General and Fractional Hypertree Decompositions: Hard and Easy Cases

Wolfgang Fischl
TU Wien
wolfgang.fischl@tuwien.ac.at

Georg Gottlob
TU Wien & University of Oxford
georg.gottlob@cs.ox.ac.uk

Reinhard Pichler
TU Wien
reinhard.pichler@tuwien.ac.at

ABSTRACT

Hypertree decompositions, as well as the more powerful generalized hypertree decompositions (GHDs), and the yet more general fractional hypertree decompositions (FHD) are hypergraph decomposition methods successfully used for answering conjunctive queries and for solving constraint satisfaction problems. Every hypergraph H has a width relative to each of these methods: its hypertree width $hw(H)$, its generalized hypertree width $ghw(H)$, and its fractional hypertree width $fhw(H)$, respectively. It is known that $hw(H) \leq k$ can be checked in polynomial time for fixed k, while checking $ghw(H) \leq k$ is NP-complete for $k \geq 3$. The complexity of checking $fhw(H) \leq k$ for a fixed k has been open for over a decade.

We settle this open problem by showing that checking $fhw(H) \leq k$ is NP-complete, even for $k = 2$. The same construction allows us to prove also the NP-completeness of checking $ghw(H) \leq k$ for $k = 2$. After that, we identify meaningful restrictions for which checking for bounded ghw or fhw becomes tractable.

ACM Reference Format:
Wolfgang Fischl, Georg Gottlob, and Reinhard Pichler. 2018. General and Fractional Hypertree Decompositions: Hard and Easy Cases. In *PODS'18: 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, June 10–15, 2018, Houston, TX, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3196959.3196962

1 INTRODUCTION AND BACKGROUND

Research Challenges Tackled. In this work we tackle computational problems on hypergraph decompositions, which play a prominent role for answering Conjunctive Queries (CQs) and solving Constraint Satisfaction Problems (CSPs), which we discuss below.

Many NP-hard graph-based problems become tractable for instances whose corresponding graphs have bounded treewidth. There are, however, many problems for which the structure of an instance is better described by a hypergraph than by a graph, for example, the above mentioned CQs and CSPs. Given that treewidth does not generalize hypergraph acyclicity[1], proper hypergraph decomposition methods have been developed, in particular, *hypertree decompositions (HDs)* [26], the more general *generalized hypertree decompositions (GHDs)* [26], and the yet more general *fractional hypertree decompositions (FHDs)* [30], and corresponding notions of width of a hypergraph H have been defined: the *hypertree width* $hw(H)$, the *generalized hypertree width* $ghw(H)$, and the *fractional hypertree width* $fhw(H)$, where for every hypergraph H, $fhw(H) \leq ghw(H) \leq hw(H)$ holds. Definitions are given in Section 2. A number of highly relevant hypergraph-based problems such as CQ-evaluation and CSP-solving become tractable for classes of instances of bounded hw, ghw, or, fhw. For each of the mentioned types of decompositions it would thus be useful to be able to recognize for each constant k whether a given hypergraph H has corresponding width at most k, and if so, to compute such a decomposition. More formally, for *decomposition* $\in$ {HD, GHD, FHD} and $k > 0$, we consider the following family of problems:

Check(*decomposition*, k)
input hypergraph $H = (V, E)$;
output *decomposition* of H of width $\leq k$ if it exists and answer 'no' otherwise.

As shown in [26], Check(HD, k) is in Ptime. However, little is known about Check(FHD, k). In fact, this has been an open problem since the 2006 paper [29], where Grohe and Marx state: "It remains an important open question whether there is a polynomial-time algorithm that determines (or approximates) the fractional hypertree width and constructs a corresponding decomposition." The 2014 journal version still mentions this as open and it is conjectured that the problem might be NP-hard. The open problem is restated in [46], where further evidence for the hardness of the problem is given by showing that "it is not expressible in monadic second-order logic whether a hypergraph has bounded (fractional, generalized) hypertree width". We will tackle this open problem here:

Research Challenge 1: Is Check(FHD, k) tractable?

Let us now turn to generalized hypertree decompositions. In [26] the complexity of Check(GHD, k) was stated as an open problem. In [27], it was shown that Check(GHD, k) is NP-complete for $k \geq 3$. For $k = 1$ the problem is trivially tractable because $ghw(H) = 1$ just means H is acyclic. However the case $k = 2$ has been left open. This case is quite interesting, because it was observed that the majority of practical queries from various benchmarks that are not acyclic have $ghw = 2$ [10, 22], and that a decomposition in such cases can be very helpful. Our second research goal is to finally settle the complexity of Check(GHD, k) completely.

Research Challenge 2: Is Check(GHD, 2) tractable?

For those problems which are known to be intractable, for example, Check(GHD, k) for $k \geq 3$, and for those others that will turn out to be intractable, we would like to find large islands of tractability that correspond to meaningful restrictions of the input hypergraph instances. Ideally, such restrictions should fulfill two main criteria: (i) they need to be *realistic* in the sense that they apply to a large

[1] We here refer to the standard notion of hypergraph acyclicity, as used in [48] and [20], where it is called α-acyclicity. This notion is more general than other types of acyclicity that have been introduced in the literature.

PODS'18, June 10–15, 2018, Houston, TX, USA

number of CQs and/or CSPs in real-life applications, and (ii) they need to be *non-trivial* in the sense that the restriction itself does not already imply bounded *hw*, *ghw*, or *fhw*. Trivial restrictions would be, for example, acyclicity or bounded treewidth. Hence, our third research problem is as follows:

Research Challenge 3: Find realistic, non-trivial restrictions on hypergraphs which entail the tractability of the $\textsc{Check}(decomp, k)$ problem for $decomp \in \{\text{GHD}, \text{FHD}\}$.

Where we do not achieve PTIME algorithms for the precise computation of a decomposition of optimal width, we would like to find tractable methods for achieving good approximations. Note that for GHDs, the problem of approximations is solved, since $ghw(H) \leq 3 \cdot hw(H) + 1$ holds for every hypergraph H [4]. In contrast, for FHDs, the best known polynomial-time approximation is cubic. More precisely, in [38], a polynomial-time algorithm is presented which, given a hypergraph H with $fhw(H) = k$, computes an FHD of width $O(k^3)$. We would like to find meaningful restrictions that guarantee significantly tighter approximations in polynomial time. This leads to the fourth research problem:

Research Challenge 4: Find realistic, non-trivial restrictions on hypergraphs which allow us to compute in PTIME good approximations of $fhw(k)$.

Background and Applications. Hypergraph decompositions have meanwhile found their way into commercial database systems such as LogicBlox [6, 9, 35, 36, 42] and advanced research prototypes such as EmptyHeaded [1, 2, 45]. Moreover, since CQs and CSPs of bounded hypertree width fall into the highly parallelizable complexity class LogCFL, hypergraph decompositions have also been discovered as a useful tool for parallel query processing with MapReduce [5]. Hypergraph decompositions, in particular, HDs and GHDs have been used in many other contexts, e.g., in combinatorial auctions [25] and automated selection of Web services based on recommendations from social networks [34]. There exist exact algorithms for computing the generalized or fractional hypertree width [41]; clearly, they require exponential time even if the optimal width is bounded by some fixed k.

CQs are the most basic and arguably the most important class of queries in the database world. Likewise, CSPs constitute one of the most fundamental classes of problems in Artificial Intelligence. Formally, CQs and CSPs are the same problem and correspond to first-order formulae using $\{\exists, \wedge\}$ but disallowing $\{\forall, \vee, \neg\}$ as connectives, that need to be evaluated over a set of finite relations: the *database relations* for CQs, and the *constraint relations* for CSPs. In practice, CQs have often fewer conjuncts (query atoms) and larger relations, while CSPs have more conjuncts but smaller relations. Unfortunately, these problems are well-known to be NP-complete [12]. Consequently, there has been an intensive search for tractable fragments of CQs and/or CSPs over the past decades. For our work, the approaches based on decomposing the structure of a given CQ or CSP are most relevant, see e.g. [8, 13–17, 24, 26, 28, 30–33, 37, 39, 40]. The underlying structure of both is nicely captured by hypergraphs. The hypergraph $H = (V(H), E(H))$ underlying a CQ (or a CSP) Q has as vertex set $V(H)$ the set of variables occurring in Q; moreover, for every atom in Q, $E(H)$ contains a hyperedge consisting of all variables occurring in this atom. From now on, we shall mainly talk about hypergraphs with the understanding that all our results are equally applicable to CQs and CSPs.

Main Results. First of all, we have investigated the above mentioned open problem concerning the recognizability of $fhw \leq k$ for fixed k. Our initial hope was to find a simple adaptation of the NP-hardness proof in [27] for recognizing $ghw(H) \leq k$, for $k \geq 3$. Unfortunately, this proof dramatically fails for the fractional case. In fact, the hypergraph-gadgets in that proof are such that both "yes" and "no" instances may yield the same *fhw*. However, via crucial modifications, including the introduction of novel gadgets, we succeed to construct a reduction from 3SAT that allows us to control the *fhw* of the resulting hypergraphs such that those hypergraphs arising from "yes" 3SAT instances have $fhw(H) = 2$ and those arising from "no" instances have $fhw(H) > 2$. Surprisingly, thanks to our new gadgets, the resulting proof is actually significantly simpler than the NP-hardness proof for recognizing $ghw(H) \leq k$ in [27]. We thus obtain the following result solving a long standing open problem:

Main Result 1: Deciding $fhw(H) \leq 2$ for hypergraphs H is NP-complete, and $\textsc{Check}(\text{FHD}, k)$ is intractable even for $k = 2$.

This result can be extended to the NP-hardness of recognizing $fhw(H) \leq k$ for arbitrarily large $k \geq 2$. Moreover, the same construction can be used to prove that recognizing ghw ≤ 2 is also NP-hard, thus killing two birds with one stone.

Main Result 2: Deciding $ghw(H) \leq 2$ for hypergraphs H is NP-complete, and $\textsc{Check}(\text{GHD}, 2)$ is intractable even for $k = 2$.

The Main Results 1 and 2 are presented in Section 3. These results close some smoldering open problems with bad news. We thus further concentrate on Research Challenges 3 and 4 in order to obtain some positive results for restricted hypergraph classes.

We first study GHDs, where we succeed to identify very general, realistic, and non-trivial restrictions that make the $\textsc{Check}(\text{GHD}, k)$ problem tractable. These results are based on new insights about the differences of GHDs and HDs and the introduction of a novel technique for expanding a hypergraph H to an edge-augmented hypergraph H' s.t. the width k GHDs of H correspond to the width k HDs of H'. The crux here is to find restrictions under which only a polynomial number of edges needs to be added to H to obtain H'. The HDs of H' can then be computed in polynomial time.

In particular, we concentrate on the *bounded edge intersection property (BIP)*, which, for a class $\mathscr{C}$ of hypergraphs requires that for some constant i, for each pair of distinct edges e_1 and e_2 of each hypergraph $H \in \mathscr{C}$, $|e_1 \cap e_2| \leq i$, and its generalization, the *bounded multi-intersection property (BMIP)*, which, informally, requires that for some constant c any intersection of c distinct hyperedges of H has at most i elements for some constant i. In [22] we report tests on a large number of known CQ and CSP benchmarks and it turns out that a very large number of instances coming from real-life applications enjoy the BIP and a yet more overwhelming number enjoys the BMIP for very low constants c and i. We obtain the following good news, which are presented in Section 4.

Main Result 3: For classes of hypergraphs fulfilling the BIP or BMIP, for every constant k, the problem $\textsc{Check}(\text{GHD}, k)$ is tractable. Tractability holds even for classes $\mathscr{C}$ of hypergraphs where for some constant c all intersections of c distinct edges of every $H \in \mathscr{C}$ of size n have $O(\log n)$ elements. Our complexity analysis reveals that the problem CHECK(GHD, k) is fixed-parameter tractable w.r.t. parameter i of the BIP.

The tractability proofs for BIP and BMIP do not directly carry over to the fractional case. However, by adding a further restriction to the BIP, we also manage to identify an interesting tractable fragment for recognizing $fhw(H) \leq k$. To this end, we consider the degree d of a hypergraph $H = (V(H), E(H))$, which is defined as the maximum number of hyperedges in which a vertex occurs, i.e., $d = \max_{v \in V(H)} |\{e \in E(H) \mid v \in E(H)\}|$. We say that a class $\mathscr{C}$ of hypergraphs has bounded degree, if there exists $d \geq 1$, such that every hypergraph $H \in \mathscr{C}$ has degree $\leq d$. We obtain the following result, which is presented in Section 5.

Main Result 4: For classes of hypergraphs fulfilling the BIP and having bounded degree, for every constant k, the problem CHECK(FHD, k) is tractable.

To get yet bigger tractable classes, we also consider approximations of an optimal FHD. Towards this goal, we establish an interesting connection between the BIP and BMIP on the one hand and the Vapnik–Chervonenkis dimension (VC-dimension) of a hypergraph on the other hand. Our research, presented in Section 6 is summarized as follows.

Main Result 5: For rather general, realistic, and non-trivial hypergraph restrictions, there exist PTIME algorithms that, for hypergraphs H with $fhw(H) = k$, where k is a constant, produce FHDs whose widths are significantly smaller than the best previously known approximation. In particular, the BIP, the BMIP, or bounded VC-dimension allow us to compute an FHD whose width is $O(k \log k)$.

An online version of this paper [21] contains full proofs and a short summary of [22].

2 PRELIMINARIES

2.1 Hypergraphs

A *hypergraph* is a pair $H = (V(H), E(H))$, consisting of a set $V(H)$ of *vertices* and a set $E(H)$ of *hyperedges* (or, simply *edges*), which are non-empty subsets of $V(H)$. We assume that hypergraphs do not have isolated vertices, i.e. for each $v \in V(H)$, there is at least one edge $e \in E(H)$, s.t. $v \in e$. For a set $C \subseteq V(H)$, we define $edges(C) = \{e \in E(H) \mid e \cap C \neq \emptyset\}$ and for a set $E \subseteq E(H)$, we define $V(E) = \{v \in e \mid e \in E\}$.

For a hypergraph H and a set $V \subseteq V(H)$, we say that a pair of vertices $v_1, v_2 \in V(H)$ is $[V]$-adjacent if there exists an edge $e \in E(H)$ such that $\{v_1, v_2\} \subseteq (e \setminus V)$. A $[V]$-path π from v to v' consists of a sequence $v = v_0, \ldots, v_h = v'$ of vertices and a sequence of edges $e_0, \ldots, e_{h-1}$ ($h \geq 0$) such that $\{v_i, v_{i+1}\} \subseteq (e_i \setminus V)$, for each $i \in [0 \ldots h-1]$. We denote by $V(\pi)$ the set of vertices occurring in the sequence $v_0, \ldots, v_h$. Likewise, we denote by $edges(\pi)$ the set of edges occurring in the sequence $e_0, \ldots, e_{h-1}$. A set $W \subseteq V(H)$ of vertices is $[V]$-connected if $\forall v, v' \in W$ there is a $[V]$-path from v to v'. A $[V]$-component is a maximal $[V]$-connected, non-empty set of vertices $W \subseteq V(H) \setminus V$.

2.2 (Fractional) Edge Covers

Let $H = (V(H), E(H))$ be a hypergraph and consider functions $\lambda\colon E(H) \to \{0, 1\}$ and $\gamma\colon E(H) \to [0, 1]$. Then, we denote by $B(\theta)$ the set of all vertices covered by θ:

$$B(\theta) = \left\{ v \in V(H) \mid \sum_{e \in E(H),\, v \in e} \theta(e) \geq 1 \right\},$$

where $\theta \in \{\lambda, \gamma\}$. The weight of function θ is defined as

$$weight(\theta) = \sum_{e \in E(H)} \theta(e).$$

Following [26], we will sometimes consider λ as a set with $\lambda \subseteq E(H)$ (i.e., the set of edges e with $\lambda(e) = 1$) and the weight as the cardinality of such a set. However, for the sake of a uniform treatment with function γ, we shall prefer to treat λ as a function.

