Coloring mixed hypertrees

Daniel Kráľ[1]⋆, Jan Kratochvíl[2]⋆⋆, Andrzej Proskurowski[3]⋆⋆⋆, and Heinz-Jürgen Voss[4]

[1] Department of Applied Mathematics, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic. E-mail: kral@kam.ms.mff.cuni.cz
[2] Department of Applied Mathematics and Institute for Theoretical Computer Science, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic. E-mail: honza@kam.ms.mff.cuni.cz
[3] Department of Computer and Information Science, University of Oregon, Eugene. E-mail: andrzej@cs.uoregon.edu
[4] Institute of Algebra, Technische Universität Dresden, Germany. E-mail: voss@math.tu-dresden.de

Abstract. A mixed hypergraph is a hypergraph with edges classified as of type 1 or type 2. A vertex coloring is strict if no edge of type 1 is totally multicolored, and no edge of type 2 monochromatic. The chromatic spectrum of a mixed hypergraph is the set of integers k for which there exists a strict coloring using exactly k different colors. A mixed hypertree is a mixed hypergraph in which every hyperedge induces a subtree of the given underlying tree. We prove that mixed hypertrees have continuous spectra (unlike general hypergraphs, whose spectra may contain gap, cf. [Jiang et al.: The chromatic spectrum of mixed hypergraphs, submitted]). We prove that determining the upper chromatic number (the maximum of the spectrum) of mixed hypertrees is NP-hard, and we identify several polynomially solvable classes of instances of the problem.

1 Introduction

Definition 1. A *mixed hypergraph* is a triple $H = (V, \mathcal{C}, \mathcal{D})$, where V is a set of vertices and $\mathcal{C}, \mathcal{D}$ are sets of hyperedges (hyperedges are subsets of the vertex set). A vertex coloring of a mixed hypergraph is *strict* if every edge $e \in \mathcal{C}$ contains two vertices of the same color, and every edge $e \in \mathcal{D}$ contains two vertices of different colors. A strict coloring that uses exactly k distinct colors is called a *strict k-coloring*. The *chromatic spectrum* of H is the set $Sp(H)$ of integers k such that H has a strict k-coloring. The spectrum of H is called *continuous* if it is an interval (in the set of integers). We denote $\overline{\chi}(H) = \max Sp(H)$ the *upper chromatic number* of H.

⋆ Research supported in part by Czech Research Grant GAČR 201/1999/0242.

⋆⋆ This author acknowledges partial support of Czech research grants GAUK 158/1999 and KONTAKT 338/99. Part of the research was carried on while visiting Technical University Dresden in June 1999 and University of Oregon in the Fall of 1999.

⋆⋆⋆ Research supported in part by National Science Foundation grants NSF-INT-9802416 and NSF-ANI-9977524.

The notion of mixed hypergraphs was introduced by Voloshin [12]. The concept is steadily gaining more interest both for potential applications and interesting theoretical results. Applications to list colorings, integer programming, scheduling and molecular biology can be found in [13, 11]. Mixed colorings of block designs of various types were considered in [3, 8, 7, 9]. Other coloring problems studied from a different point of view can be rephrased in terms of mixed hypergraphs [6].

Solving a long-standing open problem, Jiang et al. [4] showed that there are mixed hypergraphs whose spectra have gaps, i.e., are not continuous. Moreover, for every finite set of integers greater than 1 there exists a mixed hypergraph whose chromatic spectrum coincides with the given set. They also showed that spectra of interval hypergraphs are continuous. (A hypergraph is *interval* if its vertex set allows a linear ordering such that every edge is an interval in this ordering.) In this paper, we extend the former result to the class of hypertrees.

Definition 2. A *mixed hypertree* is a mixed hypergraph $H = (V, \mathcal{C}, \mathcal{D})$ such that there exists a tree T with vertex set $V(T) = V$ and such that every edge of H induces a subtree in T.

