SIAM J. DISCRETE MATH.
Vol. 32, No. 2, pp. 1189–1208

EXACT AND FIXED PARAMETER TRACTABLE ALGORITHMS FOR MAX-CONFLICT-FREE COLORING IN HYPERGRAPHS*

PRADEESHA ASHOK†, ADITI DUDEJA‡, SUDESHNA KOLAY§, AND SAKET SAURABH¶

Abstract. Conflict-free coloring of hypergraphs is a very well studied question of theoretical and practical interest. For a hypergraph $H = (U, \mathcal{F})$, a conflict-free coloring of H refers to a vertex coloring where every hyperedge has a vertex with a unique color, distinct from all other vertices in the hyperedge. In this paper, we initiate a study of a natural maximization version of this problem, namely, MAX-CFC: For a given hypergraph H and a fixed $r \geq 2$, color the vertices of U using r colors so that the number of hyperedges that are conflict-free colored is maximized. By previously known hardness results for conflict-free coloring, this maximization version is NP-hard. We study this problem in the context of both exact and parameterized algorithms. In the parameterized setting, we study this problem with respect to a natural parameter—the solution size. In particular, the question we study is the following: P-CFC: For a given hypergraph, can we conflict-free color at least k hyperedges with at most r colors, the parameter being the solution size k. We show that this problem is fixed parameter tractable by designing an algorithm with running time $2^{\mathcal{O}(k \log \log k + k \log r)}(n+m)^{\mathcal{O}(1)}$ using a novel connection to the UNIQUE COVERAGE problem and applying the method of color coding in a nontrivial manner. For the special case for hypergraphs induced by graph neighborhoods we give a polynomial kernel. Finally, we give an exact algorithm for MAX-CFC running in $\mathcal{O}(2^{n+m})$ time. All our algorithms, with minor modifications, work for a stronger version of conflict-free coloring, UNIQUE MAXIMUM COLORING.

Key words. conflict-free coloring, unique-maximum coloring, FPT algorithms, maximization algorithms

AMS subject classifications. 68Q25, 68W40, 05C85, 65F30

DOI. 10.1137/16M1107462

1. Introduction.

A hypergraph H is a pair $(U, \mathcal{F})$, where U is a set of n vertices and $\mathcal{F}$ contains m subsets of U. We call these subsets *hyperedges*. Thus a general graph is a hypergraph where every hyperedge contains exactly two vertices. A k-vertex-coloring of H for $k \in \mathbb{N}$ is a function $c : U \to \{1, 2, \ldots, k\}$. A coloring is called a *proper* coloring if none of the hyperedges are monochromatic, i.e., all the vertices of the hyperedge are not of the same color. We look at a stricter version of coloring called conflict-free coloring.

DEFINITION 1. *A vertex coloring $c : U \to \{1, 2, \ldots, k\}$ of a hypergraph $H = (U, \mathcal{F})$ is said to be* conflict-free *if for every $F \in \mathcal{F}, \exists v \in F$ such that $\forall u \in F$, $u \neq v$ implies $c(u) \neq c(v)$. In other words, every hyperedge has a uniquely colored vertex.*

The minimum number of colors required to conflict-free color the vertices of a hypergraph H is called the *conflict-free chromatic number* of H and is represented

*Received by the editors December 9, 2016; accepted for publication (in revised form) February 12, 2018; published electronically June 5, 2018. A preliminary version of this paper appeared in *Algorithms and Computation*, Lecture Notes in Comput. Sci. 9472, Springer, Berlin, 2015, pp. 271–282.

http://www.siam.org/journals/sidma/32-2/M110746.html

Funding: The research leading to these results has received partial funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 306992.

†International Institute of Information Technology, Bangalore, India (pradeesha@iiitb.ac.in).

‡Rutgers University, New Brunswick, NJ 08901 (ad1222@rutgers.edu).

§Eindhoven University of Technology, Eindhoven, Netherlands (s.kolay@tue.nl).

¶The Institute of Mathematical Sciences, Taramani, India and Department of Informatics, University of Bergen, Bergen, Norway (saket@imsc.res.in).

as $\chi_{cf}(H)$. For a given hypergraph H, the *minimum conflict-free coloring problem* refers to computing the value of $\chi_{cf}(H)$.

The concept of conflict-free coloring was introduced for hypergraphs induced by geometric regions, motivated by the frequency allocation problem in cellular networks [5]. This problem also found applications in areas like radio frequency identification and robotics. The conflict-free coloring question has been extensively studied for hypergraphs induced by various geometric regions [1, 10, 16].

The study of conflict-free coloring in general hypergraphs was initiated in [5] and in the Ph.D. thesis of Smorodinsky [15]. Pach and Tardos [14] gave an upper bound of $O(\sqrt{m})$ on the conflict-free chromatic number. This proof is derived from the work of Cheilaris [3] in his Ph.D. thesis. On the algorithmic side, the minimum conflict-free coloring problem for a general hypergraph is NP-hard by results shown in [5, 8]. Pach and Tardos [14] also studied the conflict-free coloring of hypergraphs induced by graph neighborhoods. Here the vertex set of the hypergraph corresponds to the vertex set of a general graph $G = (V, E)$ and the hyperedges are defined by the neighborhoods (open or closed) of the vertices in G. In the case of closed neighborhoods, [14] showed an upperbound of $O(\log^2 n)$ and a matching lower bound was shown in [9]. When the hyperedges are defined with respect to open neighborhoods, a tight bound of $O(\sqrt{(n)})$ was shown by Cheilaris [3]. Gargano and Rescigno [8] studied the minimum conflict-free coloring of hypergraphs induced by graph neighborhoods (both open and closed) and showed NP-completeness. Gargano and Rescigno [8] also showed that the minimum conflict-free coloring problem for these graphs becomes tractable when parameterized by the vertex cover or the neighborhood diversity number of the graph. Specifically, they gave an algorithm that decides whether a hypergraph induced by neighborhoods of a graph G can be conflict-free colored using k colors. This algorithm runs in time $2^{\mathcal{O}(kt \log k)}$, where t represents the neighborhood diversity number of G. Note that this also implies an algorithm to solve the minimum conflict-free coloring problem in hypergraphs induced by graph neighborhoods, which runs in $\mathcal{O}(n^n)$ time.

In this paper, we study a maximization version of the Minimum Conflict-Free Coloring problem.

Maximum Conflict-free Coloring(Max-CFC)
Input: A hypergraph $(U, \mathcal{F})$ on n vertices and m hyperedges, and an integer $r \geq 2$.
Output: A maximum-sized subfamily of hyperedges that can be conflict-free colored with r colors.

The NP-hardness of this problem follows from the NP-hardness reductions shown in [8]. We give an exact algorithm for this problem that runs in $\mathcal{O}(2^{m+n}) \cdot n^{\mathcal{O}(1)}$ time. As a corollary, we obtain an exact algorithm, of running time $\mathcal{O}(4^n) \cdot n^{\mathcal{O}(1)}$, for hypergraphs induced by neighborhoods in graphs. We also define a stronger variant of conflict-free coloring (cfc), namely, *unique-maximum coloring* [4].

Definition 2. *A vertex coloring $c : U \to \{1, 2, \ldots, k\}$ is said to be* unique-maximum *if for every $F \in \mathcal{F}, \exists v \in F$ such that $\forall u \in F$, $u \neq v$ implies $c(u) < c(v)$. In other words, the maximum color occuring in a hyperedge occurs uniquely. The minimum number of colors required to unique-maximum color H is called the unique-maximum chromatic number of H.*

A vertex of a hyperedge $h \in \mathcal{F}$ is said to be unique-maximum colored if it is the unique vertex that is colored with the maximum color occuring in the the hyperedge h. For a given hypergraph H, the *minimum unique-maximum coloring problem* refers to computing the minimum number of colors required to unique-maximum color H.

Similar to Max-CFC, we can define Maximum Unique-Maximum Coloring (Max-UMC) to take as input a hypergraph H and a positive integer $r \geq 2$, and output the largest subfamily of hyperedges that has a unique-maximum coloring with r colors. Our algorithms for Max-CFC, with some modification, also works for Max-UMC.

In the parameterized setting, we study Max-CFC parameterized by the solution size.

---

p-CFC **Parameter:** k
Input: A hypergraph $(U, \mathcal{F})$ on n vertices and m hyperedges, and positive integers $r \geq 2$ and k.
Question: Is there a subfamily of at least k hyperedges that can be conflict-free colored using r colors?

We also study this problem when we restrict the input hypergraph to that induced by the closed/open neighborhood of a graph G. Similarly, p-UMC is defined and studied. Note that another natural parameter for the Minimum Conflict-Free Coloring or the Max-CFC problems is the number of colors used for the conflict-free coloring. However, due to the NP-hardness result of [8], such a parameterization makes the parameterized problem para-NP-hard, which is not expected to be in fixed parameter tractable (FPT).

Our results and methods. In the realm of parameterized algorithms, we obtain the following result.

1. We show that the problem is FPT by designing a kernel with at most 4^k vertices and $\mathcal{O}(k \log k)$ hyperedges. The kernel is obtained by finding a novel connection to the Unique Coverage problem [12]. We use this one way connection to either say that the given instance for p-CFC is a YES instance or conclude that the number of hyperedges is upper bounded by $\mathcal{O}(k \log k)$. Finally, using extremal results on the set-family we bound the number of vertices (elements) to 4^k. Moreover, when we restrict the input hypergraph to that induced by the closed/open neighborhood of a graph G, then the above imply polynomial kernels for these variants.
2. A direct consequence of our kernel is an $r^{4^k}(n+m)^{\mathcal{O}(1)}$ algorithm for p-CFC. We exploit the fact that the number of hyperedges is at most $\mathcal{O}(k \log k)$ in the reduced instance to design an FPT algorithm where the running time is $2^{\mathcal{O}(k \log \log k + k \log r)}(n+m)^{\mathcal{O}(1)}$. We arrive at the required algorithm by combining the fact that we have a small number of hyperedges and using the technique of color coding introduced in [2] in a nontrivial manner.
3. All the above results, with minor modifications, hold for p-UMC.

Finally, we design an exact algorithm that solves the Max-CFC problem for general hypergraphs. This algorithm exploits structural properties of a YES instance for Max-CFC. Our algorithm runs in $\mathcal{O}(2^{m+n})$ time. The algorithm also works for the Minimum Conflict-Free coloring problem. In particular, for hypergraphs induced by graph neighborhoods, our algorithm runs in time $\mathcal{O}(4^n)$ which is a nontrivial improvement over the best known exact algorithm that runs in $\mathcal{O}(n^n)$ time [8]. The algorithm is based on dynamic programming combined with an application of subset convolution. We refer to [7] for a more detailed introduction to exact algorithms. Some minor modifications to our algorithm give an exact algorithm for Unique Maximum Coloring.