DEFINITION 2.1. *An* edge cover *(EC) of a hypergraph H is a function* $\lambda : E(H) \to \{0, 1\}$ *such that* $V(H) = B(\lambda)$. *The* edge cover number $\rho(H)$*is the minimum weight of all edge covers of H.*

Note that the edge cover number can be calculated by the following integer linear program (ILP).

$$\begin{aligned} \text{minimize:} \quad & \sum_{e \in E(H)} \lambda(e) \\ \text{subject to:} \quad & \sum_{e \in E(H),\, v \in e} \lambda(e) \geq 1, \quad \text{for all } v \in V(H) \\ & \lambda(e) \in \{0, 1\} \qquad \text{for all } e \in E(H) \end{aligned}$$

By substitung all $\lambda(e)$ by $\gamma(e)$ and by relaxing the last condition of the ILP above, we arrive at the linear program (LP) for computing the fractional edge cover number. Actually, we substitute the last condition by $\gamma(e) \geq 0$. Note that even though our weight function is defined to take values between 0 and 1, we do not need to add $\gamma(e) \leq 1$ as a constraint, because implicitly by the minimization itself the weight on an edge for an edge cover is never greater than 1. Also note that now the program above is an LP, which can be solved in PTIME, whereas finding an edge cover of weight $\leq k$ is NP-complete if k is not fixed.

DEFINITION 2.2. *A* fractional edge cover *(FEC) of a hypergraph* $H = (V(H), E(H))$ *is a function* $\gamma : E(H) \to [0, 1]$ *such that* $V(H) = B(\gamma)$. *The* fractional edge cover number $\rho^*(H)$ *of H is the minimum weight of all fractional edge covers of H. We write supp*(γ) *to denote the* support *of* γ, *i.e., supp*$(\gamma) := \{e \in E(H) \mid \gamma(e) > 0\}$.

Clearly, we have $\rho^*(H) \leq \rho(H)$ for every hypergraph H, and $\rho^*(H)$ can be much smaller than $\rho(H)$. However, below we give an example, which is important for our proof of Theorem 3.1 and where $\rho^*(H)$ and $\rho(H)$ coincide.

LEMMA 2.1. *Let* K_{2n} *be a clique of size* $2n$. *Then the equalities* $\rho(K_{2n}) = \rho^*(K_{2n}) = n$ *hold.*

PROOF. Since we have to cover each vertex with weight ≥ 1, the total weight on the vertices of the graph is $\geq 2n$. As the weight of each edge adds to the weight of at most 2 vertices, we need at least weight n on the edges to achieve $\geq 2n$ weight on the vertices. On the other hand, we can use n edges each with weight 1 to cover $2n$ vertices. Hence, in total, we get $n \leq \rho^*(K_{2n}) \leq \rho(K_{2n}) \leq n$. ◻

2.3 HDs, GHDs, and FHDs

We now define three types of hypergraph decompositions:

Definition 2.3. *A* generalized hypertree decomposition *(GHD) of a hypergraph $H = (V(H), E(H))$ is a tuple $\langle T, (B_u)_{u \in N(T)}, (\lambda)_{u \in N(T)} \rangle$, such that $T = \langle N(T), E(T) \rangle$ is a rooted tree and the following conditions hold:*

1. *for each $e \in E(H)$, there is a node $u \in N(T)$ with $e \subseteq B_u$;*
2. *for each $v \in V(H)$, the set $\{u \in N(T) \mid v \in B_u\}$ is connected in T;*
3. *for each $u \in N(T)$, λ_u is a function $\lambda_u : E(H) \to \{0, 1\}$ with $B_u \subseteq B(\lambda_u)$.*

Let us clarify some notational conventions used throughout this paper. To avoid confusion, we will consequently refer to the elements in $V(H)$ as *vertices* (of the hypergraph) and to the elements in $N(T)$ as the *nodes* of T (of the decomposition). For a node u in T, we write T_u to denote the subtree of T rooted at u. By slight abuse of notation, we will often write $u' \in T_u$ to denote that u' is a node in the subtree T_u of T. Further, we define $V(T_u) := \bigcup_{u' \in T_u} B_{u'}$ and, for a set $V' \subseteq V(H)$, we define $nodes(V', \mathcal{F}) = \{u \in T \mid B_u \cap V' \neq \emptyset\}$.

Definition 2.4. *A* hypertree decomposition *(HD) of a hypergraph $H = (V(H), E(H))$ is a GHD, which in addition also satisfies the following condition:*

(4) for each $u \in N(T)$, $V(T_u) \cap B(\lambda_u) \subseteq B_u$

Definition 2.5. *A* fractional hypertree decomposition *(FHD) [30] of a hypergraph $H = (V(H), E(H))$ is a tuple $\langle T, (B_u)_{u \in N(T)}, (\gamma)_{u \in N(T)} \rangle$, where conditions (1) and (2) of Definition 2.3 plus condition (3') hold:*

(3') for each $u \in N(T)$, γ_u is a function $\gamma_u : E(H) \to [0, 1]$ with $B_u \subseteq B(\gamma_u)$.

The width of a GHD, HD, or FHD is the maximum weight of the functions λ_u or γ_u, resp., over all nodes u in T. Moreover, the generalized hypertree width, hypertree width, and fractional hypertree width of H (denoted $ghw(H)$, $hw(H)$, $fhw(H)$) is the minimum width over all GHDs, HDs, and FHDs of H, resp. Condition (2) is called the "connectedness condition", and condition (4) is referred to as "special condition" [26]. The set B_u is often referred to as the "bag" at node u. Note that, strictly speaking, only HDs require that the underlying tree T be rooted. For the sake of a uniform treatment we assume that also the tree underlying a GHD or an FHD is rooted (with the understanding that the root is arbitrarily chosen).

We now recall two fundamental properties of the various notions of decompositions and width.

Lemma 2.2. *Let H be a hypergraph and let H' be a vertex induced subhypergraph of H, then $hw(H') \leq hw(H)$, $ghw(H') \leq ghw(H)$, and $fhw(H') \leq fhw(H)$ hold.*

Lemma 2.3. *Let H be a hypergraph. If H has a subhypergraph H' such that H' is a clique, then every HD, GHD, or FHD of H has a node u such that $V(H') \subseteq B_u$.*

Strictly speaking, Lemma 2.3 is a well-known property of tree decompositions – independently of the λ- or γ-label.

Last, we define the notion of *full* nodes. Intuitively, a node u is called full in a decomposition if it is not possible to add to the bag B_u a new vertex v without increasing the width of the decomposition.

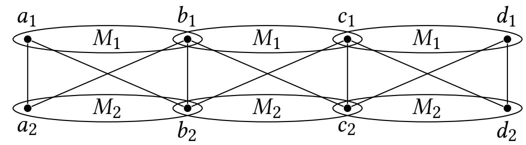

Figure 1: Basic structure of H_0 in Lemma 3.1

Definition 2.6. *Let $\mathcal{F} = \langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T} \rangle$ be an FHD of H of width $\leq k$, then a node u in T is said to be* full *in $\mathcal{F}$ (or simply* full*, if $\mathcal{F}$ is understood from the context), if for any vertex $v \in V(H) \setminus B_u$ it is the case that $\rho^*(B(\gamma_u) \cup v) > k$.*

3 NP-HARDNESS

The main result in this section is the NP-hardness of Check($decomp, k$) with $decomp \in$ {GHD, FHD} and $k = 2$. At the core of the NP-hardness proof is the construction of a hypergraph H with certain properties. The gadget in Figure 1 will play an integral part of this construction.

Lemma 3.1. *Let M_1, M_2 be disjoint sets and $M = M_1 \cup M_2$. Let $H = (V(H), E(H))$ be a hypergraph and $H_0 = (V_0, E_A \cup E_B \cup E_C)$ a subhypergraph of H with $V_0 = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\} \cup M$ and*

$$E_A = \{\{a_1, b_1\} \cup M_1, \{a_2, b_2\} \cup M_2, \{a_1, b_2\}, \{a_2, b_1\}, \{a_1, a_2\}\}$$
$$E_B = \{\{b_1, c_1\} \cup M_1, \{b_2, c_2\} \cup M_2, \{b_1, c_2\}, \{b_2, c_1\}, \{b_1, b_2\}, \{c_1, c_2\}\}$$
$$E_C = \{\{c_1, d_1\} \cup M_1, \{c_2, d_2\} \cup M_2, \{c_1, d_2\}, \{c_2, d_1\}, \{d_1, d_2\}\}$$

where no element from the set $R = \{a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$ occurs in any edge of $E(H) \setminus (E_A \cup E_B \cup E_C)$. Then, every FHD $\mathcal{F} = \langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T} \rangle$ of width ≤ 2 of H has nodes u_A, u_B, u_C s.t.:

- *$\{a_1, a_2, b_1, b_2\} \subseteq B_{u_A}$,*
- *$\{b_1, b_2, c_1, c_2\} \cup M \subseteq B_{u_B}$,*
- *$\{c_1, c_2, d_1, d_2\} \subseteq B_{u_C}$, and*
- *u_B is on the path from u_A to u_C.*

Proof Idea. The hypergraph H_0 is depicted in Figure 1. Note that H_0 contains 3 cliques of size 4, namely $\{a_1, a_2, b_1, b_2\}$, $\{b_1, b_2, c_1, c_2\}$, and $\{c_1, c_2, d_1, d_2\}$. The lemma makes heavy use of the connectedness condition and of the fact that a clique of size 4 can only be covered by a fractional edge cover of weight ≥ 2. □

Theorem 3.1. *The Check($decomp, k$) problem is* NP*-complete for $decomp \in$ {GHD, FHD} and $k = 2$.*

Proof Sketch. The problem is clearly in NP: guess a tree decomposition and check in polynomial time for each node u whether $\rho(B_u) \leq 2$ or $\rho^*(B_u) \leq 2$, respectively, holds. The NP-hardness is proved by a reduction from 3SAT. Before presenting this reduction, we first introduce some useful notation.

Notation. For $i, j \geq 1$, we denote $\{1, \ldots, i\} \times \{1, \ldots, j\}$ by $[i; j]$. For each $p \in [i; j]$, we denote by $p \oplus 1$ ($p \ominus 1$) the successor (predecessor) of p in the usual lexicographic order on pairs, that is, the order $(1, 1), \ldots, (1, j), (2, 1), \ldots, (i, 1), \ldots, (i, j)$. We refer to the first element $(1, 1)$ as min and to the last element (i, j) as max. We denote by $[i; j]^-$ the set $[i; j] \setminus \{\max\}$, i.e. $[i; j]$ without the last element.

Now let $\varphi = \bigwedge_{j=1}^{m}(L_j^1 \vee L_j^2 \vee L_j^3)$ be an arbitrary instance of 3SAT with m clauses and variables $x_1, \ldots, x_n$. From this we will construct a hypergraph $H = (V(H), E(H))$, which consists of two copies H_0, H_0' of the (sub-)hypergraph H_0 of Lemma 3.1 plus additional edges connecting H_0 and H_0'. We use the sets $Y = \{y_1, \ldots, y_n\}$ and $Y' = \{y_1', \ldots, y_n'\}$ to encode the truth values of the variables of φ. We denote by Y_l (Y_l') the set $Y \setminus \{y_l\}$ ($Y' \setminus \{y_l'\}$). Furthermore, we use the sets $A = \{a_p \mid p \in [2n+3; m]\}$ and $A' = \{a_p' \mid p \in [2n+3; m]\}$, and we define the following subsets of A and A', respectively:

$$A_p = \{a_{\min}, \ldots, a_p\} \qquad \overline{A_p} = \{a_p, \ldots, a_{\max}\}$$
$$A_p' = \{a_{\min}', \ldots, a_p'\} \qquad \overline{A_p'} = \{a_p', \ldots, a_{\max}'\}$$

In addition, we will use another set S of elements, that controls and restricts the ways in which edges are combined in a possible FHD. Such an FHD will have, implied by Lemma 3.1, two nodes u_B and u_B' such that $S \subseteq B_{u_B}$ and $S \subseteq B_{u_B'}$. From this, we will reason on the path connecting u_B and u_B'.

The concrete set S used in our construction of H is obtained as follows. Let $Q = [2n+3; m] \cup \{(0,1), (0,0), (1,0)\}$, hence Q is an extension of the set $[2n+3; m]$ with special elements $(0,1), (0,0), (1,0)$. We define S as follows:

$$S = Q \times \{1,2,3\} \times \{0,1\}.$$

An element in this set will be denoted by $(q \mid k, \tau)$, thereby we split the 3 items into 2 groups. Recall that the values $q \in Q$ are themselves pairs of integers (i, j). Intuitively, q indicates the position of a node on the "long" path π in the desired FHD or GHD. The integer k refers to a literal in the j-th clause while the values 0 and 1 of τ will be used to indicate "complementary" edges of hypergraph H in a sense to be made precise later (see Definition 3.1). We will write the wildcard $*$ to indicate that a component in some element of S can take an arbitrary value. If both k and τ may take arbitrary values, then we will use the single symbol $\circledast$ as a shorthand for $*, *$. For example, $(\min \mid \circledast)$ denotes the set of tuples $(q \mid k, \tau)$ where $q = \min = (1,1)$ and the pair (k, τ) can take an arbitrary value in $\{1,2,3\} \times \{0,1\}$. We will denote by S_p the set $(p \mid \circledast)$. For instance, $(\min \mid \circledast)$ will be denoted as $S_{\min}$. Further, for $p \in [2n+3; m]$, $k \in \{1,2,3\}$, and $\tau \in \{0,1\}$, we define singleton sets $S_p^{k,\tau} = \{(p \mid k, \tau)\}$.

Problem reduction. Let $\varphi = \bigwedge_{j=1}^{m}(L_j^1 \vee L_j^2 \vee L_j^3)$ be an arbitrary instance of 3SAT with m clauses and variables $x_1, \ldots, x_n$. From this we construct a hypergraph $H = (V(H), E(H))$ i.e., an instance of Check($decomp, k$) with $decomp \in$ {GHD, FHD} and $k = 2$.

We start by defining the vertex set $V(H)$:

$$V(H) = S \cup A \cup A' \cup Y \cup Y' \cup \{z_1, z_2\} \cup \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, a_1', a_2', b_1', b_2', c_1', c_2', d_1', d_2'\}.$$

The edges of H are defined in 3 steps. First, we take two copies of the subhypergraph H_0 used in Lemma 3.1:

- Let $H_0 = (V_0, E_0)$ be the hypergraph of Lemma 3.1 with $V_0 = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\} \cup M_1 \cup M_2$ and $E_0 = E_A \cup E_B \cup E_C$, where we set $M_1 = S \setminus S_{(0,1)} \cup \{z_1\}$ and $M_2 = Y \cup S_{(0,1)} \cup \{z_2\}$.
- Let $H_0' = (V_0', E_0')$ be the corresponding hypergraph, with $V_0' = \{a_1', a_2', b_1', b_2', c_1', c_2', d_1', d_2'\} \cup M_1' \cup M_2'$ and E_A', E_B', E_C' are the primed versions of the egde sets $M_1' = S \setminus S_{(1,0)} \cup \{z_1\}$ and $M_2' = Y' \cup S_{(1,0)} \cup \{z_2\}$.

In the second step, we define the edges which (as we will see) enforce the existence of a "long" path π between the nodes covering H_0 and the nodes covering H_0' in any GHD or FHD.

- $e_p = A_p' \cup \overline{A_p}$, for $p \in [2n+3; m]^-$,
- $e_{y_i} = \{y_i, y_i'\}$, for $1 \leq i \leq n$,
- For $p = (i,j) \in [2n+3; m]^-$ and $k \in \{1,2,3\}$:

$$e_p^{k,0} = \begin{cases} \overline{A_p} \cup (S \setminus S_p^{k,1}) \cup Y \cup \{z_1\} & \text{if } L_j^k = x_l \\ \overline{A_p} \cup (S \setminus S_p^{k,1}) \cup Y_l \cup \{z_1\} & \text{if } L_j^k = \neg x_l, \end{cases}$$

$$e_p^{k,1} = \begin{cases} A_p' \cup S_p^{k,1} \cup Y_l' \cup \{z_2\} & \text{if } L_j^k = x_l \\ A_p' \cup S_p^{k,1} \cup Y' \cup \{z_2\} & \text{if } L_j^k = \neg x_l. \end{cases}$$

Finally, we need edges that connect H_0 and H_0' with the above edges covered by the nodes of the "long" path π in a GHD or FHD:

- $e_{(0,0)}^0 = \{a_1\} \cup A \cup S \setminus S_{(0,0)} \cup Y \cup \{z_1\}$
- $e_{(0,0)}^1 = S_{(0,0)} \cup Y' \cup \{z_2\}$
- $e_{\max}^0 = S \setminus S_{\max} \cup Y \cup \{z_1\}$
- $e_{\max}^1 = \{a_1'\} \cup A' \cup S_{\max} \cup Y' \cup \{z_2\}$

This concludes the construction of the hypergraph H. In Appendix A, we provide Example A.1, which will help to illustrate the intuition underlying this construction.