Without loss of generality, we assume that every edge of our hypertree has size at least 2, and every $\mathcal{C}$-edge has size at least three (the endpoints of a $\mathcal{C}$-edge of size two must be colored by the same color and we can contract this edge). We also assume that the hypertree has at least one edge. Every 2-coloring of the underlying tree (i.e., a bipartition of T) is then a strict coloring of H, as every $\mathcal{D}$-edge contains two vertices adjacent in T (and hence colored by different colors), and every $\mathcal{C}$-edge contains at least 3 vertices (and hence at least 2 vertices of the same color by the pigeon-hole principle). Thus we have the following:

Observation 1 *For every hypertree H, 2 belongs to the chromatic spectrum of H and is its minimum member, unless H has no $\mathcal{D}$-edges.*

Our aim is to prove the following theorem:

Theorem 1. *The chromatic spectrum of any mixed hypertree is continuous, in particular $Sp(H) = [2, \overline{\chi}(H)]$ if H contains a $\mathcal{D}$-edge, and $Sp(H) = [1, \overline{\chi}(H)]$ otherwise.*

This theorem is proved in the next section. Later, we address the algorithmic questions. In general, the questions of deciding if a mixed hypergraph is strict k-colorable, or whether its upper chromatic number has a given value may be of different complexity; e.g., if H has no $\mathcal{C}$ edges then $\overline{\chi}(H) = n$ while it is NP-complete to decide whether $2 \in Sp(H)$ (the problem of bicolorability of ordinary hypergraphs). Due to Theorem 1, these two problems are equally difficult for mixed hypertrees:

Proposition 1. *The problem of determining the upper chromatic number of a mixed hypertree and the problem of deciding whether $k \in Sp(H)$, for an input integer k, are polynomially equivalent.*

Proof. Given $k \geq 2$, this k belongs to $Sp(H)$ if and only if $k \leq \overline{\chi}(H)$. On the other hand, the upper chromatic number can be found by deciding for every $k \leq n = |V|$ whether H is strict k-colorable.

2 Chromatic spectra of mixed hypertrees

First, we show that a minimal counterexample to Theorem 1 (if it existed) would have $\mathcal{D} = E(T)$, i.e. the edges of the underlying tree, $E(T)$, would be precisely the $\mathcal{D}$-edges of H.

Lemma 2. *Let H be a hypertree with disconnected spectrum and with minimum possible number of vertices. Then every tree edge is a subset of some $\mathcal{D}$-edge of H.*

Proof. Suppose H has 2 vertices, say x, y, adjacent in T which do not belong together to the same $\mathcal{D}$-edge of H. Let T_1 be the largest subtree of T which contains x but not y, and let T_2 be the largest subtree containing y but not x. Let H_i, $i = 1, 2$, be the hypertree induced by T_i, i.e., H_i contains exactly those edges of H that consist only from vertices of T_i. Both H_1 and H_2 are smaller than H and hence have continuous spectra.

Denote $c = \overline{\chi}(H)$ and consider a c-coloring f of H. Then f restricted to H_1 is a strict coloring and uses some $c_1 \leq c$ colors, and similarly, f restricted to H_2 uses some c_2 colors. The union of these two color sets is the set of c colors used on H, and hence

$$c_1 + c_2 \geq c.$$

For $i = 1, 2$ and for every $t_i, \min Sp(H_i) \leq t_i \leq c_i$, consider a t_i-coloring f_i of H_i. If necessary, rename and permute the colors so that $f_1(x) = f_2(y)$ and so that this is the only color that f_1 and f_2 have in common. Regard the union of f_1 and f_2 as coloring g of the entire H. This coloring uses $t_1 + t_2 - 1$ colors, and it is a strict coloring of H: every $\mathcal{D}$-edge of H lies entirely within H_1 or within H_2, and hence cannot be monochromatic; no $\mathcal{C}$-edge that lies totally within H_1 or within H_2 is totally multicolored since it was presented and hence well colored in H_1 or H_2, and all the remaining $\mathcal{C}$-edges are well colored because they contain both x and y (and $g(x) = g(y)$).

If we let t_1 and t_2 range over $Sp(H_1)$ and $Sp(H_2)$, respectively, we obtain strict t-colorings of H at least in the range $3 \leq t \leq c_1 + c_2 - 1$. Since $c_1 + c_2 - 1 \geq c - 1$, this covers the range $[3, c-1]$. But H has a strict 2-coloring (Lemma 1), and a strict k-coloring by the definition of $c = \overline{\chi}(H)$. Hence H has a continuous spectrum, contradicting the assumption.