2. Preliminaries. A set of consecutive integers $\{1, 2, \ldots n\}$ will be written as $[n]$ in short. We denote a hypergraph as $H = (U, \mathcal{F})$. We refer to the objects in the universe U as either vertices or elements, and each subset of $\mathcal{F}$ as a *hyperedge*. For any subfamily $\mathcal{F}' \subseteq \mathcal{F}$, we denote the elements present in the subfamily as $U(\mathcal{F}')$. Similarly, for a subset $U' \subseteq U$, $\mathcal{F}|_{U'}$ denotes the family of hyperedges obtained when we restrict each hyperedge of $\mathcal{F}$ to the subset U'. Furthermore, for a vertex $v \in U$, by $\mathsf{deg}_H(v)$ we denote the number of hyperedges the vertex v is part of. The neighborhood of a vertex $v \in U$, denoted by $\mathsf{Nbd}_H(v)$, is the subfamily of hyperedges in $\mathcal{F}$ that contain v.

Given a graph $G = (V, E)$, for a vertex $v \in V$ let $N(v)$ be the set of neighboring vertices of v in G. Let $H = (V, \mathcal{E})$ be a hypergraph defined on the vertex set V such that $\mathcal{E} = \{N(v)|v \in V\}$. Then H is called the *hypergraph induced by the neighborhoods* of the graph G.

Parameterized algorithms. The instance of a parameterized problem is a pair containing the actual problem instance of size n and a positive integer called a parameter, usually represented as k. The problem is said to be in FPT if there exists an algorithm that solves the problem in $f(k)n^{\mathcal{O}(1)}$ time, where f is a computable function. The problem is said to admit a $g(k)$-sized kernel if there exists a polynomial time algorithm that converts the actual instance to a reduced instance of size $g(k)$, while preserving the answer. When g is a polynomial function, then the problem is said to admit a polynomial kernel. A reduction rule is a polynomial time procedure that changes a given instance I_1 of a problem Π to another instance I_2 of the same problem Π. We say that the reduction rule is *safe* when I_1 is a YES instance of Π if and only if I_2 is a YES instance. Readers are requested to refer to [6] for more details.

Fast subset convolution computation. Suppose we are given a universe U with n elements. The subset convolution of two functions $f, g : 2^U \to \mathbb{Z}$ is a function $(f * g) : 2^U \to \mathbb{Z}$ such that for every $Y \subseteq U$, $(f * g)(Y) = \Sigma_{X \subseteq Y} f(X)g(Y - X)$. It is equivalent to saying that $(f * g)(Y) = \Sigma_{A \uplus B = Y} f(A)g(B)$.

PROPOSITION 3 (see [7]). *For two functions $f, g : 2^U \to \mathbb{Z}$, given all the 2^n values of f and g in the input, all the 2^n values of the subset convolution $f * g$ can be computed in $\mathcal{O}(2^n \cdot n^3)$ arithmetic operations.*

In fact, the result can be extended to a subset convolution of functions that map to any ring, instead of $(\mathbb{Z}, +, \times)$. Consider the set $\mathbb{Z} \cup \{\infty\}$, with the added relation that $\forall z \in \mathbb{Z}, \{\infty\} > z$. The *min* operator takes two elements from this set and outputs the minimum of the two elements. Notice that $\mathbb{Z} \cup \{\infty\}$, along with *min* as an additive operator and $+$ as a multiplicative operator, forms a semiring. We will call this semiring the integer min-sum semiring. The subset convolution of two functions $f, g : 2^U \to \mathbb{Z} \cup \{\infty\}$, with *min* and $+$ as the additive and multiplicative operators, becomes $(f * g)(Y) = min_{A \uplus B = Y} f(A) + g(B)$.

PROPOSITION 4 (see [7]). *Given two functions $f, g : 2^U \to \{-M, \ldots, M\}$, all the 2^n values of f and g in the input, and all the 2^n values of the subset convolution $(f * g)$ over the integer min-sum semiring can be computed in time $2^n n^{\mathcal{O}(1)} \cdot \mathcal{O}(M \log M \log \log M)$.*

For more details about subset convolutions and fast calculations of subset convolutions, please refer to [7].

Exact algorithms. Although all NP-complete problems can be solved by some brute-force algorithm, the running time of these algorithms can be extremely large even for some small input. However, for some of these problems, we can design superpolynomial algorithms which are considerably faster than brute force. Such algorithms which solve NP-complete problems optimally are called exact algorithms. At times, these may even be practical for moderate or small instance sizes.

3. FPT Algorithm for p-CFC.

We are given a hypergraph $\mathcal{H} = (U, \mathcal{F})$ as input and two positive integers, k and r. In this section, we give an FPT algorithm for p-CFC on hypergraphs, parameterized by k. In other words, we wish to find out if k hyperedges can be conflict-free colored using r colors. *For simplicity, throughout this section, we assume that we are given a simple hypergraph, that is, no hyperedges are repeated.* We first give a kernel and then use this kernel to get the desired FPT algorithm.

3.1. Kernel for p-CFC.

We begin with a simple observation that if $r > 2\sqrt{k}$, then we can conflict-free color any subfamily of k hyperedges with r colors. This follows directly from the upper bound of $\mathcal{O}(\sqrt{m})$ on the conflict-free chromatic number obtained in [14]. Thus, for the remaining section, we assume that $r \leq 2\sqrt{k}$.

We can also preprocess the input instance to detect simple YES instances of the problem, by applying the following reductions to the instance.

REDUCTION 1. *If there is a vertex $v \in U$ such that $\mathsf{deg}_H(v)$ is at least k, output a trivial* YES *instance.*

LEMMA 5. *Reduction rule* 1 *is* safe.

Proof. Reduction rule 1 is safe since we can obtain a 2 cfc for at least k hyperedges in the following way: Assigning the first color to v and the second color to all the other vertices gives us at least k conflict-free colored hyperedges. ∎

Next, we draw a connection between p-CFC and the UNIQUE HITTING SET (UHS) problem. In UHS, we take a hypergraph H and a positive integer k as input. The question is to decide whether there is a set S of vertices and a subfamily $\mathcal{F}'$ of at least k hyperedges such that each hyperedge in $\mathcal{F}'$ contains exactly 1 vertex from S. In other words, each hyperedge of $\mathcal{F}'$ needs to be uniquely hit by S.

OBSERVATION 1. *Given a hypergraph H and a positive integer k, if (H, k) is a* YES *instance for* UHS*, then $(H, k, r = 2)$ is a* YES *instance for* p-CFC*.*

Proof. Let S be a solution for (H, k) as an instance for UHS, and let $\mathcal{F}'$ be a set of at least k hyperedges that are uniquely hit by S. We color the vertices of S with the first color and the vertices of $U \setminus S$ with the second color. This coloring function conflict-free colors all hyperedges of $\mathcal{F}'$. Thus, $(H, k, 2)$ is a YES instance for p-CFC. ∎

The UHS problem, in turn, is related to the UNIQUE COVERAGE (UC) problem. In UC, we take a hypergraph H and a positive integer k as input. The question is to decide whether there is a subfamily $\mathcal{F}'$ of hyperedges and a set S of at least k vertices such that each vertex in S belongs to exactly 1 hyperedge of $\mathcal{F}'$. In other words, each vertex of S needs to be uniquely covered by $\mathcal{F}'$.

LEMMA 6. *An instance $(H = (U, \mathcal{F}), k)$ of* UHS *has an equivalent instance $(H' = (\hat{U}, \hat{\mathcal{F}}), k)$ of* UC*, where the parameter remains the same, and $|U| = |\hat{\mathcal{F}}|, |\hat{U}| = |\mathcal{F}|$. In fact, H' is the dual hypergraph of H.*

Proof. Given the instance (H, k) of UHS, we construct the equivalent instance $(H' = (\hat{U}, \hat{\mathcal{F}}), k)$ of UC in the following manner:

- For every hyperedge $h \in \mathcal{F}$, we create a new vertex u_h. $\hat{U} = \{u_h \mid h \in \mathcal{F}\}$. $|\hat{U}| = |\mathcal{F}|$.
- For each vertex $v \in U$, let $\mathcal{F}_v = \{h \mid h \in \mathcal{F}, v \in h\}$. Define$T_v = \{u_h \mid h \in \mathcal{F}_v\}$. $\hat{\mathcal{F}} = \{T_v \mid v \in U\}$. Thus each hyperedge in H' corresponds to a vertex of H and $|U| = |\hat{\mathcal{F}}|$

Suppose S was a solution of UHS for (H, k) and let $\mathcal{F}'$ be the set of at least k hyperedges that are uniquely hit by S. Then, consider the subset $\hat{U}_{\mathcal{F}'} \subseteq \hat{U}$, where $\hat{U}_{\mathcal{F}'} = \{u_h \mid h \in \mathcal{F}'\}$, and the subfamily $\hat{\mathcal{F}}_S \subseteq \hat{\mathcal{F}}$, where $\hat{\mathcal{F}}_S = \{T_v \mid v \in S\}$. By the property of S and $\mathcal{F}'$, every vertex of $\hat{U}_{\mathcal{F}'}$, which is of size at least k, is contained in exactly one hyperedge of $\hat{\mathcal{F}}_S$ and therefore (H', k) is a YES instance of UC.

Similarly, let $\hat{\mathcal{F}}'$ be a solution of UC for (H', k) and let $\hat{S}$ be the set of at least k vertices that are uniquely covered by $\hat{\mathcal{F}}'$. Then, consider the subfamily $\mathcal{F}_{\hat{S}} \subseteq \mathcal{F}$, where $\mathcal{F}_{\hat{S}} = \{h \mid u_h \in \hat{S}\}$, and the subset $U_{\hat{\mathcal{F}}'} \subseteq U$, where $U_{\hat{\mathcal{F}}'} = \{v \mid T_v \in \hat{\mathcal{F}}'\}$. By the property of $\hat{S}$ and $\hat{\mathcal{F}}'$, the vertex set $U_{\hat{\mathcal{F}}'}$ uniquely hits $\mathcal{F}_{\hat{S}}$, which is of size at least k. Therefore (H, k) is a YES instance of UHS. ∎

The UC problem has been studied in the field of parameterized complexity. When k, the number of vertices to be uniquely covered, is the parameter, the problem was shown to be in FPT in [12]. The following proposition was proved in [12], and we will shortly show how this is useful to us.

PROPOSITION 7 (see [12, Lemma 17]). *Let $(H = (U, \mathcal{F}), k)$ be an instance of* UC *such that every hyperedge has size at most $k-1$. Then there exists a constant α_{uc} such that if $|U| \geq \alpha_{uc} k \log k$ then $(H = (U, \mathcal{F}), k)$ is a* YES *instance and, furthermore, in polynomial time, it is possible to find a subfamily covering at least k elements uniquely.*

We use Proposition 7 to bound the universe size for p-CFC.

