied in [8], where it was shown that $\chi_1^*(H_n) = \lfloor \log n \rfloor + 1$ †. The on-line version of the CF coloring problem for complete interval hypergraphs, where points arrive one by one and the coloring needs to remain CF all the time, has been subsequently considered in [3–5].

The problem of CF-coloring the points of a line with respect to an arbitrary family of intervals is studied in [10]. The k-SCF coloring problem was first considered in [2] and has since then been studied in various papers under different scenarios, we refer the reader to [14] for more details on the subject. Recently, the minimum number of colors needed for k-SCF coloring the complete interval hypergraph H_n has been studied in [7], where the exact number of needed colors for $k = 2$ and $k = 3$ has been obtained. Horev et al. show that H_n admits a k-SCF coloring with $k \log_2 n$ colors, for any k [9].

Our results. In Section 2, we give an algorithm which outputs a k-SCF coloring of the points of the input interval hypergraph H, for any fixed value of $k \ge 1$. The algorithm has an approximation factor $5 - 2/k$ in the case $k \ge 2$ (approximation factor 2 in the case $k = 1$); moreover, it optimally uses k colors if for any $I, J \in \mathcal{I}$, interval I is not a subset of J and they differ in at least k points. In Section 3, we consider the problem of k-SCF coloring the complete interval hypergraph H_n. We give a very simple k-SCF coloring algorithm for H_n that uses $k\left(\left\lfloor \log \left\lceil \frac{n}{k} \right\rceil \right\rfloor + 1\right)$ colors and show a lower bound of $\left\lceil \frac{k}{2} \right\rceil \left\lceil \log \frac{n}{k} \right\rceil$ colors. In section 4, we show that

† Unless otherwise specified, all logarithms are in base 2.

the decision problem whether a given interval hypergraph can be CF-colored with at most q colors has a quasipolynomial time algorithm.

Notation. Through the rest of this paper we consider interval hypergraphs on n points. Given $I \in \mathcal{I}$, we denote the *leftmost* (minimum) and the *rightmost* (maximum) of the points of the interval I by $\ell(I) = \min\{p \mid p \in I\}$ and $r(I) = \max\{p \mid p \in I\}$, respectively. We will use the following order relation on the intervals of $\mathcal{I}$.

Definition 3. (Intervals ordering) *For all* $I, J \in \mathcal{I}$,

$$I \prec J \iff (r(I) < r(J)) \text{ or } (r(I) = r(J) \text{ and } \ell(I) > \ell(J)).$$

$I \in \mathcal{I}$ is called the i-th interval in $\mathcal{I}$ if $\mathcal{I} = \{I_1, \ldots, I_m\}$, $I_1 \prec I_2 \prec \cdots \prec I_m$, and $I = I_i$.

Given a family $\mathcal{I}$, the subfamily of intervals of $\mathcal{I}$ that are not contained in I and whose rightmost (resp. leftmost) point belongs to I is denoted by $\mathcal{L}_{\mathcal{I}}(I)$ (resp. $\mathcal{R}_{\mathcal{I}}(I)$), that is

$$\mathcal{L}_{\mathcal{I}}(I) = \{J \in \mathcal{I} \mid J \not\subseteq I,\ r(J) \in I\} \text{ and } \mathcal{R}_{\mathcal{I}}(I) = \{J \in \mathcal{I} \mid J \not\subseteq I,\ \ell(J) \in I\}$$

Clearly, $J \prec I$ (resp. $I \prec J$) for any $J \in \mathcal{L}_{\mathcal{I}}(I)$ (resp. $J \in \mathcal{R}_{\mathcal{I}}(I)$) with $J \neq I$.

We denote by $\mathcal{M}_{\mathcal{I}}(I)$ the subfamily of all the intervals contained in $I \in \mathcal{I}$

$$\mathcal{M}_{\mathcal{I}}(I) = \{J \mid J \in \mathcal{I},\ J \subset I\}.$$

2 A k-SCF coloring algorithm

We present an algorithm for k-SCF coloring any interval hypergraph $H = ([n], \mathcal{I})$. We prove that our algorithm achieves an approximation ratio 2 if $k = 1$ and an approximation ratio $5 - \frac{2}{k}$ if $k \geq 2$; we show that the algorithm is optimal when $\mathcal{I}$ consists of intervals differing in at least k points and not including any other interval in $\mathcal{I}$. We say that an interval $I \in \mathcal{I}$ is k**-colored** under coloring C if its points are colored with at least $\min\{|I|, k\}$ unique positive colors, where a color c is unique in I if there is exactly one point $p \in I$ such that $C(p) = c$. The k-SCF coloring algorithm, k-COLOR($\mathcal{I}$), is given in Fig. 1. The number of colors is upper bounded by the number of iterations performed by the algorithm times $c(k)$, where $c(k) = 2k + \lceil k/2 \rceil - 1$.

