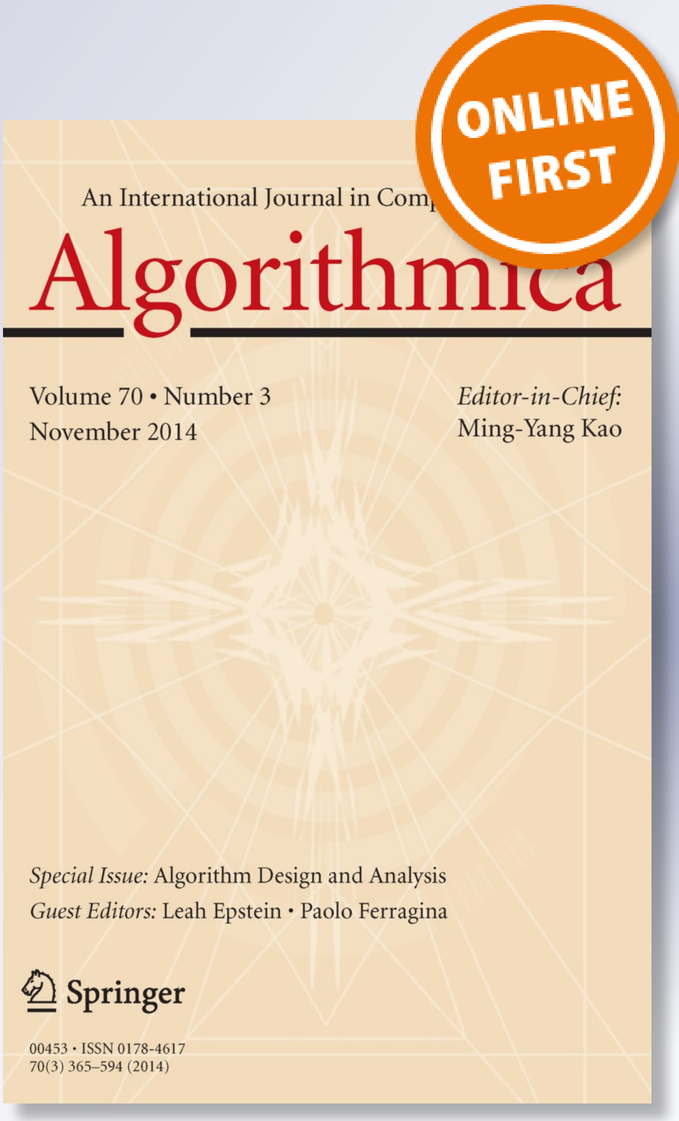
Strong Conflict-Free Coloring for Intervals

Panagiotis Cheilaris, Luisa Gargano, Adele A. Rescigno & Shakhar Smorodinsky

Algorithmica

ISSN 0178-4617

Algorithmica
DOI 10.1007/s00453-014-9929-x



Springer

Algorithmica
DOI 10.1007/s00453-014-9929-x

Strong Conflict-Free Coloring for Intervals

Panagiotis Cheilaris · Luisa Gargano · Adele A. Rescigno · Shakhar Smorodinsky

Received: 15 May 2013 / Accepted: 3 August 2014

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 approximation algorithm for the general problem; the algorithm has approximation ratio 2 when $k = 1$ and $5 - \frac{2}{k}$ when $k \geq 2$. 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 provide, in case $k = O(\text{polylog}(n))$, a quasipolynomial time algorithm to decide the existence of a k-SCF coloring that uses at most q colors.

Keywords Conflict-free coloring · Interval hypergraph · Wireless networks

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.

P. Cheilaris
Faculty of Informatics, Università della Svizzera italiana, 6900 Lugano, Switzerland

L. Gargano · A. A. Rescigno (✉)
Dipartimento di Informatica, University of Salerno, 84084 Fisciano, SA, Italy
e-mail: rescigno@dia.unisa.it

S. Smorodinsky
Mathematics Department, Ben-Gurion University, 84105 Be'er Sheva, Israel

Published online: 27 August 2014

Definition 1 *(CF coloring)* A *conflict-free vertex coloring* of a hypergraph $H = (V, \mathcal{E})$ is a function $C\colon V \to \mathbb{Z}^+$ 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 radio frequency identification (RFID) 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.

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 distinct colors $c_1, \ldots, c_{\min\{|e|,k\}} \in \mathbb{Z}^+$ such that $|\{v \mid v \in e,\ C(v) = c_i\}| = 1$, for $i = 1, \ldots, \min\{|e|, k\}$. 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.

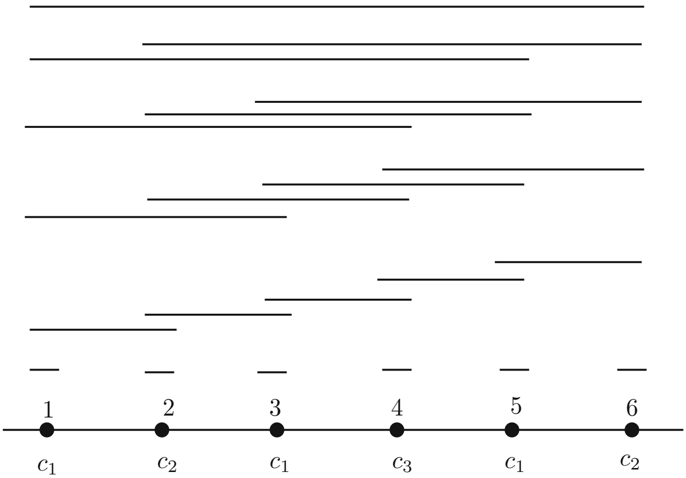


Fig. 1 The hypergraph H_6 representing all the intervals on a line with six points, and a 1-SCF coloring for the points of H_6

Remark 1 We argue 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; in particular, a CF coloring is a total 1-SCF coloring.