LEMMA 8. *Let $(H = (U, \mathcal{F}), k, r)$ be an instance of* p-CFC. *Then in polynomial time, either we can conclude that (H, k, r) is a* YES *instance of* p-CFC *or $|\mathcal{F}| \leq \alpha_{uc} k \log k$.*

Proof. Let $(H = (U, \mathcal{F}), k, r)$ be an instance of p-CFC. We first check whether Reduction 1 applies. If it does not apply then we know each element of U appears in at most $k-1$ sets. Now we consider (H, k) as an instance for UHC and apply the reduction given in Lemma 6 to obtain an equivalent instance $(H' = (\hat{U}, \hat{\mathcal{F}}), k)$ of UC. Observe that since each element of U appears in at most $k-1$ sets, we have that every hyperedge in $\hat{\mathcal{F}}$ has size at most $k-1$. Furthermore, since H is a simple hypergraph no elements in $\hat{U}$ repeat. Now we apply Proposition 7 on $(H' = (\hat{U}, \hat{\mathcal{F}}), k)$. This tells us that either $|\hat{U}| = |\mathcal{F}| \leq \alpha_{uc} k \log k$ or $(H' = (\hat{U}, \hat{\mathcal{F}}), k)$ is a YES instance of UC. In the latter case, combining Lemma 6 and Observation 1 we have that $(H = (U, \mathcal{F}), k, r)$ is a YES instance of p-CFC. ∎

Thus, from now onwards, we assume our instance to have at most $\mathcal{O}(k \log k)$ hyperedges. Using an extremal result on set systems [11], we obtain the following.

THEOREM 9. p-CFC *has a kernel with at most 4^k vertices and $\mathcal{O}(k \log k)$ hyperedges.*

For the proof of Theorem 9, we use the following result. The following definition is required for our purpose. Given a family of sets $\mathcal{F} = \{F_1, \ldots, F_m\}$, an m-tuple $(x_1, \ldots, x_m)$ is said to be a strong system of distinct representatives if all the elements x_i are distinct and $x_i \in F_i$ for all $i = 1, 2, \ldots, m$, and $x_i \notin F_j$ for all $i \neq j$. We use the following result from [11, Theorem 8.12].

PROPOSITION 10. *In any family of more than $\binom{r+s}{s}$ sets of cardinality at most r, at least $s+2$ of its members have a strong system of distinct representatives.*

Proof of Theorem 9. Given a hypergraph $H = (U, \mathcal{F})$, we consider a family of sets $\mathcal{F}'$: For each vertex in $u \in U$, we define the set $\mathcal{F}_u = \{F \in \mathcal{F} \mid u \in F\}$ in $\mathcal{F}'$. Clearly, $\mathcal{F}'$ has n sets, one for each vertex. Since the degree of every vertex $u \in U$ is bounded by k, the size of $\mathcal{F}_u$ is also bounded by k. Suppose there exists a strong system of distinct representatives for k members of $\mathcal{F}'$ i.e, there exists k hyperedges in $\mathcal{F}$ such that each of them has a vertex that does not appear in any other hyperedge. Then, by coloring these vertices by color 1 and giving everything else a different color, we can 2 conflict-free color these k hyperedges. Now by substituting $r = k-1$ and $s = k-2$ in the statement of Proposition 10, we know that if $\mathcal{F}'$ has more than $\binom{2k-3}{k-2} \le 2^{2k}$ sets i.e., if U has more than 2^{2k} vertices, we can say YES. Thus, combining this and Lemma 8, we get our result. ∎

We also get the following corollary.

COROLLARY 11. P-CFC *for hypergraphs induced by neigborhoods of graphs admits polynomial kernels.*

Corollary 11 follows from Lemma 8 and the fact that the number of hyperedges is the same as the number of vertices in hypergraphs induced by graph neigborhoods.

Theorem 9 immediately implies that P-CFC is FPT. Given an instance $(H = (U, \mathcal{F}), k, r)$ of P-CFC, by using Theorem 9, we either conclude that $(H = (U, \mathcal{F}), k, r)$ is a YES instance of P-CFC or we have that $|U| \le 4^k$. Now we look at every r-partition of U and check whether there are k hyperedges that are conflict-free colored. If we succeed for any partition then we return YES, else we conclude that the given instance is a NO instance. The running time of this algorithm is upper bounded by $r^{4^k}(|U| + |\mathcal{F}|)^{\mathcal{O}(1)}$.

Faster FPT algorithm for P-CFC. Let $N = |U| + |\mathcal{F}|$. In this section, we give the full description of an FPT algorithm for P-CFC that runs in $2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}$ time. We will assume that our input instance contains at most $\mathcal{O}(k \log k)$ hyperedges and 4^k vertices.

We first define some related concepts. Given a set $S \subseteq U$, a subfamily $\mathcal{F}'$, and a coloring $\Gamma : U \to [r]$, we say that S is a *cfc-solution* with respect to $\mathcal{F}'$ if each hyperedge h in $\mathcal{F}'$ is conflict-free colored and a uniquely colored vertex of h belongs to S. Furthermore, given such a set S and a hyperedge h, let $\mathsf{unicoleIt}_S(h)$ denote one uniquely colored vertex of h that is arbitrarily chosen from all the uniquely colored vertices of h belonging to S.

Our algorithm uses color-coding technique, introduced by Alon, Yuster, and Zwick in [2], which can be broadly defined as follows. Given an input G, suppose the problem is to find a subset H of G with some property and of size k. The color-coding technique considers a random coloring of the elements of G with k colors and tries to find a colorful copy of H.

Suppose we know the following information about the solution (if it exists):

- the subfamily $\mathcal{F}' \subseteq \mathcal{F}, |\mathcal{F}'| = k$ such that all the hyperedges in $\mathcal{F}'$ are conflict-free colored;
- a partition of the universe into p parts, $U = U_1 \uplus U_2 \cdots \uplus U_p$ such that the cfc-solution S contains exactly one element from each partition;
- $\forall F \in \mathcal{F}'$, the color assigned to the uniquely colored element in $S \cap F$;
- $\forall 1 \le i \le p$, the color assigned to the element in $S \cap U_i$.

Note that in the last two points, we do not know the element that is uniquely colored but only the color used to uniquely color it. The partition of the universe corresponds to the random coloring by the color-coding technique. The assumption that this information is known is reasonable as all of this information can be guessed in FPT time. So we work with an auxiliary problem in which all this information is known. We also maintain a list of admissible colors for each uncolored vertex that agrees with this information. The first part of the algorithm does some preprocessing that handles trivial cases and thus filters the list for each vertex. Now we make the crucial observation that for a fixed color j, if we restrict the instance to parts of the universe that are to be conflict-free colored by j (by item 2), the vertices in these parts to those for which j is an admissible color, and hyperedges to those that are to be conflict-free colored by j (by item 3) then the problem reduces to that of selecting, for each color j, a subset of vertices such that there is exactly one vertex from each hyperedge and each part of the universe. This is a special case of the UHS problem where the elements are colored and we need a colorful UHS. Using known methods, we solve the above problem for each color in the conflict-free solution.

In what follows, we formally define the auxiliary problem and give an FPT algorithm for this auxiliary problem.

Partitioned p-CFC **Parameter:** $r + p + |\mathcal{F}|$

Input: A hypergraph $(U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F})$, a function $\Psi_{\mathsf{family}} : \mathcal{F} \to [r]$, a function $\Psi_{\mathsf{parts}} : [p] \to [r]$, a subset $U' \subseteq U$, and a coloring function $\Gamma' : U' \to [r]$ for every $v \in U \setminus U'$, a list $L_v \subseteq [r]$.

Question: Does there exist a coloring function $\Gamma : U \to [r]$ such that each hyperedge is conflict-free colored, $\Gamma(U') = \Gamma'(U')$ for each $v \in U \setminus U', \Gamma(v) \in L_v$. Also, there exists a cfc-solution set S of size exactly p, for all $i \in [p]$, $|S \cap U_i| = 1$, and for every $h \in \mathcal{F}$, $\mathsf{unicolelt}_S(h) \in \bigcup_{j \in \Psi_{\mathsf{parts}}^{-1}(\Psi_{\mathsf{family}}(h))} U_j$?

In simple words, the problem definition can be explained as follows. We are given a partitioning of the universe U into p-parts and a partial coloring function Γ' on a subset U'. We are looking for a coloring $\Gamma : U \to [r]$ which extends Γ'. Each vertex v in $U \setminus U'$ has a list of admissible colors, and Γ must choose a color from L_v. Also, due to Γ, each hyperedge is conflict-free colored and there exists a cfc-solution set S such that it contains exactly one vertex from each part. Suppose the hypothetical set S is $\{x_1, x_2, \ldots, x_p\}$ (think of x_i as a variable), where $x_i \in U_i$. The function Ψ_{parts} is used to guess the color of x_i in Γ. The function Ψ_{family} divides the family $\mathcal{F}$ into r *chunks* (not to be confused with parts and coloring). The idea is that the uniquely colored vertex of $h \in \mathcal{F}$, say x_j, has been assigned the same color by Γ as h has been assigned to the chunk number by Ψ_{family}, i.e, $\Gamma(x_j) = \Psi_{\mathsf{family}}(h)$. Next we show how we can solve the Partitioned p-CFC problem.

Given an instance $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ of Partitioned p-CFC, we first do a polynomial time preprocessing of the instance. For all $v \in U'$, we must set $\Gamma(v) = \Gamma'(v)$. In the following reduction rules, we show that the input functions Ψ_{family} and Ψ_{parts} allow us to prune the list of some of the vertices. The first reduction rule deals with hyperedges h where $|\Gamma'^{-1}(\Psi_{\mathsf{family}}(h)) \cap h| = 1$.

Reduction 2. *Suppose there is a hyperedge h containing a unique vertex $w \in U'$ such that $\Psi_{\mathsf{family}}(h) = \Gamma'(w)$. Then, for every $v \in h \setminus \{w\}$ we delete $\Psi_{\mathsf{family}}(h)$ from L_v. We also delete h from $\mathcal{F}$.*

LEMMA 12. *Reduction rule* 2 *is safe.*

Proof. Suppose S is a potential cfc-solution for the given instance. Also, let $\mathsf{unicolelt}_S(h) = x$. This means that no other vertex of h should be assigned the color $\Gamma(x)$. Since $x \in \bigcup_{j \in \Psi_{\mathsf{parts}}^{-1}(\Psi_{\mathsf{family}}(h))} U_j$, it implies $\Gamma(x) = \Psi_{\mathsf{family}}(h)$. As $\Psi_{\mathsf{family}}(h) = \Gamma'(w)$, any satisfying assignment Γ must have $\Gamma(x) = \Gamma'(w)$, and it must be the case that $x = w$. Then, all other vertices $v \in U \setminus U'$ of h must get a color different from $\Gamma(w) = \Psi_{\mathsf{family}}(h)$. Thus, we have identified the vertex that will determine cfc of h and ensured that no other vertex of h can destroy the uniqueness of w. Thus, in the reduced instance, we can delete h and for every $v \in h \setminus \{w\}$, we remove $\Psi_{\mathsf{family}}(h)$ from L_v.

On the other hand, suppose the reduced instance has a satisfying coloring Γ. For each $v \in U \setminus U'$, the list of admissible colors of the original instance is a superset of the list of admissible colors from the reduced instance. Since $w \in U'$ in the reduced instance, $\Gamma(w) = \Gamma'(w)$. But in the original instance, the function Γ' was the same as in the reduced instance. Therefore, $\Psi_{\mathsf{family}}(h)$ of the original instance is the same as $\Gamma(w)$ of the reduced instance. Also, in the reduced instance, no other vertex that belonged to h contains the color $\Gamma(w)$ in its list. Hence, in the original instance, the same assignment Γ will conflict-free color h as well, and is a satisfying assignment of the original instance. Thus, reduction rule 2 is safe. ∎

We can further reduce the size of the lists by the following reduction.

