Mixed Hypertrees

Daniel Král'[*] Jan Kratochvíl[†] Andrzej Proskurowski[‡]
Heinz-Jürgen Voss[§]

Abstract

A mixed hypergraph is a triple $(V, \mathcal{C}, \mathcal{D})$ where V is its vertex set and $\mathcal{C}$ and $\mathcal{D}$ are families of subsets of V, $\mathcal{C}$–edges and $\mathcal{D}$–edges. We demand in a proper coloring that each $\mathcal{C}$–edge contains two vertices with the same color and each $\mathcal{D}$–edge contains two vertices with different colors. The feasible set of a mixed hypergraph is the set of all k's for which there exists a proper coloring using exactly k colors. A hypergraph is a hypertree if there exists a tree such that the edges of the hypergraph induce connected subgraphs of that tree.

We prove that feasible sets of mixed hypertrees are gap–free, i.e., intervals of integers, and we show that this is not true for precolored mixed hypertrees. The problem to decide whether a mixed hypertree can be colored by k colors is NP–complete in general; we investigate complexity of various restrictions of this problem and we characterize their complexity in most of the cases.

1 Introduction

Different types of coloring have been studied in the past because they are important from both the theoretical and practical points of view. A common generalization of several previously studied coloring concepts is the concept of

[*]Department of Applied Mathematics and Institute for Theoretical Computer Science (supported by Ministry of Education of Czech Republic as project LN00A056), Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic. E-mail: kral@kam.ms.mff.cuni.cz. The author acknowledges partial support by KONTAKT 338/99.

[†]Department of Applied Mathematics and Institute for Theoretical Computer Science (supported by Ministry of Education of Czech Republic as project LN00A056), Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic. E-mail: honza@kam.ms.mff.cuni.cz. The author acknowledges partial support by KONTAKT 338/99.

[‡]Department of Computer and Information Science University of Oregon, Eugene, US. E-mail: andrzej@cs.uoregon.edu.

[§]Institute of Algebra, Technische Universität Dresden, Germany. E-mail: voss@math.tu-dresden.de.

mixed hypergraphs introduced in [22]. Besides the ordinary colorings, this concept generalizes special types of colorings of different combinatorial structures [2, 3, 6, 17, 18, 19, 20], and also the concept of list coloring as described in [15].

A *mixed hypergraph* H is a triple $(V, \mathcal{C}, \mathcal{D})$ where V is the set of vertices of H, $\mathcal{C}$ is a set of subsets of V (called $\mathcal{C}$*–edges*) and $\mathcal{D}$ is a set of subsets of V (called $\mathcal{D}$*–edges*). A mixed hypergraph is called a *bihypergraph* if $\mathcal{C} = \mathcal{D}$. A mixed hypergraph is called a *co–hypergraph* if $\mathcal{D} = \emptyset$ (see [4]). A coloring c of the vertices of H is *proper* if each $\mathcal{C}$–edge of H contains at least two vertices of a $\mathcal{C}$ommon color and each $\mathcal{D}$–edge of H contains at least two vertices of $\mathcal{D}$ifferent colors. A proper coloring c using exactly k colors is a *strict* k–coloring. The *feasible set* $\mathcal{F}(H)$ of a mixed hypergraph H is the set of all k's for which there exists a strict k–coloring of H. The minimum number $\chi(H)$ of $\mathcal{F}(H)$ is called the *(lower) chromatic number* of H and the maximum number $\overline{\chi}(H)$ of $\mathcal{F}(H)$ is called the *upper chromatic number* of H. It was proved in [9] that for any set of integers A such that $1 \notin A$, there exists a mixed hypergraph whose feasible set is equal to A. This result was further extended in [12]: Besides the feasible set, one may also prescribe the numbers r_k of all different strict k–colorings of a mixed hypergraph for $k \in A$. Since mixed hypergraphs are quite a general model, it is natural to restrict attention to special classes of them in order to hope for some interesting algorithmic results.

Several classes of mixed hypergraphs have been previously studied: The class of mixed hypergraphs with bounded maximum vertex degree was investigated in [14], the class of interval mixed hypergraphs in [1, 9] and the class of planar mixed hypergraphs in [5, 10, 16]. We study the class of mixed hypertrees in this paper. A hypergraph H is a *hypertree* if there is a tree with the same vertex set such that the edges of H induce connected subtrees of it. A mixed hypergraph $(V, \mathcal{C}, \mathcal{D})$ is a *mixed hypertree* if $(V, \mathcal{C} \cup \mathcal{D})$ is a hypertree. Mixed hypertrees have been suggested in [21] as a model which can find its application in computer biology. It is possible to find a tree representation of a given (mixed) hypertree in polynomial time by an easy modification of algorithms for finding a representation of chordal graphs as intersection graphs of subtrees in a tree. A mixed hypergraph/hypertree is *reduced* if all its $\mathcal{C}$–edges have size at least three and all its $\mathcal{D}$–edges have size at least two (see [9, 13, 14]). There exists (and can be constructed in polynomial time) for each mixed hypertree a reduced one such that their proper colorings one–to–one correspond [13]. We often work with *rooted* underlying trees of (mixed) hypertrees; such a tree has a special vertex called *root* and vertices have been assigned levels according to their distance from the root. The *parent* of a vertex v (which is not the root) is a neighbor of v closer to the root. If u is a parent of v, then v is a child u. If two vertices have the same parent, they are *siblings*.

We define two following algorithmic problems for mixed hypertrees:

`STRICT COLORING`

`Input:` A mixed hypertree H and an integer k.
`Question:` Is H strict k-colorable?

And for every fixed k:

`STRICT k-COLORING`

`Input:` A mixed hypertree H.

`Question:` Is H strict k-colorable?

The problem `STRICT COLORING` of mixed hypertrees is NP–complete. Therefore we study these two problems with different constraints on input mixed hypertrees: *A vertex/edge load* of a mixed hypertree H is the maximum number of edges of H in which a single vertex of H/a single edge of a tree representation is contained in. Note that the edge load of H is equal to the maximum number of edges of H which contain a pair of the vertices of H. Thus both these parameters do not depend on a chosen tree representation and the algorithms designed for mixed hypertrees with bounded vertex/edge load are robust. Another parameter is the maximum degree of a tree representation; this parameter depends on a chosen representation and thus we always assume (when we design an algorithm for mixed hypertrees whose tree representation has small maximum degree) that we are given a representation together with the input mixed hypertree.

In this paper, we address both combinatorial properties of mixed hypertrees and algorithms for their coloring. Unlike the case of general mixed hypergraphs, feasible sets of mixed hypertrees are always intervals as stated in Corollary 1 and in Corollary 2. Moreover the lower chromatic number of a colorable mixed hypertree is at most two due to Observation 1. In Section 2 a more general problem is investigated, namely the extensions of a given precoloring of some vertices to proper colorings of the whole mixed hypertree. We prove in Theorem 1 that if a mixed hypertree with vertices precolored by k different colors admits an extension to a strict k'–coloring for $k' \geq k+2$ then it admits an extension to a strict κ–coloring for every κ, $k+2 \leq \kappa \leq k'$. This cannot be improved as shown in Theorem 2.

