AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **48** (2010), Pages 253–267

Voloshin's conjecture for C-perfect hypertrees*

Csilla Bujtás   Zsolt Tuza†

Department of Computer Science and Systems Technology
University of Pannonia
H–8200 Veszprém, Egyetem u. 10
Hungary

Abstract

In the "mixed hypergraph" model, proper coloring requires that vertex subsets of one type (called C-edges) should contain two vertices of the same color, while the other type (D-edges) should not be monochromatic. Voloshin [*Australas. J. Combin.* 11 (1995), 25–45] introduced the concept of C-perfectness, which can be viewed as a dual kind of graph perfectness in the classical sense, and proposed a characterization for C-perfect hypertrees without D-edges. (A hypergraph is called a hypertree if there exists a graph T which is a tree such that each hyperedge induces a subtree in T.)

We prove that the structural characterization conjectured by Voloshin is valid indeed, and it can even be extended in a natural way to mixed hypertrees without (or, with only few) D-edges of size 2; but not to mixed hypertrees in general. The proof is constructive and leads to a fast coloring algorithm, too. In sharp contrast to perfect graphs which can be recognized in polynomial time, the recognition problem of C-perfect hypergraphs is pointed out to be co-NP-complete already on the class of C-hypertrees.

1 Introduction

In his highly influential paper [14], Voloshin introduced many interesting notions concerning a novel type of hypergraph coloring. During the years, terminology slightly changed; we follow the one used in the research monograph [15]. In this setting, a *mixed hypergraph* is a triple $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$, where X is the *vertex set* (underlying set) and $\mathcal{C}, \mathcal{D}$ are two set systems over X, whose members are termed *C-edges* and *D-edges*, respectively. A *proper coloring* is a mapping

$$\varphi : X \to \mathbb{N}$$

* Research supported in part by the Hungarian Scientific Research Fund, OTKA grants 49613 and 81493.

† Also: Computer and Automation Institute, Hungarian Academy of Sciences, H–1111 Budapest, Kende u. 13–17, Hungary

such that $|\varphi(H)| < |H|$ for all $H \in \mathcal{C}$ and $|\varphi(H)| > 1$ for all $H \in \mathcal{D}$; that is, every C-edge contains two vertices with a common color and every D-edge contains two vertices with distinct colors. The concept of hypergraph coloring in the classical sense occurs as the particular case $\mathcal{C} = \emptyset$, when $\mathcal{H}$ is called a *D-hypergraph*. Analogously, in case of $\mathcal{D} = \emptyset$, $\mathcal{H}$ is called a *C-hypergraph*. A C-, D-, or mixed hypergraph is *r-uniform* if each of its edges has precisely r vertices.

Notation. Throughout this paper, we restrict our attention to mixed hypergraphs that are *colorable*; i.e., those admitting at least one proper coloring. The *upper chromatic number* of $\mathcal{H}$, denoted by $\bar{\chi}(\mathcal{H})$, is the largest possible number $|\varphi(X)|$ of colors, taken over all proper colorings φ of $\mathcal{H}$. If φ uses exactly $\bar{\chi}(\mathcal{H})$ colors, and we choose one vertex from each color class, we obtain a set of $\bar{\chi}$ vertices that cannot contain any C-edge as a subset. Hence, denoting by $\alpha_C(\mathcal{H})$ the maximum number of vertices in a set containing no C-edges — called the *C-stability number* — the inequality

$$\bar{\chi}(\mathcal{H}) \le \alpha_C(\mathcal{H})$$

provides a universal bound on the upper chromatic number.

Definition. A colorable mixed hypergraph $\mathcal{H}$ is called *C-perfect* if $\bar{\chi}(\mathcal{H}') = \alpha_C(\mathcal{H}')$ holds for all induced subhypergraphs $\mathcal{H}'$ of $\mathcal{H}$. (An *induced subhypergraph* is obtained by taking a set $X' \subseteq X$ and all edges of $\mathcal{H}$ that are entirely contained in X'.) A *minimally C-imperfect* mixed hypergraph is one that is colorable and not C-perfect, but each of its proper induced subhypergraphs is C-perfect.

Examples. Voloshin [14, 15] considered the following basic examples of C-perfect and C-imperfect mixed hypergraphs, all but the last type having $\alpha_C = |X| - 1$.

- A *monostar* is a mixed hypergraph in which the intersection of all C-edges consists of precisely one vertex (and D-edges are arbitrary). Monostars are *not* C-perfect.

- A *bistar* is a mixed hypergraph in which the intersection of C-edges contains a pair of vertices, say x, y, such that $\{x, y\}$ is not a D-edge. Bistars *are* C-perfect.

- A *polystar* is a mixed hypergraph with at least two C-edges, in which the intersection Y of the C-edges — called *center* — is nonempty, and every vertex pair in Y forms a D-edge. (The particular case of $|Y| = 1$ means a monostar; and if $|Y| \ge 2$, then a bistar is obtained by omitting any 2-element D-edge contained in Y. Moreover, in a C-hypergraph, every polystar is a monostar.) By *k-polystar* we mean a polystar with $|Y| = k$. Polystars are *not* C-perfect.

- A *cycloid*, denoted by $\mathcal{C}_n^r$, is an r-uniform C-hypergraph with $n > r$ vertices $x_1, \ldots, x_n$ and n C-edges of the form $\{x_i, x_{i+1}, \ldots, x_{i+r-1}\}$ ($i = 1, \ldots, n$, subscript addition taken modulo n). The cycloid $\mathcal{C}_n^r$ is C-perfect if $n \le 2r - 2$, it is minimally C-imperfect if $n = 2r - 1$, and it contains a C-monostar on $2r - 1$

vertices if $n \geq 2r$ and so in this case it is not C-perfect and not minimally C-imperfect either.