REDUCTION 3. *If there is a vertex* $v \in U_i, i \in [p]$, *and* $h \in \mathcal{F}$, *such that* $v \in h$, $\Psi_{\mathsf{family}}(h) \neq \Psi_{\mathsf{parts}}(i)$, *then we remove the color* $\Psi_{\mathsf{family}}(h)$ *from the list of* v.

LEMMA 13. *Reduction rule* 3 *is safe.*

Proof. Suppose, in a potential cfc-solution S, the uniquely colored vertex of h was x_j. By definition, $x_j \in \bigcup_{i \in \Psi_{\mathsf{parts}}^{-1}(\Psi_{\mathsf{family}}(h))} U_i$. In other words, x_j belongs to a part U_i such that $\Psi_{\mathsf{parts}}(i) = \Psi_{\mathsf{family}}(h)$ and $\Gamma(x_j) = \Psi_{\mathsf{family}}(h)$. However, for the given vertex v and hyperedge h, $\Psi_{\mathsf{family}}(h) \neq \Psi_{\mathsf{parts}}(i)$ and, hence, $v \notin \bigcup_{i \in \Psi_{\mathsf{parts}}^{-1}(\Psi_{\mathsf{family}}(h))} U_i$. Thus, $v \neq x_j$. Since x_j is uniquely colored in h, $\Gamma(v) \neq \Gamma(x_j)$. Therefore, $\Gamma(v) \neq \Psi_{\mathsf{family}}(h)$ and we can safely delete $\Psi_{\mathsf{family}}(h)$ from L_v in the reduced instance. On the other hand, the reduced instance has admissible color lists which are subsets of the admissible color lists of the original instance. Suppose the reduced instance had a coloring Γ such that each hyperedge is conflict-free colored, and there exists a cfc-solution set S of size exactly p, for all $i \in [p]$, $|S \cap U_i| = 1$ and for every $h \in \mathcal{F}$, $\mathsf{unicolelt}_S(h) \in \bigcup_{j \in \Psi_{\mathsf{parts}}^{-1}(\Psi_{\mathsf{family}}(h))} U_j$. This Γ is a satisfying coloring for the original instance as well. Thus, reduction rule 3 is safe. ∎

The next rule deals with hyperedges h, where $|\Gamma'^{-1}(\Psi_{\mathsf{family}}(h)) \cap h| \geq 2$.

REDUCTION 4. *If there are two vertices* $v, w \in U'$ *and a hyperedge* $h \in \mathcal{F}$, *such that* $\Psi_{\mathsf{family}}(h) = \Gamma'(v) = \Gamma'(w)$, *then we output a trivial* NO *instance.*

LEMMA 14. *Reduction rule* 4 *is safe.*

Proof. Suppose, in a potential cfc-solution S, the uniquely colored vertex of h was x. By definition, $\Gamma(x) = \Psi_{\mathsf{family}}(h)$. Also, by the uniqueness of x, no other vertex of h should be assigned the color $\Gamma(x)$. However, in our instance, there are already two vertices v, w of $h \cap U'$ which have been assigned $\Psi_{\mathsf{family}}(h)$ by Γ'. This means that h cannot be conflict-free colored by any satisfying assignment Γ. Thus, we correctly output a trivial NO instance. ∎

REDUCTION 5. *Suppose there is a vertex $w \in U \setminus U'$ with $L_w = \{c\}$, then we put w in U' and set $\Gamma'(w) = c$. If there is a vertex v, where $L_v = \emptyset$, then we output a trivial* NO *instance.*

LEMMA 15. *Reduction rule* 5 *is safe.*

Proof. Γ must assign a color to every vertex. If there is a vertex w with $L_w = \{c\}$, then we must set $\Gamma(w) = c$ for any satisfying assignment Γ. Thus, in the reduced instance, we fix the coloring of w by putting it in U' and setting $\Gamma'(w) = c$. On the other hand, in the reduced instance, U' is a superset of the U' of the original instance. Hence, a satisfying assignment Γ of the reduced instance is also a satisfying assignment of the original instance.

Similarly, by the correctness of the other reduction rules, if there is a vertex v where $L_v = \emptyset$, the current instance must be a NO instance. Thus we correctly output a trivial NO instance. Therefore, this reduction rule is safe. ∎

Given an instance $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ of PARTITIONED p-CFC, we apply reduction rules 2, 3, 4, 5 exhaustively. If in the process we infer that the given instance is a NO instance then we return the same. It could also happen that we get $\mathcal{F} = \emptyset$. In this case for every vertex $v \in U \setminus U'$, Γ assigns to v an element of $L(v)$ arbitrarily. Thus, from now onwards we assume that we neither conclude that the given instance is a NO instance nor obtain $\mathcal{F} = \emptyset$. We call an instance of PARTITIONED p-CFC *reduced* if reduction rules 2, 3, 4, 5 are not applicable. For simplicity, let $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma',$ $\{L_v \subseteq [r] | v \in U \setminus U'\})$ denote the reduced instance of PARTITIONED p-CFC. Observe that the reduced instance has the following properties:

1. For every vertex v, $|L_v| \geq 2$. This is because reduction rule 5 is not applicable.
2. For every hyperedge h, $|\Gamma'^{-1}(\Psi_{\mathsf{family}}(h)) \cap h| = 0$. This is because reduction rules 2 and 4 are not applicable.

We define the set $V_i \subseteq U \setminus U'$ as the set of vertices that have i in their list of admissible colors. Then, since reduction rule 3 is no longer applicable, there are two kinds of vertices in V_i: The first kind is a vertex v that has $i \in L_v$ and $\exists h \in \mathcal{F}$, $v \in U_j \cap h$ such that $\Psi_{\mathsf{family}}(h) = i, \Psi_{\mathsf{parts}}(j) = i$. The other kind of vertex v in V_i has $i \in L_v$ but for any h with $\Psi_{\mathsf{family}}(h) = i$, $v \notin h$.

To solve the reduced instance of PARTITIONED p-CFC, we will solve some r instances of an even more specialized problem that we define now.

PARTITIONED UHS
Input: A partitioned universe $U = (U_1 \uplus \ldots \uplus U_q)$ and a set family $\mathcal{F}$.
Question: Is there a set $S \subseteq U$ such that for all $h \in \mathcal{F}$, $|h \cap S| = 1$, and for all $i \in [q]$, $|U_i \cap S| = 1$?

Now we define some sets based on $V_i \subseteq U$:

1. For every $j \in [r]$, and $x \in \Psi_{\mathsf{parts}}^{-1}(j)$, let $Z_j^x = U_x \cap V_j$ and $Z_j = \bigcup_{x \in \Psi_{\mathsf{parts}}^{-1}(j)} Z_j^x$.
2. For every $j \in [r]$, and $h \in \Psi_{\mathsf{family}}^{-1}(j)$ let $h_j = h \cap V_j$ and $\mathcal{F}_j = \{h_j \mid h \in \Psi_{\mathsf{family}}^{-1}(j)\}$.

Next we relate the instance of PARTITIONED p-CFC to PARTITIONED UHS.

LEMMA 16. *Let $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ denote the reduced instance of* PARTITIONED p-CFC. *Then it is a* YES *instance of* PARTITIONED p-CFC *if and only if for all $j \in [r]$, $(\uplus_{x \in \Psi_{\mathsf{parts}}^{-1}(j)} Z_j^x, \mathcal{F}_j)$ is a* YES *instance of* PARTITIONED UHS.

Proof. First, suppose that $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ is a YES instance of Partitioned p-CFC. Then there is a satisfying assignment Γ such that each hyperedge is conflict-free colored, $\Gamma'(U') = \Gamma(U')$. For each $v \in U \setminus U', \Gamma(v) \in L_v$. Also, there exists a cfc-solution set $S = \{v_1, \ldots, v_p\}$ such that for all $x \in [p]$, $|S \cap U_x| = 1$. In the reduced instance, for all h, $|\Gamma'^{-1}(\Psi_{\mathsf{family}}(h)) \cap h| = 0$. Thus, $S \cap U' = \emptyset$. For each $j \in [r]$, we look at $S \cap V_j$. By the definition of Z_j, every vertex in $S \cap V_j$ must belong to a part in Z_j. In particular, every vertex of $S \cap \Gamma^{-1}(j)$ must belong to a part in Z_j. Also, since every vertex of S belongs to a unique part of $(U_1 \uplus U_2 \cdots \uplus U_p)$, for each $x \in [p]$ there is exactly one vertex in $S \cap Z_j^x$. Also, we know that for every $h \in \mathcal{F}$, if $\mathsf{unicolelt}_S(h) = v_h$, then $\Gamma(v_h) = \Psi_{\mathsf{family}}(h)$. Thus, for each hyperedge $h \in \mathcal{F}_j$, $\mathsf{unicolelt}_S(h) \in S \cap \Gamma^{-1}(j)$. For every other vertex $u \in h \setminus \mathsf{unicolelt}_S(h)$, $\Gamma(u) \neq j$ and, therefore, $u \notin S \cap \Gamma^{-1}(j)$. Thus, for every $j \in [r]$, $S_j = S \cap \Gamma^{-1}(j)$ is a UHS of $\mathcal{F}_j$ with the property that $\forall x \in \Psi_{\mathsf{parts}}^{-1}(j)$, $|Z_j^x \cap S_j| = 1$. Thus, $(\uplus_{x \in \Psi_{\mathsf{parts}}^{-1}(j)} Z_j^x, \mathcal{F}_j)$ is a YES instance of Partitioned UHS.

In the reverse direction, suppose $(\uplus_{x \in \Psi_{\mathsf{parts}}^{-1}(j)} Z_j^x, \mathcal{F}_j)$ is a YES instance of Partitioned UHS. Then a solution set S_j is a UHS of $\mathcal{F}_j$ with the property that $\forall x \in \Psi_{\mathsf{parts}}^{-1}(j)$, $|Z_j^x \cap S_j| = 1$. By definition, $S_j \subseteq Z_j \subseteq V_j$. First, for each vertex $v \in S_j$, we assign $\Gamma(v) = i$. For each $w \in U'$, we must set $\Gamma(w) = \Gamma'(w)$. The vertices that still need to be colored belong to $V_j \setminus S_j$ for each $j \in [r]$. First, take a vertex $w \in U_x \setminus S_j$, where $\Psi_{\mathsf{parts}}(U_x) = j$. Consider a hyperedge h such that $\Psi_{\mathsf{family}}(h) = c \neq j$. Then, since reduction rule 3 is no longer applicable, it must be the case that $c \notin L_w$. However, since reduction rule 5 is also no longer applicable, there is a color $c' \in L_w$ such that $c' \neq j$. We set $\Gamma(w) = c'$. Next, consider a vertex $w \in V_j \setminus Z_j$. Since reduction rule 3 is no longer applicable, for any h with $\Psi_{\mathsf{family}}(h) = j$, $w \notin h$. We set $\Gamma(w) = j$. Every hyperedge h has exactly one vertex in the color class $\Psi_{\mathsf{family}}(h)$, namely, the vertex in $S_{\Psi_{\mathsf{family}}(h)} \cap h$ that uniquely hit h. Thus, Γ is a satisfying assignment and $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ is a YES instance of Partitioned p-CFC. ∎

Lemma 16 allows us to reduce an instance of the Partitioned p-CFC problem to r instances of Partitioned UHS. Next, we design an algorithm for Partitioned UHS.