We define the *size* of H as the sum of the sizes of its edges, i.e., $s(H) = \sum_{e \in \mathcal{C} \cup \mathcal{D}} |e|$.

Lemma 3. *Let H be a hypertree with disconnected spectrum, and of minimum possible size. Then all $\mathcal{D}$-edges have size 2.*

Proof. Suppose H has a $\mathcal{D}$-edge e of size greater than 2. Denote $c = \overline{\chi}(H)$, and fix a strict c-coloring f of H. The edge e contains 2 vertices, say x and y, such that $f(x) \neq f(y)$ and such that x and y are adjacent in the underlying tree T (otherwise e would be monochromatic in f). Consider $H' = (V, \mathcal{C}, (\mathcal{D} \setminus e) \cup \{\{x, y\}\})$, i.e., replace e by the pair x, y. The resulting hypergraph H' is a hypertree again, because xy is an edge of T. Since f is a strict coloring of H' as well, $c \in Sp(H')$. And since $s(H') < s(H)$, and H was a minimum hypertree with disconnected spectrum, the spectrum of H' is continuous, i.e., for every $t, 2 \leq t \leq c$, H' has a strict t-coloring, which is also a strict t-coloring of H. This contradicts the assumption of H having a disconnected spectrum.

Corollary 1. *Let H be hypertree with disconnected spectrum, with minimum possible number of vertices, and with minimum possible size. Then $\mathcal{D} = E(T)$.*

We therefore restrict our attention to mixed hypertrees with $\mathcal{D} = E(T)$, and hence use the notation $H = (T, \mathcal{C}, \mathcal{D} = E(T))$. Our goal is to show that all such hypertrees have continuous spetra. That would prove Theorem 1, since if there were a counterexample, then a minimum counterexample would satisfy $\mathcal{D} = E(T)$. In order to prove this, we introduce a seemingly more general problem of finding strict colorings of *precolored* hypertrees. The input of this problem consists of a mixed hypertree $H = (T, \mathcal{C}, E(T))$ and k disjoint independent sets $S_1, S_2, \dots, S_k$ of vertices, and the task is to find a strict coloring which colors all vertices of S_i with color b_i, for each $i = 1, 2, \dots, k$. Such a coloring will be called a *precoloring extension*. (Here and later on, we refer to a set of vertices as an independent set if it is independent in the underlying tree T.)

Lemma 4. *Let H be a hypertree and let $S_1, S_2, \dots, S_k$ be disjoint nonempty independent sets of vertices of H precolored by $b_1, b_2, \dots, b_k$, respectively . If H has a precoloring extension using $c \geq k + 2$ colors, then it has a precoloring extension which uses exactly $k + 2$ colors.*

Proof. Let the precoloring extension of H and $S_1, \dots, S_k$ which uses c colors be f. For every $\mathcal{C}$-edge e, fix two different vertices $x_e, y_e \in e$ such that $f(x_e) = f(y_e)$ (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. For the resulting hypertree H', f is still a strict precoloring extension, and vice versa, every precoloring extension for H' (and $S_1, \dots, S_k$) is also a strict precoloring extension for H (and $S_1, \dots, S_k$). Note that in H', every $\mathcal{C}$-edge induces a path in the underlying tree, and therefore it makes sense talking about end-vertices of $\mathcal{C}$-edges.

Now for H', 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 edge exists. Note that the new S_i's are still disjoint after this procedure is finished, since we have only added vertices of color b_i into the set S_i.

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

Define the auxiliary graph G as the graph obtained from T by deleting S and then contracting all edges of A by collapsing their end-vertices. We claim that G has at least two vertices and is acyclic. Indeed, since the coloring f uses at least two colors other than $b_1 \dots b_k$, these two additional colors must remain in G, since vertices of different colors cannot be collapsed. Also, suppose G has a cycle. This cycle corresponds to a cycle in $T + A$ which contains at least three edges of T. Replacing every edge in A by the corresponding $(x_e y_e)$-path in T, we get a closed walk in T which traverses these (at least) three edges each only once, a contradiction with acyclicity of T.

Since G is acyclic, it can be colored by two colors, say colors b_{k+1} and b_{k+2}. And since G has at least two vertices, we can take such coloring which actually uses both colors. We claim that this coloring yields a strict precoloring extension of H' and $S_1, \dots, S_k$, and it obviously uses exactly $k+2$ colors.