An r-uniform minimally C-imperfect example, different from polystars and cycloids, was described by Král' [9] for each $r \geq 3$ on $2r$ vertices. Later, the present authors [3] have found more r-uniform minimally C-imperfect constructions for each $r \geq 4$, whose number tends to infinity with r.

Hypertrees. A *mixed hypertree* is a mixed hypergraph $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ that admits a tree graph $T = (X, E)$ — termed *host tree* — with the same vertex set X and with some edge set E, such that the vertex set of each edge $H \in \mathcal{C} \cup \mathcal{D}$ of $\mathcal{H}$ induces a subtree in T.[1] An important subclass is that of mixed *interval hypergraphs*, where the host tree T is required to be a *path*. Observe that in a mixed hypertree no k-polystars can occur with $k > 2$, because the 2-element D-edges must be edges of T, while T is a triangle-free graph.

Let us give a brief summary of what has been published on the perfectness of mixed hypertrees.

- In [14, Theorem 4.29], it was stated that a C-hypertree is C-perfect if and only if it contains no monostars as induced subhypergraphs. The "only if" part follows from the fact that monostars are not C-perfect. On the other hand, it turned out later that the original argument in [14] for the "if" part does not work, but the statement itself remained a conjecture.

- Bulgaru and Voloshin [4, Theorem 2.8] proved that a mixed interval hypergraph is perfect if and only if it has no induced polystars. This implies that the conjecture holds for interval C-hypergraphs.

- In [15, Theorem 5.17] it was proved that if a mixed hypertree does not contain any polystar as a *subhypergraph* — i.e., not only the induced polystars are excluded — then it is C-perfect. In particular, if a C-hypertree does not contain any monostar as a subhypergraph, then it is C-perfect. But this sufficient condition is not at all necessary for C-perfectness, as it is pointed out in our Proposition 1.

- Our Corollary 1 verifies the original conjecture.

1.1 New results

The main positive result of our paper is a sufficient condition for C-perfectness. In order to formulate it, we need to introduce the following notation. For a mixed hypertree $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ over a host tree T, we denote

$$\mathcal{D}_2 = \{D \in \mathcal{D} \mid |D| = 2\}$$

that can be viewed as a subforest of T (possibly edgeless).

[1] Applying a result of Duchet [6], Flament [7] and Slater [11], this is equivalent to assuming that $(X, \mathcal{C} \cup \mathcal{D})$ is an arboreal hypergraph in the terminology of [2].

Theorem 1 *Let $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ be a colorable mixed hypertree such that all but at most one vertices have degree 0 or 1 in $\mathcal{D}_2$. If $\mathcal{H}$ contains no induced polystar, then it is C-perfect, and a proper coloring of $\mathcal{H}$ with $\bar{\chi}(\mathcal{H})$ colors can be found in polynomial time.*

From the negative side, our main result is a rather unexpected one. In fact, a strong expectation is suggested in [15, p. 85] that C-perfect mixed hypertrees can be recognized and $\bar{\chi}$-colored efficiently. While the latter may be true (as we prove it for the subclass described in Theorem 1), the former is refuted by the next result.

Theorem 2 *The recognition problem of C-perfect C-hypertrees is* co-NP*-complete.*

It can be shown that hard inputs necessarily contain large edges. Indeed, some recent results of [3] imply that C-perfect C-hypertrees of *bounded edge size* can be recognized efficiently.

We observe further that the non-hereditary version, too, of the defining property $\bar{\chi} = \alpha_C$ of C-perfectness is hard to test. This fact can be verified from the proof of Proposition 12 in [10] already, but was not observed there explicitly. Moreover, our proof will yield intractability for bounded edge sizes, too.

Theorem 3 *The problem of deciding whether $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$ is* NP*-complete over the class of C-hypertrees, also when restricted to inputs of maximum edge size seven.*

Hence, in sharp contrast to Theorem 2, imposing an upper bound on edge sizes does not reduce the time complexity of testing whether $\alpha_C = \bar{\chi}$. As a matter of fact, even some strong non-approximability results are valid for C-hypertrees of bounded edge size, as proved recently in [1].

On the other hand, if both the maximum edge size and the maximum degree of the host tree are bounded then the validity of $\alpha_C = \bar{\chi}$ can be decided in polynomial time. This follows from a more general result of [1] on C-hypertrees of bounded degree, where it is shown that the degree condition alone — without restricting edge size — is sufficient for polynomial-time solvability.

Let us note further that the construction in [10] and Theorem 3 complement each other in the following way. The former shows hardness over host trees of maximum degree 3 (but the majority of edges has a very large intersection, namely nearly half of the vertices are contained in all edges larger than 3) while the latter yields intractability for edge sizes not exceeding 7 (but its host tree has a vertex adjacent to 1/3 of the others, and it is contained in all edges of size 7).

Returning to the positive side, in spite of the preceding results, the following constructive approach can be applied. It has the flavor of “robust algorithms” what means more than solvability in polynomial time. We shall put some related remarks in the concluding section.

Theorem 4 *Over the class of colorable mixed hypertrees $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ such that all but at most one vertices have degree 0 or 1 in $\mathcal{D}_2$, there exists a polynomial-time algorithm whose output is either an induced polystar subhypergraph or a proper coloring of $\mathcal{H}$ with $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$ colors.*

Note that Theorem 1 is an immediate consequence of Theorem 4.

Theorem 1 has some interesting consequences. First of all, it implies that the characterization of C-perfect C-hypertrees, as proposed in [14], is valid indeed.

Corollary 1 *A C-hypertree is C-perfect if and only if it contains no monostar as an induced subhypergraph. Moreover, C-perfect C-hypertrees can be $\bar{\chi}$-colored in polynomial time.*

Also, the exclusion of 2-element D-edges leads to a characterization.