1.1 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

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

where each (discrete) interval is a set of consecutive points. An example is given in Fig. 1.

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].

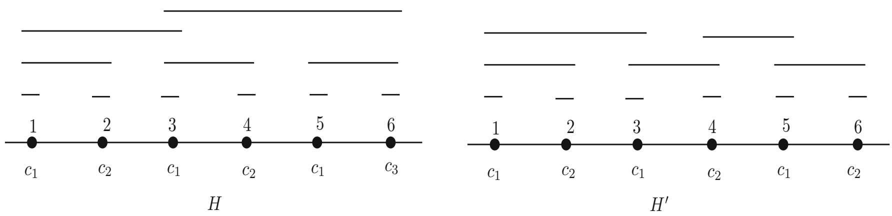


Fig. 2 The hypergraphs H and H' on $V = [6]$ with their 1-SCF-colorings

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*. Figure 2 shows a 1-SCF coloring of the interval hypergraphs H and H' on six points. It is not hard to see that any 1-SCF coloring for H needs at least three colors, while two colors suffice for H'.

Conflict-free coloring the complete interval hypergraph was first studied in [8], where it was shown that $\chi_1^*(H_n) = \lfloor \log n \rfloor + 1$.[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 [3–6].

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 $k \log n$ colors, for any k [9].

In this paper we give an algorithm which outputs a k-SCF coloring of the points of the input interval hypergraph H, for any fixed value of $k \geq 1$. The algorithm has an approximation factor $5 - 2/k$ in the case $k \geq 2$ (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. We also consider the problem of k-SCF coloring the complete interval hypergraph H_n. We give a very simple k-SCF

[1] Unless otherwise specified, all logarithms are in base 2.

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 $\left\lceil\frac{k}{2}\right\rceil\left\lfloor\log\frac{n}{k}\right\rfloor+k$ colors. Finally, we show that the decision problem whether a given interval hypergraph can be k-SCF-colored with at most q colors has a quasipolynomial time algorithm.

2 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 $I = I_i$.

We denote by $\mathcal{M}_{\mathcal{I}}(I)$ the subfamily of all the intervals $J \in \mathcal{I}$ that are strongly contained in I, that is,