We introduce and state several basic properties of the concept of witness sequences in Section 3; Section 3 is based on the conference paper [11]. We use this concept to design one of our polynomial–time algorithms of Section 4. But this concept is also interesting from the theoretical point of view: It allows us to state in Observation 2 that each proper coloring of a mixed hypertree can be modified to one which does not have vertices of the same color separated by vertices of two different colors, i.e., the same color can be forced to two vertices only if they are close enough.

We deal with algorithmic issues in the rest of the paper. Since the lower chromatic number of a colorable mixed hypertree is always either 1 or 2 (and it can be easily recognized which is the case), the only problem interesting from the algorithmic point of view is the problem to decide whether a given mixed hypertree can be properly colored by at least k colors. This problem is NP–complete in general. We present three polynomial time algorithms in Section 4: An algorithm for `STRICT` k`--COLORING` for mixed hypertrees with underlying trees of bounded maximum degrees in Proposition 1, an algorithm for `STRICT`

`COLORING` for mixed hypertrees with bounded vertex load in Proposition 2 and an algorithm for `STRICT` k`--COLORING` for mixed hypertrees containing only $\mathcal{C}$–edges in Proposition 3. On the other hand, we prove NP–completeness of the problem in the following three cases: If the edge–load is bounded in Proposition 4 and in Proposition 5 and if the degree of an underlying tree is bounded by three in Proposition 6 and in Proposition 7. If the degree of an underlying tree is bounded by two then the input mixed hypertree is an interval mixed hypergraph and its upper chromatic number can be determined in polynomial time due to the results of [1]. Our NP–completeness reductions work for both mixed hypertrees containing only $\mathcal{C}$–edges and mixed bihypertrees. We refer the reader to Table 1 for summary of our results. The above mentioned cases cover all possible combinations of the parameters except for the following two cases: The case of `STRICT` k`--COLORING` for general mixed hypertrees and the same problem for mixed hypertrees with bounded edge load. Even the latter (less general) case includes the problem of determining whether an upper chromatic number of a given co–hypergraph (mixed hypergraph containing only $\mathcal{C}$–edges, see [4]) is at least a fixed number k as stated in Proposition 8 in Section 5. We conjecture that this problem can be solved in polynomial time and a polynomial–time algorithm for any of the former two problems for mixed hypertrees would immediately yield an algorithm for this one.

2 Precolored Mixed Hypertrees

Observation 1 *Every reduced hypertree H is 2–colorable. If H contains at least one $\mathcal{D}$–edge its lower chromatic number is two. If it contains no $\mathcal{D}$–edges its lower chromatic number is one.*

Proof: Choose any vertex of T to be its root. We color the root of T arbitrarily and then we color the vertices from the root to the leaves — we color a vertex with the color different from the color of its parent. It is easy to check that this coloring is proper since any $\mathcal{D}$–edge contains an edge of the underlying tree and any $\mathcal{C}$–edge has size at least three.
□

Theorem 1 *Let H be a hypertree with an underlying tree T. let $S_1, S_2, \ldots, S_k$ ($k \geq 1$) be disjoint sets of vertices of H and let the vertices of S_i be precolored by the color b_i for $1 \leq i \leq k$. If H has a precoloring extension using $K \geq k+2$ colors then it has a precoloring extension which uses exactly $k+2$ colors.*

Proof: Let c be the precoloring extension of H and $S_1, \ldots, S_k$ such that c uses K colors. If c colors two end–vertices of an edge of T with the same color, then we may contract this edge (remove all the $\mathcal{C}$–edges containing it) — then if we find a precoloring extension of this new mixed hypertree using $k+2$ colors, then we also have a precoloring extension of the original one. Hence we may assume that c assigns end–vertices of each edge of T different colors (thus H is reduced) and we may also assume $\mathcal{D} = E(T)$.

We assume further that each S_i is non–empty (we may add to it a single vertex colored by c with the color b_i if necessary). For every $\mathcal{C}$–edge e, fix two different vertices $x_e, y_e \in e$ such that $c(x_e) = c(y_e)$ (every $\mathcal{C}$–edge contains at least one pair of such vertices). Replace every $\mathcal{C}$–edge e by the $\mathcal{C}$–edge e' consisting of all vertices of the x_e, y_e-path in T. Note that c is a precoloring extension with respect to $S_1, \ldots, S_k$ of the resulting mixed hypertree H'. Moreover every precoloring extension of H' with respect to $S_1, \ldots, S_k$ is a precoloring extension of H. Every $\mathcal{C}$–edge of H' induces a path in the underlying tree T, and therefore we may consider end-vertices of $\mathcal{C}$–edges.

Let us enlarge the sets S_i (if necessary) according to the following procedure: If there is a $\mathcal{C}$–edge e' which starts in a vertex $x_e \in S_i$ for some i, and ends in a vertex $y_e \notin S_i$, then add y_e into S_i. Repeat this step until no such $\mathcal{C}$–edge exists. Note that the new S_i's are still disjoint after this procedure is finished, because we have only added vertices of color b_i into the set S_i.

For the sake of brevity, let us write $S = \bigcup_{i=1}^{k} S_i$. Next, we define the set A of auxiliary two-element edges, which consists of the pairs $x_e y_e$, $e \in \mathcal{C}$ such that $e \setminus \{x_e, y_e\} \subseteq S$.

Define the auxiliary graph T' as the graph obtained from T by removing the vertices of S and then contracting all the auxiliary edges of A by collapsing their end-vertices. We claim that T' has at least two vertices and is acyclic. Indeed, since the coloring c uses at least two colors not used on the vertices of S, these two additional colors must remain in T' (note that the vertices of different colors cannot be contracted). Next, suppose T' has a cycle. This cycle corresponds to a cycle in $A \cup E(T \setminus S)$ which contains at least one edge, say t, of T'. Replacing each edge e in A by the corresponding $(x_e y_e)$-path in T, we get a closed walk in T which traverses the edge t precisely once, a contradiction with the acyclicity of T.

Since T' is acyclic, it can be colored by two colors, say by the colors b_{k+1} and b_{k+2}. And since H has at least two vertices, we can take such coloring which actually uses both colors. We claim that this coloring yields a proper precoloring extension of H' with respect to $S_1, \ldots, S_k$. It obviously uses exactly $k+2$ colors.

This coloring is proper on all $\mathcal{D}$–edges which are actually edges of the underlying tree T. We show that every $\mathcal{C}$–edge e' contains two vertices of the same color:

- The edge e' has at least one end-vertex in S: Then e' has both end-vertices in S by the enlargement procedure, and these end–vertices are in the same S_i, i.e., they are colored by the same color.
- The edge e' has both end–vertices in $T \setminus S$ and all other vertices in S:

Then $x_e y_e \in A$, x_e and y_e are contracted into one vertex of G, and so x_e and y_e get the same color.

- The edge e' has both the end–vertices and at least one inner vertex in $T \setminus S$: Then these 3 vertices are colored by colors b_{k+1} and b_{k+2} and two of them must get the same color.