Corollary 2 *A colorable mixed hypertree $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ with $\mathcal{D}_2 = \emptyset$ is C-perfect if and only if it contains no monostar as an induced subhypergraph. Those C-perfect mixed hypertrees with $\mathcal{D}_2 = \emptyset$ can be $\bar{\chi}$-colored in polynomial time.*

Remark 1 *It follows immediately by Corollary 2 that an r-uniform mixed hypertree with $r \geq 3$ is C-perfect if and only if it contains no monostar as an induced subhypergraph.*

Moreover, the same characterization is valid for $r = 2$ (i.e., for colorable mixed tree graphs), because of the following reasons:

(i) the upper chromatic number is equal to the number of connected components of the C-graph; and

(ii) the following sequence of equivalences is valid: this C-graph is not a matching plus isolated vertices $\Longleftrightarrow$ it contains a star — necessarily induced — with more than one edge $\Longleftrightarrow$ its C-stability number is larger than the number of its connected components. $\square$

Remark 2 *We have not made attempts to optimize the running time of the algorithms given in the proofs. It can be seen that for all positive results (Theorems 1 and 4, Corollaries 1 and 2) the universal upper bound $O(mn)$ is valid, where n and m denote the number of vertices and hyperedges, respectively. We leave it an open problem to analyze which structures admit a linear-time solution, i.e. coloring in $O(\sum_{H \in \mathcal{C} \cup \mathcal{D}} |H|)$ steps.*

It is important to note that only *induced* polystars are necessary to exclude for C-perfectness, as it is shown by the following assertion.

Proposition 1 *There exists a C-hypertree that is C-perfect and contains C-monostars as (non-induced) subhypergraphs.*

Moreover, we prove that Bulgaru and Voloshin's characterization of C-perfectness for mixed interval hypergraphs does not extend to mixed hypertrees. Our counterexample also shows that the condition on high-degree vertices of $\mathcal{D}_2$ in Theorem 1 is best possible.

Proposition 2 *There exists a mixed hypertree that is not C-perfect although it contains no induced polystars and has only two vertices of degree higher than 1 in $\mathcal{D}_2$.*

Theorems 1 and 4 are proved in the next section, and constructions for Propositions 1 and 2 are given in Section 3. Algorithmic hardness is deduced in Section 4. We mention some related open problems in the concluding section.

2 Characterization of C-perfect C-hypertrees

In this section we design a polynomial-time algorithm on the class of mixed hypertrees satisfying the conditions of Theorems 1 and 4. For each input hypertree $\mathcal{H}$ the algorithm outputs either a coloring with exactly $\alpha_C(\mathcal{H})$ colors or an induced polystar subhypergraph of $\mathcal{H}$. This proves directly Theorem 4, and implies Theorem 1.

Let us denote by τ_C the *transversal number* of $\mathcal{C}$, that is the minimum cardinality of a set S intersecting all $H \in \mathcal{C}$; and by ν_C its *matching number*, the maximum number of mutually disjoint C-edges.

It is immediate by definition that 'stable set' and 'transversal' are complementary notions: if $S \subset X$ meets all C-edges, then $X \setminus S$ does not contain any, and vice versa. Hence, the Gallai-type equality

$$\alpha_C(\mathcal{H}) + \tau_C(\mathcal{H}) = |X|$$

holds for every mixed hypergraph $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$.

Moreover, it is well-known that if $\mathcal{H}$ is a hypertree, then also the equality

$$\tau_C(\mathcal{H}) = \nu_C(\mathcal{H})$$

is valid. In his monograph, Berge [2, p. 67, Example 4] includes (without reference) a proof by induction on τ_C, but something more is true. Namely, if the host tree of the input hypertree is at hand, then all the following can be determined efficiently: the value of $\tau_C = \nu_C$, a transversal of minimum cardinality, and a matching that consists of a maximum number of pairwise disjoint edges. A standard way to do it is described in Algorithm 1. Due to information from András Gyárfás — that we gratefully acknowledge here — it is a natural generalization of Gallai's method on interval hypergraphs [8] and appears to be folklore. Let us note further that a host tree of $\mathcal{H}$ can be found in linear time if there exists one, by the results of Tarjan and Yannakakis [13]; that is, the host tree T need not be part of the input as it can be determined in preprocessing.

Having the output S and $\mathcal{M}$ of Algorithm 1 at hand, we need to distinguish between the cases $m = 1$ and $m \geq 2$.

If $m = 1$, then all C-edges of $\mathcal{H}$ share a vertex. In this case either $\mathcal{H}$ itself is a polystar, or it does not contain any polystars at all. Clearly, this can be tested efficiently.

Assume that $\mathcal{H}$ is not a polystar. Then $\bigcap_{H \in \mathcal{C}} H$ contains two vertices, say x and y, that do not form a D-edge. It is easy to locate the pair x, y in polynomial time. We assign color 1 to x and y, and mutually distinct further colors to the vertices of $X \setminus \{x, y\}$. In this way a proper coloring of $\mathcal{H}$ is obtained with $\bar{\chi} = |X| - 1$ colors.

Suppose from now on that $m \geq 2$. We now require that Algorithm 1 chooses the root x^* as a vertex of largest degree in $\mathcal{D}_2$. In this case, either $x^* = x_m$ or $x^* \notin S$. Since we have assumed $m \geq 2$, the degree distribution of $\mathcal{D}_2$ ensures that all $x_i \neq x^*$ (i.e., $1 \leq i \leq m-1$) are incident with at most one edge of $\mathcal{D}_2$.

Let us begin with the (non-proper) coloring where each vertex of $\mathcal{H}$ has a distinct color. We are going to prove the essential part of Theorems 1 and 4 in the following more explicit form:

Algorithm 1 — Minimum transversal and maximum packing of a hypertree

Require: Hypertree $\mathcal{H} = (X, \mathcal{C})$ and its host tree $T = (X, E)$.
Ensure: A smallest set $S = \{x_1, \ldots, x_m\} \subseteq X$ that meets all $H \in \mathcal{C}$, and a largest family $\mathcal{M} = \{H_1, \ldots, H_m\} \subseteq \mathcal{C}$ of mutually disjoint edges.