This coloring is strict on all $\mathcal{D}$-edges (i.e., edges of the tree T) by its definition. We will show that every $\mathcal{C}$-edge e' contains two vertices of the same color:

Case 1: e' has at least one end-vertex in S: Then e' has both end-vertices in S by the enlargment procedure, and these endpoints are in the same S_i, i.e., they are colored by the same color.

Case 2: e' has both endpoints in $T \setminus S$ and all other vertices in S: Then $x_e y_e \in A$, x_e and y_e are unified into one vertex of G, and so x_e and y_e get the same color.

Case 3: e' has at least 3 vertices 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.

Lemma 5. *Let H be a mixed hypertree and let $S_1, S_2, \dots, S_k$ be disjoint nonempty independent sets of vertices of H. If H has a precoloring extension using $c \geq k+2$ colors, then it has precoloring extensions using exactly t colors for every $t, k+2 \leq t \leq c$.*

Proof. We will prove the statement by induction on the number of vertices in $T \setminus S$ (again $S = \bigcup_{i=1}^{k} S_i$). Note also that the statement holds trivially true due to Lemma 4 if $c = k+2$, and so we assume that $c > k+2$ further on.

Let the precoloring extension of H and $S_1, \dots, S_k$ which uses c colors be f. Replace every $\mathcal{C}$-edge e by e' as in the preceding proof (e' is a x_e, y_e-path in T and $f(x_e) = f(y_e)$). Enlarge the sets S_i as in the preceding proof (if $x_e \in S_i$ and $y_e \notin S$ then add y_e into S_i and repeat). Denote S'_i the enlarged set S_i, $i = 1, \dots, k$, and $S' = \bigcup_{i=1}^{k} S'_i$. Let H' be the hypertree obtained by replacing each e by e'. Again, f is a strict precoloring extension of H' (and S') using exactly k colors, and every precoloring extension of H' (and S') is also a precoloring extension of H (and S).

H contains at least one vertex not belonging to S' (e.g., the vertex colored by a color different from $b_1, \dots, b_k$). Choose a vertex not belonging to S', call it x, such that all other vertices in $V \setminus S'$ lie in the same connected component of $T - \{x\}$. Let T_2 be this connected component and let $T_1 = T \setminus T_2$ be the subtree rooted in x that is disjoint with T_2. Finally, let y be the neighbor of x in T_2.

Now let C be the set of endpoints of $\mathcal{C}$-edges e' that have x as the other endpoint, formally

$$C = \{y_e : x_e = x, e \in \mathcal{C}\} \cup \{x_e : y_e = x, e \in \mathcal{C}\}.$$

We distinguish three cases:

Case 1: $f(x) = b_i$ for some i. Then redefine $S'_i := S'_i \cup C \cup \{x\}$. The hypertree H' has a strict precoloring extension for this new $S'_1, \dots, S'_k$ (namely f) and has less unprecolored vertices than H (with precoloring $S_1, \dots, S_k$). Hence by induction hypothesis, H' has precoloring extensions using exactly t colors for every $k+2 \le t \le c$. Each of these extensions is also a strict precoloring extension for H and $S_1, \dots, S_k$.

Case 2: $f(x) \neq b_i$ for all $i = 1, 2, \dots, k$ and $C = \emptyset$. Consider H_2 as the hypertree on T_2 with all edges e' that lie entirely within T_2 as the $\mathcal{C}$-edges of H_2. All the $\mathcal{C}$-edges e' not entirely lying in T_2 will be colored properly, since their both end points are in S'_i for the same i and thus the following construction assign their ends the same color (b_i). Set $S''_i = S'_i \cap T_2$ and say that k' of these are nonempty. The hypertree H_2 has less unprecolored vertices than H (at least x is missing) and we may use induction hypothesis. The coloring f restricted to H_2 is a strict precoloring extension and it uses at least $k' + c - k - 1$ colors (all the new colors used by f are used in T_2 except possibly for the color of x). But since $c - k \ge 3$, $k' + c - k - 1 \ge k' + 2$ as we need for Lemma 4. It follows that H_2 has a precoloring extension which uses exactly t colors for every t, $k' + 2 \le t \le k' + c - k - 1$. Each such coloring g' can be re-extended to a coloring g of H by giving x the color $g(x) = b_{k+2}$ if $g'(y) = b_{k+1}$ and setting $g(x) = b_{k+1}$ otherwise. Such a coloring g then uses $t + k - k'$ colors, and this number ranges from $k + 2$ (for $t = k' + 2$) to $c - 1$ (for $t = k' + c - k - 1$). This is what we needed, as a precoloring extension using c colors exists by the assumption itself.