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

3 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. 3. 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 round-robin manner to the ordered sequence (from the minimum to the maximum) of the selected points. The intervals that have k unique colors among the $c(k)$ colors assigned to the points in P_t are k-colored at the end of step t and are inserted in the set $\mathcal{Y}_t$ 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_t = I   [The intervals in I_t will be k-colored at some step t' ≥ t]
while I_t ≠ ∅
    Execute the following step t
     - Set Y_t = ∅   [The intervals in Y_t ⊆ I_t become k-colored during step t]
     - Let P_t be the set returned by SELECT(I_t) and assume that
           the points in P_t are p_0 < p_1 < ... < p_{n_t}
     - for i = 0 to n_t
           Assign to p_i color c_i = (t − 1)c(k) + (i mod c(k)) + 1
     - for each I ∈ I_t
           if I has k unique colors among (t − 1)c(k) + 1, ..., tc(k) then Y_t = Y_t ∪ {I}
     - I_{t+1} = I_t \ Y_t
     - 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]
    Set P_t(I) = ∅
    if |I ∩ P_t| < min{|I|, k} then
       - Let P_t(I) be the set of largest min{|I|, k} − |I ∩ P_t| points of I \ P_t
       - P_t = P_t ∪ P_t(I)
Return P_t
```

Fig. 3 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. 4. 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 the colors c_1, c_2, c_3, c_4 in round-robin manner. Only three intervals remain in $\mathcal{I}_2$; all the others are in $\mathcal{Y}_1$, each having two unique colors among the assigned colors at the end of step 1. SELECT($\mathcal{I}_2$) returns $P_2 = \{14, 15, 23\}$ and these points are colored with the colors c_5, c_6, c_7. Now $\mathcal{I}_3 = \mathcal{I}_2 \backslash \mathcal{Y}_2 = \emptyset$ and the algorithm ends.

In the following, we will prove the following theorem.

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

3.1 Correctness of Algorithm k-COLOR

We denote by P_t the set of points returned by SELECT($\mathcal{I}_t$). Furthermore, we denote by $P_t(i)$ the subset of points selected during iterations 1 up to i; that is, if $\mathcal{I}_t$ contains

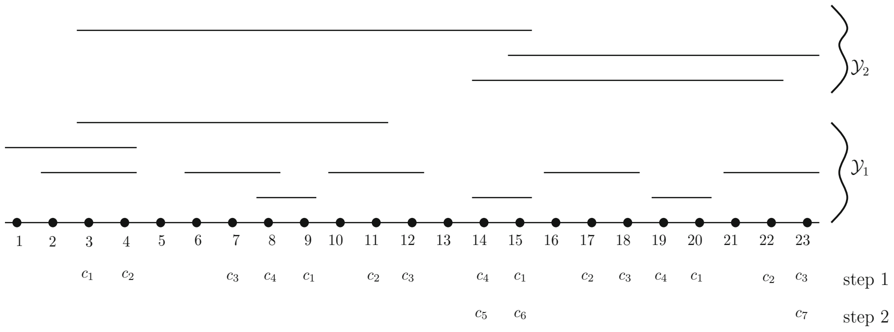


Fig. 4 Example coloring by k-COLOR for $k = 2$

$I_1 \prec I_2 \prec \cdots \prec I_{m_t}$, then

$$P_t(i) = P_t(I_1) \cup P_t(I_2) \cdots \cup P_t(I_i)$$

(see lines 18–21 of SELECT($\mathcal{I}_t$)), for all $i = 1, \ldots, m_t$.

Lemma 1 *Let $t \geq 1$. For each interval $I \in \mathcal{I}_t$, it holds*

(a) $|I \cap P_t| \geq \min\{|I|, k\}$;
(b) if $|I \cap P_t| \leq 2c(k) - k$ then $I \in \mathcal{Y}_t$;
(c) if $t > 1$ then $|I| \geq 2c(k) - k + 1 \geq 4k - 1$.

Proof Point (a) follows since the algorithm SELECT($\mathcal{I}_t$) assures that $|I \cap P_t| \geq \min\{|I|, k\}$ are selected in each interval in $\mathcal{I}_t$. Now, we prove (b).
Let P_t be the set returned by SELECT($\mathcal{I}_t$) and assume that the points in P_t are $p_0 < p_1 < \cdots < p_{n_t}$. The step t of the algorithm k-COLOR($\mathcal{I}$) assigns colors to the points in P_t from the set $C = \{(t-1)c(k) + 1, (t-1)c(k) + 2, \ldots, tc(k)\}$ in round-robin manner; namely point p_i gets color $(t-1)c(k) + (i \mod c(k)) + 1$, for $i = 1, \ldots, n_t$. If $|I \cap P_t| \leq 2c(k) - k$, then at least $k \leq 2c(k) - |I \cap P_t|$ (or $|I \cap P_t|$ if less than k) among the colors in C assigned to the points in $I \cap P_t$ are unique. This, together with point (a), assures that I is contained in $\mathcal{Y}_t$ and proves (b).

Point (c) is an immediate consequence of (b) with $t = 1$ and the definition of $c(k)$. □

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

Proof By contradiction, let $J \in \mathcal{I}_t \backslash \mathcal{Y}_t$ and $J \subset I$. Since $J \in \mathcal{I}_t \backslash \mathcal{Y}_t$ we have that, at the end of step t of the algorithm k-COLOR($\mathcal{I}$), the interval J has not k unique colors among the $c(k)$ colors assigned at step t. It follows that $|J \cap P_t| > 2c(k) - k$ must hold. Hence, $|I \cap P_t| \geq |J \cap P_t| > 2c(k) - k$ and I also cannot have k unique colors among the $c(k)$ colors assigned at step t. This contradicts the hypothesis $I \in \mathcal{Y}_t$. □

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

Proof Let $\mathcal{L}_{\mathcal{I}}(I)$ (resp. $\mathcal{R}_{\mathcal{I}}(I)$) be the subfamily of intervals of $\mathcal{I}$ that are not contained in I and whose rightmost (resp. leftmost) point belongs to I; that is

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

Let $L \in \mathcal{I}_t$ be the last interval in $\mathcal{L}_{\mathcal{I}_t}(I)$, that is, $J \prec L$ for any $J \in \mathcal{L}_{\mathcal{I}_t}(I)$ with $J \neq L$. Assume L is the ℓ-th interval in $\mathcal{I}_t$. We first notice that $|L \cap I \cap P_t(\ell)| \leq k$. Otherwise, if $|L \cap I \cap P_t(\ell)| > k$ then there would exist an interval $K \subset L \cap I$ since SELECT($\mathcal{I}_t$) selects points only if they are useful to supply k selected points to an interval; this is not possible since $\mathcal{M}_{\mathcal{I}_t}(I) = \emptyset$.