Fix any root x^* in T
$\mathcal{C}' \leftarrow \mathcal{C}$
for all $H \in \mathcal{C}'$ find $x_H \in H$ nearest to x^* in T
$i \leftarrow 1$
while $\mathcal{C}' \neq \emptyset$ **do**
  Determine $H \in \mathcal{C}'$ such that x_H is farthest from x^* in T
  $x_i \leftarrow x_H$
  $H_i \leftarrow H$
  $\mathcal{C}' \leftarrow \mathcal{C}' - x_i$ // remove all edges containing x_i //
  $i \leftarrow i + 1$
endwhile
$m \leftarrow i - 1$
$S \leftarrow \{x_1, \ldots, x_m\}$, $\mathcal{M} \leftarrow \{H_1, \ldots, H_m\}$

($\star$) *For every $\mathcal{H}$ satisfying the conditions of Theorems 1 and 4, either an induced polystar with center containing some $x_i \in S$ can be found, or a proper α_C-coloring can be obtained by identifying a suitably chosen pair of colors inside each $H_i \in \mathcal{M}$, where S and $\mathcal{M}$ are the output of Algorithm 1.*

The identification of colors will be done sequentially for $i = 1, \ldots, m$ (unless an induced polystar is found in some step), and the soundness of the algorithm will be proved by induction on m. The basic case $m = 1$ has already been settled. The iterative step can be handled by the following procedure that either locates an induced polystar or determines how two colors can be identified properly inside H_i. The key steps are summarized in Algorithm 2, whose input is taken from the output of the preceding iteration. Initially we set $\mathcal{H}_0 = \mathcal{H} = (X, \mathcal{C}, \mathcal{D})$. The number $m = \nu_C(\mathcal{H})$ and the sets $S = \{x_1, \ldots, x_m\} \subseteq X$ and $\mathcal{M} = \{H_1, \ldots, H_m\} \subseteq \mathcal{C}$ are determined from Algorithm 1 with specifically chosen root x^*.

In a less formal description, let us consider first the particular case of $i = 1$. We then take the subhypergraph induced by $X \setminus \{x_2, \ldots, x_m\}$. Since S is a transversal, all edges of $\mathcal{C}'$ contain x_1, moreover $\mathcal{C}'$ is nonempty because $H_1 \in \mathcal{C}'$. We set $Y = \bigcap_{H \in \mathcal{C}'} H$.

If $|Y| = 1$, then $X \setminus \{x_2, \ldots, x_m\}$ induces a monostar because all C-edges $H \in \mathcal{C} \setminus \mathcal{C}'$ meet $\{x_2, \ldots, x_m\}$. Hence, the algorithm halts with an induced monostar found efficiently in $\mathcal{H}$. Similarly, if $|Y| = 2$ and $Y \in \mathcal{D}_2$, then an induced 2-polystar has been found. In both cases, $x_1 \in Y$ holds.

Let $\mathcal{H}^- := \mathcal{H} - \{x_2, \ldots, x_m\}$, and assume that it is neither monostar nor 2-polystar. Then, beside $x_1 \in Y$, there is at least one further vertex, say y', in $Y \setminus \{x_1\}$. Moreover, if $x_1 y' \in \mathcal{D}_2$, then — since $\mathcal{H}^-$ is not a 2-polystar — there exists a further vertex $y'' \in Y \setminus \{x_1, y'\}$. In either case, there exists $y_1 \in Y \setminus \{x_1\}$ such that

Algorithm 2 — Elimination of C-edges containing x_i but none of $x_{i+1}, \ldots, x_m$

Require: Hypertree $\mathcal{H}_{i-1} = (X_{i-1}, \mathcal{C}_{i-1}, \mathcal{D}_{i-1})$ from iteration $i-1$.

Ensure: Either a vertex $y_i \in H_i \setminus \{x_i\}$ receiving the color of x_i, or induced polystar $\mathcal{H}_{i-1} - \{x_{i+1}, \ldots, x_m\}$.

$\mathcal{C}' \leftarrow \{H \in \mathcal{C}_{i-1} \mid x_j \notin H$ for all $i < j \leq m\}$

$Y \leftarrow \bigcap_{H \in \mathcal{C}'} H$

if $|Y| = 1$, or $|Y| = 2$ and $Y \in \mathcal{D}_2$

  then $\mathcal{H}_{i-1} - \{x_{i+1}, \ldots, x_m\}$ is induced polystar — STOP

  else Determine $y_i \in Y \setminus \{x_i\}$ such that $x_i y_i \notin \mathcal{D}_2$

Identify the colors of x_i and y_i

$\mathcal{H}_i \leftarrow \mathcal{H}_{i-1} - y_i$ // delete y_i and all edges containing it //

$x_1 y_1 \notin \mathcal{D}_2$, because x^* is the only possible vertex of degree higher than 1 in $\mathcal{D}_2$, and $x^* \notin H_1$. Note that y_1 can always be chosen as a child or grandchild of x_1.