Case 3: $f(x) \neq b_i$ for all $i = 1, 2, \dots, k$ and $C \neq \emptyset$. Then $C \subset T_2$ and $C \cap S' = \emptyset$. Consider H_2 as the hypertree on T_2 with all edges e' that lie entirely within T_2 as the $\mathcal{C}$-edges of H_2; the other $\mathcal{C}$-edges will be colored properly for the same reason as in the previous case. Set $S''_i = S'_i \cap T_2$, $S'_{k+1} = C \cup \{x\}$ and let k' of the sets $S''_i, i \le k$ be nonempty. The hypertree H_2 has less unprecolored vertices than H (at least x is missing), but now $k'+1 \le k+1$ is the number of colors used for the precoloring. By induction hypothesis, since H_2 has a strict precoloring extension (namely f restricted to H_2) which uses exactly $c - k + k' \ge k' + 3$ colors, H_2 has a precoloring extension which uses exactly t colors for every t, $k' + 3 \le t \le k' + c - k$. Each such coloring g' can be re-extended to a coloring g of H by giving x the color $g(x) = b_{k+2}$ if $g'(y) = b_{k+1}$ and setting $g(x) = b_{k+1}$ otherwise (note that in the former case all $\mathcal{C}$ edges starting in x and ending in C are well colored because they contain vertex $y \in T_2$ of the same color b_{k+1} as the endpoint in C). Such a coloring g then uses $t + k - k'$ colors, and this number ranges from $k + 3$ (for $t = k' + 3$) to c (for $t = k' + c - k$). This is what we needed, since a precoloring extension using $k + 2$ colors exists by Lemma 4.

For the proof of Theorem 1, use Lemma 5 with $k = 0$ and $c = \overline{\chi}(H)$.

3 Dynamic programming algorithm

One might expect that deciding strict k-colorability of mixed hypertrees should be solvable in polynomial time due to the structure of the underlying tree. We will see in the next section that this is not true (unless P=NP). In this section we study the more or less straightforward dynamic programming algorithm to decide the existence of a strict coloring of a mixed hypertree H. Henceforth, n will denote the number of vertices of H, k the number of colors, s the maximum size of a hyperedge and l_e (l_v) the maximum H-load of an edge (a vertex) of the underlying tree T, where the load of an edge (a vertex) is the number of hyperedges that include both of its end points (the vertex itself).

Consider a given mixed hypertree $H = (T, \mathcal{C}, \mathcal{D})$ on an underlying tree T rooted in a leaf $r \in V(T)$. Let $e = \{u, v\}$ be an edge of the underlying tree T with v being closer to r than u (in T). The hypergraph induced by the vertices of T in the same connected component of $T - \{v\}$ as u will be denoted by H_e. (This subhypertree contains exactly those hyperedges of H that have all vertices in H_e.) Let $e_1, e_2, \dots, e_m$ be the hyperedges containing v and not fully contained in H_e. We maintain boolean arrays $\Phi^e(a_1, a_2, b_1, b_2, \dots, b_m, c)$ indexed by $a_1, a_2 \in \{1, 2, \dots, k\}$, $b_1, \dots, b_m \in \{0, 1, \dots, k\}$ and $c \in \{1, 2, \dots, k\}$ such that

$$\Phi^e(a_1, a_2, b_1, b_2, \dots, b_m, c) = \mathsf{true}$$

if and only if H_e allows a strict coloring $\phi :\longrightarrow \{1, 2, \dots, k\}$ such that

1. $\phi(v) = a_1, \phi(u) = a_2$,
2. for every $j = 1, 2, \dots, m$, if $b_j > 0$ then $e_j \cap V(H_e)$ contains a vertex of color b_j, and if $b_j = 0$ then e_j is satisfied within $V(H_e)$, i.e., $e_j \cap V(H_e)$ contains two vertices of the same color if $e_j \in \mathcal{C}$ and $e_j \cap V(H_e)$ contains two vertices of different colors if $e_j \in \mathcal{D}$,
3. ϕ uses exactly c colors on $V(H_e)$.