At each step t of the algorithm a subset P_t of points of $[n]$ is selected (through algorithm SELECT), then $c(k)$ colors are assigned in cyclic sequence to the ordered sequence (from the minimum to the maximum) of the selected points. The intervals that are k-colored at the end of the step t are inserted in the set $\mathcal{X}_t$ and discarded. The algorithm ends when all the intervals in $\mathcal{I}$ have been discarded. At each step t a new set of $c(k)$ colors is used.
A point $p \in [n]$ can be re-colored several times during different steps of the k-COLOR algorithm; its color at the end of algorithm is the last assigned one.

```
k-COLOR(I):
Set t = 1.
I_1 = I.   [I_t is the set of intervals not k-colored by the beginning of step t]
X_1 = ∅.   [X_t ⊂ I_t contains the intervals that become k-colored during step t]
while I_t ≠ ∅
      Execute the following step t
        1.  Let P_t = {p_0, p_1, ..., p_{n_t}} be the set returned by SELECT(I_t)
        2.  for i = 0 to n_t
                 Assign to p_i color c_i = (t − 1)c(k) + (i mod c(k)) + 1
        3.  for each I ∈ I_t
                 if I is k-colored then  X_t = X_t ∪ {I}
        4.  I_{t+1} = I_t \ X_t
        5.  t = t + 1

SELECT(I_t):
Set P_t = ∅.   [P_t represents the set of selected points at step t]
for each I ∈ I_t by increasing order according to relation ≺ [see Def.3]
     if |I ∩ P_t| < min{|I|, k} then
          1. Let P_t(I) be the set of largest min{|I|, k} − |I ∩ P_t| points of I \ P_t
          2. P_t = P_t ∪ P_t(I)
Return P_t
```

Fig. 1. The k-SCF coloring algorithm for $H = ([n], \mathcal{I})$

The algorithm SELECT($\mathcal{I}_t$) considers intervals in $\mathcal{I}_t$ according to the $\prec$ relation and selects points so that P_t has at least $\min\{|I|, k\}$ points in each interval. Namely, if I is the i-th interval, then it is considered at the i-th iteration of the **for** loop and if less than $\min\{|I|, k\}$ points of I have been already selected, then the algorithm adds the missing $\min\{|I|, k\} - |I \cap P_t|$ points of I to P_t (such points are the largest unselected ones of I).

Example 1. Consider $H = ([23], \mathcal{I})$, where $\mathcal{I}$ is the set of 13 intervals given in Fig. 2. Run k-COLOR($\mathcal{I}$) with $k = 2$; hence $c(2) = 4$ colors are used at each iteration. Initially, $\mathcal{I}_1 = \mathcal{I}$ and SELECT($\mathcal{I}_1$) returns $P_1 = \{3, 4, 7, 8, 9, 11, 12, 14, 15, 17, 18, 19, 20, 22, 23\}$ whose points are colored with c_1, c_2, c_3, c_4 in cyclic sequence. Only 3 intervals remain in $\mathcal{I}_2$; all the others are in $\mathcal{X}_1$, being 2-colored at the end of step 1. SELECT($\mathcal{I}_2$) returns $P_2 = \{14, 15, 23\}$ and these points are colored with c_5, c_6, c_7. Now $\mathcal{I}_3 = \mathcal{I}_2 \setminus \mathcal{X}_2 = \emptyset$ and the algorithm ends.

In the following, we will sketch a proof of the following theorem.

Theorem 1. *Algorithm k-COLOR($\mathcal{I}$) is a polynomial k-SCF coloring algorithm that uses less than $\frac{c(k)}{\lceil k/2 \rceil}\chi^*_k(H)$ colors on the interval hypergraph $H = ([n], \mathcal{I})$.*

2.1 Correctness of Algorithm k-COLOR

We denote by P_t the set of points returned by SELECT($\mathcal{I}_t$).