Evidently, the conditions imposed on $\mathcal{H}$ remain valid for every induced subhypergraph and, in particular, for $\mathcal{H} - y_1$, too. Moreover, let us observe that applying Algorithm 1 to $\mathcal{H} - y_1$ we can get the output $S_1 = \{x_2, \ldots, x_m\}$ and $\mathcal{M}_1 = \{H_2, \ldots, H_m\}$. Indeed, vertex y_1 is contained in all C-edges disjoint from $\{x_2, \ldots, x_m\}$, whereas $\mathcal{H} - y_1$ still involves the mutually disjoint C-edges $H_2, \ldots, H_m$. Consequently, $\tau_C(\mathcal{H} - y_1) = m - 1$ and therefore S_1 is a minimum transversal set and $\mathcal{M}_1$ is a maximum edge packing for the $\mathcal{C}$-edges of $\mathcal{H} - y_1$. Every additional C-edge, contained in $\mathcal{H} - y_1$ but not in $\mathcal{H} - x_1$, is incident with at least one of the vertices $x_2, \ldots, x_m$. Hence, in each step Algorithm 1 can find the same x_i farthest from x^*, no matter whether it is applied to $\mathcal{H} - x_1$ or to $\mathcal{H} - y_1$. Although in some cases the edge set $\mathcal{M}$ could be chosen in various ways, $H_2, \ldots, H_m$ always remains one of the possible outputs when the input is the hypergraph $\mathcal{H} - y_1$. Consequently, since τ_C and ν_C are decreased from m to $m-1$, the induction hypothesis can be applied to $\mathcal{H} - y_1$ with sets $S_1, \mathcal{M}_1$. As regards Algorithm 2 this means that the i-th iteration will be performed on the subhypergraph induced by

$$X_{i-1} = X \setminus \{y_j \mid 1 \leq j < i\}$$

for $i = 2, \ldots, m-1$, and eventually the structure is reduced to the case of $\tau_C = 1$.

We now prove that the coloring of $\mathcal{H}$ determined by the algorithm is proper (unless an induced polystar with center containing x_i is found in the i-th iteration, for some $i \leq m$).

- By the induction hypothesis, a proper color partition is constructed for $\mathcal{H} - y_1$, such that each H_i $(2 \leq i \leq m)$ contains precisely one vertex pair with identified color, and all the other color classes are singletons. This guarantees proper coloring for every C- and D-edge contained in $\mathcal{H} - y_1$.

- Identifying the colors of x_1 and y_1 yields a proper coloring for every C-edge inside $\mathcal{H}^-$.

- If some $H \in \mathcal{C}$ is incident with y_1 and meets $\{x_2, \dots, x_m\}$, then it contains all internal vertices of a path from y_1 to some x_i ($2 \le i \le m$). By the choice of H_1 and x_1, such a path necessarily passes through x_1, and hence H is properly colored by the monochromatic pair $\{x_1, y_1\}$.

- Due to Algorithm 2 and the induction hypothesis, the monochromatic vertex pairs are contained in mutually disjoint hyperedges and none of them forms a $\mathcal{D}_2$-edge. Hence, no monochromatic D-edges can occur.

Summarizing, we either obtain an induced polystar in step 3 halting some iteration of Algorithm 2, or construct a coloring that establishes the equality $\bar{\chi}(\mathcal{H}) = \alpha_C(\mathcal{H})$. In particular, if $\mathcal{H}$ contains no induced polystar, then $\bar{\chi} = \alpha_C$ is valid in every induced subhypergraph of $\mathcal{H}$, hence yielding C-perfectness. Moreover, one can see that all the steps above can be executed in polynomial time. This completes the proof of Theorems 1 and 4. □

3 Constructions

To simplify notation, below we list the vertices of an edge without separating commas and without braces. For instance, the edge $\{v_1, v_2, v_3\}$ will be written simply as $v_1v_2v_3$.

Proof of Proposition 1. Consider the C-hypertree $\mathcal{H}_6$ with six vertices $v_1, \dots, v_6$ and four C-edges $v_1v_2v_3, v_1v_2v_4v_6, v_2v_3v_4v_5, v_4v_5v_6$, from which the first three and also the last three determine a (non-induced) monostar. The host tree has edge set

$$\{v_1v_2, v_2v_3, v_2v_4, v_4v_5, v_4v_6\}.$$

A proper 4-coloring is obtained e.g. by assigning color 1 to v_1 and v_2, color 2 to v_4 and v_5, color 3 to v_3, and color 4 to v_6. Since $v_1v_2v_3$ and $v_4v_5v_6$ are disjoint edges, the independence number cannot be larger than 4. Moreover, the removal of any vertex yields either just one edge or two edges sharing two vertices, and so $\mathcal{H}_6$ is C-perfect. □

Proof of Proposition 2. Consider the following mixed hypertree $\mathcal{H}_{12} = (X, \mathcal{C}, \mathcal{D})$ with vertex set $X = Y \cup Z$ where $Y = \{y_1, \dots, y_5\} \cup \{y^*\}$ and $Z = \{z_1, \dots, z_5\} \cup \{z^*\}$, over the host tree $T = (X, E)$ whose edge set is

$$E = \{y^*z^*\} \cup \{y^*y_i \mid 1 \le i \le 5\} \cup \{z^*z_i \mid 1 \le i \le 5\}.$$

Let

$$\mathcal{D} = \mathcal{D}_2 = E \setminus \{y^*z^*\}$$

and

$$\mathcal{C} = \{Y, Z\} \cup \{H_i \mid 1 \le i \le 5\}$$

where

$$H_i = y^*z^*y_iy_{i+1}y_{i+2}z_iz_{i+1}z_{i+2},$$

subscript addition taken modulo 5. Here y^* and z^* are the two vertices of degree higher than 1 in $\mathcal{D}_2$.

Since Y and Z are disjoint edges, we see that $\alpha_C \le 10$, and in fact $\alpha_C = 10$ holds because $X \setminus \{y^*, z^*\}$ contains no C-edges. On the other hand, we claim that $\bar{\chi} < 10$. Indeed, a coloring φ with 10 colors should satisfy $|\varphi(Y)| = |\varphi(Z)| = 5$ (because "C-edge" means $|\varphi(Y)| \le 5$, $|\varphi(Z)| \le 5$). Moreover, due to the presence of $\mathcal{D}_2$-edges, the two monochromatic vertex pairs should appear inside $Y \setminus \{y^*\}$ and $Z \setminus \{z^*\}$. However, each pair $\{y_i, y_j\}$ and $\{z_i, z_j\}$ $(1 \le i < j \le 5)$ is contained in at most two of the five C-edges H_i. Thus, at least one H_i has $|\varphi(H_i)| = |H_i|$ whenever $|\varphi(X)| = 10$. Consequently, $\mathcal{H}_{12}$ is not C-perfect.