To prove the correctness of our problem reduction, we have to show the two equivalences: first, that $ghw(H) \leq 2$ if and only if φ is satisfiable and second, that $fhw(H) \leq 2$ if and only if φ is satisfiable. We prove the two directions of these equivalences separately.

Proof of the "if"-direction. We will first assume that φ is satisfiable. It suffices to show that then H has a GHD of width ≤ 2, because $fhw(H) \leq ghw(H)$ holds. Let σ be a satisfying truth assignment. Let us fix for each $j \leq m$, some $k_j \in \{1,2,3\}$ such that $\sigma(L_j^{k_j}) = 1$. By l_j, we denote the index of the variable in the literal $L_j^{k_j}$, that is, $L_j^{k_j} = x_{l_j}$ or $L_j^{k_j} = \neg x_{l_j}$. For $p = (i,j)$, let k_p refer to k_j and let $L_p^{k_p}$ refer to $L_j^{k_j}$. Finally, we let Z be the set $\{y_i \mid \sigma(x_i) = 1\} \cup \{y_i' \mid \sigma(x_i) = 0\}$.

A GHD $\mathcal{G} = \langle T, (B_u)_{u \in T}, (\lambda_u)_{u \in T} \rangle$ of width 2 for H is constructed as follows. T is a path u_C, u_B, u_A, $u_{\min \ominus 1}$, $u_{\min}, \ldots, u_{\max}$, u_A', u_B', u_C'. The construction is illustrated in Figure 2. The precise definition of B_u and λ_u is given in Table 1. Clearly, the GHD has width ≤ 2. We now show that $\mathcal{G}$ is indeed a GHD of H:

(1) For each edge $e \in E$, there is a node $u \in T$, such that $e \subseteq B_u$:
   - $\forall e \in E_X : e \subseteq B_{u_X}$ for all $X \in \{A, B, C\}$,
   - $\forall e' \in E_X' : e' \subseteq B_{u_X'}$ for all $X \in \{A, B, C\}$,
   - $e_p \subseteq B_{u_p}$ for $p \in [2n+3; m]^-$,
   - $e_{y_i} \subseteq B_{u_{\min \ominus 1}}$ or $e_{y_i} \subseteq B_{u_{\max}}$ depending on Z,
   - $e_p^{k,0} \subseteq B_{u_{\min \ominus 1}}$ for $p \in [2n+3; m]^-$,
   - $e_p^{k,1} \subseteq B_{u_{\max}}$ for $p \in [2n+3; m]^-$,
   - $e_{(0,0)}^0 \subseteq B_{u_{\min \ominus 1}}$, $e_{(0,0)}^1 \subseteq B_{u_{\max}}$,
   - $e_{\max}^0 \subseteq B_{u_{\min \ominus 1}}$ and $e_{\max}^1 \subseteq B_{u_{\max}}$.

   All of the above inclusions can be verified in Table 1.

(2) For each vertex $v \in V$, the set $\{u \in T \mid v \in B_u\}$ induces a connected subtree of T, which is easy to verify in Table 1.

(3) For each $u \in T$, $B_u \subseteq B(\lambda_u)$: The only inclusion which cannot be easily verified in Table 1 is $B_{u_p} \subseteq B(\lambda_{u_p})$. In fact,

$u \in T$	B_u	λ_u
u_C	$\{d_1, d_2, c_1, c_2\} \cup Y \cup S \cup \{z_1, z_2\}$	$\{c_1, d_1\} \cup M_1$, $\{c_2, d_2\} \cup M_2$
u_B	$\{c_1, c_2, b_1, b_2\} \cup Y \cup S \cup \{z_1, z_2\}$	$\{b_1, c_1\} \cup M_1$, $\{b_2, c_2\} \cup M_2$
u_A	$\{b_1, b_2, a_1, a_2\} \cup Y \cup S \cup \{z_1, z_2\}$	$\{a_1, b_1\} \cup M_1$, $\{a_2, b_2\} \cup M_2$
$u_{\min \ominus 1}$	$\{a_1\} \cup A \cup Y \cup S \cup Z \cup \{z_1, z_2\}$	$e^0_{(0,0)}, e^1_{(0,0)}$
$u_{p \in [2n+3;m]^-}$	$A'_p \cup \overline{A_p} \cup S \cup Z \cup \{z_1, z_2\}$	$e_p^{k_p,0}, e_p^{k_p,1}$
$u_{\max}$	$\{a'_1\} \cup A' \cup Y' \cup S \cup Z \cup \{z_1, z_2\}$	$e^0_{\max}, e^1_{\max}$
u'_A	$\{a'_1, a'_2, b'_1, b'_2\} \cup Y' \cup S \cup \{z_1, z_2\}$	$\{a'_1, b'_1\} \cup M'_1$, $\{a'_2, b'_2\} \cup M'_2$
u'_B	$\{b'_1, b'_2, c'_1, c'_2\} \cup Y' \cup S \cup \{z_1, z_2\}$	$\{b'_1, c'_1\} \cup M'_1$, $\{b'_2, c'_2\} \cup M'_2$
u'_C	$\{c'_1, c'_2, d'_1, d'_2\} \cup Y' \cup S \cup \{z_1, z_2\}$	$\{c'_1, d'_1\} \cup M'_1$, $\{c'_2, d'_2\} \cup M'_2$

Table 1: Definition of B_u and λ_u for GHD of H.

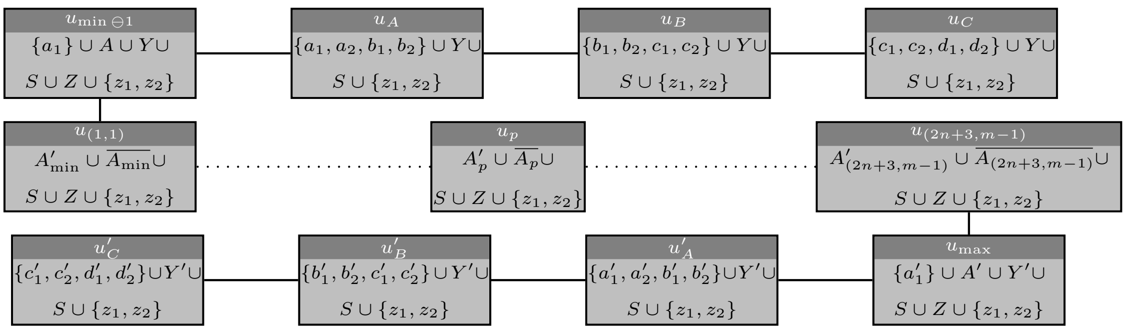

Figure 2: Intended path of the FHD of hypergraph H in the proof of Theorem 3.1

this is the only place in the proof where we make use of the assumption that φ is satisfiable. First, notice that the set $A'_p \cup \overline{A_p} \cup S \cup \{z_1, z_2\}$ is clearly a subset of $B(\lambda_{u_p})$. It remains to show that $Z \subseteq B(\lambda_{u_p})$. Assume that $L_p^{k_p} = x_{l_j}$, for some $p \in [2n+3; m]^-$. Thus, $\sigma(x_{l_j}) = 1$ and therefore $y'_{l_j} \notin Z$. But, by definition of $e_p^{k_p,0}$ and $e_p^{k_p,1}$, vertex y'_{l_j} is the only element of $Y \cup Y'$ not contained in $B(\lambda_{u_p})$. Since $Z \subseteq (Y \cup Y')$ and $y'_{l_j} \notin Z$, we have that $Z \subseteq B(\lambda_{u_p})$. It remains to consider the case $L_p^{k_p} = \neg x_{l_j}$, for some $p \in [2n+3; m]^-$. Thus, $\sigma(x_{l_j}) = 0$ and again $y_{l_j} \notin Z$. But, by definition of $e_p^{k_p,0}$ and $e_p^{k_p,1}$, vertex y_{l_j} is the only element of $Y \cup Y'$ not contained in $B(\lambda_{u_p})$. Since $Z \subseteq (Y \cup Y')$ and $y_{l_j} \notin Z$, we have that $Z \subseteq B(\lambda_{u_p})$.

Two crucial lemmas. Before we give a proof sketch of the "only if"-direction, we define the notion of complementary edges and state two important lemmas related to this notion.

Definition 3.1. *Let e and e' be two edges from the hypergraph H as defined before. We say e' is the* complementary *edge of e (or, simply, e, e' are complementary edges) whenever*

- *$e \cap S = S \setminus S'$ for some $S' \subseteq S$ and*
- *$e' \cap S = S'$.*

Observe that for every edge in our construction that covers $S \setminus S'$ for some $S' \subseteq S$ there is a complementary edge that covers S', for example $e_p^{k,0}$ and $e_p^{k,1}$, $e^0_{(0,0)}$ and $e^1_{(0,0)}$, and so on. In particular there is no edge that covers S completely. Moreover, consider arbitrary subsets S_1, S_2 of S, s.t. (syntactically) $S \setminus S_i$ is part of the definition of e_i for some $e_i \in E(H)$ with $i \in \{1, 2\}$. Then S_1 and S_2 are disjoint.

We now give two lemmas needed for the "only if"-direction.

Lemma 3.2. *Let $\mathcal{F} = \langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T} \rangle$ be an FHD of width ≤ 2 of the hypergraph H constructed above. For every node u with $S \cup \{z_1, z_2\} \subseteq B_u$ and every pair e, e' of complementary edges, it holds that $\gamma_u(e) = \gamma_u(e')$.*

Proof Sketch. First, we try to cover z_1 and z_2 with weight 2. To do this, we split the set of edges into disjoint sets $E^0 = \{e \in E(H) \mid z_1 \in e\}$ to cover z_1 and $E^1 = \{e \in E(H) \mid z_2 \in e\}$ to cover z_2 (no edge contains both z_1 and z_2). Then $\Sigma_{e \in E^0} \gamma_u(e) = 1$ and $\Sigma_{e \in E^1} \gamma_u(e) = 1$ must hold. An inspection of E^0 and E^1 shows that, in order to also cover S while not exceeding the weight of 2, every pair e, e' of complementary edges must satisfy $\gamma_u(e) = \gamma_u(e')$. ◻

Lemma 3.3. *Let $\mathcal{F} = \langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T} \rangle$ be an FHD of width ≤ 2 of the hypergraph H constructed above and let $p \in [2n+3; m]^-$. For every node u with $S \cup A'_p \cup \overline{A_p} \cup \{z_1, z_2\} \subseteq B_u$, the condition $\gamma_u(e) = 0$ holds for all edges e in $E(H)$ except for $e_p^{k,0}$ and $e_p^{k,1}$ with $k \in \{1, 2, 3\}$, i.e. the only way to cover $S \cup A'_p \cup \overline{A_p} \cup \{z_1, z_2\}$ with weight ≤ 2 is by using only edges $e_p^{k,0}$ and $e_p^{k,1}$ with $k \in \{1, 2, 3\}$.*

Proof Sketch. Now, in addition to the vertices to be covered in Lemma 3.2, also A'_p and $\overline{A_p}$ have to be covered. Similar as in the proof of Lemma 3.2, we identify sets of edges E^1_p and E^0_p able to cover A'_p and $\overline{A_p}$, respectively. Now, by Lemma 3.2, the only way to also cover $S \cup \{z_1, z_2\}$ is by using complementary edges, which in those sets are only the edges $e_p^{k,0}$ and $e_p^{k,1}$ with $k \in \{1,2,3\}$. ◻

Proof of the "only if"-direction. It remains to show that φ is satisfiable if H has a GHD (FHD) of width ≤ 2. Due to the inequality $fhw(H) \leq ghw(H)$, it suffices to show that φ is satisfiable if H has an FHD of width ≤ 2. For this, we let $\mathcal{F} = \langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T} \rangle$ be such an FHD. Let u_A, u_B, u_C and u'_A, u'_B, u'_C be the nodes that are guaranteed by Lemma 3.1 with M_i, M'_i as defined above. Recall that in the proof of Lemma 3.1 we observed that the nodes u_A, u_B, u_C and u'_A, u'_B, u'_C are full. We state several properties of the path connecting u_A and u'_A. The proofs of these claims can be found in Appendix B. They rely on Lemmas 3.2 and 3.3. Particularly, the proofs of Claims E, H and I use the fact that the same weight has to be put on complementary edges (Lemma 3.2) and that a total weight of 1 has to be put on the edges $e_p^{k,0}$ and $e_p^{k,1}$ with $k = \{1,2,3\}$.

Claim A. *The nodes u'_A, u'_B, u'_C (resp. u_A, u_B, u_C) are not on the path from u_A to u_C (resp. u'_A to u'_C).*

Claim B. *The following equality holds:*

$$nodes(A \cup A', \mathcal{F}) \cap \{u_A, u_B, u_C, u'_A, u'_B, u'_C\} = \emptyset.$$

We are now interested in the sequence of nodes $\hat{u}_i$ that cover the edges $e^0_{(0,0)}, e_{\min}, e_{\min \oplus 1}, \dots$. Before we formulate Claim C, it is convenient to introduce the following notation. To be able to refer to the edges $e^0_{(0,0)}, e_{\min}, e_{\min \oplus 1}, \dots, e_{\max \ominus 1}, e^1_{\max}$ in a uniform way, we use $e_{\min \ominus 1}$ as synonym of $e^0_{(0,0)}$ and $e_{\max}$ as synonym of $e^1_{\max}$. We thus get the natural order $e_{\min \ominus 1} < e_{\min} < e_{\min \oplus 1} < \dots < e_{\max \ominus 1} < e_{\max}$ on these edges.

Claim C. *The FHD $\mathcal{F}$ has a path containing nodes $\hat{u}_1, \dots, \hat{u}_N$ for some N, such that the edges $e_{\min \ominus 1}, e_{\min}, e_{\min \oplus 1}, \dots, e_{\max \ominus 1}, e_{\max}$ are covered in this order. More formally, there is a mapping $f : \{\min \ominus 1, \dots, \max\} \rightarrow \{1, \dots, N\}$, s.t.*

- *$\hat{u}_{f(p)}$ covers e_p and*
- *if $p < p'$ then $f(p) \leq f(p')$.*

By a path containing nodes $\hat{u}_1, \dots, \hat{u}_N$ *we mean that $\hat{u}_1$ and $\hat{u}_N$ are nodes in $\mathcal{F}$, such that the nodes $\hat{u}_2, \dots, \hat{u}_{N-1}$ lie (in this order) on the path from $\hat{u}_1$ to $\hat{u}_N$. Of course, the path from $\hat{u}_1$ to $\hat{u}_N$ may also contain further nodes, but we are not interested in whether they cover any of the edges e_p.*

So far we have shown, that there are three disjoint paths from u_A to u_C, from u'_A to u'_C and from $\hat{u}_1$ to $\hat{u}_N$, resp. It is easy to see, that u_A is closer to the path $\hat{u}_1, \dots, \hat{u}_N$ than u_B and u_C, since otherwise u_B and u_C would have to cover a_1 as well, which is impossible since they are full. The same also holds for u'_A. In the next claims we will argue that the path from u_A to u'_A goes through some $\hat{u}$ of the path from $\hat{u}_1$ to $\hat{u}_N$. For this we introduce the short-hand notation $\pi(\hat{u}_1, \hat{u}_N)$ for the path from $\hat{u}_1$ to $\hat{u}_N$. Next, we state some important properties of $\pi(\hat{u}_1, \hat{u}_N)$ and the path from u_A to u'_A.