$\square$

The immediate corollary of the preceding lemma is the following:

Corollary 1 *The feasible set of any mixed hypertree is gap–free.*

Proof: If there exists a mixed hypertree whose feasible set contains a gap, then there is such a reduced mixed hypertree H. Since its feasible set contains a gap, $\overline{\chi}(H) \geq 4$ due to Observation 1. Let t be a fixed integer, $2 < t < \overline{\chi}(H)$, let c be any coloring of H using $\overline{\chi}(H)$ colors and let $S_1, \ldots, S_{t-2}$ be any $t-2$ color classes of this coloring. If we apply Theorem 1 to H precolored with respect to $S_1, \ldots, S_{t-2}$, we get a proper coloring (a precoloring extension with respect to $S_1, \ldots, S_{t-2}$) using exactly t colors. Thus the feasible set of H does not contain a gap which contradicts the choice of H.
$\square$

We have seen that for a mixed hypertree H precolored with k colors, the feasible set equals either $\{k\} \cup [k+2, \overline{\chi}(H)]$ or $[k, \overline{\chi}(H)]$. One may ask if the latter is always the case, i.e., if the feasible sets of precolored mixed hypertrees are always gap-free. This is not true as proven in the next theorem:

Theorem 2 *For each $k \geq 2$, there exists a mixed hypertree precolored with k colors such that this precoloring can be extended to a strict k–coloring and a strict $(k+2)$–coloring, but it cannot be extended to a strict $(k+1)$–coloring.*

Proof: We first present a counterexample for $k = 2$. Consider the following precolored mixed hypertree H:

$$V(H) = \{a, b, c, d, \alpha, \beta, \gamma, \delta\}$$

$$\mathcal{C}(H) = \{a\alpha\beta, b\beta\alpha, c\gamma\delta, d\delta\gamma, \alpha\beta\gamma, \beta\gamma\delta\}$$

$$\mathcal{D}(H) = \{a\alpha, b\beta, c\gamma, d\delta, \beta\gamma\}$$

$$S_1 = \{a, c\}$$

$$S_2 = \{b, d\}$$

The tree with the edge–set $\{a\alpha, b\beta, c\gamma, d\delta, \alpha\beta, \beta\gamma, \gamma\delta\}$ shows that H is a hypertree. The precoloring of H can be extended to a strict 2–coloring by assigning the common color of a and c to β and δ and the color of b and d to α and γ.

On the other hand, the precoloring of H can be extended to a strict 4–coloring by assigning the same new color both to α and β and another new color both to γ and δ. We prove that these are the only possible precoloring extensions of H with respect to S_1 and S_2.
Claim: In any proper precoloring extension of H with respect to S_1 and S_2, the vertices α and β get either the same color different from the color of a or b, or α gets the color of b and β gets the color of a.

The proof of this claim can be easily done by checking all possible cases. The same claim is true also for the vertices γ and δ due to the symmetry of H and the precoloring. For a proper precoloring extension using 3 colors, there are only the following three possibilities:

- The vertices α and β are colored with a new color, γ with the color of a and δ with the color b.
- The vertices γ and δ are colored with a new color, α with the color of b and β with the color a.
- The vertices α, β, γ and δ are all colored with the same new color.

None of these extension is proper: In the first two cases one of the $\mathcal{C}$–edges $\alpha\beta\gamma$ and $\beta\gamma\delta$ is multicolored, while in the last case the vertices β and γ of a $\mathcal{D}$–edge get the same color.

The proof can be easily extended to the case that $k > 2$. It is enough to add $4(k-2)$ vertices a_i, b_i, c_i and d_i for $3 \leq i \leq k$ together with the $\mathcal{D}$–edges αa_i, βb_i, γc_i and δd_i for $3 \leq i \leq k$. We precolor the vertices a_i, b_i, γ_i and d_i with the same new color, i.e., $S_i = \{a_i, b_i, c_i, d_i\}$, for $3 \leq i \leq k$. The arguments used above can be used also in this case and thus the precolored mixed hypertree has only two precoloring extensions, namely one using k colors and one using $k+2$ colors.

□

3 Concept of Witness Sequences

We introduce the concept of witness sequences in this section.

Let $H = (V, \mathcal{C}, \mathcal{D})$ be a mixed hypertree with an underlying tree $T = (V, E)$ which we consider to be rooted at a fixed vertex; let level(v) be the distance of v from the root. We call any sequence of distinct vertices $w_1, \ldots, w_k$, $0 =$ level$(w_1) <$ level$(w_2) \leq$ level$(w_3) \leq \ldots \leq$ level(w_k) of vertices V a *witness sequence* (note that w_1 has to the root).

Once we have a witness sequence of length k, we want to find a strict k–coloring c of H (if it exists) such that the colors of the vertices $w_1, \ldots, w_k$ are mutually different. In order to do this, we divide the vertices of H into three sets: L is the set of the vertices introducing new colors (i.e., vertices in the witness sequence) and their siblings:

$$L = \{v | \exists i, \mathrm{parent}(v) = \mathrm{parent}(w_i) \vee v = w_i\} \tag{1}$$

R is the set of the neighbors of the vertices of L:

$$R = \{v | v \notin L \wedge \exists u \in L, uv \in E\} \tag{2}$$

And O is the set of other vertices of H:

$$O = V \setminus (L \cup R) \tag{3}$$

Later, we color the vertices of O as described in the two–colorability proof above.

Next, we construct a mapping $\lambda : V \to \{\mathrm{F}, \mathrm{G}, 1, \ldots, k\}$. We define a special set $\Lambda(v)$ for each vertex v as follows:

$$\Lambda(v) = \{\mathrm{F}, \mathrm{G}\} \cup \{i | \mathrm{parent}(v) = \mathrm{parent}(w_i) \vee v = w_i\} \text{ if } v \in L \tag{4}$$

$$\Lambda(v) = \{\mathrm{F}, \mathrm{G}\} \text{ if } v \in R \tag{5}$$

$$\Lambda(v) = \{\mathrm{F}\} \text{ if } v \in O \text{ and } \{v, \mathrm{parent}(v)\} \in \mathcal{C} \tag{6}$$

$$\Lambda(v) = \{\mathrm{G}\} \text{ if } v \in O \text{ and } \{v, \mathrm{parent}(v)\} \notin \mathcal{C} \tag{7}$$

We require that $\lambda(v) \in \Lambda(v)$ for each vertex v. The meaning of the function λ is the following: The function λ assigns the number i to the vertex/ices introducing the color i, to a vertex which should be colored with the same color as its parent the value F and to a vertex which should be colored with the color different from the color of its parent the value G. We describe a construction of a coloring $\tilde{c}$ which is proper for some choice of λ if there is a proper coloring of H assigning to $w_1, \ldots, w_k$ mutually different colors and for which this sequence is minimal in the sense defined later. The coloring $\tilde{c}$ of H is based on a witness sequence $w_1, \ldots, w_k$ and a mapping λ:

$$\tilde{c}(v) = 1 \text{ if } w_2 \text{ is an successor of } v \tag{8}$$

$$\tilde{c}(v) = \lambda(v) \text{ if } \lambda(v) \in \{1, \ldots, k\} \tag{9}$$

$$\tilde{c}(v) = \tilde{c}(\mathrm{parent}(v)) \text{ if } \lambda(v) = \mathrm{F} \tag{10}$$

$$\tilde{c}(v) = \tilde{c}^\star(\mathrm{parent}(v)) \text{ if } \lambda(v) = \mathrm{G} \tag{11}$$

where $\tilde{c}^\star(u)$ is following:

- $\tilde{c}^\star(u) = 2$ if u is the root (note that $\tilde{c}(u) = 1$ in this case)
- $\tilde{c}^\star(u) = \tilde{c}(\mathrm{parent}(u))$ if $\tilde{c}(u) \neq \tilde{c}(\mathrm{parent}(u))$
- $\tilde{c}^\star(u) = \tilde{c}^\star(\mathrm{parent}(u))$ if $\tilde{c}(u) = \tilde{c}(\mathrm{parent}(u))$

Note that all inner vertices v on the path from w_2 to w_1 get color $\tilde{c}(v) = 1$. The last rule of the definition $\tilde{c}$ says that $\tilde{c}(v) := \tilde{c}(w)$ for the first vertex w met when walking from parent(v) to w_1 such that $\tilde{c}(w) \neq \tilde{c}(\text{parent}(v))$, and $\tilde{c}(v) = 2$ if such w does not exist (i.e., if all vertices on the path from parent(v) to w_1 are colored with color 1).