If $I \in \mathcal{I}_t$ then it is considered at iteration $\ell + 1$ and at most $k - |L \cap I \cap P_t(\ell)|$ points are added to P_t. Hence

$$|I \cap P_t(h)| \leq k, \text{ for } h = \begin{cases} \ell & \text{if } I \notin \mathcal{I}_t, \\ \ell + 1 & \text{otherwise.} \end{cases} \tag{1}$$

Now, consider any $R \in \mathcal{R}_{\mathcal{I}_t}(I)$ and let r be such that R is the r-th interval in $\mathcal{I}_t$. Observe that $R \in \mathcal{R}_{\mathcal{I}_t}(I)$ implies that R has at least one point outside I and $r > h$. If $P_t(r) \backslash P_t(h) \neq \emptyset$ then at least one of such points is selected at iteration r. This implies $|I \cap R \cap P_t(r)| \leq k - |(R \backslash I) \cap P_t| \leq k - 1$. This together with (1) gives the lemma.

If $P_t(r) \backslash P_t(h) = \emptyset$ then $|I \cap P_t| = |I \cap P_t(h)|$ and, by (1), the lemma holds. □

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 In order to prove the theorem, we show the following statement for each $t \geq 1$.

S(t): *At the end of step t of algorithm k-COLOR($\mathcal{I}$), each interval $I \in \bigcup_{i=1}^{t} \mathcal{Y}_i$ is k-colored.*

The proof is by induction on t. For $t = 1$, the statement trivially follows by definition of $\mathcal{Y}_1$. Consider now $t \geq 2$. Assume the statement be true for each $i \leq t - 1$. 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 $I \in \mathcal{Y}_t$, then I is k-colored by definition of $\mathcal{Y}_t$.

Consider then $I \in \mathcal{Y}_i$ for some $i \leq t - 1$. By the inductive hypothesis I is k-colored at the end of step $t - 1$. We will prove that, at the end of step t, the interval I has k unique colors, among $1, \ldots, t \cdot c(k)$. Indeed, by Lemma 2 we know that $\mathcal{M}_{\mathcal{I}_i}(I) \subseteq \mathcal{Y}_i$; 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. □

3.2 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 considered, 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 $\mathcal{M}_{\mathcal{I}}(I_1) \cup \{I_1\}$ (resp.*

$\mathcal{M}_{\mathcal{I}}(I_2) \cup \{I_2\}$). Then the number of colors used by any optimal k-SCF coloring of $\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}$ is

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

Proof Suppose w.l.o.g. that $\chi_1 \leq \chi_2$. Recalling that $I_1, I_2 \in \mathcal{M}_{\mathcal{I}}(I)$, we immediately get $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}) \geq \chi_2$. Moreover, consider any k-SCF coloring which colors the points in I using $\chi_2 + c$ colors. Since $I_1 \cap I_2 = \emptyset$, we have that at least $\chi_1 - c$ colors are used more than once to color points in $I_1 \cup I_2$. Hence, at most $(\chi_2 + c) - (\chi_1 - c) = (\chi_2 - \chi_1) + 2c$ colors can be unique in I. In order for the coloring to be k-SCF, the inequality $(\chi_2 - \chi_1) + 2c \geq k$ must be true. This immediately gives $c \geq \left\lceil \frac{k-|\chi_2-\chi_1|}{2} \right\rceil$ and (2) holds. □

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 $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I_1) \cup \{I_1\})$ and $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I_2) \cup \{I_2\})$ are at least χ, then the number of colors used in any optimal k-SCF coloring of $\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}$ is $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}) \geq \chi + \lceil k/2 \rceil$.*

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.

Lemma 4 *Let $t \geq 2$. For each $I \in \mathcal{I}_t$,*

(i) there exist $J', J'' \in \mathcal{I}_{t-1}$ such that $J', J'' \subset I$ and $J' \cap J'' = \emptyset$;
(ii) if $t = 2$ then

$$\max\left\{ \left| \bigcup_{\substack{J \in \mathcal{M}_{\mathcal{I}}(I) \\ |J| < k}} J \right|, \max_{\substack{J', J'' \in \mathcal{M}_{\mathcal{I}}(I) \\ J' \cap J'' = \emptyset}} \{|J'| + |J''|\} \right\} \geq k + 1. \tag{3}$$

Proof We already know, by Lemma 3 and (b) of Lemma 1, that if $\mathcal{M}_{\mathcal{I}_{t-1}}(I) = \emptyset$ then I is k-colored at the end of step $t-1$ and $I \notin \mathcal{I}_t$. Hence, we can assume that $\mathcal{M}_{\mathcal{I}_{t-1}}(I) \neq \emptyset$ for any $I \in \mathcal{I}_t$.

We prove now (i); we proceed by contradiction. Suppose that there exists $I \in \mathcal{I}_t$ that does not include two disjoint intervals in $\mathcal{I}_{t-1}$, that is,

$$\bigcap_{K \in \mathcal{M}_{\mathcal{I}_{t-1}}(I)} K \neq \emptyset. \tag{4}$$

Let $K_1, K_2 \in \mathcal{M}_{\mathcal{I}_{t-1}}(I)$ be such that

$$K_1 \prec K' \prec K_2, \qquad \text{for each } K' \subset I. \tag{5}$$

Let K_1 and K_2 be the $(r_1 + 1)$-th and r_2-th interval in $\mathcal{I}_{t-1}$, respectively.

Reasoning as in Lemma 3, if we disregard the points added to P_{t-1} at iterations $r_1 + 1, \ldots, r_2$ of SELECT($\mathcal{I}_{t-1}$) when intervals in $\mathcal{M}_{\mathcal{I}_{t-1}}(I)$ are considered, then

$$\left|I \cap (P_{t-1} \backslash (P_{t-1}(r_2) \backslash P_{t-1}(r_1)))\right| \leq 2k - 1. \tag{6}$$