Claim D. *In the FHD $\mathcal{F}$ of H of width ≤ 2 the path from u_A to u'_A has non-empty intersection with $\pi(\hat{u}_1, \hat{u}_N)$.*

Claim E. *In the FHD $\mathcal{F}$ of H of width ≤ 2 there are two distinguished nodes $\hat{u}$ and $\hat{u}'$ in the intersection of the path from u_A to u'_A with $\pi(\hat{u}_1, \hat{u}_N)$, s.t. $\hat{u}$ is the node closer to u_A than to u'_A. Then, $\hat{u}$ is closer to $\hat{u}_1$ than to $\hat{u}_N$.*

Claim F. *In the FHD $\mathcal{F}$ of H of width ≤ 2 the path $\pi(\hat{u}_1, \hat{u}_N)$ has at least 3 nodes $\hat{u}_i$, i.e., $N \geq 3$.*

Claim G. *In the FHD $\mathcal{F}$ of H of width ≤ 2 all the nodes $\hat{u}_2, \dots, \hat{u}_{N-1}$ are on the path from u_A to u'_A.*

By Claim C, the decomposition $\mathcal{F}$ contains a path $\hat{u}_1 \cdots \hat{u}_N$ that covers the edges $e_{\min \ominus 1}, e_{\min}, e_{\min \oplus 1}, \dots, e_{\max \ominus 1}, e_{\max}$ in this order. We next strengthen this property by showing that every node $\hat{u}_i$ covers exactly one edge e_p.

Claim H. *Each of the nodes $\hat{u}_1, \dots, \hat{u}_N$ covers exactly one of the edges $e_{\min \ominus 1}, e_{\min}, e_{\min \oplus 1}, \dots, e_{\max \ominus 1}, e_{\max}$.*

We can now associate with each $\hat{u}_i$ with $1 \leq i \leq N$ the corresponding edge e_p and write u_p to denote the node that covers the edge e_p. By Claim G, we know that all of the nodes $u_{\min} \dots, u_{\max \ominus 1}$ are on the path from u_A to u'_A. Hence, by the connectedness condition, all these nodes cover $S \cup \{z_1, z_2\}$.

We are now ready to construct a satisfying truth assignment σ of φ. For each $i \leq 2n+3$, let X_i be the set $B_{u_{(i,1)}} \cap (Y \cup Y')$. As $Y \subseteq B_{u_A}$ and $Y' \subseteq B_{u'_A}$, the sequence $X_1 \cap Y, \dots, X_{2n+3} \cap Y$ is non-increasing and the sequence $X_1 \cap Y', \dots, X_{2n+3} \cap Y'$ is non-decreasing. Furthermore, as all edges $e_{y_i} = \{y_i, y'_i\}$ must be covered by some node in $\mathcal{F}$, we conclude that for each i and j, $y_j \in X_i$ or $y'_j \in X_i$. Then, there is some $s \leq 2n+2$ such that $X_s = X_{s+1}$. Furthermore, all nodes between $u_{(s,1)}$ and $u_{(s+1,1)}$ cover X_s. We derive a truth assignment for $x_1, \dots, x_n$ from X_s as follows. For each $l \leq n$, we set $\sigma(x_l) = 1$ if $y_l \in X_s$ and otherwise $\sigma(x_l) = 0$. Note that in the latter case $y'_l \in X_s$.

Claim I. *The constructed truth assignment σ is a model of φ.*

Claim I completes the proof of Theorem 3.1. ◻

We conclude this section by mentioning that the above reduction is easily extended to $k + \ell$ for arbitrary $\ell \geq 1$: for integer values ℓ, simply add a clique of 2ℓ fresh vertices $v_1, \dots, v_{2\ell}$ to H and connect each v_i with each "old" vertex in H. To achieve a rational bound $k + \ell/q$ with $\ell > q$, we add ℓ fresh vertices and add hyperedges $\{v_i, v_{i \oplus 1}, \dots, v_{i \oplus (q-1)}\}$ with $i \in \{1, \dots, \ell\}$ to H, where $a \oplus b$ denotes $a + b$ modulo ℓ. Again, we connect each v_i with each "old" vertex in H. With this construction we can give NP-hardness proofs for any (fractional) $k \geq 3$. For all fractional $k < 3$ (except for $k = 2$) different gadgets and ideas might be needed to prove NP-hardness of Check(FHD,k), which we leave for future work.

4 EFFICIENT COMPUTATION OF GHDS

As discussed in Section 1 we are interested in finding a realistic and non-trivial criterion on hypergraphs that makes the Check(GHD, k) problem tractable for fixed k. We thus propose here such a simple property, namely the bounded intersection of two or more edges.

Definition 4.1. *The* intersection width *$iwidth(H)$ of a hypergraph H is the maximum cardinality of any intersection $e_1 \cap e_2$ of two distinct edges e_1 and e_2 of H. We say that a hypergraph H has the* i-bounded intersection property (i-BIP) *if $iwidth(H) \leq i$ holds.*

Let $\mathcal{C}$ be a class of hypergraphs. We say that $\mathcal{C}$ has the bounded intersection property (BIP) *if there exists some integer constant i such that every hypergraph H in $\mathcal{C}$ has the i-BIP. Class $\mathcal{C}$ has the* logarithmically-bounded intersection property (LogBIP) *if for each* *of its elements H, iwidth(H) is $O(\log n)$, where n denotes the size of the hypergraph H.*

The BIP criterion is indeed non-trivial, as several well-known classes of unbounded *ghw* enjoy the 1-BIP, such as cliques and grids. Moreover, our empirical study [22] (summarized in [21]) suggests that the overwhelming number of CQs enjoys the 2-BIP (i.e., one hardly joins two relations over more than 2 attributes). To allow for a yet bigger class of hypergraphs, the BIP can be relaxed as follows.

Definition 4.2. *The c-*multi-intersection width *c-miwidth(H) of a hypergraph H is the maximum cardinality of any intersection $e_1 \cap \cdots \cap e_c$ of c distinct edges $e_1, \ldots, e_c$ of H. We say that a hypergraph H has the i-*bounded *c-*multi-intersection property *(ic-*BMIP) *if c-miwidth(H) ≤ i holds.*

Let $\mathscr{C}$ be a class of hypergraphs. We say that $\mathscr{C}$ has the bounded multi-intersection property (BMIP) *if there exist constants c and i such that every hypergraph H in $\mathscr{C}$ has the ic-BMIP. Class $\mathscr{C}$ of hypergraphs has the* logarithmically-bounded multi-intersection property (LogBMIP) *if there is a constant c such that for the hypergraphs $H \in \mathscr{C}$, c-miwidth(H) is $O(\log n)$, where n denotes the size of the hypergraph H.*

The LogBMIP is the most liberal restriction on classes of hypergraphs introduced in Definitions 4.1 and 4.2. The main result in this section will be that the Check(GHD, k) problem with fixed k is tractable for any class of hypergraphs satisfying this criterion.

Towards this result, first recall that the difference betwen HDs and GHDs lies in the "special condition" required by HDs. Assume a hypergraph $H = (V(H), E(H))$ and an arbitrary GHD $\mathcal{H} = \langle T, (B_u)_{u\in T}, (\lambda_u)_{u\in T}\rangle$ of H. Then $\mathcal{H}$ is not necessarily an HD, since it may contain a special condition violation (SCV), i.e.: there can exist a node u, an edge $e \in \lambda_u$ and a vertex $v \in V$, s.t. $v \in e$ (and, hence, $v \in B(\lambda_u)$), $v \notin B_u$ and $v \in V(T_u)$. Clearly, if we could be sure that $E(H)$ also contains the edge $e' = e \cap B_u$, then we would simply replace e in λ_u by e' and would thus get rid of this SCV.

Now our goal is to define a polynomial-time computable function f which, to each hypergraph H and integer k, associates a set $f(H, k)$ of additional hyperedges such that $ghw(H) = k$ iff $hw(H') = k$ with $H = (V(H), E(H))$ and $H' = (V(H), E(H) \cup f(H, k))$. From this it follows immediately that $ghw(H)$ is computable in polynomial time. Moreover, a GHD of the same width can be easily obtained from any HD of H'. The function f is defined in such a way that $f(H, k)$ only contains subsets of hyperedges of H, thus f is a *subedge function* as described in [27]. It is easy to see and well-known [27] that for each subedge function f, and each H and k, $ghw(H) \leq hw(H \cup f(H, k)) \leq hw(H)$. Moreover, for the "limit" subedge function f^+ where $f^+(H, k)$ consists of all possible non-empty subsets of edges of H, we have that $hw(H \cup f^+(H, k)) = ghw(H)$ [3, 27]. Of course, in general, f^+ contains an exponential number of edges. The important point is that our function f will achieve the same, while generating a polynomial and Ptime-computable set of edges only.

We start by introducing a usefuly property of GHDs, which we will call *bag-maximality*. Let $\mathcal{H} = \langle T, (B_u)_{u\in T}, (\lambda_u)_{u\in T}\rangle$ be a GHD of some hypergraph $H = (V(H), E(H))$. For each node u in T, we have $B_u \subseteq B(\lambda_u)$ by definition of GHDs and, in general, $B(\lambda_u) \setminus B_u$ may be non-empty. We observe that it is sometimes possible to take some vertices from $B(\lambda_u) \setminus B_u$ and add them to B_u without violating the connectedness condition. Of course, such an addition of vertices to B_u does not violate any of the other conditions of GHDs. Morevoer, it does not increase the width. We call a GHD *bag-maximal*, if for every node u in T, adding a vertex $v \in B(\lambda_u) \setminus B_u$ to B_u would violate the connectedness condition. It is easy to verify that if H has a GHD of width $\leq k$, then it also has a bag-maximal GHD of width $\leq k$. Indeed, just take an arbitrary GHD $\mathcal{H}$ of width $\leq k$ and, if $\mathcal{H}$ is not bag-maximal, add vertices from $B(\lambda_u)$ to B_u for every node $u \in T$ where this is possible. So from now on, we will restrict ourselves w.l.o.g. to bag-maximal GHDs.

Before we prove a crucial lemma, we introduce some useful notation: in a GHD $\mathcal{H} = \langle T, (B_u)_{u\in T}, (\lambda_u)_{u\in T}\rangle$ of a hypergraph $H = (V(H), E(H))$, let $u \in T$, $e \in \lambda_u$, and $e \setminus B_u \neq \emptyset$. Let u^* be the node closest to u with $e \subseteq B_u$ and let $\pi = (u_0, u_1, \ldots, u_\ell)$ with $u_0 = u$ and $u_\ell = u^*$ denote the path of nodes connecting u and u^*. We shall refer to π as the *critical path* of (u, e).

Lemma 4.1. *Let $\mathcal{H} = \langle T, (B_u)_{u\in T}, (\lambda_u)_{u\in T}\rangle$ be a bag-maximal GHD of a hypergraph $H = (V(H), E(H))$, let $u \in T$, $e \in \lambda_u$, and $e \setminus B_u \neq \emptyset$. Let $\pi = (u_0, u_1, \ldots, u_\ell)$ with $u_0 = u$ be the critical path of (u, e). Then the following equality holds.*

$$e \cap B_u = \quad e \cap \bigcap_{i=1}^{\ell} B(\lambda_{u_i})$$

Proof. "⊆": Given that $e \subseteq B_{u_\ell}$ and by the connectedness condition, $e \cap B_u$ must be a subset of B_{u_i} for every $i \in \{1, \ldots, \ell\}$. Therefore, $e \cap B_u \subseteq e \cap \bigcap_{i=1}^{l} B(\lambda_{u_i})$ holds.
"⊇": Assume to the contrary that there exists some vertex $v \in e$ with $v \notin B_u$ but $v \in \bigcap_{i=1}^{\ell} B(\lambda_{u_i})$. By $e \in B_{u_\ell}$, we have $v \in B_{u_\ell}$. By the connectedness condition, along the path $u_0, \ldots, u_\ell$ with $u_0 = u$, there exists $\alpha \in \{0, \ldots, \ell - 1\}$, s.t. $v \notin B_{u_\alpha}$ and $v \in B_{u_{\alpha+1}}$. However, by the assumption, $v \in \bigcap_{i=1}^{\ell} B(\lambda_{u_i})$ holds. In particular, $v \in B(\lambda_{u_\alpha})$. Hence, we could safely add v to B_{u_α} without violating the connectedness condition nor any other GHD condition. This contradicts the bag-maximality of $\mathcal{H}$. □

We are now ready to prove the main result of this section.

Theorem 4.1. *For every hypergraph class $\mathscr{C}$ that enjoys the LogBMIP, and for every constant $k \geq 1$, the Check(GHD, k) problem is tractable, i.e., given a hypergraph H, it is feasible in polynomial time to check ghw(H) ≤ k and, if so, to compute a GHD of width k of H.*

Sketch. Let $\mathcal{H} = \langle T, (B_u)_{u\in T}, (\lambda_u)_{u\in T}\rangle$ be a bag-maximal GHD of a hypergraph $H = (V(H), E(H))$, let $u \in T$, $e \in \lambda_u$, and $e \setminus B_u \neq \emptyset$. Let $\pi = (u_0, u_1, \ldots, u_\ell)$ with $u_0 = u$ be the critical path of (u, e). By Lemma 4.1, the equality $e \cap B_u = \quad e \cap \bigcap_{i=1}^{\ell} B(\lambda_{u_i})$ holds. For $i \in \{1, \ldots, \ell\}$, let $\lambda_{u_i} = \{e_{i1}, \ldots, e_{ij_i}\}$ with $j_i \leq k$. Then $e \cap \bigcap_{i=1}^{\ell} B(\lambda_{u_i})$ and, therefore, also $e \cap B_u$, is of the form

$$e \cap (e_{11} \cup \cdots \cup e_{1j_1}) \cap \cdots \cap (e_{\ell 1} \cup \cdots \cup e_{\ell j_\ell}).$$

We aim at a stepwise transformation of this intersection of unions into a union of intersections via distributivity of ∪ and ∩. For $i \in \{0, \ldots, \ell\}$, let $I_i = e \cap \bigcap_{\alpha=1}^{i} B(\lambda_{u_\alpha}) = e \cap \bigcap_{\alpha=1}^{i} (e_{\alpha 1} \cup \cdots \cup e_{\alpha j_\alpha})$. In order to compute the sets $I_0, \ldots, I_\ell$ as unions of intersections, the Algorithm `Union-of-Intersections-Tree` in Figure 3 constructs the "$\bigcup\bigcap$-tree". In a loop over all $i \in \{1, \ldots, \ell\}$, we thus compute trees $\mathcal{T}_i$ such that each node p in $\mathcal{T}_i$ is labelled by a set *label*(p) of edges. By $int(p)$ we denote the intersection of the edges in $label(p)$. The parent-child relationship between a node p and its child nodes $p_1, \dots, p_\gamma$ corresponds to a splitting step, where the intersection $int(p)$ is replaced by the union $(int(p) \cap e_{\alpha 1}) \cup \dots \cup (int(p) \cap e_{\alpha j_\alpha})$. It can be verified that, in the tree $\mathcal{T}_i$, the union of $int(p)$ over all *leaf nodes* of $\mathcal{T}_i$ yields precisely the union-of-intersections representation of I_i.

We observe that, in the tree $\mathcal{T}_\ell$, each node has at most k child nodes. Nevertheless, $\mathcal{T}_\ell$ can become exponentially big since we have no appropriate bound on the length ℓ of the critical path. Recall that we are assuming the LogBMIP, i.e., there exists a constant $c > 1$, s.t. any intersection of $\geq c$ edges of H has at most $a \log n$ elements, where a is a constant and n denotes the size of H. Now let $\mathcal{T}^*$ be the *reduced* $\bigcup\bigcap$*-tree*, which is obtained from $\mathcal{T}_\ell$ by cutting off all nodes of depth greater than $c - 1$. Clearly, $\mathcal{T}^*$ has at most k^{c-1} leaf nodes and the total number of nodes in $\mathcal{T}^*$ is bounded by $(c-1)k^{c-1}$.

The set $f(H, k)$ of subedges that we add to H will consist in all possible sets I_ℓ that we can obtain from all possible critical paths $\pi = (u_0, u_1, \dots, u_\ell)$ in all possible bag-maximal GHDs $\mathcal{H}$ of width $\leq k$ of H. We only discuss the polynomial bound on the number of possible sets I_ℓ. The polynomial-time computability of this set of sets is then easy to see. The set of all possible sets I_ℓ is obtained by first considering all possible reduced $\bigcup\bigcap$-trees $\mathcal{T}^*$ and then considering all sets I_ℓ that correspond to some extension $\mathcal{T}_\ell$ of $\mathcal{T}^*$.