The sequence $w_1, \ldots, w_k$ of vertices V is a *witness sequence with respect to a strict k'–coloring* $c : V \to \{1, \ldots, k'\}$ if $c(w_i) \neq c(w_j)$ for $i \neq j$ (it could be $k \leq k'$). A witness sequence is *minimal with respect to* c if $\sum_i \text{level}(w_i)$ is minimal. We say that the mapping λ is *consistent* with a strict k'–coloring c and a minimal witness sequence $w_1, \ldots, w_k$ $(k \leq k')$ with respect to c if it satisfies:

- $\lambda(v) = c(v)$ if $c(v) \in \Lambda(v)$
- $\lambda(v) = \text{F}$ if $c(v) = c(\text{parent}(v))$ and $\text{F} \in \Lambda(v)$
  (This holds in particular when $\{v, \text{parent}(v)\}$ is a C–edge.)
- $\lambda(v) = \text{G}$ otherwise

If parent(v) does not exist, the second condition does not apply. Note that if $w_1, \ldots, w_k$ is a minimal witness sequence with respect to c then a consistent mapping λ exists and is uniquely determined.

The following theorem actually gives the sense to all the previous definitions which introduced witness sequences:

Theorem 3 *If $w_1, \ldots, w_k$ is a minimal witness sequence with respect to a strict k'–coloring c, $2 \leq k \leq k'$, of a mixed hypertree H and if λ is consistent with c then $\tilde{c}$ based on $w_1, \ldots, w_k$ and λ is a strict k–coloring.*

Proof: The coloring $\tilde{c}$ clearly uses exactly k colors. It remains to prove that $\tilde{c}$ is proper. Thus it is enough to prove that each edge of $\mathcal{D}$ contains two vertices colored by $\tilde{c}$ with different colors and each edge of $\mathcal{C}$ contains two vertices colored by $\tilde{c}$ with the same colors.

Let e be any edge of $\mathcal{D}$ and let u and v be two vertices of e colored by c with different colors. We can assume w.l.o.g. that u and v are neighbors and that e.g. $u = \text{parent}(v)$. Note that w_2 is not an successor of v, since otherwise it would hold that $c(u) = c(v) = 1$ due to the minimality of the witness sequence and our assumption that $0 = \text{level}(w_1) < \text{level}(w_2) \leq \ldots \leq \text{level}(w_k)$. We claim that $\tilde{c}(u) \neq \tilde{c}(v)$. We distinguish several cases:

- $\lambda(v) = \text{F}$
  This case is impossible due to consistency of λ with c.
- $\lambda(v) = \text{G}$
  It holds that $\tilde{c}(v) = \tilde{c}^{\star}(\text{parent}(v)) \neq \tilde{c}(\text{parent}(v))$ — note that w_2 is not an successor of v since otherwise $c(u) = c(v) = 1$ due to the minimality of the witness sequence.

- $\lambda(v) \in \{1, \ldots, k\}$
  There is no predecessor of v colored by $\tilde{c}$ with the color $\lambda(v) = c(v)$ due to the minimality of the witness sequence; it especially holds that $\tilde{c}(u) \neq \tilde{c}(v)$, in this case.

We have just proved that all the $\mathcal{D}$–edges contain two vertices colored by different colors. So, we focus our attention on the $\mathcal{C}$–edges.

Let e be any edge of $\mathcal{C}$. If e contains two vertices whose successor is w_2, then these two vertices of e are colored by $\tilde{c}$ with color 1. Next, we assume that there is at most one vertex whose successor is w_2; clearly this vertex, if it exists, is the vertex of e which is the nearest one in e to the root. Let u and v be two nearest vertices of e colored by c with the same color. It is not necessary that $\tilde{c}(u) = \tilde{c}(v)$. We first state a useful observation which is going to be used several times in the proof:

Claim 1 *If $e \in \mathcal{C}$ contains a vertex u such that the following two conditions are satisfied:*

- $\lambda(u) \in \{\mathrm{F}, \mathrm{G}\}$
- $\{u, \mathrm{parent}(u), \mathrm{parent}(\mathrm{parent}(u))\} \subseteq e$ **or** $\{u, \mathrm{parent}(u)\} = e$

Then e contains two vertices colored by $\tilde{c}$ with the same color.

The case that $\lambda(u) = \mathrm{F}$ is trivial. If $\lambda(u) = \mathrm{G}$ then e cannot be $\{u, \mathrm{parent}(u)\}$ due to the definition of $\lambda(u)$ and it is $\{u, \mathrm{parent}(u), \mathrm{parent}(\mathrm{parent}(u))\} \subseteq e$. Let $u' = \mathrm{parent}(u)$ and $u'' = \mathrm{parent}(u') = \mathrm{parent}(\mathrm{parent}(u))$. If $\tilde{c}(u') = \tilde{c}(u'')$ the claim is clear. If $\tilde{c}(u') \neq \tilde{c}(u'')$ then $\tilde{c}(u) = \tilde{c}^{\star}(u') = \tilde{c}(u'')$ and the claim also holds.

We continue the proof of the theorem. We distinguish several cases to prove that e contains two vertices colored by $\tilde{c}$ with the same color:

- **The vertex v is a predecessor of u.** (The case that u is a predecessor of v is symmetric.)
  In this case e contains all the vertices between u and v, i.e. it contains $\mathrm{parent}(u)$ in particular. We distinguish several cases:
  - $\lambda(u) = \mathrm{F}$
    Then u and $\mathrm{parent}(u)$ are colored with the same color.
  - $\lambda(u) = \mathrm{G}$
    If v is not a parent of u, then e contains two vertices colored by $\tilde{c}$ with the same color due to Claim 1. Let us consider the remaining case that v is the parent of u (and thus $\tilde{c}(u) \neq \tilde{c}(v)$). Due to the consistency of λ with c, $\mathrm{F} \notin \Lambda(u)$ and thus $u \in O$. Then, e must contain at least three vertices, in particular it contains either a child of u, a sibling of u or a grand–parent of u:

* e **contains a child of** u**.** Call it w.
  Since $u \in O$, we clearly have $w \in R \cup O$ and e contains two vertices colored by $\tilde{c}$ with the same color due to Claim 1 applied to w.
* e **contains a sibling of** u**.** Call it w.
  Since $u \in O$, we clearly have $w \in R \cup O$ and thus $\lambda(w)$ is either F or G. If $\lambda(w) = \mathrm{F}$ then $\tilde{c}(w) = \tilde{c}(v)$ and the vertices v and w are two vertices of e colored by $\tilde{c}$ with the same color. If $\lambda(w) = \lambda(u) = \mathrm{G}$ then $\tilde{c}(u) = \tilde{c}(w) = \tilde{c}^\star(v)$ and thus u and w are two vertices of e colored by $\tilde{c}$ with the same color.
* e **contains a grand–parent of** u**.**
  In this case e contains two vertices colored by $\tilde{c}$ with the same color due to Claim 1.