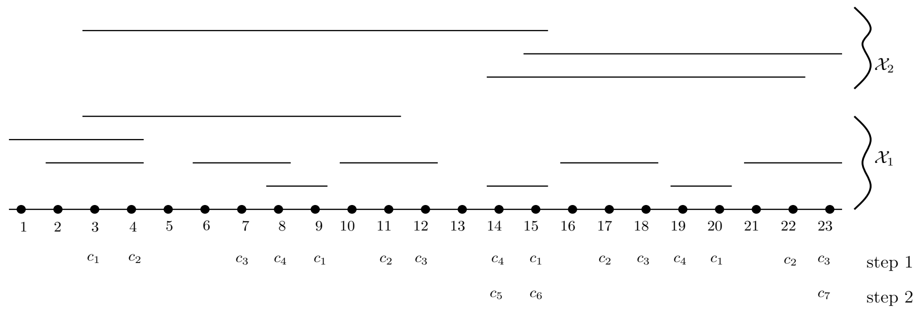

Fig. 2. Example coloring by k-COLOR for $k = 2$

Lemma 1. *Let $I \in \mathcal{I}_t$, $t \geq 1$.*
a) $|I \cap P_t| \geq \min\{|I|, k\}$; **b)** $I \in \mathcal{X}_t$ *if* $|I \cap P_t| \leq 4k - 2$; **c)** $|I| \geq k$, *for* $t \geq 2$.

Lemma 2. *If $I \in \mathcal{X}_t$ then $\mathcal{M}_{\mathcal{I}_t}(I) \subseteq \mathcal{X}_t$.*

Lemma 3. *If $\mathcal{M}_{\mathcal{I}_t}(I) = \emptyset$, then $|I \cap P_t| \leq 2k - 1$.*

With the help of the above, we can prove correctness of the algorithm.

Theorem 2. *Given interval hypergraph $H = ([n], \mathcal{I})$, algorithm k-COLOR($\mathcal{I}$) produces a k-SCF coloring of H.*

Proof. We show by induction the following statement for each $t \geq 1$: *At the end of step t of algorithm k-COLOR($\mathcal{I}$), each interval $I \in \bigcup_{i=1}^{t} \mathcal{X}_i$ is k-colored.*
For $t = 1$, the statement trivially follows. Assume the statement be true for each $i \leq t - 1$ and $t \geq 2$. We prove that it holds for t. Notice that, by c) of Lemma 1, for any $I \in \mathcal{I}_t$ it holds $\min\{|I|, k\} = k$. Clearly, if $I \in \mathcal{X}_t$, then I is k-colored by definition of $\mathcal{X}_t$. Consider then $I \in \mathcal{X}_i$ for some $i \leq t - 1$. By the inductive hypothesis I is k-colored at the end of step $t - 1$. By Lemma 2 we know that $\mathcal{M}_{\mathcal{I}_i}(I) \subseteq \mathcal{X}_i$; which implies that $\mathcal{M}_{\mathcal{I}_t}(I) = \emptyset$. Moreover, by Lemma 3, we have $|I \cap P_t| \leq 2k - 1 < c(k)$. This means that even if some points are recolored, all the assigned colors will be unique in I. □

2.2 Analysis of algorithm k-COLOR($\mathcal{I}$)

In this section we evaluate the approximation factor of the algorithm k-COLOR. We first give a lower bound tool (see also [7]). Since the vertex set $[n]$ is usually implied, we use the shorthand notation $\chi_k^*(\mathcal{I}) = \chi_k^*(([n], \mathcal{I}))$.

Theorem 3. *Let $I_1, I_2, I \in \mathcal{I}$ with $I_1, I_2 \subset I$ and $I_1 \cap I_2 = \emptyset$. Let χ_1 (resp. χ_2) be the number of colors used by an optimal k-SCF coloring of $\mathcal{M}_{\mathcal{I}}(I_1)$ (resp. $\mathcal{M}_{\mathcal{I}}(I_2)$). Then the number of colors used by any optimal k-SCF coloring of $\mathcal{M}_{\mathcal{I}}(I)$ is*

$$\chi_k^*(\mathcal{M}_{\mathcal{I}}(I)) \geq \begin{cases} \max\{\chi_1, \chi_2\} & \textit{if } k \leq |\chi_2 - \chi_1|, \\ \max\{\chi_1, \chi_2\} + \left\lceil \frac{k - |\chi_2 - \chi_1|}{2} \right\rceil & \textit{otherwise.} \end{cases}$$