Lemma 17. Partitioned UHS, *where the number of hyperedges is m, the universe size is n, and a $q \leq m$ partitioning of the universe is given, is* FPT *parameterized by m. The running time of the algorithm is $2^{(m+q)} \cdot (n+m)^{\mathcal{O}(1)}$.*

Proof. We are given as input a hypergraph $H = (U, \mathcal{F})$ and a q-partition $U = (U_1 \uplus \cdots \uplus U_q)$ of the universe. We define a function $\mathcal{A}$ that takes as input a pair $(\mathcal{E}, C)$, where $\mathcal{E} \subseteq \mathcal{F}$ and $C \subseteq [q]$. The function outputs 1 if exactly the subfamily $\mathcal{E}$ can be uniquely hit by a set S, where the vertices of S come from $\bigcup_{i \in C} U_i$, and $\forall i \in C$, $|U_i \cap S| = 1\}$, and 0 otherwise. We want to emphasize that only the hyperedges of $\mathcal{E}$ should be hit by S and the hyperedges of $\mathcal{F} \setminus \mathcal{E}$ should not be hit by S. Define for each vertex v that belongs to a part $U_i, i \in C$, a subfamily $\mathcal{E}_v = \{h \mid v \in h\}$. We define the function $\mathcal{A}$ using the following recurrence relation:

$$\mathcal{A}(\mathcal{E}, C) = \begin{cases} \max\limits_{c \in C, v \in U_c, \mathcal{E}_v \subseteq \mathcal{E}} \mathcal{A}(\mathcal{E} \setminus \mathcal{E}_v, C \setminus \{c\}) & \text{if } \mathcal{E} \neq \emptyset,\, C \neq \emptyset, \\ 1 & \text{if } \mathcal{E} = \emptyset,\, C = \emptyset, \\ 0 & \text{if } \mathcal{E} = \emptyset,\, C \neq \emptyset, \\ 0 & \text{if } C = \emptyset,\, \mathcal{E} \neq \emptyset. \end{cases}$$

We prove the correctness of this recurrence by induction on the size of the set $C \subseteq [q]$. In the base case, when $\mathcal{E} = \emptyset, C = \emptyset$, then trivially this family has been uniquely hit and therefore $\mathcal{A}(\mathcal{E}, C) = 1$. When $C = \emptyset$, then a nonempty subfamily cannot be uniquely hit and therefore $\mathcal{A}(\mathcal{E}, C) = 0$ for any nonempty subfamily $\mathcal{E}$. When $\mathcal{E} = \emptyset$, but $C \neq \emptyset$, then the family cannot have a UHS S with the property $\forall i \in C,\ |U_i \cap S| = 1$. Therefore, $\mathcal{A}(\mathcal{E}, C) = 0$ for any set $C \neq \emptyset$.

Now, let $|C| \geq 1$. Suppose we have correctly calculated $\mathcal{A}(\mathcal{E}', C')$ for all pairs $(\mathcal{E}', C')$, where $|C'| < |C|$ and $\mathcal{E}' \subseteq \mathcal{F}$. There can be two cases:

1. Suppose $\mathcal{A}(\mathcal{E}, C) = 1$. Then there is a solution set S where it is true that $\forall i \in C,\ |U_i \cap S| = 1$. Take one $i \in C$ and let $v_i \in S \cap U_i$. Then $S - \{v_i\}$ uniquely hits exactly the subfamily $\mathcal{E} \setminus \mathcal{E}_{v_i}$ and $\forall j \in C \setminus \{i\},\ |U_j \cap S| = 1$. Then, by the induction hypothesis, $\mathcal{A}(\mathcal{E} \setminus \mathcal{E}_{v_i}, C \setminus \{i\}) = 1$. Hence, we correctly calculate $\mathcal{A}(\mathcal{E}, C)$.
2. On the other hand, suppose $\mathcal{A}(\mathcal{E}, C) = 0$. Then, there is no UHS S that hits exactly the subfamily $\mathcal{E}$ such that $\forall i \in C,\ |U_i \cap S| = 1$. Then we claim that there is no subproblem $(\mathcal{E} \setminus \mathcal{E}_v, C \setminus \{c\})$ such that $\mathcal{A}(\mathcal{E} \setminus \mathcal{E}_v, C \setminus \{c\}) = 1$. Assume such a subproblem existed and let S' be the UHS for that subfamily. Then S' hits all hyperedges in $\mathcal{E} \setminus \mathcal{E}_v$ and no hyperedge outside this subfamily. Then $S' \cup v$ will be a UHS for $(\mathcal{E}, C)$, a contradiction.

Thus the recurrence is correct.

It is enough to solve this recurrence for every pair $(\mathcal{E}, C)$. The given instance is a YES instance of PARTITIONED UHS if $\mathcal{A}(\mathcal{F}, [q]) = 1$. In order to calculate $\mathcal{A}(\mathcal{E}, C)$, we look up the values of subproblems corresponding to each $c \in C$ and $v \in U_c$. There are at most n such subproblems. There are $2^{(m+q)}$ such pairs $(\mathcal{E}, C)$. Thus, the running time for solving the recurrence is $2^{(m+q)}(n+m)^{\mathcal{O}(1)}$. ∎

Lemmas 16, 17, and the safeness of the reduction rules 2, 3, 4, 5 together result in the following algorithm for PARTITIONED P-CFC.

LEMMA 18. PARTITIONED P-CFC *can be solved in time* $2^{p+|\mathcal{F}|} \cdot N^{\mathcal{O}(1)}$.

Next, using Lemma 18 and the method of the color coding technique of [2] we give an algorithm for P-CFC. Towards this we need the following notion of a perfect hash family. A perfect hash family is a family of functions, whose domain is a universe U of n elements and range is a set of k elements, and with the following property: For every k-sized subset $S \subseteq U$, there is a function ζ in the family that maps S to the range injectively. That is, every element of S maps to a different number in $[k]$. The following proposition shows that such families can be constructed [13].

PROPOSITION 19. *For any n and $k \leq n$, an (n,k)-perfect hash family of size $e^k k^{\mathcal{O}(\log k)} \log n$ can be deterministically computed in time $e^k k^{\mathcal{O}(\log k)} n \log n$.*

Our main theorem is the following.

THEOREM 20. P-CFC *can be solved in time* $2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}$.

Proof. Let $((U, \mathcal{F}), k, r)$ be an instance of P-CFC. Recall that $|U| = n$, $|\mathcal{F}| = m$, and $N = n+m$. Given an instance we first apply Theorem 9 and obtain an equivalent instance with at most 4^k vertices and $\mathcal{O}(k \log k)$ hyperedges. We run through all $p \leq k$. Since the number of hyperedges in the input instance is $\alpha_{uc} k \log k$, the number of subfamilies of size k is $\binom{\alpha_{uc} k \log k}{k} \leq (\frac{e\alpha_{uc} k \log k}{k})^k \leq (e\alpha_{uc} \log k)^k$. We guess a subfamily $\mathcal{F}'$ of hyperedges that will be conflict free colored. That is, we are trying to find a coloring $\Gamma : U \to [r]$ such that each hyperedge h in $\mathcal{F}'$ is conflict-free colored. Let S be a hypothetical cfc-solution corresponding to it. In other words, for each hyperedge h in $\mathcal{F}'$, a uniquely colored vertex of h (with respect to Γ) belongs to S. We guess the size of $|S|$, say $p \leq k$. For a fixed p, let $\mathfrak{F}$ be an (n, p)-perfect hash family of size $e^p p^{\mathcal{O}(\log p)} \log n$. By the property of $\mathfrak{F}$, we know that there exists a function $\zeta \in \mathfrak{F}$ that maps S to $[p]$ injectively. Let $U_1, \ldots, U_p$ denote the partition of U given by ζ. Observe that after this we will be seeking for a cfc-solution S such that $|S \cap U_i| = 1$ for all $i \in [p]$.

Next for each hyperedge h in $\mathcal{F}'$, we guess the color of a vertex in h that is uniquely colored by Γ. There are r^k such guesses. Thus, after this guess, we define a function $\Psi_{\mathsf{family}} : \mathcal{F}' \to [r]$ such that h is assigned the color of the vertex in h that will be uniquely colored by Γ. Finally, for the potential solution set S we guess the color of each vertex given by Γ. Since we are looking for a cfc-solution set S, such that $\forall i \in [p],\ |U_i \cap S| = 1$ it is equivalent to say that we guess an r partitioning of the p parts in $U = (U_1 \uplus \cdots \uplus U_p)$. That is, the vertex of S inside U_i will be assigned to each color by Γ. To express this guess, we define another function $\Psi_{\mathsf{parts}} : [p] \to [r]$ such that $\Psi_{\mathsf{parts}}(j) = i$ if the vertex x in $S \cap U_j$ will have $\Gamma(x) = i$. Thus, there are r^p guesses for the coloring of the potential solution set S by Γ. At the end of this sequence of guesses, we have fixed a choice of hyperedges that are to be r conflict-free colored, a coloring of the potential solution set S (without actually knowing the vertices of S, this essentially means a partitioning of the parts of U), and a partitioning of the hyperedges according to which color of Γ will determine that the hyperedge is conflict-free colored. This results in the following instance of PARTITIONED P-CFC: $((U = (U_1 \uplus U_2 \cdots \uplus U_p), \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U' = \emptyset, (\forall v \in U : L_v = [r]))$. By Lemma 18 we know that we can solve this in time $2^{p+k} \cdot N^{\mathcal{O}(1)} \leq 4^k \cdot N^{\mathcal{O}(1)}$. Thus the overall running time for P-CFC is upper bounded by the number of guesses and the running time of an algorithm for PARTITIONED P-CFC. Thus, the running time of the algorithm is upper bounded by

$$\binom{\alpha_{uc} k \log k}{k} \times k \times |\mathfrak{F}| \times r^k \times r^k \times 4^k \cdot N^{\mathcal{O}(1)} = 2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}.$$

This concludes the proof. ∎

4. FPT algorithm for P-UMC. Recall that P-UMC takes as input a hypergraph $(U, \mathcal{F})$ on n vertices and m hyperedges, and positive integers $r \geq 2$ and k and determines whether there is a subfamily of at least k hyperedges that can be unique-maximum colored using r colors. A vertex coloring is a unique-maximum coloring if every hyperedge h in the hypergraph has a unique vertex that is colored with the maximum color occurring in h.

Let $N = |U| + |\mathcal{F}|$. In this section, we describe an FPT algorithm for P-UMC that runs in $2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}$ time. This algorithm is very similar to the algorithm for P-CFC. However, for the sake of completeness, we chalk out the algorithm for P-UMC, while explicitly giving details of the steps where the algorithm deviates from the one for P-CFC.

The results of Lemma 8 and Theorem 9 can be modified for P-UMC.