– $\lambda(u) \in \{1, \ldots, k\}$
  If $\lambda(u) = i$ then we could get by substituting w_i (a member of the witness sequence with $\mathrm{level}(w_i) = \mathrm{level}(u) > \mathrm{level}(v)$) with v a witness sequence with a smaller level sum, contradicting the minimality of the witness sequence.

- **Neither** u **is a predecessor of** v **nor** v **is a predecessor of** u**.**
  Let w be the nearest common predecessor of u and v. We distinguish several cases to prove that e contains two vertices colored by $\tilde{c}$ with the same color:

  – $\lambda(u) = \mathrm{F}$ **or** $\lambda(v) = \mathrm{F}$
    Then u (resp. v) and its parent belong to e and they are both colored with the same color by the coloring $\tilde{c}$.
  – $\lambda(u) = \mathrm{G}$ **and** $\lambda(v) = \mathrm{G}$
    If $\mathrm{parent}(u) \neq w$ or $\mathrm{parent}(v) \neq w$ then e contains two vertices of the same color due to Claim 1; otherwise it holds that $w = \mathrm{parent}(u) = \mathrm{parent}(v)$. Then, it holds that $\tilde{c}(u) = \tilde{c}(v) = \tilde{c}^\star(w)$ and thus u and v are two vertices of e colored by $\tilde{c}$ with the same color.
  – $\lambda(u) \in \{1, \ldots, k\}$ **and** $\lambda(v) = \mathrm{G}$ (the case that $\lambda(u) = \mathrm{G}$ and $\lambda(v) \in \{1, \ldots, k\}$ is symmetric)
    If u and v are siblings, then due to the consistency of λ with c, it has to hold that $\lambda(u) = \lambda(v)$. Thus u and v are not siblings. If $\mathrm{parent}(v) \neq w$ then e contains two vertices of the same color due to Claim 1. We may therefore assume that $w = \mathrm{parent}(v)$. Since w is the nearest common predecessor of u and v, the level of u is greater than the level of v. It follows from the consistency of λ with c and the witness sequence that u has a sibling w_i, where $i = c(u) = c(v) = c(w_i)$, contradicting the minimality of the witness sequence ($\mathrm{level}(v) > \mathrm{level}(w_i)$).

– $\lambda(u) \in \{1, \ldots, k\}$ **and** $\lambda(v) \in \{1, \ldots, k\}$
  If u and v are not siblings, then $\Lambda(u) \cap \Lambda(v) \cap \{1, 2, \ldots, k\} = \emptyset$. It follows from the consistency of λ with c and the witness sequence that $c(u) \in \Lambda(u) \cap \{1, 2, \ldots, k\}$ and $c(v) \in \Lambda(v) \cap \{1, 2, \ldots, k\}$ and hence $c(u) \neq c(v)$, a contradiction. Hence u and v are siblings. The consistency of λ implies that $\lambda(u) = \lambda(v)$, and hence $\tilde{c}(u) = \tilde{c}(v)$ and u and v are two vertices of e colored by $\tilde{c}$ with the same color.

□

Corollary 2 *If a hypertree H has a strict k'–coloring, then it also has a strict k–coloring for all $2 \leq k \leq k'$. In particular, the feasible set of any mixed hypertree is gap–free.*

Proof: Let c be a strict k'–coloring of H and let $w_1, \ldots, w_k$ be a minimal witness sequence with respect to c. Let λ be consistent with c. Then the coloring $\tilde{c}$ based on the witness sequence $w_1, \ldots, w_k$ and mapping λ is a strict k–coloring. Set $k' = \overline{\chi}(H)$ to derive that $\mathcal{F}(H)$ is gap-free.
□

Observation 2 *Let H a mixed hypertree with an underlying tree T. Then for each $k \in \mathcal{F}(H)$, there exists a strict k–coloring c satisfying the following: If u and v are two vertices of H such that $c(u) = c(v)$, then there either exists a vertex w on the path between u and v such that $c(u) = c(v) = c(w)$ or this path is monochromatic, i.e., all its inner vertices have the same color assigned by c.*

Proof: If $k = 1$, the statement is trivial. Let $2 \leq k \in \mathcal{F}(H)$ be fixed throughout the proof. Let c be a strict k–coloring of H and let $w_1, \ldots, w_k$ be a minimal witness sequence with respect to c. Let λ be consistent with c. Then $\tilde{c}$ corresponding to $w_1, \ldots, w_k$ and λ satisfies the conditions of the statement of the observation.
□

4 Algorithms

In this section, we present three polynomial time algorithms for deciding strict k-colorability of mixed hypertrees for restricted inputs. In particular, we study the cases that maximum vertex load l_v, maximum edge load l_e, the number k of colors or the maximum degree Δ of the underlying tree (which is given as a part of the input in this case) are bounded. Note that $l_e \leq l_v \leq l_e\Delta$ and hence bounded vertex load implies bounded edge load, and bounded the maximum degree together with edge load imply bounded vertex load.

In the first algorithm, we use the concept of witness sequences from the previous section:

Proposition 1 *Let k and Δ be fixed integers.* STRICT k--COLORING *can be solved in polynomial time for mixed hypertrees with underlying trees of maximum degree at most Δ.*

Proof: The algorithm consists in checking all possible witness sequences. We describe it formally:

```
for all sequences w_1,...,w_k of distinct vertices of H such
        that 0 = level(w_1) < level(w_2) <= ... <= level(w_k)
     do
        construct sets L, R and O for w_1,...,w_k by rules (1)-(3)
        for every vertex v construct the sets Λ(v) by rules (4)-(7)
        for all mappings λ : V -> {F,G,1,...,k} such that λ(v) ∈ Λ(v)
           do
              construct c̃ based on w_1,...,w_k and λ by rules (8)-(11)
              if c̃ is a proper coloring then
                 output YES
                 halt
           done
     done
output NO
halt
```

We claim that the algorithm outputs YES if and only if H has a strict k-coloring; YES is output only if the last checked coloring $\tilde{c}$ is a strict k-coloring. Suppose that H has a strict k-coloring, say c. Let $w_1, \ldots, w_k$ be a minimal witness sequence with respect to c and let λ be the mapping consistent with c and $w_1, \ldots, w_k$. It follows from Theorem 5 that the algorithm finds the strict k-coloring $\tilde{c}$ that is based on these $w_1, \ldots, w_k$ and λ.