One can also see that $\mathcal{H}_{12}$ is not a monostar but it is minimally C-imperfect. Indeed, deleting any one vertex we obtain either just one C-edge (Y or Z) with five D-edges, or three C-edges (H_i, H_{i+1}, Y or H_i, H_{i+1}, Z for some $i \le 5$, where H_6 means H_1) with nine D-edges. Those three C-edges form a bistar (with non-$\mathcal{D}_2$-pair $y_{i+1}y_{i+2}$ or $z_{i+1}z_{i+2}$), which is C-perfect. □

4 Intractability

In this section we prove that the recognition problems of C-perfectness and of $\alpha_C = \bar{\chi}$ are hard for C-hypertrees.

Proof of Theorem 2. By Corollary 1, a C-hypertree $\mathcal{H}$ is <u>not</u> C-perfect if and only if it contains an induced monostar. For any given $\mathcal{H}' \subseteq \mathcal{H}$ it can be tested in linear time whether $\mathcal{H}'$ is an induced monostar; hence, membership in co-NP is clear.

To prove co-NP-hardness, we apply reduction from 3-SAT. Let $\Phi = C_1 \wedge \cdots \wedge C_m$ be an instance of 3-SAT, that is a Boolean formula in conjunctive normal form, with $m \ge 1$ clauses C_j over n variables $x_1, \ldots, x_n$, and with exactly three literals from three distinct variables per clause. It can be assumed without loss of generality that each x_i occurs as both a positive and a negative literal in Φ (for otherwise the instance could be simplified in linear time, while the problem is NP-complete).

We first construct a host tree $T = (X, E)$ on $m + 4n + 1$ vertices,

$$X = \{c^*\} \cup \{y_j \mid 1 \le j \le m\} \cup \{x_i', x_i'', t_i, f_i \mid 1 \le i \le n\}$$

where y_j represents clause C_j and x_i', x_i'', t_i, f_i correspond to variable x_i. The edge set of the host tree is

$$E = \{c^* y_j \mid 1 \le j \le m\} \cup \{c^* x_i',\ x_i' x_i'',\ x_i'' t_i,\ x_i'' f_i \mid 1 \le i \le n\}.$$

We now construct a C-hypertree $\mathcal{H}$ over T. It will have $n + 3m$ C-edges, one H_i for each variable x_i $(1 \le i \le n)$ and one H_j^ℓ for each literal of each clause C_j $(1 \le j \le m$, $1 \le \ell \le 3)$. The "small" C-edges are

$$H_i = \{x_i', x_i'', t_i, f_i \mid 1 \le i \le n\}.$$

All "large" C-edges H_j^ℓ contain c^* and $m - 1$ of the y-vertices:

$$y_{j'} \in H_j^\ell \iff 1 \le j, j' \le m,\ j \ne j'.$$

Moreover, each H_j^ℓ contains three further vertices that induce a subpath in T, as follows.

- If the ℓ-th literal $(1 \le \ell \le 3)$ of clause C_j $(1 \le j \le m)$ is x_i, then

$$x_i', x_i'', f_i \in H_j^\ell$$

- If the ℓ-th literal $(1 \le \ell \le 3)$ of clause C_j $(1 \le j \le m)$ is $\neg x_i$, then

$$x_i', x_i'', t_i \in H_j^\ell$$

We view the vertices t_i, f_i as representatives of a truth assignment $a : \{x_1, \ldots, x_n\} \to \{\mathsf{T}, \mathsf{F}\}$ of Φ, with the meaning

$$t_i \longleftrightarrow a(x_i) = \mathsf{T}, \qquad f_i \longleftrightarrow a(x_i) = \mathsf{F}.$$

This establishes an obvious bijection between truth assignments of Φ and subsets $S \subset \{t_i, f_i \mid 1 \le i \le n\}$ containing precisely one vertex from each pair $\{t_i, f_i\}$. Consequently, for such subsets S, the following property is valid.

($\star\star$) The ℓ-th literal of the j-th clause satisfies C_j if and only if the corresponding leaf contained in S is not an element of H_j^ℓ.

Since 3-SAT is NP-complete, the co-NP-complete status of C-perfectness will follow, once we show that $\mathcal{H}$ contains an induced monostar if and only if Φ is satisfiable.

(1) *If Φ is satisfiable, then $\mathcal{H}$ contains an induced monostar.*

Suppose that the truth assignment $a : \{x_1, \ldots, x_n\} \to \{\mathsf{T}, \mathsf{F}\}$ satisfies each clause C_j of Φ. Then the set S of leaves corresponding to $\{a(x_i), \ldots, a(x_n)\}$ — that intersects all H_i $(1 \le i \le n)$ — is disjoint from at least one of the three subtrees H_j^1, H_j^2, H_j^3 (the one whose literal satisfies C_j). Consequently, in the subhypergraph $\mathcal{H}'$ induced by $X \setminus S$, the intersection of edges does not contain any y_j. Due to our assumption, each variable appears as both a positive and a negative literal in Φ, implying that no vertex x_i' is contained in all hyperedges of $\mathcal{H}'$. Therefore, c^* is the only vertex shared by all edges of $\mathcal{H}'$, and hence $\mathcal{H}'$ is an induced monostar in $\mathcal{H}$.

(2) *If $\mathcal{H}$ contains an induced monostar, then Φ is satisfiable.*

Suppose that $\mathcal{H}' = (X', \mathcal{C}')$ is an induced monostar in $\mathcal{H}$. We first observe that no small edge H_i can occur in $\mathcal{H}'$. Indeed, if $H_i \in \mathcal{C}'$, then all edges of $\mathcal{H}'$ meet H_i, but then all of them contain $\{x_i', x_i''\}$, contradicting the assumption that $\mathcal{H}'$ is a monostar.