THEOREM 21. P-UMC *has a kernel with at most* 4^k *vertices and* $\mathcal{O}(k \log k)$ *sets.*

Therefore, we assume that our input instance for P-UMC contains $\mathcal{O}(k \log k)$ hyperedges and 4^k vertices.

Given a set $S \subseteq U$, a subfamily $\mathcal{F}'$, and a coloring $\Gamma : U \to [r]$, we say that S is a *unique-maximum-solution* if each hyperedge h in $\mathcal{F}'$ is unique-maximum colored and the vertex, which is unique-maximum colored in h, belongs to S. Furthermore, given such a set S and a hyperedge h, let $\mathsf{unicolelt}_S(h)$ denote the unique-maximum colored vertex of h that belongs to S. Our strategy for the FPT algorithm is the same as in the case of p-CFC. We define an auxiliary problem and give an FPT algorithm for this problem. Finally, we reduce our problem to the auxiliary problem with some guesses and by using the color coding technique, introduced by Alon, Yuster, and Zwick in [2], to obtain the desired algorithm for p-UMC.

Partitioned p-UMC **Parameter:** $r + p + |\mathcal{F}|$
Input: A hypergraph $(U = U_1 \uplus U_2 \cdots U_p, \mathcal{F})$, a function $\Psi_{\mathsf{family}} : \mathcal{F} \to [r]$, a function $\Psi_{\mathsf{parts}} : [p] \to [r]$, a subset $U' \subseteq U$, and a coloring function $\Gamma' : U' \to [r]$ for every $v \in U \setminus U'$, a list $L_v \subseteq [r]$.
Question: Does there exist a coloring function $\Gamma : U \to [r]$ such that each hyperedge is unique-maximum colored, $\Gamma(U') = \Gamma'(U')$. For each $v \in U \setminus U', \Gamma(v) \in L_v$. Also, the unique-maximum-solution set S, defined by Γ, is of size exactly p. For all $i \in [p]$, $|S \cap U_i| = 1$, and for every $h \in \mathcal{F}$, $\mathsf{unicolelt}_S(h) \in \bigcup_{j \in \Psi^{-1}_{\mathsf{parts}}(\Psi_{\mathsf{family}}(h))} U_j$?

Let us explain the problem definition. We are given a partitioning of the universe U into p-parts and a partial coloring function Γ' on a subset U'. We are looking for a coloring $\Gamma : U \to [r]$ which extends Γ'. Each vertex v in $U \setminus U'$ has a list of admissible colors, and Γ must choose a color from L_v. Also, due to Γ, each hyperedge is unique-maximum colored and the unique-maximum-solution set S, due to Γ, is such that it contains exactly one vertex from each part. Suppose the hypothetical set S is $\{x_1, x_2, \ldots, x_p\}$ (the x_i's can be thought of as variables), where $x_i \in U_i$. The function Ψ_{parts} is used to guess the color of x_i in Γ. The function Ψ_{family} divides the family $\mathcal{F}$ into r chunks (again the chunks are different from parts and color classes). The idea is that the unique-maximum colored vertex of $h \in \mathcal{F}$, say x_j, has been assigned the same color by Γ as h has been assigned to the chunk number by Ψ_{family}, i.e, $\Gamma(x_j) = \Psi_{\mathsf{family}}(h)$. Next we show how we can the Partitioned p-UMC problem.

Given an instance $((U = U_1 \uplus U_2 \cdots U_p, \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ of Partitioned p-UMC, we first do a polynomial time preprocessing of the instance. For all $v \in U'$, we must set $\Gamma(v) = \Gamma'(v)$. In the following reduction rules, we show that the input functions Ψ_{family} and Ψ_{parts} allow us to prune the list of some of the vertices. The first reduction rule is necessary for a unique-maximum coloring that supports the function Ψ_{family}.

Reduction 6. *For each hyperedge h, and each vertex $v \in h$, remove the colors $r \geq i > \Psi_{\mathsf{family}}(h)$ from $L(v)$.*

Lemma 22. *Reduction rule* 6 *is safe.*

Proof. The required unique-maximum coloring Γ should be such that for each hyperedge h, the unique-maximum colored vertex of h must receive the same color as $\Psi_{\mathsf{family}}(h)$. Therefore, no vertex of h can be given a color which is higher in order than $\Psi_{\mathsf{family}}(h)$. This implies that the reduction rule is safe. ∎

The next rule also ensures the properties of unique-maximum coloring.

Reduction 7. *Given a hyperedge h, if there is a vertex $w \in h \cap U'$ such that $\Gamma'(w) > \Psi_{\mathsf{family}}(h)$, then we output a trivial* NO *instance.*

Lemma 23. *Reduction rule* 7 *is safe.*

Proof. The definition of the problem requires Γ to be an extension of Γ'. It also requires the unique-maximum colored vertex of each hyperedge h to be colored by

$\Psi_{\mathsf{family}}(h)$. By definition of unique-maximum coloring, all other vertices of h must receive a color of lower order than $\Psi_{\mathsf{family}}(h)$. Therefore, for a YES instance, it must be the case that for each vertex w in $h \cap U'$, $\Gamma'(w) \leq \Psi_{\mathsf{family}}(h)$. This implies the correctness of the reduction rule. ∎

The next few reduction rules are similar to the rules described for the FPT algorithm of p-UMC.

Reduction 8. *Suppose there is a hyperedge h containing a unique vertex $w \in U'$ such that $\Psi_{\mathsf{family}}(h) = \Gamma'(w)$. Then, for every $v \in h \setminus \{w\}$ we delete $\Psi_{\mathsf{family}}(h)$ from L_v. We delete h from $\mathcal{F}$.*

The proof of correctness for this rule is very similar to that of reduction rule 2.

Reduction 9. *If there is a vertex $v \in U_i, i \in [p]$, and $h \in \mathcal{F}$, such that $v \in h$, $\Psi_{\mathsf{family}}(h) \neq \Psi_{\mathsf{parts}}(i)$, then we remove the color $\Psi_{\mathsf{family}}(h)$ from the list of v.*

This proof of correctness is similar to that of reduction rule 3.

Reduction 10. *If there are two vertices $v, w \in U'$ and a hyperedge $h \in \mathcal{F}$, such that $\Psi_{\mathsf{family}}(h) = \Gamma'(v) = \Gamma'(w)$, then we say* NO.

The proof of correctness for this reduction rule is similar to reduction rule 4.

Reduction 11. *Suppose there is a vertex $w \in U \setminus U'$ with $L_w = \{c\}$, then we put w in U' and set $\Gamma'(w) = c$. If there is a vertex v where $L_v = \emptyset$, then we say* NO.

This follows from the safeness of reduction rule 5.

Given an instance $((U = U_1 \uplus U_2 \cdots U_p, \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ of Partitioned p-UMC, we apply the above reduction rules exhaustively. If in the process we infer that the given instance is a NO instance then we return the same. It could also happen that we get $\mathcal{F} = \emptyset$. In this case, for every vertex $v \in U \setminus U'$, Γ assigns to v an element of $L(v)$ arbitrarily. Thus, from now onwards we assume that we neither conclude that the given instance is a NO instance nor obtain $\mathcal{F} = \emptyset$. We call an instance of Partitioned p-UMC reduced if the above reduction rules are not applicable. For simplicity, let $((U = U_1 \uplus U_2 \cdots U_p, \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ denote the reduced instance of Partitioned p-UMC. Observe that the reduced instance has the following properties:

1. For every vertex v, $|L_v| \geq 2$. This is because reduction rule 11 is no longer applicable.
2. For every hyperedge h, $|\Gamma'^{-1}(\Psi_{\mathsf{family}}(h)) \cap h| = 0$. This is because reduction rules 7, 8, and 10 are no longer applicable.
3. For every hyperedge h, and each vertex $v \in h$, $L(v)$ contains colors that are of order at most $\Psi_{\mathsf{family}}(h)$. This is because reduction rule 6 is no longer applicable.

We define the set $V_i \subseteq U \setminus U'$ as the set of vertices that have i in their list of admissible colors. Then, because reduction rule 9 is no longer applicable, there are two kinds of vertices in V_i: It could be that the vertex v has $i \in L_v$ and $\exists h \in \mathcal{F},\ v \in U_j \cap h$ such that $\Psi_{\mathsf{family}}(h) = i, \Psi_{\mathsf{parts}}(j) = i$. Or, the vertex v has $i \in L_v$ but for any h with $\Psi_{\mathsf{family}}(h) = i$, $v \notin h$, and for any h that contains v, $\Psi_{\mathsf{family}}(h) > i$.

To solve the reduced instance of Partitioned p-UMC, we will again solve r instances of Partitioned UHS. We define some sets based on $V_i \subseteq U$:

1. For every $j \in [r]$ and $x \in \Psi^{-1}_{\mathsf{parts}}(j)$ let $Z^x_j = U_x \cap V_j$ and $Z_j = \bigcup_{x \in \Psi^{-1}_{\mathsf{parts}}(j)} Z^x_j$.
2. For every $j \in [r]$ and $h \in \Psi^{-1}_{\mathsf{family}}(j)$ let $h_j = h \cap V_j$ and $\mathcal{F}_j = \{h_j \mid h \in \Psi^{-1}_{\mathsf{family}}(j)\}$.

Next we relate the instance of PARTITIONED P-CFC to PARTITIONED UHS.

LEMMA 24. *Let* $((U = U_1 \uplus U_2 \cdots U_p, \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U', \Gamma', \{L_v \subseteq [r] | v \in U \setminus U'\})$ *denote the reduced instance of* PARTITIONED P-UMC. *Then it is a* YES *instance of* PARTITIONED P-UMC *if and only if for all* $j \in [r]$, $(\uplus_{x \in \Psi^{-1}_{\mathsf{parts}}(j)} Z^x_j, \mathcal{F}_j)$ *is a* YES *instance of* PARTITIONED UHS.

This proof is very similar to the proof of Lemma 16. Lemmas 24 and 17 together result in the following algorithm for PARTITIONED P-CFC.

LEMMA 25. PARTITIONED P-UMC *can be solved in time* $2^{p+|\mathcal{F}|} \cdot N^{\mathcal{O}(1)}$.

Next, using Lemma 25 and the method of the color coding technique of [2] we give an algorithm for P-UMC.

THEOREM 26. P-UMC *can be solved in time* $2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}$.

Proof. Let $((U, \mathcal{F}), k, r)$ be an instance of P-UMC. Recall that $|U| = n$, $|\mathcal{F}| = m$, and $N = n + m$. Given an instance we first apply Theorem 21 and obtain an equivalent instance with at most 4^k vertices and $\mathcal{O}(k \log k)$ hyperedges. We run through all $p \leq k$. Since the number of hyperedges in the input instance is $\alpha_{uc} k \log k$, the number of subfamilies of size k is $\binom{\alpha_{uc} k \log k}{k} \leq (\frac{e \alpha_{uc} k \log k}{k})^k \leq (e \alpha_{uc} \log k)^k$. We guess a subfamily $\mathcal{F}'$ of hyperedges that will be unique-maximum colored. That is, we are trying to find a coloring $\Gamma : U \to [r]$ such that each hyperedge h in $\mathcal{F}'$ is unique-maximum colored. Let S be the hypothetical unique-maximum-solution corresponding to it. In other words, for each hyperedge h in $\mathcal{F}'$, the unique-maximum colored vertex of h (with respect to Γ) belongs to S. We guess the size of $|S|$, say $p \leq k$. For a fixed p, let $\mathfrak{F}$ be the family of (n, p)-perfect hash family of size $e^p p^{\mathcal{O}(\log p)} \log n$. By the property of $\mathfrak{F}$, we know that there exists a function $\zeta \in \mathfrak{F}$ that maps S to $[p]$ injectively. Let $U_1, \ldots, U_p$ denote the partition of U given by ζ. Observe that after this we will be seeking for the unique-maximum-solution S with the property that $|S \cap U_i| = 1$ for all $i \in [p]$.