We want now to evaluate the number of points added to P_{t-1} at any iteration of SELECT($\mathcal{I}_{t-1}$) in which intervals $K_1, \ldots, K_2$ are considered, that is $|P_{t-1}(r_2) \backslash P_{t-1}(r_1)|$.

At most k points are added to P_{t-1} at iteration $r_1 + 1$, in order to complete K_1, that is

$$|P_{t-1}(r_1 + 1) \backslash P_{t-1}(r_1)| \leq k.$$

Moreover at any iteration from $r_1 + 2$ to r_2 we only need to complete each interval up to K_2. The hypothesis (4) implies that each point selected in $K \in \mathcal{M}_{\mathcal{I}_{t-1}}(I)$ at any iteration from $r_1 + 2$ to r_2 belongs to each $K' \in \mathcal{M}_{\mathcal{I}_{t-1}}(I)$ such that $K \prec K'$, then the number of points added to P_{t-1} is

$$|P_{t-1}(r_2) \backslash P_{t-1}(r_1 + 1)| \leq \begin{cases} k - 1 & \text{if } |P_{t-1}(r_1 + 1) \backslash P_{t-1}(r_1)| > 0, \\ k & \text{if } |P_{t-1}(r_1 + 1) \backslash P_{t-1}(r_1)| = 0. \end{cases}$$

In conclusion, the number of points added to P_{t-1} when intervals $K_1, \ldots, K_2$ are considered is

$$|P_{t-1}(r_2) \backslash P_{t-1}(r_1)| \leq 2k - 1.$$

From this and (6) we have that

$$|I \cap P_{t-1}| \leq 4k - 2 \leq 2c(k) - k.$$

By (b) of Lemma 1, this implies that $I \in \mathcal{Y}_{t-1}$ contradicting the hypothesis $I \in \mathcal{I}_t = \mathcal{I}_{t-1} \backslash \mathcal{Y}_{t-1}$.

We prove now (ii). Assume $t = 2$ and suppose that (3) does not hold for $I \in \mathcal{I}_2$. Let $J_1 \prec \cdots \prec J_\ell$ be all the intervals in $\mathcal{I}$ such that $J_i \subset I$ and $|J_i| < k$, for $i = 1, \ldots, \ell$. Hence, we have that

$$\left|\bigcup_{i=1}^{\ell} J_i\right| \leq k, \quad \text{and} \quad J_i \cap J \neq \emptyset \tag{7}$$

for $i = 1, \ldots, \ell$ and for any $J \subset I$ with $|J| \geq k$ (otherwise, we have $|J| + |J_i| \geq |J| + 1 \geq k + 1$ and (3) would be true).

Consider now the intervals K_1, K_2 as defined in (5). We can then proceed as in the proof of i) and, by using (7) deduce that

$$|P_{t-1}(r_2) \backslash P_{t-1}(r_1)| \leq \begin{cases} |J_1| + (k-1) + (|J_\ell| - 1) & \text{if } K_1 = J_1 \text{ and } J_\ell = K_2, \\ |J_1| + |\cup_{i=2}^{\ell} J_i| + (k - |\cup_{i=2}^{\ell} J_i| - 1)) & \text{if } K_1 = J_1 \text{ and } J_\ell \prec K_2, \\ k + |J_\ell| - 1 & \text{if } K_1 \prec J_1. \end{cases}$$

Hence $|P_{t-1}(r_2) \backslash P_{t-1}(r_1)| \leq 2k - 2$ which leads to the contradiction $I \notin \mathcal{I}_2$. □

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 at least* $\max\{k,\ k+1+(t-2)\left\lceil\frac{k}{2}\right\rceil\}$ *colors.*

Proof For $t = 1$ the lemma is clearly true since any k-SCF coloring algorithm on $\mathcal{I}_1 = \mathcal{I}$ uses at least k colors.

Consider now $t \geq 2$. We prove by induction on t, that for each $I \in \mathcal{I}_t$

$$\chi_k^*(\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}) \geq k+1+(t-2)\left\lceil\frac{k}{2}\right\rceil.$$

Consider first $t = 2$. By recalling that any interval J must contain $\min\{|J|, k\}$ unique colors, and using (ii) of Lemma 4, we have that k colors cannot be sufficient for the desired coloring of $\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}$. Hence the bound holds for $t = 2$.
Assume now that the bound holds for $t-1 \geq 2$; we prove it for t. Consider any $I \in \mathcal{I}_t$. Let $I', I'' \in \mathcal{I}_{t-1}$ be the two disjoint sub-intervals of I whose existence is granted by (i) of Lemma 4. Moreover, (c) of Lemma 1 implies $|I'|, |I''| \geq k$. By using the inductive hypothesis on I' and I'' we have $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I') \cup \{I'\}) \geq k+1+(t-3)\left\lceil\frac{k}{2}\right\rceil$ and $\chi_k^*(\mathcal{M}_{\mathcal{I}}(I'') \cup \{I''\}) \geq k+1+(t-3)\left\lceil\frac{k}{2}\right\rceil$. Hence, Corollary 1 implies

$$\chi_k^*(\mathcal{M}_{\mathcal{I}}(I) \cup \{I\}) \geq k+1+(t-3)\left\lceil\frac{k}{2}\right\rceil + \left\lceil\frac{k}{2}\right\rceil = k+1+(t-2)\left\lceil\frac{k}{2}\right\rceil$$

and the desired bound holds for I. □

We remark that the algorithm can be implemented in time $O(kn \log n)$, since in each step SELECT($\mathcal{I}_t$) can be implemented in $O(kn)$ time (indeed one does not actually need to process all the intervals having the same right endpoint but only the k shortest ones) and the number of steps is upper bounded by $O(\log n)$, the worst case being the complete interval hypergraph. This together with the following Theorems 2 and 4, 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 upper bounded by $\frac{c(k)}{\lceil k/2\rceil}\chi_k^*(\mathcal{I})$.*

Proof Let δ be the last step of the algorithm k-COLOR($\mathcal{I}$). Then the total number of colors used by the algorithm is $ALG(\mathcal{I}) \leq \delta \cdot c(k)$; since each step of the algorithm uses a different set of $c(k)$ colors. By Lemma 5, any optimal k-SCF coloring algorithm on $\mathcal{I}_\delta \subseteq \mathcal{I}$ uses at least $\max\{k,\ k+1+(\delta-2)\left\lceil\frac{k}{2}\right\rceil\}$ colors. This obviously implies $\chi_k^*(\mathcal{I}) \geq \max\{k,\ k+1+(\delta-2)\left\lceil\frac{k}{2}\right\rceil\}$. Hence,