The number of possible reduced $\bigcup\bigcap$-trees $\mathcal{T}^*$ for given H and k is bounded by $m * m^{(c-1)k^{c-1}}$, where m denotes the number of edges in $E(H)$. It remains to determine the number of possible sets I_ℓ that one can get from possible extensions $\mathcal{T}_\ell$ of $\mathcal{T}^*$. Clearly, if a leaf node in $\mathcal{T}^*$ is at depth $< c - 1$, then no descendants at all of this node have been cut off. In contrast, a leaf node p in $\mathcal{T}^*$ at depth $c - 1$ may be the root of a whole subtree in $\mathcal{T}_\ell$. Let $U(p)$ denote the union of the intersections represented by all leaf nodes below p. By construction of $\mathcal{T}_\ell$, $U(p) \subseteq int(p)$ holds. Moreover, by the LogBMIP, $|int(p)| \leq a \log n$ for some constant a. Hence, $U(p)$ takes one out of at most $2^{a \log n} = n^a$ possible values. In total, the number of possible sets I_ℓ (and, hence, $|f(H, k)|$) is bounded by $m * m^{(c-1)k^{c-1}} * n^{a(c-1)k^{c-1}}$ for some constant a. □

We have defined in Section 1 the degree d of a hypergraph H. A class $\mathscr{C}$ of hypergraphs has *bounded degree* if there exists some integer constant d s.t. every hypergraph H in $\mathscr{C}$ has degree $\leq d$.

The class of hypergraphs of bounded degree is an interesting special case of the class of hypergraphs enjoying the BMIP. Indeed, suppose that each vertex in a hypergraph H occurs in at most d edges for some constant d. Then the intersection of $d+1$ hyperedges is always empty. The following corollary is thus immediate.

Corollary 4.1. *For every class $\mathscr{C}$ of hypergraphs of bounded degree, for each constant k, the problem Check(GHD, k) is tractable.*

The upper bound $|f(H, k)|$ in the proof sketch of Theorem 4.1, improves to $m^{k+1} \cdot 2^{k \cdot i}$ for the important special case of the BIP. We thus get the following parameterized complexity result.

Theorem 4.2. *For each constant k, the Check(GHD, k) problem is fixed-parameter tractable w.r.t. the parameter i for hypergraphs enjoying the BIP, i.e., in this case, Check(GHD, k) can be solved in time $O(h(i) \cdot poly(n))$, where $h(i)$ is a function depending on the intersection width i only and $poly(n)$ is a function that depends polynomially on the size n of a given hypergraph H.*

ALGORITHM `Union-of-Intersections-Tree`

Input: GHD $\mathcal{H}$ of H, edge e, critical path π
Output: $\bigcup\bigcap$-tree T_ℓ

```
begin
// Initialization: compute (N, E) for T_0
   Let π = (u_0, ..., u_ℓ);
   N := {e};
   E := ∅;
   T := (N, E);
// Compute T_i from T_{i-1} in a loop over i
   For i := 1 To ℓ Do
      For Each leaf node p of T Do
         If label(p) ∩ λ_{u_i} = ∅ Then
            Let λ_{u_i} = {e_{i1}, ..., e_{ij_i}};
            Create new nodes {p_1, ..., p_{j_i}};
            For α := 1 To j_i Do label(p_α) := label(p_α) ∪ {e_{iα}};
            N := N ∪ {p_1, ..., p_{j_i}};
            E := E ∪ {(p, p_1), ..., (p, p_{j_i})};
      T := (N, E);
   Return T;
end
```

Figure 3: Algorithm to compute the $\bigcup\bigcap$-tree

5 EFFICIENT COMPUTATION OF FHDS

In Section 4, we have shown that under certain conditions (with the BIP as most specific and the LogBMIP as most general condition) the problem of computing a GHD of width k can be reduced to the problem of computing an HD of width k. The key to this problem reduction was to repair the special condition violations in the given GHD. When trying to carry over these ideas from GHDs to FHDs, we encounter *two major challenges*: Can we repair special condition violations in an FHD by ideas similar to GHDs? Does the special condition in case of FHDs allow us to carry the hypertree decomposition algorithm from [26] over to FHDs?

As for the first challenge, it turns out that FHDs behave substantially differently from GHDs. Suppose that there is a special condition violation (SCV) in some node u of an FHD. Then there must be some hyperedge $e \in E(H)$, such that $\gamma_u(e) > 0$ and $B(\gamma_u)$ contains some vertex v with $v \in e \setminus B_u$. Moreover, e is covered by some descendant node u_0 of u. For GHDs, we exploit the BIP essentially by distinguishing two cases: either $\lambda_{u'}(e) = 1$ for every node u' on the path π from u to u_0 or there exists a node u' on path π with $\lambda_{u'}(e) = 0$. In the former case, we simply add all vertices $v \in e \setminus B_u$ to B_u (in the proof of Theorem 4.1 this is taken care of by assuming bag-maximality). In the latter case, we can apply the BIP to the edges e_j with $\lambda_{u'}(e_j) = 1$ since we now know that they are all distinct from e. In case of FHDs, this argument does not work anymore, since it may well happen that $\gamma_{u'}(e) > 0$ holds for every node u' on the path π but, nevertheless, we are not allowed to add all vertices of e to every bag $B_{u'}$. The simple reason for this is that $\gamma_{u'}(e) > 0$ does not imply $e \subseteq B(\gamma_{u'})$ in the fractional case.

As for the second challenge, it turns out that even if we restrict to FHDs satisfying the special condition, there remains another obstacle compared to the HD algorithm from [26]: a crucial step of the top-down construction of an HD is to "guess" the k edges with $\lambda_u(e) = 1$ for the next node u in the HD. However, for a fractional cover γ_u, we do not have such a bound on the number of edges with non-zero weight. In fact, it is easy to exhibit a family $(H_n)_{n\in\mathbb{N}}$ of hypergraphs where it is advantageous to have unbounded $supp(H_n)$ even if $(H_n)_{n\in\mathbb{N}}$ enjoys the BIP, as the example illustrates:

Example 5.1. Consider the family $(H_n)_{n\in\mathbb{N}}$ of hypergraphs with $H_n = (V_n, E_n)$ defined as follows:

$V_n = \{v_0, v_1, \dots, v_n\}$

$E_n = \{\{v_0, v_i\} \mid 1 \le i \le n\} \cup \{\{v_1, \dots, v_n\}\}$

Clearly $iwidth(H_n) = 1$, but an optimal fractional edge cover of H_n is obtained by the following mapping γ with $supp(\gamma) = E_n$:

$\gamma(\{v_0, v_i\}) = 1/n$ for each $i \in \{1, \dots, n\}$ and

$\gamma(\{v_1, \dots, v_n\}) = 1 - (1/n)$

where $weight(\gamma) = 2 - (1/n)$, which is optimal in this case. ⋄

Nevertheless, in this section, we use the ingredients from our tractability results for the Check(GHD, k) problem to prove a similar (slightly weaker though) tractability result for the Check(FHD, k) problem. More specifically, we shall show below that the Check(FHD, k) problem becomes tractable for fixed k, if we impose the two restrictions BIP *and* bounded degree on the hypergraphs under investigation. Thus, the main result of this section is:

Theorem 5.1. *For every hypergraph class $\mathscr{C}$ that enjoys the BIP and has bounded degree, and for every constant $k \ge 1$, the Check(FHD, k) problem is tractable, i.e., given a hypergraph $H \in \mathscr{C}$, it is feasible in polynomial time to check $fhw(H) \le k$ and, if this holds, to compute an FHD of width k of H.*

We now develop the necessary machinery to finally give a proof sketch of Theorem 5.1. The crucial concept, which we introduce next, will be that of a *c-bounded fractional part*. Intuitively, FHDs with c-bounded fractional part are FHDs, where the fractional edge cover γ_u in every node u is "close to an edge cover" – with the possible exception of up to c vertices in the bag B_u. For the special case $c = 0$, an FHD with c-bounded fractional part is simply a GHD.

It is convenient to first introduce the following notation: let $\gamma : E(H) \to [0, 1]$ and let $S \subseteq supp(\gamma)$. We write $\gamma|_S$ for the *restriction of γ to S*, i.e., $\gamma|_S(e) = \gamma(e)$ if $e \in S$ and $\gamma|_S(e) = 0$ otherwise.

Definition 5.1. *Let $\mathcal{F} = \langle T, (B_u)_{u\in T}, (\gamma_u)_{u\in T}\rangle$ be an FHD of some hypergraph H and let $c \ge 0$. We say that $\mathcal{F}$ has* c-bounded fractional part *if in every node $u \in T$, the following property holds:*

Let $S = \{e \in E(H) \mid \gamma_u(e) = 1\}$ and $B_u = B_1 \cup B_2$ with $B_1 = B_u \cap B(\gamma_u|_S)$ and $B_2 = B_u \setminus B_1$. Then $|B_2| \le c$.

We next generalize the special condition (i.e., (4) of Defintion 2.4) to FHDs. Hence, we define the *weak special condition*. It requires that the special condition must be satisfied by the integral part of each fractional edge cover. For the special case $c = 0$, an FHD with c-bounded fractional part satisfying the weak special condition is thus simply a GHD satisfying the special condition, i.e., a HD.

Definition 5.2. *Let $\mathcal{F} = \langle T, (B_u)_{u\in T}, (\gamma_u)_{u\in T}\rangle$ be an FHD of some hypergraph H. We say that $\mathcal{F}$ satisfies the* weak special condition *if in every node $u \in T$, the following property holds: for $S = \{e \in E(H) \mid \gamma_u(e) = 1\}$, we have $B(\gamma_u|_S) \cap V(T_u) \subseteq B_u$.*

We now present the two key lemmas for classes C of hypergraphs with the BIP and bounded degree, namely: (1) if a hypergraph $H \in C$ has an FHD of width $\le k$, then it also has an FHD of width $\le k$ with c-bounded fractional part (where c only depends on k, d, and the bound i on the intersection width, but not on the size of H) and (2) we can extend H to a hypergraph H' by adding polynomially many edges, such that H' has an FHD of width $\le k$ with c-bounded fractional part satisfying the weak special condition.

Lemma 5.1. *Let $\mathscr{C}$ be a hypergraph class that enjoys the BIP and has bounded degree and let $k \ge 1$. For every hypergraph $H \in \mathscr{C}$, the following property holds:*

If H has an FHD of width $\le k$, then H also has an FHD of width $\le k$ with c-bounded fractional part, where c only depends on width k, degree d, and intersection width i (but not on the size of H).

Proof Sketch. Consider an arbitrary node u in an FHD $\mathcal{F} = \langle T, (B_u)_{u\in T}, (\gamma_u)_{u\in T}\rangle$ of H and let γ_u be an optimal fractional cover of B_u. Let $B_2 \subseteq B_u$ be the fractional part of B_u, i.e., for $S = \{e \in E(H) \mid \gamma_u(e) = 1\}$, we have $B_1 = B_u \cap B(\gamma_u|_S)$ and $B_2 = B_u \setminus B_1$.

By the bound d on the degree and bound k on the weight of γ_u, there exists a subset $R \subseteq supp(\gamma_u)$ with $|R| \le k \cdot d$, s.t. $B_2 \subseteq V(R)$, i.e., every vertex $x \in B_2$ is contained in at least one edge $e \in R$.

One can then show that only "constantly" many edges (where this constant m depends on k, d, and i) are needed so that every vertex $x \in B_2$ is contained in at least two edges in $supp(\gamma_u)$. Let this set of edges be denoted by R^* with $|R^*| \le m$. Then every vertex $x \in B_2$ is contained in some e_j plus one more edge in $R^* \setminus \{e_j\}$. Hence, by the BIP, we have $|e_j| \le m \cdot i$ and, therefore, by $B_2 \subseteq e_1 \cup \dots \cup e_n$, we have $|B_2| \le n \cdot m \cdot i \le k \cdot d \cdot m \cdot i$. □

Lemma 5.2. *Let $c \ge 0$, $i \ge 0$, and $k \ge 1$. There exists a polynomial-time computable function $f_{(c,i,k)}$ which takes as input a hypergraph H with $iwidth(H) \le i$ and yields as output a set of subedges of $E(H)$ with the following property: If H has an FHD of width $\le k$ with c-bounded fractional part then H' has an FHD of width $\le k$ with c-bounded fractional part satisfying the weak special condition, where $H' = (V(H), E(H) \cup f_{(c,i,k)}(H))$.*

Proof Sketch. Let i denote the bound on the intersection width of the hypergraphs in $\mathscr{C}$. Analogously to the proof of Theorem 4.1, it suffices to add those edges to $E(H)$ which are obtained as a subset of the intersection of an edge $e \in E(H)$ with some bag B_u in the FHD. The bag B_u in turn is contained in the union of at most k edges different from e (namely the edges e_j with $\gamma_u(e_j) = 1$) plus at most c additional vertices. The intersection of an edge e with up to k further edges has at most $k \cdot i$ elements. In total, we thus just need to add all subedges e' of e with $|e'| \le k \cdot i + c$ for every $e \in E(H)$. Clearly, this set of subedges is polynomially bounded (since we are considering k, i, and c as constants) and can be computed in polynomial time. □

We are now ready to give a proof sketch of Theorem 5.1.

Proof Sketch of Theorem 5.1. The tractability of Check(FHD, k) is shown by adapting the alternating logspace algorithm from [26]. The key steps in that algorithm are (A) to guess a set S of ℓ edges with $\ell \le k$ (i.e., the edge cover λ_s of a node s in the

ALTERNATING ALGORITHM k-`frac-decomp`

Input: hypergraph H, integer $c \geq 0$.
Output: "Accept", if H has an FHD of width $\leq k$ with c-bounded fractional part and weak special condition;
"Reject", otherwise.

Procedure k-`fdecomp` (C_r, W_r: Vertex-Set, R: Edge-Set)
begin
// Step (A) – **Guess & Check**
1) **Guess**:
1.a) Guess a set $S \subseteq E(H)$ with $|S| = \ell$, s.t. $\ell \leq k$;
1.b) Guess a set $W_s \subseteq (V(R) \cup C_r)$ with $|W_s| \leq c$;
2) **Check**:
2.a) $W_s \cap V(s) = \emptyset$;
2.b) $\exists\gamma$ with $W_s \subseteq B(\gamma)$ and $weight(\gamma) \leq k - \ell$;
2.c) $\forall e \in edges(C_R)$: $e \cap (V(R) \cup W_r) \subseteq (V(S) \cup W_s)$;
2.d) $(V(S) \cup W_s) \cap C_r \neq \emptyset$;
// Step (B) – **Decompose**
3) **If** one of these checks fails **Then Halt** and **Reject**;
Else
Let $\mathcal{C} := \{C \subseteq V(H) \mid C$ is a $[V(S) \cup W_s]$-component and $C \subseteq C_r\}$;
4) **If for each** $C \in \mathcal{C}$: k-`fdecomp` (C, W_s, S)
Then Accept
Else Reject
end

begin (* Main *)
Accept if k-`fdecomp` ($V(H)$, $\emptyset$, $\emptyset$)
end

Figure 4: Alternating algorithm to decide if $fhw \leq k$

construction of the HD) and (B) to compute all $[B_s]$-components to recursively continue the top-down construction of the HD.

We now show how this algorithm can be adapted to compute an FHD of width k. The adapted algorithm is given in Figure 4.

In step (A) – **Guess & Check** – we now have to guess a set S of ℓ edges plus a set W_s of up to c vertices from outside $V(S)$. Morever, it is important to verify in Ptime (by linear programming) that W_s indeed has a fractional cover of width $k - \ell$.

For step (B) – **Decompose** – the crucial property used in the algorithm of [26] is that, if we construct an HD (i.e., a GHD satisfying the *special condition*), then the $[B_s]$-components and the $[B(\lambda_s)]$-components coincide. Analogously, we can show that if an FHD with c-bounded fractional part satisfies the *weak special condition*, then the $[B_1 \cup B_2]$-components and the $[B(\gamma_s|_S) \cup B_2]$-components coincide, where $B_1 = B_s \cap B(\gamma_s|_S)$ and $B_2 = B_s \setminus B_1$. Hence, analogously to the algorithm of [26], the components to be considered in the recursion of this algorithm are fully determined by S and W_s, where both $|S|$ and $|W_s|$ are bounded by a constant. □

We conclude this section by exhibiting a simple further class of hypergraphs with tractable Check(FHD, k) problem, namely the class $\mathscr{C}$ of hypergraphs with bounded rank, i.e., there exists a constant r, such that for every $H \in \mathscr{C}$ and every $e \in E(H)$, we have $|E| \leq r$. Note that in this case, a fractional edge cover of weight k can cover at most $c = k \cdot r$ vertices. Hence, every FHD of such a hypergraph trivially has c-bounded fractional part. Moreover, in step (1) of the algorithm sketched in the proof of Theorem 5.1, we may simply skip the guess of set S (i.e, we do not need the weak special condition) and just guess a set W of vertices with $|W| \leq c$. The following corollary is thus immediate.