Corollary 1. *Let $I_1, I_2, I \in \mathcal{I}$ with $I_1 \subset I$, $I_2 \subset I$ and $I_1 \cap I_2 = \emptyset$. If both $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I_1))$ and $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I_2))$ are at least χ, then the number of colors used in any optimal k-SCF coloring of $\mathcal{M}_{\mathcal{I}}(I)$ is $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I)) \geq \chi + \lceil k/2 \rceil$.*

In order to assess the approximation factor of the k-COLOR algorithm, we need the following result on the family $\mathcal{I}_t$ of intervals that still need to be k-colored after step t of the algorithm.

Lemma 4. *For each $I \in \mathcal{I}_t$, there exist at least two intervals $I', I'' \in \mathcal{I}_{t-1}$ such that $I', I'' \subset I$ and $I' \cap I'' = \emptyset$.*

In the following we assume that there exists at least an interval $I \in \mathcal{I}$ with $|I| \geq k$. Notice that if $|I| < k$ for each $I \in \mathcal{I}$, then each interval in $\mathcal{I}$ is k-colored after the first step of the algorithm k-COLOR($\mathcal{I}$) (even using for $c(k)$ the smaller value $\max\{|I| \mid I \in \mathcal{I}\}$).

Lemma 5. *Any k-SCF coloring algorithm on $\mathcal{I}_t$ needs $k + (t-1)\left\lceil \frac{k}{2} \right\rceil$ colors.*

We remark that the algorithm can be implemented in time $O(kn\ _2n)$, since in each step SELECT($\mathcal{I}_t$) can be implemented in $O(kn)$ time (one does not actually need to separately consider all the intervals having the same right endpoint but only the k shortest ones) and the number of steps is upper bounded by $O(\log_2 n)$, the worst case being the complete interval hypergraph. This together with the following Theorem 4 and Theorem 2, proves the desired Theorem 1.

Theorem 4. *Consider the interval hypergraph $H = ([n], \mathcal{I})$. Then the total number of colors used by k-COLOR($\mathcal{I}$) is less than $\frac{c(k)}{\lceil k/2 \rceil}\chi_k^*(\mathcal{I})$.*

For a special class of interval hypergraphs, we show that the algorithm is optimal.

Theorem 5. *If for any $I, J \in \mathcal{I}$ such that $J \prec I$ and $I \cap J \neq \emptyset$ it holds $I \not\subseteq J$ and $|I \setminus J| \geq k$, then the algorithm k-COLOR($\mathcal{I}$), running with $c(k) = k$ on interval hypergraph $H = ([n], \mathcal{I})$, optimally uses k colors.*

3 A k-SCF coloring algorithm for H_n

In this section we present a k-SCF-coloring algorithm for the complete interval hypergraph $H_n = ([n], \mathcal{I}^{[n]})$. When $k = 1$ the algorithm reduces to the one in [8]. We assume that $n = hk$ for some integer $h \geq 1$. If $(h-1)k < n < hk$ then we can add the points $n+1, n+2, \ldots, hk$.

A simple k-SCF-coloring algorithm for H_n can be obtained by partitioning the $n = hk$ points of V in blocks $B(1), B(2), \cdots, B(h)$ of k points and coloring their points recursively with the colors in the sets $C_1, \cdots, C_{\lfloor \log h \rfloor + 1}$, where $C_t = \{k(t-1)+1, \cdots, kt\}$, for $1 \leq t \leq \lfloor \log h \rfloor + 1$. The points in the median block $B(\lfloor \frac{h+1}{2} \rfloor)$ are colored with colors in C_1, then the points in the blocks $B(1), \cdots, B(\lfloor \frac{h+1}{2} \rfloor - 1)$ and in the blocks $B(\lfloor \frac{h+1}{2} \rfloor + 1), \cdots, B(h)$ are recursively colored with the same colors in the sets $C_2, \cdots, C_{\lfloor \log h \rfloor + 1}$. Formally, the algorithm is given in Fig. 3. It starts calling (k,n)-COLOR$(1, h, 1)$.

The proof that algorithm (k,n)-COLOR$(1,h,1)$ provides a k-SCF coloring for H_n can be easily derived by that presented in [8, 14]. Furthermore, since at each of the $\lfloor \log h \rfloor + 1$ recursive steps of algorithm (k,n)-COLOR a new set of k colors is used, we have that the number of colors is at most $k(\lfloor \log h \rfloor + 1)$. Hence, we get the following result.