$$\frac{ALG(\mathcal{I})}{\chi_k^*(\mathcal{I})} \leq \frac{\delta c(k)}{\max\{k,\ k+1+(\delta-2)\left\lceil\frac{k}{2}\right\rceil\}} \leq \frac{c(k)}{\left\lceil\frac{k}{2}\right\rceil}.$$

□

3.3 Tight Instances for the k-SCF Approximation Algorithm

We define a family of intervals $\mathcal{J}_\delta$, for $\delta \geq 1$, inducing a hypergraph that binds the algorithm to use at least $\frac{c(k)}{k}$ times the number of colors of an optimal coloring. Given a set of intervals $\mathcal{J}$, we define

- $\mathcal{J}^{+c}$, for some integer $c \geq 1$, to be the set of the intervals in $\mathcal{J}$ each shifted by c points to the right, that is,

$$\mathcal{J}^{+c} = \{\{a+c, a+1+c, \ldots, b+c\} \mid \{a, a+1, \ldots, b\} \in \mathcal{J}\}.$$

- $\ell(\mathcal{J})$ (resp. $r(\mathcal{J})$) to be the leftmost (resp. rightmost) of the points of any interval in $\mathcal{J}$

$$\ell(\mathcal{J}) = \min\{\ell(I) \mid I \in \mathcal{J}\} \qquad r(\mathcal{J}) = \max\{r(I) \mid I \in \mathcal{J}\}.$$

- *length* of $\mathcal{J}$, denoted by $\ell en(\mathcal{J})$, to be

$$\ell en(\mathcal{J}) = r(\mathcal{J}) - \ell(\mathcal{J}) + 1.$$

Now, we can define the desired tight instance for the k-SCF approximation algorithm.

Definition 4 Define $\mathcal{J}_1$ as the set of size 2 and $\ell en(\mathcal{J}_1) = 4k$ given by

$$\mathcal{J}_1 = \{\{1, \ldots, 2k\}, \{2k+1, \ldots, 4k\}\}.$$

For $\delta \geq 1$, define the set of intervals $\mathcal{J}_{\delta+1}$ as the set of $\ell en(\mathcal{J}_{\delta+1}) = 4\, \ell en(\mathcal{J}_\delta)$ given by

$$\begin{aligned}\mathcal{J}_{\delta+1} = {} & \mathcal{J}_\delta \cup \mathcal{J}_\delta^{+\ell en(\mathcal{J}_\delta)} \cup \mathcal{J}_\delta^{+2\,\ell en(\mathcal{J}_\delta)} \cup \mathcal{J}_\delta^{+3\,\ell en(\mathcal{J}_\delta)} \cup \\ & \cup \{\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 2\,\ell en(\mathcal{J}_\delta)\} \cup \{3\,\ell en(\mathcal{J}_\delta)/2 \\ & - k + 1, \ldots, 4\,\ell en(\mathcal{J}_\delta)\}.\end{aligned}$$

By Definition 4, we easily have $\ell en(\mathcal{J}_\delta) = k2^{2\delta}$. In the following we consider the hypergraph $H = (V, \mathcal{J}_\delta)$ where $V = \{1, \ldots, k2^{2\delta}\}$. Figure 5 shows $H = (V, \mathcal{J}_3)$, for $k = 1$, with an optimal 1-SCF coloring and the 1-SCF coloring produced by algorithm 1-COLOR.

Note that, except for intervals $\{\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 2\,\ell en(\mathcal{J}_\delta)\}$ and $\{3\,\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 4\,\ell en(\mathcal{J}_\delta)\}$, $\mathcal{J}_{\delta+1}$ can be partitioned into four $\mathcal{J}_\delta$ components, i.e., $\mathcal{J}_\delta$, $\mathcal{J}_\delta^{+\ell en(\mathcal{J}_\delta)}$, $\mathcal{J}_\delta^{+2\,\ell en(\mathcal{J}_\delta)}$, $\mathcal{J}_\delta^{+3\,\ell en(\mathcal{J}_\delta)}$. We use this partition to assign each interval in $\mathcal{J}_{\delta+1}$ a *level*: The intervals in each of the four $\mathcal{J}_\delta$ components of $\mathcal{J}_{\delta+1}$, have the same level as the corresponding intervals in $\mathcal{J}_\delta$, and intervals $\{\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 2\,\ell en(\mathcal{J}_\delta)\}$ and $\{3\,\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 4\,\ell en(\mathcal{J}_\delta)\}$ have level $\delta + 1$. The intervals of $\mathcal{J}_3$ shown in Fig. 5 are arranged for levels.

By the construction of $\mathcal{J}_\delta$, we immediately have:

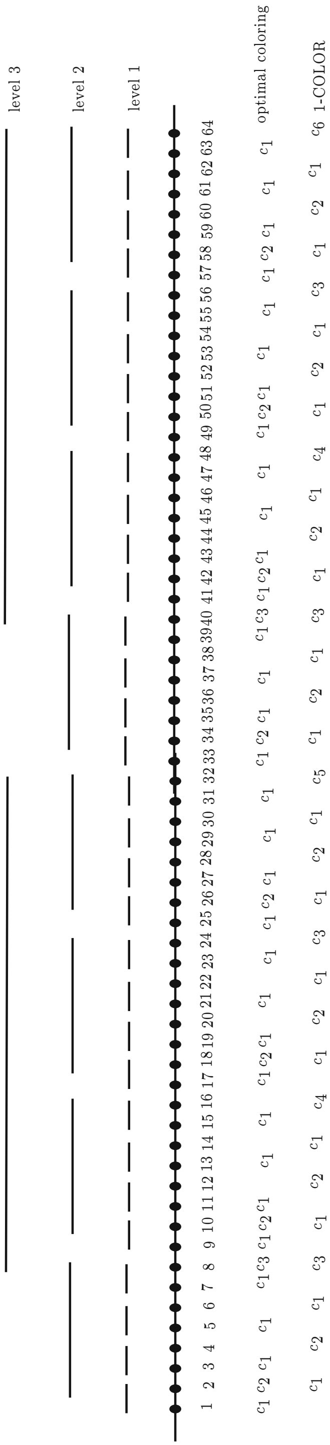


Fig. 5 1-SCF colorings for $H = (V, \mathcal{J}_3)$

Fact 1 *The intervals at level t, with $t \geq 1$, are pairwise disjoint.*

Fact 2 *Each interval at level t, with $t \geq 2$, includes the rightmost k points of exactly four intervals at level $t-1$.*

Lemma 6 *Let $\delta \geq 1$. Given the interval hypergraph $H = (V, \mathcal{J}_\delta)$, there is a k-SCF coloring of H that uses $k\delta$ colors.*

Proof We first define recursively the coloring of $\mathcal{J}_\delta$ and then we prove by induction that it is a k-SCF coloring of $\mathcal{J}_\delta$. Recall that initially all nodes are colored 0.

Consider $\mathcal{J}_1$ and color points $1, \ldots, k$ and $2k+1, \ldots, 3k$ respectively with $c_1, \ldots, c_k$. The intervals $\{1, \ldots, 2k\}$ and $\{2k+1, \ldots, 4k\}$ in $\mathcal{J}_1$ are obviously k-SCF colored. For $\delta \geq 1$, we recall that, excluding intervals $\{\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 2\,\ell en(\mathcal{J}_\delta)\}$ and $\{3\,\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 4\,\ell en(\mathcal{J}_\delta)\}$, $\mathcal{J}_{\delta+1}$ is partitioned in four $\mathcal{J}_\delta$ components. Hence, we use the k-SCF coloring of $\mathcal{J}_\delta$ to color the points covered by $\mathcal{J}_{\delta+1}$ from left to right as follows:

- Use the coloring of $\mathcal{J}_\delta$ and recolor points $\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, \ell en(\mathcal{J}_\delta)/2$ with colors $c_{k\delta+1}, \ldots, c_{k\delta+k}$.
- Concatenate the coloring of $\mathcal{J}_\delta$.