Next for each hyperedge h in $\mathcal{F}'$, we guess the color of the vertex in h that is unique-maximum colored by Γ. There are r^k such guesses. Thus, after this guess, we define a function $\Psi_{\mathsf{family}} : \mathcal{F}' \to [r]$ such that h is assigned the color of the vertex in h that will be unique-maximum colored by Γ. Finally, for the potential solution set S we guess the color of each vertex given by Γ. Since we are looking for a unique-maximum-solution set S, such that $\forall i \in [p]$, $|U_i \cap S| = 1$ it is equivalent to say that we guess an r partitioning of the p parts in $U = (U_1, \ldots, U_p)$. That is, the vertex of $S \in U_i$ will be assigned a color by Γ. To express this guess, we define another function $\Psi_{\mathsf{parts}} : [p] \to [r]$ such that $\Psi_{\mathsf{parts}}(j) = i$ if the vertex x in $S \cap U_j$ will have $\Gamma(x) = i$. Thus, there are r^p guesses for the coloring of the potential solution set S by Γ. At the end of this sequence of guesses, we have fixed a choice of hyperedges that are to be r unique-maximum colored, a coloring of the potential solution set S (without actually knowing the vertices of S, this essentially means a partitioning of the parts of U) and a partitioning of the hyperedges according to which color of Γ will determine that the hyperedge is unique-maximum colored. This results in the following instance of PARTITIONED P-UMC: $((U = U_1 \uplus U_2 \cdots U_p, \mathcal{F}'), \Psi_{\mathsf{family}}, \Psi_{\mathsf{parts}}, U' = \emptyset, (\forall v \in U : L_v = [r]))$. By Lemma 25 we know that we can solve this in time $2^{p+k} \cdot N^{\mathcal{O}(1)} \leq 4^k \cdot N^{\mathcal{O}(1)}$. Thus the overall running time for P-UMC is upper bounded by the number of guesses and the running time of an algorithm for PARTITIONED P-UMC. Thus, the running time of the algorithm is upper bounded by

$$\binom{\alpha_{uc} k \log k}{k} \times k \times |\mathfrak{F}| \times r^k \times r^k \times 4^k \cdot N^{\mathcal{O}(1)} = 2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}.$$

This concludes the proof. ∎

5. Exact algorithm for Max-CFC.

In this section, we give an exact algorithm for solving Max-CFC for hypergraphs. We give a recurrence on subproblems, using which we can give a dynamic programming algorithm to solve the problem. However, a much faster algorithm can be designed using the technique of subset convolutions on functions.

Theorem 27. Max-CFC *for hypergraphs can be solved by an exact algorithm that runs in* $\mathcal{O}(2^{(m+n)})$ *time.*

Proof. Let $H = (U, \mathcal{F})$ be the input hypergraph. Suppose, for a given hypergraph, there is a procedure to decide whether there exists an r-coloring that is conflict free. Then, we can generate all subsets $\mathcal{F}'$ of $\mathcal{F}$, such that there exists an r-coloring of vertices of $(U(\mathcal{F}'), \mathcal{F}')$ that is conflict free, by running this procedure for all subsets. Then solving the Max-CFC problem reduces to picking the maximum sized subsets among those.

We now give a procedure to find the minimum number of colors required to conflict-free color a given hypergraph, $(U', \mathcal{F}')$, where $U' \subseteq U$ and $\mathcal{F}' \subseteq \mathcal{F}$. Let χ' be an r-coloring on U' and let $\mathcal{F}'$ be conflict-free colored by χ'. Then χ' partitions U' into r sets, $(U_1 \uplus U_2 \uplus \cdots \uplus U_r)$, such that the following property is true:

$$\forall F \in \mathcal{F}', \exists i \in [r] \text{ such that } |F \cap U_i| = 1.$$

Let $\mathcal{F}_1$ be the set of hyperedges such that $\forall F \in \mathcal{F}_1, |F \cap U_1| = 1$. In other words, all the hyperedges in $\mathcal{F}_1$ have a unique vertex colored by color 1. Then, if we correctly guessed U_1, solving whether $\mathcal{F}'$ has an r cfc in U is equivalent to solving the subproblem of whether $\mathcal{F}' \setminus \mathcal{F}_1$ has an $r-1$ cfc in $U \setminus U_1$.

We aim at calculating a function $\mathcal{C} : 2^U \times 2^{\mathcal{F}} \to \mathbb{Z}$. For a given $X \subseteq U, \mathcal{E} \subseteq \mathcal{F}$, we want $\mathcal{C}(X, \mathcal{E})$ to be the minimum number of colors needed to conflict-free color the hypergraph $(X, \mathcal{E})$. We give the following recurrence relation to find $\mathcal{C}(X, \mathcal{E})$:

$$\mathcal{C}(X, \mathcal{E}) = \begin{cases} min_{X' \subseteq X : \exists h \in \mathcal{E}, |h \cap X'| = 1} \{1 + \mathcal{C}(X \setminus X', \mathcal{E} \setminus \mathcal{E}')\} & \text{if } X \neq \phi, \\ 0 & \text{if } X = \phi, \end{cases} \tag{1}$$

where $\mathcal{E}' = \{h \in \mathcal{E} | |h \cap X'| = 1\}$.

We prove the correctness of the recurrence by induction on the size of X. When $|X| = 0$, the recurrence correctly returns 0.

Now assume $|X| > 0$. Assume $X' \subseteq X$ is a color class of a cfc that uses $\chi_{cf}((X, \mathcal{E}|_X))$ colors. Then X' uniquely colors all hyperedges that contain exactly one element from X'. $\mathcal{E}'$ represents the family of these hyperedges. The remaining hyperedges $\mathcal{E} \setminus \mathcal{E}'$ need to be uniquely covered by color classes in $X \setminus X'$. By the induction hypothesis, $\mathcal{C}(X \setminus X', \mathcal{E} \setminus \mathcal{E}') = \chi_{cf}((X \setminus X', \mathcal{E} \setminus \mathcal{E}'|_{X \setminus X'}))$. Hence, $\mathcal{C}(X \setminus X', \mathcal{E} \setminus \mathcal{E}') + 1$ returns the the value of $\chi_{cf}((X, \mathcal{E}|_X))$. Since the recurrence considers all possible subsets of X, one of them is the correct guess for X' and returns the minimum value.

Let us analyze the running time for computing the function $\mathcal{C}$. Let the input pairs $(X, \mathcal{E})$ be ordered in the following lexicographic manner. The input pairs $(X_1, \mathcal{E}_1) \leq$

$(X_2, \mathcal{E}_2)$ if either $|X_1| < |X_2|$ or if $|X_1| = |X_2|$ and $|\mathcal{E}_1| \leq |\mathcal{E}_2|$. We use a bottom-up algorithm for filling in the table entries for each input pair $(X, \mathcal{E})$, where the input pairs are ordered as above. For a subset X of size i, once we fix a subset $X' \subseteq X$ we also fix the subfamily $\mathcal{E}'$ and, consequently, the subfamily $\mathcal{E} \setminus \mathcal{E}'$. Therefore, once we fix $\mathcal{E}$ and X, we need to look up at most 2^i entries from the table to calculate $\mathcal{C}(X, \mathcal{E})$. By the ordering on the input pairs, and the definition of the function $\mathcal{C}$, the entries that have to be looked up have already been calculated before the need to calculate $\mathcal{C}(X, \mathcal{E})$. Thus, the time required to calculate $\mathcal{C}(X, \mathcal{E})$, where $|X| = i$ is $\mathcal{O}(2^i)$. To compute the function $\mathcal{C}$, for each fixed subset X, we need to calculate $\mathcal{C}(X, \mathcal{E})$ for all subfamilies $\mathcal{E} \subseteq \mathcal{F}$, which are at most 2^m subproblems that need to be calculated with respect to the fixed set X. Last, there are at most $\binom{n}{i}$ subsets X of size exactly i. Thus, we can compute the function $\mathcal{C}$ in time $\mathcal{O}(2^m \cdot \Sigma_{0 \leq i \leq n} \binom{n}{i} 2^i) = \mathcal{O}(3^n 2^m)$. Once this is done, finding a largest subfamily $\mathcal{F}' \subseteq \mathcal{F}$, such that $(U(\mathcal{F}'), \mathcal{F}')$ can be r conflict-free colored, can be done in $\mathcal{O}(2^n \cdot 2^m)$ time. Hence, we could solve the Max-CFC problem in $\mathcal{O}(3^n \cdot 2^m)$ time. However, we can improve the running time by quite a bit.

Let us relax the definition of $\mathcal{C}$ to be the function which takes a pair $(U', \mathcal{F}')$, where $U' \subseteq U, \mathcal{F}' \subseteq \mathcal{F}$, and, when $\chi_{cf}((U', \mathcal{F}'|_{U'})) \leq r$, correctly maps it to $\chi_{cf}((U', \mathcal{F}'|_{U'}))$. Otherwise, it could map $(U', \mathcal{F}')$ to some value between $n+1$ and $r(n+1)$, thereby clearly indicating that $(U', \mathcal{F}'|_{U'})$ is not r conflict-free colorable. Then, too, we can identify subfamilies $\mathcal{F}'$ such that $\mathcal{C}((U(\mathcal{F}'), \mathcal{F}')) \leq r$ and pick one subfamily which has that largest size. From now on, by $\mathcal{C}$, we will refer to this new definition of $\mathcal{C}$.

To facilitate the calculation of this newly defined $\mathcal{C}$, we define another function f which takes as input a pair $(X \subseteq U, \mathcal{E} \subseteq \mathcal{F})$. When $X = \emptyset$, for a subfamily $\mathcal{E}$ we define the function $f(X, \mathcal{E}) = 0$ if $\mathcal{E} = \emptyset$ and $f(X, \mathcal{E}) = n+1$ otherwise. For each nonempty $X \subseteq U$ and $\mathcal{E} \subseteq \mathcal{F}$, $f(X, \mathcal{E}) = 1$ if for each $h \in \mathcal{E}, |h \cap X| = 1$. Otherwise, $f(X, \mathcal{E}) = n+1$. Notice that it takes $\mathcal{O}(2^{(n+m)})$ time to calculate the function f. Using this function f, we are ready to define the function $\mathcal{C}(X, \mathcal{E})$ as follows:

$$\mathcal{C}(X, \mathcal{E}) = \min_{(X_1 \uplus X_2 \uplus \ldots \uplus X_r) = X; (\mathcal{E}_1 \uplus \ldots \uplus \mathcal{E}_r) = \mathcal{E}} f(X_1, \mathcal{E}_1) + \cdots + f(X_r, \mathcal{E}_r) \tag{2}$$

Finally, we identify a subfamily $\mathcal{F}'$ of $\mathcal{F}$, of largest size, such that $\mathcal{C}(U(\mathcal{F}'), \mathcal{F}')$ is r conflict-free colorable.