From now on, we assume $H_i \notin \mathcal{C}'$ for all $1 \le i \le n$. Then c^* is contained in all edges of $\mathcal{H}'$, therefore the intersection of the edges of $\mathcal{H}'$ cannot contain any y_j.

Suppose, without loss of generality, that $\mathcal{H}'$ is an induced monostar with as many edges as possible and without isolated vertices. We cannot have $\{t_i, f_i\} \subset X'$ for any $1 \le i \le n$, because $H_i \nsubseteq X'$. On the other hand, the maximality of $\mathcal{H}'$ implies that at least one of t_i and f_i is contained in X', since otherwise we could take e.g. a clause C_j where x_i is a positive literal, and insert the corresponding edge H_j^ℓ into $\mathcal{H}'$.

Thus, $|X' \cap \{t_i, f_i\}| = 1$ holds for all i, and the same is true for the set $S = X \setminus X'$. Hence, S corresponds to some truth assignment a of Φ. Since

$$y_j \notin \bigcap_{H \in \mathcal{C}'} H \qquad \forall\, 1 \le j \le m,$$

we obtain that for each j there exists a value $\ell = \ell(j)$ such that $H_j^\ell \in \mathcal{C}'$ and, equivalently, $H_j^\ell \cap S = \emptyset$. Therefore, by $(\star\star)$, the ℓ-th literal in the j-th clause satisfies C_j. Since this holds for each j $(1 \le j \le m)$, we obtain $\Phi(a) = \mathsf{T}$. This completes the proof of the theorem. $\square$

Proof of Theorem 3. Applying Algorithm 1, the value of $\alpha_C = |X| - \tau_C$ can be obtained efficiently, hence the equality $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$ can be verified by a proper α_C-coloring in non-deterministic polynomial time. (Alternatively, a proper k-coloring together with $n-k$ mutually disjoint C-edges can also witness the equality.) Therefore, the problem is in NP.

In the proof of NP-hardness, we apply a polynomial-time reduction from 3-SAT. In flavor it is similar to the construction given in [10] but, as we already mentioned in the Introduction, the structure obtained is different.

For every Boolean 3-CNF formula Φ, a C-hypertree $\mathcal{H}$ with maximum edge size 7 will be constructed, such that $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$ holds if and only if Φ is satisfiable. Let $\Phi = C_1 \wedge \cdots \wedge C_m$ be any instance of 3-SAT, with variables $x_1, \dots, x_n$. It can be supposed without loss of generality that the three literals in each clause C_j of Φ correspond to exactly three distinct variables. The $(3n+1)$-element vertex set of $\mathcal{H}$ is defined as

$$X = \{c^*\} \cup \{x'_i, t_i, f_i \mid 1 \le i \le n\}$$

where the vertices x'_i, t_i, f_i correspond to variable x_i. The host tree $T = (X, E)$ has the edge set

$$E = \{c^* x'_i, x'_i t_i, x'_i f_i \mid 1 \le i \le n\}.$$

The three-element C-edges of $\mathcal{H}$ are $H_i = \{x'_i, t_i, f_i\}$, for all $1 \le i \le n$. Moreover, for every clause C_j, one "large" C-edge F_j is defined that contains c^* and six further vertices, two for each literal of C_j:

- If C_j contains the positive literal x_i, then F_j contains x'_i and t_i.
- If C_j contains the negative literal $\neg x_i$, then F_j contains x'_i and f_i.

Since $H_1, \dots, H_n$ are disjoint edges and $x'_1, \dots, x'_n$ is a transversal of $\mathcal{H}$, we have $\nu_C(\mathcal{H}) = \tau_C(\mathcal{H}) = n$, and therefore $\alpha_C(\mathcal{H}) = 2n + 1$ holds.

(1) *If Φ is satisfiable, then $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$.*

Assume a truth assignment $a : \{x_1, \dots, x_n\} \to \{\mathsf{T}, \mathsf{F}\}$ that satisfies each clause C_j of Φ. To obtain a $(2n+1)$-coloring, create the 2-element color class $\{x'_i, t_i\}$ if $a(x_i) = \mathsf{T}$, and $\{x'_i, f_i\}$ if $a(x_i) = \mathsf{F}$, for each i $(1 \le i \le n)$. The remaining vertices are singleton color classes. It is clear that the hyperedges H_i are properly colored and, since every clause C_j is satisfied by a, every C-edge F_j contains a monochromatic pair.

(2) *If $\alpha_C(\mathcal{H}) = \bar{\chi}(\mathcal{H})$, then Φ is satisfiable.*

Consider a $\bar{\chi}$-coloring φ of $\mathcal{H}$. It uses exactly $|X| - n$ colors, consequently each of the disjoint C-edges $H_1, \dots, H_n$ contains a monochromatic vertex pair and the other color classes are singletons. If there occurs a monochromatic pair of the form $\{t_i, f_i\}$, it is not contained in any F_j, hence the coloring remains proper if it is replaced by $\{t_i, x_i'\}$. Thus, we can assume for all $1 \leq i \leq n$ that either $\varphi(x_i') = \varphi(t_i)$ or $\varphi(x_i') = \varphi(f_i)$ holds. Define the truth assignment a as $a(x_i) = \mathsf{T}$ if $\varphi(x_i') = \varphi(t_i)$, and $a(x_i) = \mathsf{F}$ if $\varphi(x_i') = \varphi(f_i)$. By assumption, every hyperedge F_j is properly colored by a monochromatic pair, which can be either (x_i, t_i) or (x_i, f_i) for an appropriate value of $i = i(j)$. Corresponding to this monochromatic pair, clause C_j is satisfied by the positive or negative literal of variable x_i, respectively. □

5 Concluding remarks

Open problems. We have characterized and efficiently colored the C-perfect members of a proper subclass of mixed hypertrees. The structure class considered is much wider than that of C-hypertrees, but the following problems remain open.