- Use again the coloring of $\mathcal{J}_\delta$ and recolor points $3\,\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 3\,\ell en(\mathcal{J}_\delta)/2$ with colors $c_{k\delta+1}, \ldots, c_{k\delta+k}$.
- Concatenate the coloring of $\mathcal{J}_\delta$.

The first and the third $\mathcal{J}_\delta$ component of $\mathcal{J}_{\delta+1}$ are k-SCF colored since we have introduced k new colors, $c_{k\delta+1}, \ldots, c_{k\delta+k}$, in the k-SCF coloring given by induction. Furthermore, each of the remaining two $\mathcal{J}_\delta$ components of $\mathcal{J}_{\delta+1}$ are k-SCF colored by induction. Finally, intervals $\{\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 2\,\ell en(\mathcal{J}_\delta)\}$ and $\{3\,\ell en(\mathcal{J}_\delta)/2 - k + 1, \ldots, 4\,\ell en(\mathcal{J}_\delta)\}$ are k-SCF colored because of colors $c_{k\delta+1}, \ldots, c_{k\delta+k}$ that occur uniquely. □

Lemma 7 *Let $\delta \geq 1$ and $H = (V, \mathcal{J}_\delta)$. The number of colors used by k-COLOR($\mathcal{J}_\delta$) is $c(k)\delta$.*

Proof The algorithm k-COLOR uses a set of $c(k) = 2k + \lceil \frac{k}{2} \rceil - 1$ new colors at each step. We will prove that k-COLOR($\mathcal{J}_\delta$) stops after δ steps. In particular, we will prove by induction on the steps of the algorithm that at the end of step t, for $1 \leq t \leq \delta$, the set of the k-colored intervals $\mathcal{Y}_t$ contains all the intervals at level t in $\mathcal{J}_\delta$.

At the first step $t = 1$ of k-COLOR($\mathcal{J}_\delta$), algorithm SELECT($\mathcal{I}_1 = \mathcal{J}_\delta$) selects the k rightmost points of each interval of length $2k$. Such points are colored with $c_1, \ldots, c_{c(k)}$ in round-robin manner. Since each interval at level 1 has length $2k$ and using Fact 1, we can state that all the intervals at level 1 are in $\mathcal{Y}_1$ at the end of step 1. All the other intervals have, by Fact 2, at least $4k$ points selected that are colored in round-robin manner with $c(k)$ colors. Since $4k > 2c(k) - k$ we have that all the intervals at levels at least 2 are not k-colored at the end of step 1.

By induction, the intervals not yet k-colored at the beginning of step t, i.e., the intervals in $\mathcal{I}_t = \mathcal{J}_\delta - (\mathcal{Y}_1 \cup \cdots \cup \mathcal{Y}_{t-1})$, are the intervals at levels $t, t+1, \ldots, \delta$ in $\mathcal{J}_\delta$. The algorithm SELECT($\mathcal{I}_t$) selects the rightmost k points of each interval at level

t. Such points are colored with $c_{c(k)(t-1)+1}, \ldots, c_{c(k)t}$ in round-robin manner. Hence, Fact 1 all the intervals at level t are in $\mathcal{Y}_t$ at the end of step t. Furthermore, by Fact 2, each interval at level at least $t+1$ has at least $4k > 2c(k) - k$ points selected and then all the intervals at levels $t+1, \ldots, \delta$ are not k-colored at the end of step t. □

By Lemmas 6 and 7 we have the following result.

Theorem 5 *Consider the interval hypergraph $H = (V, \mathcal{J}_\delta)$ with $\delta \geq 1$. The total number of colors used by k-COLOR($\mathcal{J}_\delta$) is lower bounded by $\frac{c(k)}{k}\chi_k^*(\mathcal{J}_\delta)$ where*

$$\frac{c(k)}{k} = \begin{cases} \left(\frac{5}{2} - \frac{1}{2k}\right) & \text{if } k \text{ is odd,} \\ \left(\frac{5}{2} - \frac{1}{k}\right) & \text{if } k \text{ is even.} \end{cases}$$

4 An Optimal Algorithm for Strongly Non-Nested Intervals

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

We say that $\mathcal{I}$ is a set of *k-non-nested intervals* if for any $I, J \in \mathcal{I}$ such that $J \prec I$ and $I \cap J \neq \emptyset$ it holds $|I \backslash J| \geq k$.

Theorem 6 *If $\mathcal{I}$ is a set of k-non-nested intervals then the algorithm k-COLOR($\mathcal{I}$), running with $c(k) = k$ on interval hypergraph $H = ([n], \mathcal{I})$, optimally uses k colors.*

Proof We will prove that at the end of the first step of k-COLOR($\mathcal{I}$) for each $I \in \mathcal{I} = \mathcal{I}_1$ it holds $|I \cap P_1| = \min\{|I|, k\}$. This will imply that if we run algorithm k-COLOR($\mathcal{I}$) with $c(k) = k$ then each interval in $\mathcal{I}$ is optimally k-colored.
Let I be the i-th interval in $\mathcal{I}$. By (a) of Lemma 1, we know that $|I \cap P_1| \geq \min\{|I|, k\}$. Moreover, as in the proof of Lemma 3, we can prove $|I \cap P_1(i)| \leq \min\{|I|, k\}$.

Now, consider any $R \in \mathcal{R}(I)$ and let R be the r-th interval in $\mathcal{I}$. Since $|R \backslash I| \geq k \geq \min\{|R|, k\} - |R \cap P_1(r-1)|$ we have $|I \cap P_1(r)| \leq \min\{|I|, k\}$. □

5 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 $B(1), B(2), \ldots, B(h)$ of k points and coloring their points recursively with the colors in the sets $C_1, \ldots, C_{\lfloor \log h \rfloor + 1}$, where $C_t = \{k(t-1)+1, \ldots, kt\}$, 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 $B(1), \ldots, B(\lfloor \frac{h+1}{2} \rfloor - 1)$ and in the blocks $B(\lfloor \frac{h+1}{2} \rfloor + 1), \ldots, B(h)$ are recursively colored with the same colors in the sets $C_2, \ldots, C_{\lfloor \log h \rfloor + 1}$. Formally, the algorithm is given in Fig. 6. It starts calling (k, n)-COLOR$(1, h, 1)$.