Next we estimate the running time of this algorithm. Let n (resp. m) be the number of vertices (resp. edges) of H and let d be the maximum degree of T. There are $O(n^{k-1})$ choices of the witness sequence in the first step. Given $w_1, \ldots, w_k$, the sets L, R, O and $\Lambda(v), v \in V$, can be constructed in time $O(n + kd^2)$. It is clear that $|L| \leq k(d-1) \leq kd$ and $|R| \leq k(d-1)^2 + k \leq kd^2$. There are exactly $|\Lambda(v)| \leq k+2$ choices of $\lambda(v)$ for $v \in L$, two choices of $\lambda(v)$ for $v \in R$ and one choice of $\lambda(v)$ for $v \in O$. Hence the total number of choices of λ is $\prod_{v \in V} |\Lambda(v)| \leq (k+2)^{kd} 2^{kd^2}$. For each $w_1, \ldots, w_k$ and λ, the coloring $\tilde{c}$ can be constructed in time $O(n^2)$ and then checked in time $O(nm)$ if it is proper. The running time of the whole algorithm is thus majorized by $O(n^{k-1}(n+kd^2+(k+2)^{kd}2^{kd^2}(n^2+nm))) = O(n^{k+1}m)$ if k and d are bounded.
□

The next algorithm is based on a more or less straightforward dynamic programming approach:

Proposition 2 *Let l_v be a fixed integer. The problem* STRICT COLORING *can be solved in polynomial time for mixed hypertrees with vertex load bounded by l_v. Moreover, this problem is fixed parameter tractable.*

Proof: Fix an underlying tree T of H rooted in a vertex of degree one. Note that the edge load of H is bounded by l_v as well. In the description of the algorithm, we use colors $1, 2, \ldots, k$ and, to avoid confusion, we distinguish *edges* of the underlying tree and *hyperedges* (edges of H). For an edge $e = uv$, with u being closer to the root of T than v, T_e denotes the subtree of T rooted in v which is induced by the vertices which belong to the same component of $T \setminus v$ as u. H_e denotes the hypertree with the underlying tree T_e consisting of the hyperedges of H fully contained in T_e. For each vertex v of H, we fix an ordering $e_1^v, \ldots, e_l^v$ of the hyperedges containing v (note that $l \leq l_v$).

Let e be an edge of T and let v be its vertex closer to the root. Recall that $e_1^v, \ldots, e_l^v$ are the hyperedges containing v. We define $\Phi_e(x_1, \ldots, x_l)$ for $x_i \in \{\emptyset, C, D, 1, \ldots, l_v\}$ to be the maximum number of colors used by a proper coloring c of H_e which assigns color 1 to the vertex v and which satisfies the following:

- If $x_i = C$ then c colors two vertices of $e_i^v \cap V(T_e)$ with the same color.
- If $x_i = D$ then c colors two vertices of $e_i^v \cap V(T_e)$ with different colors.
- If $x_i = \kappa \in \{1, \ldots, l_v\}$ then c colors a vertex of $e_i^v \cap V(T_e)$ with the color κ.
- If $x_i = \emptyset$ then no requirements are imposed on $e_i^v \cap V(T_e)$.

The algorithm will compute the functions Φ_e by brute force in bottom up fashion. Note that for each edge e, the function Φ_e can be described by a table of its values of size $(3 + l_v)^{l_v}$.

Let v be a vertex of T and let $e_0, e_1, \ldots, e_m$ be the edges adjacent to v such that e_0 is the edge joining v to its parent. Let us assume that we have computed $\Phi_{e_1}, \ldots, \Phi_{e_m}$. When computing Φ_{e_0}, we first compute functions $\Phi_1^{e_0}, \ldots, \Phi_m^{e_0}$ which are defined similarly to the functions $\Phi_{e_1}, \ldots, \Phi_{e_m}$ as follows: $\Phi_i^{e_0}(x_1, \ldots, x_l)$ is the maximum possible number of colors of a proper coloring (under the constraints posed by the variables $x_1, \ldots, x_l$) of the mixed hypergraph whose vertices are vertices of $T_{e_1} \cup \ldots \cup T_{e_i}$ and whose edges are the edges of H fully contained in its vertex set. Each $\Phi_i^{e_0}$ can be by a straightforward approach computed from $\Phi_{i-1}^{e_0}$ and Φ_{e_i} in time $O((3+l_v)^{2l_v})$ for $2 \leq i \leq m$ ($\Phi_1^{e_0}$ is just Φ_{e_1}). At this step, only vertices whose colors are explicitly mentioned among variables x_i have to preserve their colors; the remaining vertices can change their original colors to other colors (under the condition that all the vertices colored with the same color change to the same new one). Thus we actually consider also the colorings using colors which are not among the colors

$1, \ldots, l_v$. On the other hand, l_v colors are sufficient to model all possible interactions between colors of vertices in edges contained in different subtrees (it is enough to model $\mathcal{C}$–edges, since for each $\mathcal{D}$–edge there are in a proper coloring two adjacent vertices with different colors).

The function Φ_{e_0} can be then computed from the function $\Phi_m^{e_0}$ by considering two possibilities — the end–vertices of the edge e_0 have either the same color or different colors. The upper chromatic number of a given mixed hypertree can be easily determined from the values of function Φ_f where f is the only edge adjacent to the root of the tree. We may conclude that the problem `STRICT COLORING` for a mixed hypertree with n vertices, m hyperedges and maximum vertex load l_v can be solved in time $O((n+m)(3+l_v)^{2l_v})$.

□

Our last algorithm is a version of the algorithm of Proposition 1 and it is based on a concept of witness sequences simplified for the case of mixed hypertrees containing only $\mathcal{C}$–edges:

Proposition 3 *Let k be a fixed integer. The problem* `STRICT` k-`COLORING` *can be solved in polynomial time for mixed hypertrees containing only $\mathcal{C}$–edges.*

Proof: Let H be a given mixed hypertree which contains only $\mathcal{C}$–edges. Let $v_1, \ldots, v_k$ be a minimal witness sequence with respect to a strict k-coloring c. We define a coloring c' as follows: $c'(v) = i$ if v_i is the nearest predecessor of v which belongs to the witness sequence. We claim that c is a proper coloring of H. Note that each vertex which is not contained in the witness sequence has the same color as its parent. If a $\mathcal{C}$–edge e is not properly colored by c', then all its vertices, except possibly its vertex closest to the root, are members of the witness sequence. This contradicts the minimality of the witness sequence with respect to c, because the vertex closest to the root would have to be colored with the same color as some other vertex of e by c. Hence it suffices to try all $O(n^k)$ possible witness sequences, construct the coloring c' for each of them and check if at least one of them is proper.