Lemma 6. *At the end of algorithm (k,n)-COLOR$(1, \lceil n/k \rceil, 1)$ each $I \in \mathcal{I}$ is k-SCF colored and the number of used colors is at most $k\left(\lfloor \log \lceil \frac{n}{k} \rceil \rfloor + 1\right)$.*

```
(k,n)-COLOR(a,b,t):
if a ≤ b then
    m = ⌊(a+b)/2⌋
    Color the k points in B(m) with the k colors in C_t.
    (k,n)-COLOR(1, m − 1, t + 1).
    (k,n)-COLOR(m + 1, b, t + 1).
```

Fig. 3. The k-SCF coloring algorithm for H_n

We remark that [9] shows that $\chi_k^*(H_n) \leq k \log n$ (as a specific case of a more general framework); however, we present the (k,n)-COLOR algorithm since it is very simple and gives a slightly better bound.

By Corollary 1 and considering that, for the complete interval hypergraph H_n, for each $I \in \mathcal{I}$, any of its subintervals $I' \subset I$ also belongs to $\mathcal{I}$, we get the following lower bound on $\chi^*(H_n)$.

Corollary 2. $\chi_k^*(H_n) \geq \lceil \frac{k}{2} \rceil \lceil \log \frac{n}{k} \rceil$.

Lemma 6 together with Corollary 2 proves that (k,n)-COLOR uses at most twice the minimum possible number of colors.

4 A quasipolynomial time algorithm

Consider the decision problem CFSubsetIntervals: *"Given an interval hypergraph H and a natural number q, is it true that $\chi_1^*(H) \leq q$?"* Notice that the above problem is non-trivial only when $q < \lfloor \log n \rfloor + 1$; if $q \geq \lfloor \log n \rfloor + 1$ the answer is always yes, since $\chi_1^*(H_n) = \lfloor \log n \rfloor + 1$.

Algorithm DECIDE-COLORS (Fig. 4) is a *non-deterministic* algorithm for CFSubsetIntervals. The algorithm scans points from 1 to n, tries for every point non-deterministically every color in $\{0, \ldots, q\}$, and checks if all intervals in $\mathcal{I}$ ending at the current point have the conflict-free property. If some interval in $\mathcal{I}$ has not the conflict-free property under a non-deterministic assignment, the algorithm answers 'no'. If all intervals in $\mathcal{I}$ have the conflict-free property under some non-deterministic assignment, the algorithm answers 'yes'.

We check if an interval in $\mathcal{I}$ that ends at the current point, say t, has the conflict-free property in the following space-efficient way. For every color c in

$\{0, \ldots, q\}$, we keep track of:
(a) the closest point to t colored with c in variable p_c, and
(b) the second closest point to t colored with c in variable s_c.
Then, color c is occurring exactly one time in $[j, t] \in \mathcal{I}$ if and only if $s_c < j \le p_c$.

```
DECIDE-COLORS(q, I)
for c = 0 to q
    s_c = 0, p_c = 0.
for t = 1 to n
    Choose c non-deterministically from {0, ..., q}.
    s_c = p_c, p_c = t.
    for j ∈ {j | [j, t] ∈ I}
        IntervalConflict = True.
        for c = 1 to q
            if s_c < j ≤ p_c then IntervalConflict = False
        if IntervalConflict then return NO
return YES
```

Fig. 4. A non-deterministic algorithm deciding whether $\chi_1^*(H) \le q$

Lemma 7. *The space complexity of algorithm DECIDE-COLORS is* $O(\log^2 n)$.

Proof. Since $q = O(\log n)$ and each point position can be encoded with $O(\log n)$ bits, the arrays p and s (indexed by color) take space $O(\log^2 n)$. All other variables in the algorithm can be implemented in $O(\log n)$ space. Therefore the above non-deterministic algorithm has space complexity $O(\log^2 n)$.

Theorem 6. CFSubsetIntervals *has a quasipolynomial time deterministic algorithm.*

Proof. By standard computational complexity theory arguments (see, e.g., [13]), we can transform DECIDE-COLORS to a deterministic algorithm solving the same problem with time complexity $2^{O(\log^2 n)}$, i.e., CFSubsetIntervals has a quasipolynomial time deterministic algorithm.