The space needed to maintain these arrays is $O(nk^{3+l_e})$. It is also clear that H allows a strict coloring using exactly k colors if and only if

$$\Phi^{r_e}(a_1, a_2, k) = \mathsf{true}$$

for some a_1 and a_2 where r_e is the tree-edge containing the root r (note that $m = 0$ in the case of the root since all hyperedges are fully contained in $H_{r_e} = H$).

The initilization of the arrays Φ for the leaves of T is obvious. We will only add hints how to update the arrays. Straightforward bottom-up strategy is used: consider for vertex x with children $y_1, y_2, \dots, y_g$, a true entry of the arrays Φ^{xy_i} for each i and derive the information for Φ^{zx}, where z is the parent of x. This would be, however, exponential in g, the number of children of x, i.e., exponential in d. Our aim is to be a little more careful here. Let $e_1, \dots, e_m$ be the hyperedges that contain x. We introduce another array Ψ^x_j, $1 \le j \le g$, such that

$$\Psi^x_j(a_1, a_2, b_1, \dots, b_m, c) = \mathsf{true}$$

if and only if the hypertree H^x_j induced by $\{z\} \cup \bigcup_{i=1}^{j} H_{xy_i}$ allows a strict coloring $\psi :\longrightarrow \{1, 2, \dots, k\}$ such that

1. $\psi(z) = a_1, \phi(x) = a_2$,
2. for every $h = 1, 2, \ldots, m$, if $b_h > 0$ then $e_h \cap V(H_j^x)$ contains a vertex of color b_h, and if $b_h = 0$ then e_h is satisfied within $V(H_j^x)$,
3. ψ uses exactly c colors on $V(H_j^x)$.

The "horizontal" recursion step $j \rightarrow j+1$ consists of combining true entries of Ψ_j^x with a true entry of $\Phi^{xy_{j+1}}$, which can be done in time $O(k^{3+l_e}).O(k^{2+l_v}).O(2^{l_v}).O(k) = O(2^{l_v} k^{6+l_e+l_v})$, since there are at most $O(k^{3+l_e})$ true entries $\Phi^{xy_{j+1}}$, combined with at most $O(k^{2+l_v})$ true entries Ψ_j^x (we only consider those that use the same color for x), and for this choice of two entries we check all hyperedges containing x. If a hyperedge is fully contained in H_j^x we check if it is satisfied in $\Phi^{xy_{j+1}}$ or in Ψ_j^x or by their combination. For hyperedges that are not fully contained in H_j^x, we choose which color will be propagated (henceforth the factor $O(2^{l_v})$). Finally, we compare the colors appearing in $\Phi^{xy_{j+1}}$ and in Ψ_j^x. Say $\Phi^{xy_{j+1}}$ explicitly names k_1 colors, Ψ_j^x names k_2 colors, and k_3 of these are in common. In total, $\Phi^{xy_{j+1}}$ uses c_1 colors and Ψ_j^x uses c_2 colors. The anonymous $c_1 - k_1$ colors may be those from $k_2 - k_3$ explicit colors for Ψ_j^x, but some or all of them may be completely new. This consideration gives an interval of possible numbers of colors used in Ψ_{j+1}^x, and the range of the interval can be determined in time linear in k. For all these values of parameters we set Ψ_{j+1}^x to `true`.

Skipping the tedious details, we have sketched the argument for the following proposition:

Theorem 2. *It can be decided in time $O(n \cdot 2^{l_v} k^{6+l_e+l_v}))$ whether H allows a k-strict coloring.*

A straightforward corollary reads that the upper chromatic number of a mixed hypertree can be determined in polynomial time if the maximum vertex load l_v is bounded.

4 Upper chromatic number is NP-hard

The following theorem shows that the hardness of the decision problem of the upper chromatic number for mixed hypertrees is caused by the $\mathcal{C}$-edges.