□

5 NP–completeness

The following proposition without a bound on the edge–load was proved in [7]; the proof with a bound on the edge–load can be found in [13]:

Proposition 4 *The problem* `STRICT COLORING` *is NP-complete for mixed hypertrees containing only $\mathcal{C}$-edges, and even for those with the maximum edge load bounded by four.*

We prove a modification of Proposition 4 for mixed bihypertrees:

Proposition 5 *The problem* `STRICT COLORING` *is NP-complete for mixed bihypertrees of maximum edge load bounded by twelve.*

Proof: We show a reduction from the decision problem on the independence number of a cubic graph (see [8]) is presented: Let G be a cubic graph with the vertex set $V(G) = \{w_1, \ldots, w_n\}$ and the edge set $E(G)$. Let T be a tree with the vertex set equal to $V = \{\star, u_1, \ldots, u_n, v_1, \ldots, v_n\}$ and the edge set equal to $\{\star u_i, u_i v_i | 1 \leq i \leq n\}$. We define a mixed bihypertree H on the vertex set V as follows: The edges of H are precisely the sets $\{\star, u_i, v_i, u_j\}$ for all i and j such that $w_i w_j$ is an edge of G (since H is a mixed bihypertree, all its edges are both $\mathcal{C}$ and $\mathcal{D}$–edges). We claim that $\overline{\chi}(H) = n + \alpha + 1$ where α is the size of the independent set of G. Note that each edge of the tree T is contained either in 6 or 12 edges of H (the edges incident to the root are contained in 12, the other edges in 6 edges of H).

We prove the claim: Let $w_{i_1}, \ldots, w_{i_\alpha}$ be an independent set of G. Let c be the following coloring of the vertices of H:

$$c(\star) = 0$$

$$c(v_j) = k \text{ for all } k = 1, \ldots, n$$
$$c(u_{i_j}) = n + j \text{ for all } j = 1, 2, \ldots, \alpha$$
$$c(u_k) = 0 \text{ for all } k \neq i_1, \ldots, i_\alpha$$

Let $\{\star, u_i, v_i, u_j\}$ be an edge of H. Either $c(u_i) = 0$ or $c(u_j) = 0$, since both w_i and w_j cannot be in the considered independent set and $c(\star) = 0 \neq c(v_i)$. Hence c is a strict $(n + \alpha + 1)$–coloring of H. On the other hand, let c be a coloring using β colors, $\beta \geq n$. We construct an independent set of G of size $\alpha = \beta - n - 1$. Let $R_0 = \star, \ldots, R_{\beta-1}$ (we assume that $c(\star) = 0$) be the minimal witness sequence with respect to T rooted at $\star$. We first define a new β-strict coloring $\tilde{c}$:

- We set $\tilde{c}(\star) = 0$.
- If both u_i and v_i are among $R_1, \ldots, R_{\beta-1}$ (a ray of Type 1), we set $\tilde{c}(u_i) = c(u_i)$ and $\tilde{c}(v_i) = c(v_i)$.
- If only u_i is among $R_1, \ldots, R_{\beta-1}$ (a ray of Type 2), we set $\tilde{c}(u_i) = 0$ and $\tilde{c}(v_i) = c(u_i)$.
- If only v_i is among $R_1, \ldots, R_{\beta-1}$ (a ray of Type 3), we set $\tilde{c}(u_i) = 0$ and $\tilde{c}(v_i) = c(v_i)$.
- If neither u_i nor v_i is among $R_1, \ldots, R_{\beta-1}$ (a ray of Type 4), we set $\tilde{c}(u_i) = 0$ and $\tilde{c}(v_i)$ to a completely new color.

Note that if c uses $\overline{\chi}(H)$ colors, the last case cannot occur. Let $\{\star, u_i, v_i, u_j\}$ be an edge of H; $\tilde{c}(\star) \neq \tilde{c}(v_i)$ due to the definition of $\tilde{c}$. If u_i and u_j belong both to rays of Type 1, then the original coloring c cannot be proper. Thus at least one of them belongs to a ray of Type 2, 3 or 4 and is colored by 0, the same color as the vertex $\star$. We have just proven that $\tilde{c}$ is a proper coloring. Let $A = \{w_i | \tilde{c}(u_i) \neq c(\star)\}$. The set A is an independent set of G, since $\tilde{c}$ is a proper coloring, and its size is exactly α, since vertices $\star, v_1, \ldots, v_n$ are colored by mutually different $n+1$ colors. This finishes the proof of the claim.
□

Proposition 6 *The problem* STRICT COLORING *is NP-complete for mixed hypertrees containing $\mathcal{C}$–edges only, even if the input hypertree is given together with an underlying tree of maximum degree at most 3.*

Proof: We present a reduction from 3–SAT (cf. [8]). Let Φ be a given formula with n variables $x_1, \ldots, x_n$. Let T be a tree whose all internal vertices have degree 3 and which has exactly n leaves; let $v_1, \ldots, v_n$ be the leaves. Let V be the set of the vertices of T. We add to T other $2n$ vertices v_i^{T} and v_i^{F} for $1 \leq i \leq n$ together with the edges $v_i v_i^{\mathrm{T}}$ and $v_i v_i^{\mathrm{F}}$ for $1 \leq i \leq n$. The $\mathcal{C}$–edges of the constructed mixed hypertree are the following:

- $\{v_i, v_i^{\mathrm{T}}, v_i^{\mathrm{F}}\}$ for $1 \leq i \leq n$
- $V \cup \{v_i^X, v_j^Y, v_k^Z\}$ for each clause containing the variables x_i, x_j and x_k; X is T if the occurrence of x_i in the clause is positive, F otherwise; Y and Z are chosen in a similar manner.

We claim that the upper chromatic number of the constructed mixed hypertree is $|V| + n$ iff Φ has a satisfying assignment. Note that the upper chromatic number of the constructed mixed hypertree cannot be more than $|V| + n$ since it contains n disjoint $\mathcal{C}$–edges (those corresponding to the variables) and it has $|V| + 2n$ vertices.

Assume that Φ has a satisfying assignment. We color the vertices of the mixed hypertree as follows: The vertices of V are colored with $|V|$ mutually different colors. If x_i is true, we assign the vertex v_i^{T} the color of v_i and the vertex v_i^{F} a completely new color. If x_i is false, we assign the color of v_i to the vertex v_i^{F} and a completely new color to the vertex v_i^{T}. This yields a proper coloring of the mixed hypertree with $|V| + n$ colors.

Assume that the upper chromatic number of the mixed hypertree is $|V| + n$ and let c be a proper coloring of it using this number of colors. The mixed hypertree has $|V| + 2n$ vertices and it contains n disjoint $\mathcal{C}$–edges of size three which correspond to the variables of Φ each having exactly one vertex in common with V; thus c assigns mutually different colors to the vertices of V. Moreover, the vertices v_i, v_i^{T} and v_i^{F} for each $1 \leq i \leq n$ can be colored as described in one of the following three possibilities:

- $c(v_i) = c(v_i^{\mathrm{T}})$ and v_i^{F} is the only vertex colored with the color $c(v_i^{\mathrm{F}})$.
- $c(v_i) = c(v_i^{\mathrm{F}})$ and v_i^{T} is the only vertex colored with the color $c(v_i^{\mathrm{T}})$.
- The vertices v_i^{T} and v_i^{F} are the only two vertices colored with the color $c(v_i^{\mathrm{T}}) = c(v_i^{\mathrm{F}}) \neq c(v_i)$.