Corollary 5.1. *For every hypergraph class with bounded rank and every constant $k \geq 1$, the Check(FHD, k) problem is tractable.*

6 EFFICIENT APPROXIMATION OF FHW

In the previous section, we have seen that the computation of FHDs poses additional challenges compared with the computation of GHDs. Consequently, we needed a stronger restriction (combining BIP and bounded degree) on the hypergraphs under consideration to achieve tractability. We have to leave it as an open question for future research if the BIP alone or bounded degree alone suffice to ensure tractability of the Check(FHD, k) problem for fixed $k \geq 1$.

In this section, we turn our attention to approximations of the *fhw*. We know from [38] that a tractable cubic approximation of the *fhw* always exists, i.e.: for $k \geq 1$, there exists a polynomial-time algorithm that, given a hypergraph H with $fhw(H) \leq k$, finds an FHD of H of width $O(k^3)$. In this section, we search for conditions which guarantee a better approximation of the *fhw* and which are again realistic.

A natural first candidate for restricting hypergraphs are the BIP and, more generally, the BMIP from the previous section. Indeed, by combining some classical results on the Vapnik-Chervonenkis (VC) dimension with some novel observations, we will show that the BMIP yields a better approximation of the *fhw*. To this end, we first recall the definition of the VC-dimension of hypergraphs.

Definition 6.1 ([43, 47]). *Let $H = (V(H), E(H))$ be a hypergraph, and $X \subseteq V$ a set of vertices. Denote by $E(H)|_X = \{X \cap e \mid e \in E(H)\}$. X is called* shattered *if $E(H)|_X = 2^X$. The* Vapnik-Chervonenkis dimension (VC dimension) vc(H) *of H is the maximum cardinality of a shattered subset of V.*

We now provide a link between the VC-dimension and our first approximation result for the *fhw*.

Definition 6.2. *Let $H = (V(H), E(H))$ be a hypergraph. A* transversal *(also known as* hitting set*) of H is a subset $S \subseteq V(H)$ that has a non-empty intersection with every edge of H. The* transversality *$\tau(H)$ of H is the minimum cardinality of all transversals of H.*

Clearly, $\tau(H)$ corresponds to the minimum of the following integer linear program: find a mapping $w : V \rightarrow \mathbb{R}_{\geq 0}$ which minimizes $\Sigma_{v \in V(H)} w(v)$ under the condition that $\Sigma_{v \in e} w(v) \geq 1$ holds for each hyperedge $e \in E$.

The fractional transversality *τ^* of H is defined as the minimum of the above linear program when dropping the integrality condition. Finally, the* transversal integrality gap *tigap(H) of H is the ratio $\tau(H)/\tau^*(H)$.*

Recall that computing the mapping λ_u for some node u in a GHD can be seen as searching for a minimal edge cover ρ of the vertex set B_u, whereas computing γ_u in an FHD corresponds to the search for a minimal fractional edge cover ρ^* [30]. Again, these problems can be cast as linear programs where the first problem has the integrality condition and the second one has not. Further, we can define the *cover integrality gap* $cigap(H)$ of H as the ratio $\rho(H)/\rho^*(H)$. With this we state a first approximation result for *fhw*.

Theorem 6.1. *Let $\mathscr{C}$ be a class of hypergraphs with VC-dimension bounded by some constant d and let $k \geq 1$. Then there exists a polynomial-time algorithm that, given a hypergraph $H \in \mathscr{C}$ with $fhw(H) \leq k$, finds an FHD of H of width $O(k \cdot \log k)$.*

Proof. The proof proceeds in several steps.

Reduced hypergraphs. We are interested in hypergraphs that are *essential* in the following sense: let $H = (V, E)$ be a hypergraph and let $v \in V$. Then the edge-type of v is defined as $etype(v) = \{e \in E \mid v \in e\}$. We call H *essential* if there exists no pair (v, v') of distinct vertices with the same edge-type. Every hypergraph H can be transformed into an essential hypergraph H' by exhaustively applying the following rule: if there are two vertices v, v' with $v \neq v'$ and $etype(v) = etype(v')$, then delete v'. It is easy to verify that $hw(H) = hw(H')$, $ghw(H) = ghw(H')$, and $fhw(H') = fhw(H')$ hold for any hypergraph H with corresponding essential hypergraph H'. Hence, w.l.o.g., we only consider *essential* hypergraphs.

Dual hypergraphs. Given a hypergraph $H = \{V, E)$, the dual hypergraph $H^d = (W, F)$ is defined as $W = E$ and $F = \{\{e \in E \mid v \in e\} \mid v \in V\}$. We are assuming that H is *essential*. Then $(H^d)^d = H$ clearly holds. Moreover, the following relationships between H and H^d are well-known and easy to verify (see, e.g., [19]):

(1) The edge coverings of H and the transversals of H^d coincide.

(2) The fractional edge coverings of H and the fractional transversals of H^d coincide.

(3) $\rho(H) = \tau(H^d)$, $\rho^*(H) = \tau^*(H^d)$, and $cigap(H) = tigap(H^d)$.

VC-dimension. By a classical result (see [11, 18]), for every hypergraph $H = (V(H), E(H))$, we have

$$tigap(H) = \tau(H)/\tau^*(H) \leq 2\text{vc}(H)\log(11\tau^*(H))/\tau^*(H).$$

Moreover, in [7], it is shown that $\text{vc}(H^d) < 2^{\text{vc}(H)+1}$ always holds. In total, we thus get

$$\begin{aligned} cigap(H) = tigap(H^d) &\leq 2\text{vc}(H^d)\log(11\tau^*(H^d))/\tau^*(H^d) \\ &< 2^{\text{vc}(H)+2}\log(11\rho^*(H))/\rho^*(H). \end{aligned}$$

Approximation of fhw by ghw. Suppose that H has an FHD $\langle T, (B_u)_{u\in V(T)}, (\lambda)_{u\in V(T)}\rangle$ of width k. Then there exists a GHD of H of width $O(k \cdot \log k)$. Indeed, we can find such a GHD by leaving the tree structure T and the bags B_u for every node u in T unchanged and replacing each fractional edge cover γ_u of B_u by an optimal integral edge cover λ_u of B_u. By the above inequality, we thus increase the weight at each node u only by a factor $O(\log k)$. Moreover, we know from [4] that computing an HD instead of a GHD increases the width only by the constant factor 3. □

One drawback of the VC-dimension is that deciding if a hypergraph has VC-dimension $\leq v$ is intractable [44]. However, Lemma 6.1 establishes a relationship between BMIP and VC-dimension. Together with Theorem 6.1, Corollary 6.1 is immediate.

Lemma 6.1. *If a class $\mathscr{C}$ of hypergraphs has the BMIP then it has bounded VC-dimension. However, there exist classes $\mathscr{C}$ of hypergraphs with bounded VC-dimension that do not have the BMIP.*

Corollary 6.1. *Let $\mathscr{C}$ be a class of hypergraphs enjoying the BMIP and let $k \geq 1$. Then there exists a polynomial-time algorithm that, given $H \in \mathscr{C}$ with $fhw(H) \leq k$, finds an FHD (actually, even a GHD) of H of width $O(k \cdot \log k)$.*

We would like to identify classes of hypergraphs that allow for a yet better approximation of the *fhw*. Below we show that the hypergraphs of bounded degree indeed allow us to approximate the *fhw* by a constant factor in polynomial time. We proceed in two steps. First, in Lemma 6.2, we establish a relationship between *fhw* and *ghw* via the degree. Then we make use of Corollary 4.1 from the previous section on the computation of a GHD to get the desired approximation of *fhw* in Corollary 6.2.

Lemma 6.2. *Let H be an arbitrary hypergraph and let d denote the degree of H. Then the following holds: $ghw(H) \leq d \cdot fhw(H)$.*

Corollary 6.2. *Let $\mathscr{C}$ be a class of hypergraphs whose degree is bounded by some constant $d \geq 1$ and let $k \geq 1$. Then there exists a polynomial-time algorithm that, given a hypergraph $H \in \mathscr{C}$ with $fhw(H) \leq k$, finds an FHD (actually, a GHD) of H of width $\leq d \cdot k$.*

7 CONCLUSION AND FUTURE WORK

In this paper we have settled the complexity of deciding $fhw(H) \leq k$ for fixed constant $k \geq 2$ and $ghw(H) \leq k$ for $k = 2$ by proving the NP-completeness of both problems. This gives negative answers to two open problems. On the positive side, we have identified rather mild restrictions such as the BIP, LogBIP, BMIP, and LogBMIP, which give rise to a Ptime algorithm for the Check(GHD, k) problem. Moreover, we have shown that the combined restriction of BIP and bounded degree ensures tractability also of the Check(FHD, k) problem. As our empirical analyses reported in [22] show, these restrictions are very well-suited for instances of CSPs and, even more so, of CQs. We believe that they deserve further attention.

Our work does not finish here. We plan to explore several further issues regarding the computation and approximation of the fractional hypertree width. We find the following questions particularly appealing: (i) Does the special condition defined by Grohe and Marx [30] lead to tractable recognizability also for FHDs, i.e., in case we define "$sfhw(H)$" as the smallest width an FHD of H satisfying the special condition, can $sfhw(H) \leq k$ be recognized efficiently? (ii) Our tractability result in Section 5 for the Check(FHD, k) problem is weaker than for Check(GHD, k), in that we need the combined restriction of the BIP and bounded degree. Actually, very recently [23], we could show that bounded degree alone suffices to ensure tractability of Check(FHD, k). It is open if the BIP alone (or, more generally, the BMIP) also suffices. (iii) In case that the BIP (or BMIP) does not guarantee tractability of Check(FHD, k), it is interesting to investigate if the BIP (or BMIP) at least ensures a polynomial-time approximation of $fhw(H)$ up to a constant factor. Or can non-approximability results be obtained under reasonable complexity-theoretic assumptions?

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), Programme Grant EP/M025268/ VADA: Value Added Data Systems — Principles and Architecture as well as by the Austrian Science Fund (FWF):P25518-N23 and P30930-N35.

REFERENCES

[1] Christopher R. Aberger, Susan Tu, Kunle Olukotun, and Christopher Ré. 2016. EmptyHeaded: A Relational Engine for Graph Processing. In *Proceedings of SIGMOD 2016*. ACM, 431–446.

[2] Christopher R. Aberger, Susan Tu, Kunle Olukotun, and Christopher Ré. 2016. Old Techniques for New Join Algorithms: A Case Study in RDF Processing. *CoRR* abs/1602.03557 (2016). http://arxiv.org/abs/1602.03557

[3] Isolde Adler. 2004. Marshals, monotone marshals, and hypertree-width. *Journal of Graph Theory* 47, 4 (2004), 275–296.

[4] Isolde Adler, Georg Gottlob, and Martin Grohe. 2007. Hypertree width and related hypergraph invariants. *Eur. J. Comb.* 28, 8 (2007), 2167–2181.

[5] Foto N. Afrati, Manas Joglekar, Christopher Ré, Semih Salihoglu, and Jeffrey D. Ullman. 2014. GYM: A Multiround Join Algorithm In MapReduce. *CoRR* abs/1410.4156 (2014).

[6] Molham Aref, Balder ten Cate, Todd J. Green, Benny Kimelfeld, Dan Olteanu, Emir Pasalic, Todd L. Veldhuizen, and Geoffrey Washburn. 2015. Design and Implementation of the LogicBlox System. In *Proceedings of SIGMOD 2015*. ACM, 1371–1382.

[7] Patrick Assouad. 1983. Densité et dimension. *Annales de l'Institut Fourier* 33, 3 (1983), 233–282.

[8] Albert Atserias, Martin Grohe, and Dániel Marx. 2013. Size Bounds and Query Plans for Relational Joins. *SIAM J. Comput.* 42, 4 (2013), 1737–1767.

[9] Nurzhan Bakibayev, Tomás Kociský, Dan Olteanu, and Jakub Závodný. 2013. Aggregation and Ordering in Factorised Databases. *PVLDB* 6, 14 (2013), 1990–2001.

[10] Angela Bonifati, Wim Martens, and Thomas Timm. 2017. An Analytical Study of Large SPARQL Query Logs. *PVLDB* 11, 2 (2017), 149–161. http://www.vldb.org/pvldb/vol11/p149-bonifati.pdf

[11] H. Brönnimann and M. T. Goodrich. 1995. Almost optimal set covers in finite VC-dimension. *Discrete & Computational Geometry* 14, 4 (01 Dec 1995), 463–479. https://doi.org/10.1007/BF02570718

[12] Ashok K. Chandra and Philip M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of STOC 1977*. ACM, 77–90.

[13] Chandra Chekuri and Anand Rajaraman. 2000. Conjunctive query containment revisited. *Theor. Comput. Sci.* 239, 2 (2000), 211–229.

[14] Hubie Chen and Víctor Dalmau. 2005. Beyond Hypertree Width: Decomposition Methods Without Decompositions. In *Proceedings of CP 2005 (Lecture Notes in Computer Science)*, Vol. 3709. Springer, 167–181.

[15] David A. Cohen, Peter Jeavons, and Marc Gyssens. 2008. A unified theory of structural tractability for constraint satisfaction problems. *J. Comput. Syst. Sci.* 74, 5 (2008), 721–743. https://doi.org/10.1016/j.jcss.2007.08.001

[16] Víctor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. 2002. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Proceedings of CP 2002 (Lecture Notes in Computer Science)*, Vol. 2470. Springer, 310–326.

[17] Rina Dechter and Judea Pearl. 1989. Tree Clustering for Constraint Networks. *Artif. Intell.* 38, 3 (1989), 353–366.

[18] Guo-Li Ding, Paul Seymour, and Peter Winkler. 1994. Bounding the vertex cover number of a hypergraph. *Combinatorica* 14, 1 (1994), 23–34.

[19] Pierre Duchet. 1996. Hypergraphs. In *Handbook of combinatorics (vol. 1)*. MIT Press, 381–432.

[20] Ronald Fagin. 1983. Degrees of acyclicity for hypergraphs and relational database schemes. *J. ACM* 30, 3 (1983), 514–550.

[21] Wolfgang Fischl, Georg Gottlob, and Reinhard Pichler. 2016. General and Fractional Hypertree Decompositions: Hard and Easy Cases. *CoRR* abs/1611.01090 (2016). http://arxiv.org/abs/1611.01090

[22] Woflgang Fischl, Georg Gottlob, and Reinhard Pichler. 2016. Generalized and Fractional Hypertree Decompositions: Empirical Results. forthcoming report. (2016).

[23] Wolfgang Fischl, Georg Gottlob, and Reinhard Pichler. 2017. Tractable Cases for Recognizing Low Fractional Hypertree Width. *viXra.org e-prints* viXra:1708.0373 (2017). http://vixra.org/abs/1708.0373

[24] Eugene C. Freuder. 1990. Complexity of K-Tree Structured Constraint Satisfaction Problems. In *Proceedings of AAAI 1990*. AAAI Press / The MIT Press, 4–9.

[25] Georg Gottlob and Gianluigi Greco. 2013. Decomposing combinatorial auctions and set packing problems. *J. ACM* 60, 4 (2013), 24.

[26] Georg Gottlob, Nicola Leone, and Francesco Scarcello. 2002. Hypertree Decompositions and Tractable Queries. *J. Comput. Syst. Sci.* 64, 3 (2002), 579–627. https://doi.org/10.1006/jcss.2001.1809

[27] Georg Gottlob, Zoltán Miklós, and Thomas Schwentick. 2009. Generalized Hypertree Decompositions: NP-hardness and Tractable Variants. *J. ACM* 56, 6, Article 30 (Sept. 2009), 32 pages. https://doi.org/10.1145/1568318.1568320

[28] Martin Grohe. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54, 1 (2007).

[29] Martin Grohe and Dániel Marx. 2006. Constraint solving via fractional edge covers. In *Proceedings of SODA 2006*. ACM Press, 289–298.