Theorem 3. *It is NP-complete to decide whether $\overline{\chi}(H) \geq k$, for k part of the input, even for hypertrees $H = (V, \mathcal{C}, \emptyset)$ with maximum edge load and maximum hyperedgesize 3.*

Proof. Given a cubic graph G containing n vertices, define a tree T with $n+1$ vertices (let w be a new vertex not contained in $V(G)$) by

$$V(T) = V(G) \cup \{w\}$$

and edges

$$E(T) = \{vw : v \in V(G)\}$$

and define a mixed hypertree H on the same vertex set by

$$\mathcal{C} = \{\{uwv\} : uv \in E(G)\}$$

$$\mathcal{D} = \emptyset.$$

All hyperedges of H have size three, and since the edge load of a tree-edge vw equals the degree of v in G, the load of every tree-edge is three as well. We claim that $\overline{\chi}(H) = \alpha(G) + 1$ and thus this construction yields a reduction from the well known NP-complete problem INDEPENDENT SET (which remains NP-complete for cubic graphs, cf. [5]).

Let f be a coloring of H using $k+1$ colors. Let $v_1, \ldots, v_k$ be vertices of H such that $f(v_i) \neq f(v_j)$ for all $i \neq j$ and $f(v_i) \neq f(w)$ for all $1 \leq i \leq k$. If there were an edge v_iv_j, then the edge $\{v_iwv_j\} \in \mathcal{C}$ would not be colored properly. Thus the vertices $v_1, \ldots, v_k$ create an independent set of G and $\alpha(G) \geq k$.

On the other hand, let $v_1, \ldots, v_k$ be an independent set of G and $b_1, \ldots, b_{k+1}$ be $k+1$ different colors. Define a coloring f of H by $f(v_i) = b_i$, $1 \leq i \leq k$, and $f(v) = b_{k+1}$ for all $v \in V(H) \setminus \{v_1, \ldots, v_k\}$. This coloring is strict, since for every edge $uwv \in \mathcal{C}$ either u or v is different from all v_i, $1 \leq i \leq k$ (otherwise $v_1, \ldots, v_k$ would not be an independent set), and thus its color is the same as the color of w. This implies that $\overline{\chi}(H) \geq k+1$. We conclude that $\alpha(G) + 1 = \overline{\chi}(H)$.

5 Conclusion

Several questions concerning the computational complexity remain open. E.g., we have shown that strict colorability is polynomial if the loads of vertices are bounded, while the problem is NP-complete for unbounded vertex loads. The complexity of the case when the number of colors is fixed (but the load of vertices unbounded) is presently open.

Another most probably hard problem is to decide if the chromatic spectrum of a given mixed hypergraph is continuous.

References

1. R. Ahlswede, Coloring hypergraphs: a new approach to multi-user source coding, I and II, *Journal of Combinatorics, Information, and System Sciences* **4** (1979), 76–115, and **5** (1980), 220–268.
2. E. Bulgaru and V. Voloshin, Mixed interval hypergraphs, *Discrete Applied Math.* **77** (1997), 24–41.
3. T. Etzion and A. Hartman, Towards a large set of Steiner quadruple systems, *SIAM J. Discrete Math.* **4** (1991), 182–195.
4. T. Jiang, D. Mubayi, Zs. Tuza, V. Voloshin, D.B. West, The chromatic spectrum of mixed hypergraphs (submitted).
5. M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete problems, *Theor. Comput. Sci.* **1**, 237–267.
6. H. Lefmann, V. Rödl, and R. Thomas, Monochromatic vs. multicolored paths, *Graphs Combin.* **8** (1992), 323–332.

7. L. Milazzo, On upper chromatic number for SQS(10) and SQS(16), *Le Matematiche* **L** (Catania, 1995), 179–193.
8. L. Milazzo and Zs. Tuza, Upper chromatic number of Steiner triple and quadruple systems, *Discrete Math.* **174** (1997), 247–259.
9. L. Milazzo and Zs. Tuza, Strict colorings for classes of Steiner triple systems, *Discrete Math.* **182** (1998), 233–243.
10. A. Niculitsa, V. Voloshin, Mixed hypertrees (submitted).
11. Zs. Tuza and V. Voloshin, Uncolorable mixed hypergraphs, *Discrete Applied Math.* (to appear).
12. V. Voloshin, On the upper chromatic number of a hypergraph, *Australasian J. Comb.* **11** (1995), 25–45.
13. V. Voloshin, Mixed hypergraphs as models for real problems (in preparation).