In the first case, let x_i be true; in the second case, let x_i be false; in the third case, let the value of x_i be arbitrary. The just obtained assignment satisfies Φ, since each $\mathcal{C}$–edge which corresponds to a clause, has to contain at least two vertices of the same color and these vertices has to be v_i and v_i^X for some $1 \leq i \leq n$ and $X \in \{\mathrm{T}, \mathrm{F}\}$.
□

Proposition 7 *The problem* STRICT COLORING *is NP-complete for mixed bihypertrees, even if the input bihypertree is given together with an underlying tree of maximum degree at most 3.*

Proof: If in the mixed hypertree H of the proof of Proposition 6 each $\mathcal{C}$–edge is also a $\mathcal{D}$–edge, then the proof of Proposition 7 is obtained.
□

The last proposition of this section does not provide an NP–completeness result, but it shows that polynomial–time algorithms for the last two remaining open problems for general mixed hypertrees (cf. Table 1) would provide a polynomial–time algorithm for coloring co–hypergraphs with a fixed number of colors:

Proposition 8 *The problem of determining whether the upper chromatic number of a given co–hypergraph is at least k can be polynomially reduced to* STRICT $(k+1)$-COLORING *of mixed hypertrees with maximum edge load* 4.

Proof: Let H be a given co–hypergraph. We may assume that its maximum vertex degree is at most three (see [14] for details). Let $v_1, \ldots, v_n$ be the vertices of H. We create a mixed hypertree on a star whose central vertex is a new vertex $\star$ and whose leaves are the vertices $v_1, \ldots, v_n$. The $\mathcal{D}$–edges of this mixed hypertree are all the pairs $\{\star, v_i\}$ for $1 \leq i \leq n$ and its $\mathcal{C}$–edges are the sets $\{\star\} \cup C$ for each C which is a $\mathcal{C}$–edge of H.

Any proper coloring of the constructed mixed hypertree assigns the vertex $\star$ a color which is not used to color any other vertex and thus the coloring of the vertices $v_1, \ldots, v_n$ of the mixed hypertree is a proper coloring of H. We immediately conclude that $\overline{\chi}(H)$ is the upper chromatic number of the constructed mixed hypertree decreased by one.
□

Bounded parameters:	—	l_e	l_v l_v, l_e	Δ, l_e Δ, l_v, (l_e)	Δ
General mixed hypertrees					
k is fixed	?	?	P	P	P
k is part of input	NPC	NPC	P	P	NPC
Mixed bihypertrees					
k is fixed	?	?	P	P	P
k is part of input	NPC	NPC	P	P	NPC
Mixed hypertrees with only $\mathcal{D}$–edges	P	P	P	P	P
Mixed hypertrees with only $\mathcal{C}$–edges					
k is fixed	P	P	P	P	P
k is part of input	NPC	NPC	P	P	NPC

Table 1: Complexity of determining whether the upper chromatic number of a given mixed hypertree is at least k for various classes of mixed hypertrees; Δ is the maximum degree of the underlying tree, l_v/l_e is the maximum vertex/edge load.

6 Conclusion

We investigated the properties of precoloring extensions of mixed hypertrees. Further we introduced concept of witness sequences and we used this concept to prove several results on mixed hypertrees (Corollary 2, Observation 2) and to design an algorithm for coloring mixed hypertrees. Besides this algorithm, we designed two other algorithms, and we provided several NP–completeness reductions. The summary of our algorithmic results can be found in Table 1 (note that for mixed hypertrees H containing only $\mathcal{D}$–edges $\overline{\chi}(H) = n$ where n is the number of vertices of H).

Acknowledgement

The second and fourth authors thank Angelica Nuclitsu for discussions.

References

[1] E. Bulgaru and V. Voloshin, Mixed interval hypergraphs, Discrete Applied Math., 77, 1997, 24–41.

[2] C. Colbourn, J. Dinitz, A. Rosa: Bicoloring Triple Systems, Electronic J. Combin. 6# 1, paper 25, 16 pages.

[3] Ch. J. Colbourn, A. Rosa: Triple Systems, Clarendon Press, Oxford, 1999, sect. 18.6. Strict colouring and the upper chromatic number, 340–341.

[4] K. Diao, P. Zhao, H. Zhou: About the upper chromatic number of a co–hypergraph, Discrete Mathemetics 220, 2000, 67–73.

[5] Z. Dvořák, D. Král': On Planar Mixed Hypergraphs, Electronic J. Combin. 8 (1), 2001, #R35.

[6] T. Etzion and A. Hartman: Towards a large set of Steiner quadruple systems, SIAM J. Discrete Math. 4 (1991), 182–195.

[7] E. Flocos: The Upper Chromatic Number of Simple Co-Monostars, Anale stiintifice ale Universitatii de Stat din Moldova, Seria Stiinte Reale, Chisinau, 1997, 22–27.

[8] M. R. Garey, D. S. Johnson: Computers and Intractability, A Guide to the Theory of NP–completeness, Freeman, San Franscisco, Cal., 1979.

[9] T. Jiang, D. Mubayi, Zs. Tuza, V. Voloshin and D. B. West: Chromatic spectrum is broken, 6th Twente Workshop on Graphs and Combinatorial Optimization, 26–28, May, 1999, H. J. Broersma, U. Faigle and J. L. Hurink (eds.), University of Twente, May, 1999, 231–234.

[10] D. Kobler, A. Kündgen: Gaps in the chromatic spectrum of face–constrained plane graphs, Electronic J. Combin. 8 (1), 2001, #N3.

[11] D. Král': On Complexity of Colouring Mixed Hypertrees, Proceedings 13th International Symposium on Fundamentals of Computing Theory, 1st International Workshop on Efficient Algorithms, LNCS vol. 2138, 2001, pp. 516–524.

[12] D. Král': On feasible sets of mixed hypergraphs, submitted.

[13] D. Král', J. Kratochvíl, A. Proskurowski, H.-J. Voss: Coloring mixed hypertrees, Proceedings 26th Workshop on Graph-Theoretic Concepts in Computer Science, LNCS vol. 1928, 2000, 279–289.

[14] D. Král', J. Kratochvíl, H.-J. Voss: Complexity Note on Mixed Hypergraphs, Proceedings 26th International Symposium on Mathematical Foundations of Computer Science, LNCS vol. 2136, 2001, 474–486.

[15] J. Kratochvíl, Z. Tuza, M. Voigt: New trends in the theory of graph colorings: Choosability and list coloring, in: Contemporary Trends in Discrete Mathematics (from DIMACS and DIMATIA to the future), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 49, American Mathematical Society, Providence, RI, 1999, 183-197.

[16] A. Kündgen, E. Mendelsohn, V. Voloshin: Colouring planar mixed hypergraphs, Electronic J. Combin. 7, 2000, #R60.

[17] H. Lefmann, V. Rödl, and R. Thomas: Monochromatic vs. multicolored paths, Graphs Combin. 8 (1992), 323–332.

[18] L. Milazzo: On upper chromatic number for SQS(10) and SQS(16), Le Matematiche L (Catania, 1995), 179–193.

[19] L. Milazzo and Zs. Tuza: Upper chromatic number of Steiner triple and quadruple systems, Discrete Math. 174 (1997), 247–259.

[20] L. Milazzo and Zs. Tuza: Strict colorings for classes of Steiner triple systems, Discrete Math. 182 (1998), 233–243.

[21] V. Voloshin: On the upper chromatic number of a mixed hypergraph, Australasian Journal of Combinatorics 11, 1995, 25–45.

[22] V. Voloshin: The mixed hypergraphs, Computer Science Journal of Moldova 1, 1993, 45–52.