5 Conclusions, further work, and open problems

The exact complexity of computing an optimal k-SCF-coloring for an interval hypergraph remains an open problem. We have presented an algorithm with approximation ratio $5 - 2/k$ when $k \ge 2$ and 2 when $k = 1$. In a longer version of our work, we will include a proof that our analysis of the approximation ratio is tight when $k = 1$ and $k = 2$; when $k \ge 3$, we have an instance that forces the algorithm to use $(5 - 1/k)/2 > 2$ times the optimal number of colors. One might try to improve the approximation ratio, find a polynomial time approximation scheme, or even find a polynomial time exact algorithm. The last possibility is supported by the fact that the decision version of the 1-SCF problem, CFSubsetIntervals, is unlikely to be NP-complete, unless NP-complete problems have quasipolynomial time algorithms. Furthermore, we have shown that the algorithm optimally uses k colors if for any $I, J \in \mathcal{I}$, interval I is not contained in J and they differ for at least k points. For the complete interval hypergraph H_n, we have presented a k-SCF coloring using at most two times the optimal number of colors. It would be interesting to close this gap.

Finally, we introduced a SCF-coloring function $C\colon V \to \mathbb{N}$, for which vertices colored with '0' can not act as uniquely-colored vertices in a hyperedge. Naturally, one could try to study the bi-criteria optimization problem, in which there two minimization goals: (a) the number of colors used, $\max_{v\in V} C(v)$ (minimization of frequency spectrum use) and (b) the number of vertices with positive colors, $|\{v \in V \mid C(v) > 0\}|$ (minimization of activated base stations).

References

1. Abam, M.A., de Berg, M., Poon, S.H.: Fault-tolerant conflict-free coloring. In: Proc. 20th Canadian Conference on Computational Geometry (CCCG) (2008)
2. Abellanas, M., Bose, P., Garcia, J., Hurtado, F., Nicolas, M., Ramos, P.A.: On properties of higher order Delaunay graphs with applications. In: Proc. 21st European Workshop on Computational Geometry (EWCG). pp. 119–122 (2005)
3. Bar-Noy, A., Cheilaris, P., Olonetsky, S., Smorodinsky, S.: Online conflict-free colouring for hypergraphs. Combin. Probab. Comput. 19, 493–516 (2010)
4. Bar-Noy, A., Cheilaris, P., Smorodinsky, S.: Deterministic conflict-free coloring for intervals: from offline to online. ACM Trans. Alg. 4(4) (2008)
5. Chen, K., Fiat, A., Levy, M., Matoušek, J., Mossel, E., Pach, J., Sharir, M., Smorodinsky, S., Wagner, U., Welzl, E.: Online conflict-free coloring for intervals. SIAM J. Comput. 36, 545–554 (2006)
6. Chen, K., Kaplan, H., Sharir, M.: Online conflict free coloring for halfplanes, congruent disks, and axis-parallel rectangles. ACM Trans. Alg. 5(2) (2009)
7. Cui, Z., Hu, Z.C.: k-conflict-free coloring and k-strong-conflict-free coloring for one class of hypergraphs and online k-conflict-free coloring. ArXiv abs/1107.0138 (2011)
8. Even, G., Lotker, Z., Ron, D., Smorodinsky, S.: Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. SIAM J. Comput. 33, 94–136 (2003)
9. Horev, E., Krakovski, R., Smorodinsky, S.: Conflict-free coloring made stronger. In: Proc. 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT). pp. 105–117 (2010)
10. Katz, M., Lev-Tov, N., Morgenstern, G.: Conflict-free coloring of points on a line with respect to a set of intervals. Comput. Geom. 45, 508–514 (2012)
11. Lev-Tov, N., Peleg, D.: Conflict-free coloring of unit disks. Discrete Appl. Math. 157(7), 1521–1532 (2009)
12. Nguyen, H.L., Nguyen, U.T.: Algorithms for bandwidth efficient multicast routing in multi-channel multi-radio wireless mesh networks. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC). pp. 1107–1112 (2011)
13. Papadimitriou, C.: Computational Complexity. Addison Wesley (1993)
14. Smorodinsky, S.: Conflict-free coloring and its applications. ArXiv abs/1005.3616 (2010)
15. Zeng, G., Wang, B., Ding, Y., Xiao, L., Mutka, M.: Efficient multicast algorithms for multichannel wireless mesh networks. IEEE Trans. Parallel Distrib. Systems 21, 86–99 (2010)