```
(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. 6 k-SCF-coloring for H_n

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 8 *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* $k\left(\left\lfloor \log \left\lceil \frac{n}{k} \right\rceil \right\rfloor + 1\right)$.

We remark that [9] shows that $\chi_k^*(H_n) \le k \log n$ (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 $\chi_k^*(H_n)$.

Corollary 2 $\chi_k^*(H_n) \ge \left\lceil \frac{k}{2} \right\rceil \left\lfloor \log \frac{n}{k} \right\rfloor + k$.

Lemma 8 together with Corollary 2 proves that (k, n)-COLOR uses at most twice the minimum possible number of colors.

6 A Quasipolynomial Time Algorithm

Consider the decision problem k-SCFSubsetIntervals:

> *"Given interval hypergraph* $H = ([n], \mathcal{I})$ *and integer* q*, is it true that* $\chi_k^*(H) \le q$*?"*

Algorithm DECIDE-COLORS (Fig. 7) is a *non-deterministic* algorithm for decision problem k-SCFSubsetIntervals.

The algorithm scans points from 1 to n, tries for every point non-deterministically every color in $\{0, \dots, q\}$, and checks if all intervals in $\mathcal{I}$ ending at the current point have the k-strong conflict-free property. If some interval in $\mathcal{I}$ does not have the k-strong conflict-free property under a non-deterministic assignment, the algorithm answers 'no'. If all intervals in $\mathcal{I}$ have the k-strong 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 k-strong conflict-free property in the following space-efficient way. For every color c in $\{0, \dots, q\}$, we keep track of:

(a) the closest point to t colored with c in variable p_c, and

```
DECIDE-COLORS(n, I, k, q)
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}
        CountUnique = 0.
        for c = 1 to q
            if s_c < j ≤ p_c then increment CountUnique.
        if CountUnique < min(k, t − j + 1) then return NO.
return YES.
```

Fig. 7 A non-deterministic algorithm deciding whether $\chi_k^*(H) \leq q$

(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 $s_c < j \leq p_c$. We count the number of uniquely occurring colors in $[j, t]$ in variable CountUnique.

Lemma 9 *The space complexity of algorithm DECIDE-COLORS is* $O(q \log n)$.

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

Theorem 7 k-SCFSubsetIntervals *has a quasipolynomial time deterministic algorithm, when* $k = O(\text{polylog}(n))$.

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 $2^{O(q \log n)}$. From Sect. 5, we know that $\chi_k^*(H_n) \leq k \left\lceil \log \left(\left\lceil \frac{n}{k} \right\rceil + 1 \right) \right\rceil$ for the complete interval hypergraph H_n and since $\chi_k^*(H) \leq \chi_k^*(H_n)$, it is only sensible to run the deterministic algorithm for values of q bounded by the above value; for greater values, the answer is always yes. Therefore, when $k = O(\text{polylog}(n))$, the deterministic algorithm has time complexity $2^{O(\text{polylog}(n))}$, i.e., k-SCFSubsetIntervals has a quasipolynomial time deterministic algorithm. □

7 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 $k \geq 2$ and 2 when $k = 1$. We have shown that our analysis of the approximation ratio is tight for $k = 1$; when $k \geq 2$, we have given 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 k-SCF problem, k-SCFSUBSETINTERVALS, is unlikely to be NP-complete, unless NP-complete problems have quasipolynomial time algorithms.

For the complete interval hypergraph H_n, we have presented a k-SCF coloring using at most twice the optimal number of colors. It would be interesting to close this gap.

We studied the SCF-coloring function $C: V \to \mathbb{N}$, for which vertices colored with '0' cannot act as uniquely-colored vertices in a hyperedge. One could try to study the bi-criteria optimization problem, in which there are two minimization goals:

(a) the number $\max_{v\in V} C(v)$ of colors used (minimization of frequency spectrum use)
(b) the number $|\{v \in V \mid C(v) > 0\}|$ of vertices with positive colors (minimization of activated base stations).

Acknowledgments We are grateful to the anonymous reviewers for their comments and suggestions, which significantly helped us improve the quality of the paper.

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. Comb. 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. Algorithms **4**(4), 44 (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 congruent disks, and axis-parallel rectangles. ACM Trans. Algorithms **5**(2) (2009). doi:10.1145/1497290.1497292
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 (2011). arXiv:1107.0138
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, Boston (1993)
14. Smorodinsky, S.: Conflict-free coloring and its applications (2010). arXiv:1005.3616
15. Zeng, G., Wang, B., Ding, Y., Xiao, L., Mutka, M.: Efficient multicast algorithms for multichannel wireless mesh networks. IEEE Trans. Parallel Distrib. Syst. **21**, 86–99 (2010)