[30] Martin Grohe and Dániel Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Trans. Algorithms* 11, 1 (2014), 4:1–4:20.

[31] Martin Grohe, Thomas Schwentick, and Luc Segoufin. 2001. When is the evaluation of conjunctive queries tractable?. In *Proceedings of STOC 2001*. ACM, 657–666.

[32] Marc Gyssens, Peter Jeavons, and David A. Cohen. 1994. Decomposing Constraint Satisfaction Problems Using Database Techniques. *Artif. Intell.* 66, 1 (1994), 57–89.

[33] Marc Gyssens and Jan Paredaens. 1984. A Decomposition Methodology for Cyclic Databases. In *Advances in Data Base Theory: Volume 2*. Springer, 85–122.

[34] Khayyam Hashmi, Zaki Malik, Erfan Najmi, and Abdelmounaam Rezgui. 2016. SNRNeg: A social network enabled negotiation service. *Information Sciences* 349 (2016), 248–262.

[35] Mahmoud Abo Khamis, Hung Q. Ngo, Christopher Ré, and Atri Rudra. 2015. Joins via Geometric Resolutions: Worst-case and Beyond. In *Proceedings of PODS 2015*. 213–228.

[36] Mahmoud Abo Khamis, Hung Q. Ngo, and Atri Rudra. 2016. FAQ: Questions Asked Frequently. In *Proceedings of PODS 2016*. 13–28.

[37] Phokion G. Kolaitis and Moshe Y. Vardi. 2000. Conjunctive-Query Containment and Constraint Satisfaction. *J. Comput. Syst. Sci.* 61, 2 (2000), 302–332.

[38] Dániel Marx. 2010. Approximating Fractional Hypertree Width. *ACM Trans. Algorithms* 6, 2, Article 29 (2010), 17 pages. https://doi.org/10.1145/1721837.1721845

[39] Dániel Marx. 2011. Tractable Structures for Constraint Satisfaction with Truth Tables. *Theory Comput. Syst.* 48, 3 (2011), 444–464.

[40] Dániel Marx. 2013. Tractable Hypergraph Properties for Constraint Satisfaction and Conjunctive Queries. *J. ACM* 60, 6 (2013), 42.

[41] Lukas Moll, Siamak Tazari, and Marc Thurley. 2012. Computing hypergraph width measures exactly. *Inf. Process. Lett.* 112, 6 (2012), 238–242. https://doi.org/10.1016/j.ipl.2011.12.002

[42] Dan Olteanu and Jakub Závodnỳ. 2015. Size bounds for factorised representations of query results. *ACM Trans. Database Syst.* 40, 1 (2015), 2.

[43] Norbert Sauer. 1972. On the density of families of sets. *J. Combinatorial Theory (A)* 13, 1 (1972), 145–147.

[44] Ayumi Shinohara. 1995. Complexity of Computing Vapnik-Chervonenkis Dimension and Some Generalized Dimensions. *Theor. Comput. Sci.* 137, 1 (1995), 129–144. https://doi.org/10.1016/0304-3975(94)00164-E

[45] Susan Tu and Christopher Ré. 2015. Duncecap: Query plans using generalized hypertree decompositions. In *Proceedings of SIGMOD 2015*. ACM, 2077–2078.

[46] René van Bevern, Rodney G. Downey, Michael R. Fellows, Serge Gaspers, and Frances A. Rosamond. 2015. Myhill-Nerode Methods for Hypergraphs. *Algorithmica* 73, 4 (2015), 696–729. https://doi.org/10.1007/s00453-015-9977-x

[47] Vladimir Vapnik and Alexey Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* 16 (1971), 264–280.

[48] Mihalis Yannakakis. 1981. Algorithms for Acyclic Database Schemes. In *Proceedings of VLDB 1981*. IEEE Computer Society, 82–94.

A NP-HARDNESS: EXAMPLE

Example A.1. Suppose that an instance of 3SAT is given by the propositional formula $\varphi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$, i.e.: we have $n = 3$ variables and $m = 2$ clauses. From this we construct a hypergraph $H = (V(H), E(H))$. First, we instantiate the sets Q, A, A', S, Y, and Y' from our problem reduction.

$$\begin{aligned}
A &= \{a_{(1,1)}, a_{(1,2)}, a_{(2,1)}, a_{(2,2)}, \dots, a_{(9,1)}, a_{(9,2)}\}, \\
A' &= \{a'_{(1,1)}, a'_{(1,2)}, a'_{(2,1)}, a'_{(2,2)}, \dots, a'_{(9,1)}, a'_{(9,2)}\}, \\
Q &= \{(1,1), (1,2), (2,1), (2,2), \dots, (9,1), (9,2)\} \cup \\
&\quad \{(0,1), (0,0), (1,0)\} \\
S &= Q \times \{1,2,3\} \times \{0,1\}, \\
Y &= \{y_1, y_2, y_3\}, \text{ and } Y' = \{y'_1, y'_2, y'_3\}.
\end{aligned}$$

According to our problem reduction, the set $V(H)$ of vertices of H is

$$\begin{aligned}
V(H) = S \cup A \cup A' \cup Y \cup Y' \cup \{z_1, z_2\} \cup \\
\{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\} \cup \\
\{a'_1, a'_2, b'_1, b'_2, c'_1, c'_2, d'_1, d'_2\}.
\end{aligned}$$

The set $E(H)$ of edges of H is defined in several steps. First, the edges in H_0 and H_0' are defined: We thus have the subsets $E_A, E_B, E_C, E_A', E_B', E_C' \subseteq E(H)$, whose definition is based on the sets $M_1 = S \setminus S_{(0,1)} \cup \{z_1\}$, $M_2 = Y \cup S_{(0,1)} \cup \{z_2\}$, $M_1' = S \setminus S_{(1,0)} \cup \{z_1\}$, and $M_2' = Y' \cup S_{(1,0)} \cup \{z_2\}$. The definition of the edges

$$\begin{aligned} e_p &= A_p' \cup \overline{A_p} \text{for } p \in \{(1,1),(1,2),\dots(8,1),(8,2),(9,1)\}, \\ e_{y_i} &= \{y_i, y_i'\} \qquad \text{for } 1 \le i \le 3, \\ e^0_{(0,0)} &= \{a_1\} \cup A \cup S \setminus S_{(0,0)} \cup Y \cup \{z_1\}, \\ e^1_{(0,0)} &= S_{(0,0)} \cup \{z_2\}, \\ e^0_{(9,2)} &= S \setminus S_{(9,2)} \cup \{z_1\}, \text{ and} \\ e^1_{(9,2)} &= \{a_1'\} \cup A' \cup S_{(9,2)} \cup Y' \cup \{z_2\} \end{aligned}$$

is straightforward. We concentrate on the edges $e_p^{k,0}$ and $e_p^{k,1}$ for $p \in \{(1,1),(1,2),\dots(8,1),(8,2),(9,1)\}$ and $k \in \{1,2,3\}$. These edges play the key role for covering the bags of the nodes along the "long" path π in any FHD or GHD of H. Recall that this path can be thought of as being structured in 9 blocks. Consider an arbitrary $i \in \{1,\dots,9\}$. Then $e^{k,0}_{(i,1)}$ and $e^{k,1}_{(i,1)}$ encode the k-th literal of the first clause and $e^{k,0}_{(i,2)}$ and $e^{k,1}_{(i,2)}$ encode the k-th literal of the second clause (the latter is only defined for $i \le 8$). These edges are defined as follows: the edges $e^{1,0}_{(i,1)}$ and $e^{1,1}_{(i,1)}$ encode the first literal of the first clause, i.e., the positive literal x_1. We thus have

$$\begin{aligned} e^{1,0}_{(i,1)} &= \overline{A_{(i,1)}} \cup (S \setminus S^{1,1}_{(i,1)}) \cup \{y_1,y_2,y_3\} \cup \{z_1\} \text{ and} \\ e^{1,1}_{(i,1)} &= A'_{(i,1)} \cup S^{1,1}_{(i,1)} \cup \{y_2',y_3'\} \cup \{z_2\} \end{aligned}$$

The edges $e^{2,0}_{(i,1)}$ and $e^{2,1}_{(i,1)}$ encode the second literal of the first clause, i.e., the negative literal $\neg x_2$. Likewise, $e^{3,0}_{(i,1)}$ and $e^{3,1}_{(i,1)}$ encode the third literal of the first clause, i.e., the positive literal x_3. Hence,

$$\begin{aligned} e^{2,0}_{(i,1)} &= \overline{A_{(i,1)}} \cup (S \setminus S^{2,1}_{(i,1)}) \cup \{y_1,y_3\} \cup \{z_1\}, \\ e^{2,1}_{(i,1)} &= A'_{(i,1)} \cup S^{2,1}_{(i,1)} \cup \{y_1',y_2',y_3'\} \cup \{z_2\} \\ e^{3,0}_{(i,1)} &= \overline{A_{(i,1)}} \cup (S \setminus S^{3,1}_{(i,1)}) \cup \{y_1,y_2,y_3\} \cup \{z_1\}, \text{ and} \\ e^{3,1}_{(i,1)} &= A'_{(i,1)} \cup S^{3,1}_{(i,1)} \cup \{y_1',y_2'\} \cup \{z_2\} \end{aligned}$$

Analogously, the edges $e^{1,0}_{(i,2)}$ and $e^{1,1}_{(i,2)}$ (encoding the first literal of the second clause, i.e., $\neg x_1$), the edges $e^{2,0}_{(i,2)}$ and $e^{2,1}_{(i,2)}$ (encoding the second literal of the second clause, i.e., x_2), and the edges $e^{3,0}_{(i,2)}$ and $e^{3,1}_{(i,2)}$ (encoding the third literal of the second clause, i.e., $\neg x_3$) are defined as follows:

$$\begin{aligned} e^{1,0}_{(i,2)} &= \overline{A_{(i,2)}} \cup (S \setminus S^{1,1}_{(i,2)}) \cup \{y_2,y_3\} \cup \{z_1\}, \\ e^{1,1}_{(i,2)} &= A'_{(i,2)} \cup S^{1,1}_{(i,2)} \cup \{y_1',y_2',y_3'\} \cup \{z_2\}, \\ e^{2,0}_{(i,2)} &= \overline{A_{(i,2)}} \cup (S \setminus S^{2,1}_{(i,2)}) \cup \{y_1,y_2,y_3\} \cup \{z_1\}, \\ e^{2,1}_{(i,2)} &= A'_{(i,2)} \cup S^{2,1}_{(i,2)} \cup \{y_1',y_3'\} \cup \{z_2\} \\ e^{3,0}_{(i,2)} &= \overline{A_{(i,2)}} \cup (S \setminus S^{3,1}_{(i,2)}) \cup \{y_1,y_2\} \cup \{z_1\}, \text{ and} \\ e^{3,1}_{(i,2)} &= A'_{(i,2)} \cup S^{3,1}_{(i,2)} \cup \{y_1',y_2',y_3'\} \cup \{z_2\}, \end{aligned}$$

where $S^{k,1}_{(i,j)}$ with $j \in \{1,2\}$ and $k \in \{1,2,3\}$ is defined as the singleton $S^{k,1}_{(i,j)} = \{((i,j) \mid k,1)\}$. The crucial property of these pairs of edges $e^{k,0}_{(i,j)}$ and $e^{k,1}_{(i,j)}$ is that they together encode the k-th literal of the j-th clause in the following way: if the literal is of the form x_l (resp. of the form $\neg x_l$), then $e^{k,0}_{(i,j)} \cup e^{k,1}_{(i,j)}$ covers all of $Y \cup Y'$ except for y_l' (resp. except for y_l).

Clearly, φ is satisfiable, e.g., by the truth assignment σ with $\sigma(x_1) = \text{true}$ and $\sigma(x_2) = \sigma(x_3) = \text{false}$. Hence, for the problem reduction to be correct, there must exist a GHD (and thus also an FHD) of width 2 of H. In Figure 2, the tree structure T plus the bags $(B_t)_{t \in T}$ of such a GHD is displayed. Moreover, in Table 1, the precise definition of B_t and λ_t of every node $t \in T$ is given. The set Z in the bags of this GHD is defined as $Z = \{y_i \mid \sigma(x_i) = \text{true}\} \cup \{y_i' \mid \sigma(x_i) = \text{false}\}$. In this example, for the chosen truth assignment σ, we thus have $Z = \{y_1, y_2', y_3'\}$. The bags B_t and the edge covers λ_t for each $t \in T$ are explained below.

The nodes u_C, u_B, u_A to cover the edges of the subhypergraph H_0 and the nodes u_A', u_B', u_C' to cover the edges of the subhypergraph H_0' are clear by Lemma 3.1. The purpose of the nodes $u_{\min \ominus 1}$ and $u_{\max}$ is mainly to make sure that each edge $\{y_i, y_i'\}$ is covered by some bag. Recall that the set Z contains exactly one of y_i and y_i' for every i. Hence, the node $u_{\min \ominus 1}$ (resp. $u_{\max}$) covers each edge $\{y_i, y_i'\}$, such that $y_i' \in Z$ (resp. $y_i \in Z$).

We now have a closer look at the nodes $u_{(1,1)}$ to $u_{(9,1)}$ on the "long" path π. More precisely, let us look at the nodes $u_{(i,1)}$ and $u_{(i,2)}$ for some $i \in \{1,\dots,8\}$, i.e., the "i-th block". It will turn out that the bags at these nodes can be covered by edges from H because φ is satisfiable. Indeed, our choice of $\lambda_{u_{(i,1)}}$ and $\lambda_{u_{(i,2)}}$ is guided by the literals satisfied by the truth assignment σ, namely: for $\lambda_{u_{(i,j)}}$, we have to choose some k_j, such that the k_j-th literal in the j-th clause is true in σ. For instance, we may define $\lambda_{u_{(i,1)}}$ and $\lambda_{u_{(i,2)}}$ as follows:

$$\lambda_{u_{(i,1)}} = \{e^{1,0}_{(i,1)}, e^{1,1}_{(i,1)}\} \qquad \lambda_{u_{(i,2)}} = \{e^{3,0}_{(i,2)}, e^{3,1}_{(i,2)}\}$$

The covers $\lambda_{u_{(i,1)}}$ and $\lambda_{u_{(i,2)}}$ were chosen because the first literal of the first clause and the third literal of the second clause are true in σ. Now let us verify that $\lambda_{u_{(i,1)}}$ and $\lambda_{u_{(i,2)}}$ are indeed covers of $B_{u_{(i,1)}}$ and $B_{u_{(i,2)}}$, respectively. By the definition of the edges $e^{k,0}_{(i,j)}, e^{k,1}_{(i,j)}$ for $j \in \{1,2\}$ and $k \in \{1,2,3\}$, it is immediate that $e^{k,0}_{(i,j)} \cup e^{k,1}_{(i,j)}$ covers $\overline{A_{(i,j)}} \cup A'_{(i,j)} \cup S \cup \{z_1, z_2\}$. The only non-trivial question is if $\lambda_{u_{(i,j)}}$ also covers Z. Recall that by definition, $(e^{1,0}_{(i,1)} \cup e^{1,1}_{(i,1)}) \supseteq (Y \cup Y') \setminus \{y_1'\}$. Our truth assignment σ sets $\sigma(x_1) = \text{true}$. Hence, by our definition of Z, we have $y_1 \in Z$ and $y_1' \notin Z$. This means that $e^{1,0}_{(i,1)} \cup e^{1,1}_{(i,1)}$ indeed covers Z and, hence, all of $B_{u_{(i,1)}}$. Note that we could have also chosen $\lambda_{u_{(i,1)}} = \{e^{2,0}_{(i,1)}, e^{2,1}_{(i,1)}\}$, since also the second literal of the first clause (i.e., $\neg x_2$) is true in σ. In this case, we would have $(e^{2,0}_{(i,1)} \cup e^{2,1}_{(i,1)}) \supseteq (Y \cup Y') \setminus \{y_2\}$ and Z indeed does not contain y_2. Conversely, setting $\lambda_{u_{(i,1)}} = \{e^{3,0}_{(i,1)}, e^{3,1}_{(i,1)}\}$ would fail, because in this case, $y_3' \notin (e^{3,0}_{(i,1)} \cup e^{3,1}_{(i,1)})$ since x_3 occurs positively in the first clause. On the other hand, we have $y_3' \in Z$ by definition of Z, because $\sigma(x_3) = \text{false}$ holds.