Correctness. The correctness for the procedure described above is similar to the previous arguments. First, we show that, when X is nonempty, the function $\mathcal{C}(X, \mathcal{E})$ determines whether $(X, \mathcal{E}|_X)$ can be r conflict-free colored or not. Also, when the tuple is r conflict-free colorable, the function returns the minimum number of colors required. We prove the correctness by case analysis of X. When $|X| = 1$, then for any subfamily $\mathcal{F}'$, $f(X, \mathcal{F}') = 1$ if and only if every hyperedge in $\mathcal{F}'$ contains the vertex of X. Thus, the hypothesis is true for the base case.

Now assume $|X| > 1$. First, suppose $\chi_{cf}((X, \mathcal{E}|_X)) \leq r$. Assume $(X_1 \uplus X_2 \uplus \cdots \uplus X_r)$ is a cfc that realizes $\chi_{cf}((X, \mathcal{E}|_X)) \leq r$. Some of the X_i's could be empty sets if $\chi_{cf}((X, \mathcal{E}|_X)) < r$. Each nonempty X_i uniquely colors all hyperedges that contain exactly one element from X_i. Without loss of generality, we may assume that X_1 is nonempty. Let $\mathcal{E}_1 \subseteq \mathcal{E}$ represent the subfamily of hyperedges that contains exactly one element from X_1. For $i > 1$, if X_i is empty, then we set $\mathcal{E}_i = \emptyset$. When X_i is nonempty, let $\mathcal{E}_i \subseteq \mathcal{E} - \bigcup_{1 \leq j \leq i-1} \mathcal{E}_j$ be the subfamily of hyperedges that contain exactly one element from X_i. Notice that for all $1 \leq i \leq r$, when X_i is nonempty, $f(X_i, \mathcal{E}_i) = 1$. When $X_i = \emptyset$, from the definition of f, $f(X_i, \mathcal{E}_i) = 0$. Hence,

$\mathcal{C}(X,\mathcal{E}) = \min_{(X_1 \uplus X_2 \uplus \ldots \uplus X_r)=X;(\mathcal{E}_1 \uplus \ldots \uplus \mathcal{E}_r)=\mathcal{E}} f(X_1,\mathcal{E}_1) + \cdots + f(X_r,\mathcal{E}_r)$ returns the minimum number of colors required to conflict-free color $(X, \mathcal{E}|_X)$.

On the other hand, if $\mathcal{C} > r$, then for any r-partition $(X_1 \uplus X_2 \uplus \cdots \uplus X_r)$ of X, and r-partition $(\mathcal{E}_1 \uplus \cdots \uplus \mathcal{E}_r)$ of $\mathcal{E}$, there will be at least one tuple $(X_i, \mathcal{E}_i)$ such that $f(X_i, \mathcal{E}_i) = n+1$. Therefore, we correctly calculate that $\mathcal{C}(X,\mathcal{E}) > r$. Notice that $(n+1) \leq \mathcal{C}(X,\mathcal{E}) \leq r(n+1)$.

A subfamily $\mathcal{F}'$ which is r conflict-free colorable and of largest size should have $\mathcal{C}(U(\mathcal{F}'),\mathcal{F}') \leq r$. Thus, we go through all such tuples corresponding to subfamilies, and determine a largest subfamily that is r conflict-free colorable.

Running time. The first step of the algorithm is to compute the function $\mathcal{C} : 2^U \times 2^{\mathcal{F}} \to \mathbb{Z} \cup \{\infty\}$ for all subsets of U and all subfamilies of $\mathcal{F}$. As stated earlier, it takes $\mathcal{O}(2^{(n+m)})$ time to calculate the function $f : 2^U \times 2^{\mathcal{F}} \to \mathbb{Z} \cup \{\infty\}$. The definition of $\mathcal{C}$ involves taking the minimum over the sum of r values from the range of f. The range of f is bounded between 0 and $n+1$, which makes the range of $\mathcal{C}$ bounded between 0 and $r(n+1)$. Using Proposition 4, computing the function $\mathcal{C}$ is equivalent to computing a sequence of $\log r$ subset convolutions over the integer min-sum semiring in the following way: We first calculate the subset convolution $f * f$ and obtain a function $g_1 : 2^U \times 2^{\mathcal{F}} \to \mathbb{Z} \cup \{\infty\}$, then we compute the subset convolution $g_1 * g_1$ to obtain a function $g_2 : 2^U \times 2^{\mathcal{F}} \to \mathbb{Z} \cup \{\infty\}$ and so on for $\log r$ steps. Hence, the algorithm for computing the function $\mathcal{C}$ runs in $\mathcal{O}(2^{n+m})$ time.

Finally, the algorithm runs through all subfamilies $\mathcal{F}'$ and finds out the largest sized subfamily such that $(U(\mathcal{F}'),\mathcal{F}')$ is r conflict-free colored. Thus the total running time of the algorithm is $\mathcal{O}(2^{n+m})$. ∎

It is to be noted that by setting $r = n$, $\mathcal{C}(U,\mathcal{F})$ returns the minimum number of colors required to conflict-free color the given hypergraph.

COROLLARY 28. *Given a hypergraph H, $\chi_{cf}(H)$ can be found in $\mathcal{O}(2^n 2^m)$ time.*

COROLLARY 29. *The* MAX-CFC *problem on hypergraphs induced by neighborhoods of graphs can be solved in $\mathcal{O}(4^n)$ time.*

6. Exact algorithm for unique maximum coloring.

We now give an exact algorithm for solving MAX-UMC on hypergraphs. It can be seen that the dynamic algorithm that we gave in Section 5, with minor changes, can be used to solve this problem.

LEMMA 30. *There exists an exact algorithm to solve* MAX-UMC *with running time $\mathcal{O}(3^n \cdot 2^m)$.*

Proof. Given a hypergraph $(U',\mathcal{F}')$, let χ' be an r-coloring on U' and let $\mathcal{F}'$ be unique-maximum colored by χ'. Then χ' partitions U' into r sets, $(U_1 \uplus U_2 \uplus \cdots \uplus U_r)$, such that the following property is true:

$$\forall F \in \mathcal{F}, \exists i \in [r] \text{ such that } |F \cap U_i| = 1 \text{ and } \forall j > i, F \cap U_j = \emptyset.$$

This is similar to partitions given by a cfc except for the last part. Let $\mathcal{U}$ be the function that takes a tuple $(X \subseteq U, \mathcal{E} \subseteq \mathcal{F})$ and maps it to the minimum number of colors required for unique-maximum coloring $(X,\mathcal{E}|_X)$. We give a recurrence very similar to that in the previous section:

(3)
$$\mathcal{U}(X,\mathcal{E}) = \begin{cases} \min_{X' \subseteq X, \forall E \in \mathcal{E}, |E \cap X'|=0 \vee |E \cap X'|=1} \{1 + \mathcal{U}(X \setminus X', \mathcal{E} \setminus \mathcal{E}')\} & \text{if } X \neq \phi, \\ 0 & \text{if } X = \phi, \end{cases}$$

where $\mathcal{E}' = \{E \in \mathcal{E} | |E \cap X'| = 0 \vee |E \cap X'| = 1\}$.

The correctness of this recurrence can be seen in a similar way. Assume X' is the maximum color class in $\mathcal{U}$. Then X' uniquely colors all hyperedges that contain exactly one element from X'. The remaining hyperedges can be optimally colored by $\mathcal{U}(X \setminus X', \mathcal{E} \setminus \mathcal{E}')$. Since we are considering all subsets of X, we get an optimal solution. $\square$

We would like to note that the fast subset convolution technique will also give rise to an algorithm for Max-UMC. However, the running time of this algorithm is not better than the above dynamic programming algorithm.

7. Conclusion. We studied the Max-CFC and the Max-UMC problems and gave exact algorithms for the two problems. We also looked at p-CFC and p-UMC, and gave an FPT algorithm that runs in time $2^{\mathcal{O}(k \log\log k + k \log r)} \cdot N^{\mathcal{O}(1)}$. Here, k is the number of hyperedges that are r conflict-free colored, and N is the size of the input instance. It would be interesting to show lower bounds for FPT algorithms for p-CFC and p-UMC. We also obtain an exponential vertex kernel for the problem, and it is open whether a polynomial kernel for the problem exists or not.

REFERENCES

[1] D. Ajwani, K. Elbassioni, S. Govindarajan, and S. Ray, *Conflict-free coloring for rectangle ranges using $o(n^{.382})$ colors*, Discrete Comput. Geom., 48 (2012), pp. 39–52.

[2] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. ACM, 42 (1995), pp. 844–856, http://doi.org/10.1145/210332.210337.

[3] P. Cheilaris, *Conflict-Free Coloring*, Ph.D. thesis, CUNY, New York, 2009.

[4] P. Cheilaris and G. Tóth, *Graph unique-maximum and conflict-free colorings*, J. Discrete Algorithms, 9 (2011), pp. 241–251.

[5] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky, *Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002, IEEE Computer Society, Los Alamitos, CA, 2002, pp. 691–700.

[6] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts Theoret. Comput. Sci. EATCS Ser., Springer, Berlin, 2006.

[7] F. V. Fomin and D. Kratsch, *Exact Exponential Algorithms*, Springer, Heidelberg, 2010.

[8] L. Gargano and A. A. Rescigno, *Complexity of conflict-free colourings of graphs*, Theoret. Comput. Sci., 566 (2015), pp. 39–49.

[9] R. Glebov, T. Szabó, and G. Tardos, *Conflict-free colouring of graphs*, Combin. Probab. Comput., 23 (2014), pp. 434–448.

[10] S. Har-Peled and S. Smorodinsky, *Conflict-free coloring of points and simple regions in the plane*, Discrete Comput. Geom., 34 (2005), pp. 47–70.

[11] S. Jukna, *Extremal Combinatorics: With Applications in Computer Science*, Springer, Heidelberg, 2011.

[12] N. Misra, H. Moser, V. Raman, S. Saurabh, and S. Sikdar, *The parameterized complexity of unique coverage and its variants*, Algorithmica, 65 (2013), pp. 517–544.

[13] M. Naor, L. J. Schulman, and A. Srinivasan, *Splitters and near-optimal derandomization*, in 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 1995, IEEE Computer Society, New York, 1995, pp. 182–191.

[14] J. Pach and G. Tardos, *Conflict-free colourings of graphs and hypergraphs*, Combin. Probab. Comput., 18 (2009), pp. 819–834.

[15] S. Smorodinsky, *Combinatorial Problems in Computational Geometry*, Ph.D. thesis, Tel Aviv University, Tel Aviv, 2003.

[16] S. Smorodinsky, *Conflict-Free Coloring and its Applications*, Springer, Berlin, 2013.