Problem 1 *Characterize the minimally C-imperfect mixed hypertrees.*

Problem 2 *Determine the algorithmic complexity of $\bar{\chi}$-coloring C-perfect mixed hypertrees.*

It is quite interesting to compare Theorem 2 and Corollary 1. They say that coloring is easy, recognition is hard, but nevertheless there is a concise description of the class in terms of forbidden induced substructures. Such situations do not occur very frequently. The open question is whether or not they extend to Problems 1 and 2, too.

Necessary or sufficient conditions. Our constructions show that at least some variant of the $\mathcal{D}_2$-degree condition is required for deriving the conclusion of Theorem 1. Although this is not necessarily the case for Theorem 4, the present version of Algorithm 2 certainly needs such an assumption. For instance, if the algorithm is applied to the hypertree $\mathcal{H}_{12}$ constructed in Section 3, the procedure terminates with neither a $\bar{\chi}$-coloring nor an induced polystar.

C-perfectness. We think that there is some hope to characterize C-perfect r-uniform C-hypergraphs, although it has not yet been completed even for $r = 3$. But the general characterization problem of C-perfect (or that of minimally C-imperfect) mixed hypergraphs appears to be rather complicated; Proposition 2 may be viewed as an indication in this direction.

Robust algorithms. It is worth comparing Theorems 1 and 4 in the context of Cameron and Edmonds's work [5]. Generally speaking, a "non-robust" algorithm over a structure class $\mathcal{S}$ determines efficiently an easily recognizable solution for each member of $\mathcal{S}$. With the terminology of [12], a "robust algorithm" gives more: it admits inputs from an efficiently recognizable wider class $\mathcal{S}^+ \supset \mathcal{S}$ and, in polynomial time, for each member of $\mathcal{S}^+$, it either determines a solution or presents an efficiently recognizable certificate showing that the input is in $\mathcal{S}^+ \setminus \mathcal{S}$. While both methods prove the "existentially polytime theorem" that every member of $\mathcal{S}$ admits a solution, a robust algorithm is definitely stronger, even though it does not necessarily recognize the class $\mathcal{S}$.

In some cases it is immediate to extend a non-robust algorithm to a robust one. For instance, the independence number of a bipartite graph can be determined in polynomial time, and in unrestricted input graphs one can efficiently find either an odd cycle or a largest independent set. As noted in [5], however, the comparison of chromatic number χ and clique number ω is much more complicated: currently there is no approach known with fewer than hundreds of pages altogether that would (1) prove that every graph without odd holes and antiholes is perfect, and (2) find a clique and a coloring of the same size or (3) exhibit an odd hole or antihole. A robust algorithm would require (2) and (3), possibly without recognizing Berge graphs but still proving (1).

In our main results, $\mathcal{S}^+$ is the easily recognizable class of mixed hypertrees under the given $\mathcal{D}_2$-condition, and $\mathcal{S}$ consists of the C-perfect members of $\mathcal{S}^+$. Contrary to $\mathcal{S}^+$, however, the structures in $\mathcal{S}$ are hard to recognize, what makes it even more interesting that Algorithm 2 is robust. Putting it into another context, structurally our Theorem 2 provides an example where dropping the condition of C-perfectness transforms an intractable problem to a polynomial-time solvable one. This is just the opposite of what one expects typically. And it can never happen with graph perfectness: for any class $\mathcal{S}^+$ of graphs, if membership in $\mathcal{S} = \{G \in \mathcal{S}^+ \mid G$ is perfect$\}$ is hard to test, then so is membership in $\mathcal{S}^+$, too.

Acknowledgement

We are indebted to Vitaly Voloshin who kindly directed our attention to his conjecture on C-perfect C-hypertrees.

References

[1] C. Bazgan, Cs. Bujtás and Zs. Tuza, Approximability of the upper chromatic number of hypergraphs, Manuscript, 2010.

[2] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, North-Holland, 1989.

[3] Cs. Bujtás and Zs. Tuza, C-perfect hypergraphs, *J. Graph Theory* 64(2) (2010), 132–149.

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

[5] K. Cameron and J. Edmonds, Colouring some classes of perfect graphs robustly, *Electronic Notes in Discrete Math.* 22 (2005), 553–556.

[6] P. Duchet, Thesis. Université Paris 6, 1979. (See also in: Problèmes Combinatoires et Théorie des Graphes, Colloq. Internat., Orsay, 1976, CNRS, Paris, 1978, 117–118.)

[7] C. Flament, Hypergraphes arborés. *Discrete Math.* 21 (1978), 223–226.

[8] T. Gallai, unpublished. (Early 1950's.)

[9] D. Král', A counter-example to Voloshin's hypergraph co-perfectness conjecture, *Australas. J. Combin.* 27 (2003), 25–41.

[10] D. Král', J. Kratochvíl, A. Proskurowski and H.-J. Voss, Coloring mixed hypertrees, *Discrete Applied Math.* 154(4) (2006), 660–672 .

[11] J. B. Slater, A characterization of soft hypergraphs, *Canad. Math. Bull.* 21 (1978), 335–337.

[12] J. P. Spinrad, *Efficient Graph Representations*, Amer. Math. Soc., 2003.

[13] R. E. Tarjan and M. Yannakakis, Simple linear time algorithms to test chordability of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, *SIAM J. Comput.* 13 (1984), 566–579.

[14] V. I. Voloshin, On the upper chromatic number of a hypergraph, *Australas. J. Combin.* 11 (1995), 25–45.

[15] V. I. Voloshin, *Coloring Mixed Hypergraphs: Theory, Algorithms and Applications*, Fields Institute Monographs, Amer. Math. Soc., 2002.

(Received 23 Mar 2010)