Checking that $\lambda_{u_{(i,2)}}$ as defined above covers Z is done analogously. Note that in the second clause, only the third literal is satisfied by σ. Hence, setting $\lambda_{u_{(i,2)}} = \{e^{3,0}_{(i,2)}, e^{3,1}_{(i,2)}\}$ is the only option to cover $B_{u_{(i,2)}}$ (in particular, to cover Z). Finally, note that σ as defined above is not the only satisfying truth assignment of φ. For instance, we could have chosen $\sigma(x_1) = \sigma(x_2) = \sigma(x_3) = \text{true}$. In this case, we would define $Z = \{y_1, y_2, y_3\}$ and the covers $\lambda_{u_{(i,j)}}$ would have to be chosen according to an arbitrary choice of one literal per clause that is satisfied by this assignment σ. ◇

B NP-HARDNESS: CLAIMS A-I

Proof of Claim A. We only show that none of the nodes u'_i with $i \in \{A, B, C\}$ is on the path from u_A to u_C. The other property is shown analogously. Suppose to the contrary that some u'_i is on the path from u_A to u_C. Since u_B is also on the path between u_A and u_C we distinguish two cases:

- Case (1) u'_i is on the path between u_A and u_B; then $\{b_1, b_2\} \subseteq B_{u'_i}$. This contradicts that u'_i is already full.
- Case (2) u'_i is on the path between u_B and u_C; then $\{c_1, c_2\} \subseteq B_{u'_i}$, which again contradicts that u'_i is already full.

The paths from u_A to u_C and from u'_A to u'_C are indeed disjoint. □

Proof of Claim B. Suppose there is a u_i (analog. for u'_i) for some $i \in \{A, B, C\}$, s.t. $u_i \in \mathit{nodes}(A \cup A', \mathcal{F})$; then there is some $a \in (A \cup A')$, s.t. $a \in B_{u_i}$. This contradicts the fact that u_i is full. □

Proof of Claim C. Suppose that no such path exists. Let $p \geq \min$ be the maximal value such that there is a path containing nodes $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_l$, which cover $e_{\min \ominus 1}, \ldots, e_p$ in this order. Clearly, there exists a node $\hat{u}$ that covers $e_{p\oplus 1} = A'_{p\oplus 1} \cup \overline{A_{p\oplus 1}}$. We distinguish four cases:

- Case (1): $\hat{u}$ is on the path from $\hat{u}_1$ to $\hat{u}_l$. Hence, $\hat{u}$ is between two nodes $\hat{u}_i$ and $\hat{u}_{i+1}$ for some $1 \leq i < l$ or $\hat{u} = \hat{u}_{i+1}$ for some $1 \leq i < l-1$. The following arguments hold for both cases. Now, there is some $q \leq p$, such that e_q is covered by $\hat{u}_{i+1}$ and $e_{q\ominus 1}$ is covered by $\hat{u}_i$. Therefore, $\hat{u}$ covers $\overline{A_q}$ either by the connectedness condition (if $\hat{u}$ is between $\hat{u}_i$ and $\hat{u}_{i+1}$) or simply because $\hat{u} = \hat{u}_{i+1}$. Hence, in total, $\hat{u}$ covers $A'_{p\oplus 1} \cup \overline{A_q}$ with $A'_{p\oplus 1} = \{a'_{\min}, \ldots, a'_{p\oplus 1}\}$ and $\overline{A_q} = \{a_q, a_{q\oplus 1}, \ldots, a_p, a_{p\oplus 1}, \ldots a_{\max}\}$. Then, $\hat{u}$ covers all edges $e_q, e_{q\oplus 1}, \ldots, e_{p\oplus 1}$. Therefore, the path containing nodes $\hat{u}_1, \ldots, \hat{u}_i, \hat{u}$ covers $e_{\min \ominus 1}, \ldots, e_{p\oplus 1}$ in this order, which contradicts the maximality of p.
- Case (2): There is a u^* on the path from $\hat{u}_1$ to $\hat{u}_l$, such that the paths from $\hat{u}_1$ to $\hat{u}$ and from $\hat{u}$ to $\hat{u}_l$ go through u^*. Then, u^* is either between $\hat{u}_i$ and $\hat{u}_{i+1}$ for some $1 \leq i < l$ or $u^* = \hat{u}_{i+1}$ for some $1 \leq i < l-1$. The following arguments hold for both cases. There is some $q \leq p$, such that e_q is covered by $\hat{u}_{i+1}$ and $e_{q\ominus 1}$ is covered by $\hat{u}_i$. By the connectedness condition, u^* covers
  - $A'_p = \{a'_{\min}, \ldots, a'_p\}$, since u^* is on the path from $\hat{u}$ to $\hat{u}_l$, and
  - $\overline{A}_q = \{a_q, \ldots, a_p, a_{p\oplus 1}, \ldots a_{\max}\}$, since u^* is on the path from $\hat{u_A}$ to $\hat{u}_{i+1}$ or $u^* = \hat{u}_{i+1}$.

  Then u^* covers all edges $e_q, e_{q\oplus 1}, \ldots, e_p$. Therefore, the path containing the nodes $\hat{u}_1, \ldots, \hat{u}_i, u^*, \hat{u}$ covers $e_{\min \ominus 1}, \ldots, e_{p\oplus 1}$ in this order, which contradicts the maximality of p.
- Case (3): $\hat{u}_1$ is on the path from $\hat{u}$ to all other nodes $\hat{u}_i$, with $1 < i \leq l$. By the connectedness condition, $\hat{u}_1$ covers A'_p. Hence, in total $\hat{u}_1$ covers $A'_p \cup A$ with $A'_p = \{a'_{\min}, \ldots, a'_p\}$ and $A = \{a_{\min}, \ldots, a_{\max}\}$. Then $\hat{u}_1$ covers all edges $e_{\min \ominus 1}, \ldots, e_p$. Therefore, the path containing nodes $\hat{u}_1$ and $\hat{u}$ covers $e_{\min \ominus 1}, \ldots, e_{p\oplus 1}$ in this order, which contradicts the maximality of p.
- Case (4): $\hat{u} = \hat{u}_1$, hence, $\hat{u}_1$ covers $A'_{p\oplus 1} \cup A$ with $A'_{p\oplus 1} = \{a'_{\min}, \ldots, a'_{p\oplus 1}\}$ and $A = \{a_{\min}, \ldots, a_{\max}\}$. Then, $\hat{u}_1$ covers all $e_{\min \ominus 1}, \ldots, e_{p\oplus 1}$, which contradicts the maximality of p. □

Proof of Claim D. Suppose to the contrary that the path from u_A to u'_A is disjoint from $\pi(\hat{u}_1, \hat{u}_N)$. We distinguish three cases:

- Case (1): u_A is on the path from u'_A to $\pi(\hat{u}_1, \hat{u}_N)$. Then, by the connectedness condition, u_A must contain a'_1, which contradicts the fact that u_A is full.
- Case (2): u'_A is on the path from u_A to $\pi(\hat{u}_1, \hat{u}_N)$. Analogously to Case (1), we get a contradiction by the fact that then u'_A must contain a_1.
- Case (3): There is a node u^* on the path from u_A to u'_A, which is closest to $\pi(\hat{u}_1, \hat{u}_N)$, i.e., u^* lies on the path from u_A to u'_A and both paths, the one connecting u_A with $\pi(\hat{u}_1, \hat{u}_N)$ and the one connecting u'_A with $\pi(\hat{u}_1, \hat{u}_N)$, go through u^*. Hence, by the connectedness condition, the bag of u^* contains $S \cup \{z_1, z_2, a_1, a'_1\}$. By Lemma 3.2, in order to cover $S \cup \{z_1, z_2\}$ with weight ≤ 2, we are only allowed to put non-zero weight on pairs of complementary edges. However, then it is impossible to achieve also weight ≥ 1 on a_1 and a'_1 at the same time. □

Proof of Claim E. First, we show that $\hat{u}$ and $\hat{u}'$ are indeed distinguished. Suppose towards a contradiction that they are not, i.e. $\hat{u} = \hat{u}'$. But then, by connectedness $\hat{u}$ has to cover $S \cup \{z_1, z_2, a_1, a'_1\}$. By Lemma 3.2, we know that, to cover $S \cup \{z_1, z_2\}$ with weight ≤ 2, we are only allowed to put non-zero weight on pairs of complementary edges. However, then it is impossible to achieve also weight ≥ 1 on a_1 and on a'_1 at the same time.

Second, suppose towards a contradiction that $\hat{u}$ is closer to u_N. As before, by connectedness $\hat{u}$ has to cover $S \cup \{z_1, z_2, a_1, a'_1\}$, which is impossible with weight ≤ 2. □

Proof of Claim F. First, it is easy to verify that $N \geq 2$ must hold. Otherwise, a single node would have to cover $\{e_{\min \ominus 1}, e_{\min}, e_{\min \oplus 1}, \ldots, e_{\max \ominus 1}, e_{\max}\}$ and also $S \cup \{z_1, z_2, a_1, a'_1\}$. However, we have already seen in Case (3) of the proof of Claim D that not even $S \cup \{z_1, z_2, a_1, a'_1\}$ can be covered using weight ≤ 2.

It remains to prove $N \geq 3$. Suppose to the contrary that $N = 2$. Observe that by the reduction every hypergraph has at least the edges $e_{\min \ominus 1}$, $e_{\min}$ and $e_{\max}$, and that $\hat{u}_1$ covers at least $e_{\min \ominus 1}$ and $\hat{u}_2$ covers at least $e_{\max}$. We distinguish 4 cases, based on the intersection with the path from u_A to u'_A. Recall nodes $\hat{u}$ and $\hat{u}'$ from Claim E.

- Case (1) - $\hat{u} = \hat{u}_1$ and $\hat{u}' = \hat{u}_2$: By connectedness and definition $\hat{u}_1$ covers the vertices $S \cup A \cup \{a_1, z_1, z_2\}$ and $\hat{u}_N$ covers the vertices $S \cup A' \cup \{a'_1, z_1, z_2\}$. The edge $e_{\min}$ is also covered in either $\hat{u}_1$ or $\hat{u}_2$. If $e_{\min}$ is covered in $\hat{u}_1$ then $\hat{u}_1$ has to cover additionally the vertex $a'_{\min}$ which is impossible with weight ≤ 2. Similar, if $e_{\min}$ is covered in $\hat{u}_2$ then $\hat{u}_2$ has to cover additionally the vertices A which is impossible with weight ≤ 2.

- Case (2) - $\hat{u} = \hat{u}_1$ and $\hat{u}'$ is on the path from $\hat{u}_1$ to $\hat{u}_2$: By connectedness and definition $\hat{u}_1$ covers the vertices $S \cup A \cup \{a_1, z_1, z_2\}$ and $\hat{u}'$ covers the vertices $S \cup \{a'_1, z_1, z_2\}$. The edge $e_{\min}$ is also covered in either $\hat{u}_1$ or $\hat{u}_2$. If $e_{\min}$ is covered in $\hat{u}_1$ then $\hat{u}_1$ has to cover additionally the vertex $a'_{\min}$ which is impossible with weight ≤ 2. Similar, if $e_{\min}$ is covered in $\hat{u}_2$ then by connectedness $\hat{u}'$ has to cover additionally the vertices A which is impossible with weight ≤ 2.
- Case (3) - $\hat{u}$ is on the path from $\hat{u}_1$ to $\hat{u}_2$ and $\hat{u}' = \hat{u}_2$: By connectedness and definition $\hat{u}$ covers the vertices $S \cup \{a_1, z_1, z_2\}$ and $\hat{u}_2$ covers the vertices $S \cup A' \cup \{a'_1, z_1, z_2\}$. The edge $e_{\min}$ is also covered in either $\hat{u}_1$ or $\hat{u}_2$. If $e_{\min}$ is covered in $\hat{u}_1$ then by connectedness $\hat{u}$ has to cover additionally the vertex $a'_{\min}$ which is impossible with weight ≤ 2. Similar, if $e_{\min}$ is covered in $\hat{u}_2$ then $\hat{u}_2$ has to cover additionally the vertices A which is impossible with weight ≤ 2.
- Case (4) - $\hat{u}$ is on the path from $\hat{u}_1$ to $\hat{u}_2$ and $\hat{u}'$ is on the path from $\hat{u}_1$ to $\hat{u}_2$: By connectedness and definition $\hat{u}$ covers the vertices $S \cup \{a_1, z_1, z_2\}$ and $\hat{u}'$ covers the vertices $S \cup \{a'_1, z_1, z_2\}$. The edge $e_{\min}$ is also covered in either $\hat{u}_1$ or $\hat{u}_2$. If $e_{\min}$ is covered in $\hat{u}_1$ then by connectedness $\hat{u}$ has to cover additionally the vertex $a'_{\min}$ which is impossible with weight ≤ 2. Similar, if $e_{\min}$ is covered in $\hat{u}_2$ then by connectedness $\hat{u}'$ has to cover additionally the vertices A which is impossible with weight ≤ 2.

Hence, the path $\pi(\hat{u}_1, \hat{u}_N)$ has at least 3 nodes $\hat{u}_i$. □

Proof of Claim G. We have to show that $\hat{u}_2$ is on the path from u_A to any node $\hat{u}_i$ with $i > 2$ and $\hat{u}_{N-1}$ is on the path from u'_A to any $\hat{u}_i$ with $i < N-1$. We only prove the first property since the two properties are symmetric. Suppose to the contrary that there exists some $i > 2$ such that $\hat{u}_2$ is *not* on the path from u_A to $\hat{u}_i$. We distinguish two cases:

- Case (1): $\hat{u}_N$ is on the path from $\hat{u}_2$ to u_A. Then $\hat{u}_N$ is also on the path from $\hat{u}_1$ to u_A. Hence, by the connectedness condition, $\hat{u}_N$ has to cover the following (sets of) vertices:
  - a_1, since the path between $\hat{u}_1$ and u_A goes through $\hat{u}_N$,
  - $S \cup \{z_1, z_2\}$, since the path from u_A to u'_A passes $\pi(\hat{u}_1, \hat{u}_N)$,
  - A', since $\hat{u}_N$ covers $e_{\max} = e^1_{\max}$.

  By Lemma 3.2, we know that, to cover $S \cup \{z_1, z_2\}$ with weight ≤ 2, we are only allowed to put non-zero weight on pairs of complementary edges. However, then it is impossible to achieve also weight ≥ 1 on A' and on a_1 at the same time.
- Case (2): There is some $\hat{u}$ on the path from $\hat{u}_i$ to $\hat{u}_{i+1}$ for some i with $2 \leq i < N$, such that u_A is closest to $\hat{u}$ among all nodes on $\pi(\hat{u}_1, \hat{u}_N)$. This also includes the case that $\hat{u} = \hat{u}_i$ holds. By definition of $\hat{u}_i$ and $\hat{u}_{i+1}$, there is a $p \in [2n+3; m]$, such that both $\hat{u}_i$ and $\hat{u}_{i+1}$ cover a'_p. Then, by the connectedness condition, $\hat{u}$ covers the following (sets of) vertices:
  - a'_p, since $\hat{u}$ is on the path from $\hat{u}_i$ to $\hat{u}_{i+1}$,
  - a_1, since $\hat{u}$ is on the path from $\hat{u}_1$ to u_A,
  - $S \cup \{z_1, z_2\}$, since $\hat{u}$ is on the path from u_A to u'_A.

  Again, by Lemma 3.2, we know that, to cover $S \cup \{z_1, z_2\}$ with weight ≤ 2, we are only allowed to put non-zero weight on pairs of complementary edges. However, then it is impossible to achieve also weight ≥ 1 on a'_p and a_1 at the same time. □

Proof of Claim H. We prove this property for the "outer nodes" $\hat{u}_1, \hat{u}_N$ and for the "inner nodes" $\hat{u}_2 \cdots \hat{u}_{N-1}$ separately.

We start with the "outer nodes". The proof for $\hat{u}_1$ and $\hat{u}_N$ is symmetric. We thus only work out the details for $\hat{u}_1$. Suppose to the contrary that $\hat{u}_1$ not only covers $e_{\min \ominus 1}$ but also some further edges e_p with $p \geq \min$. Then it also covers $e_{\min}$. We distinguish two cases:

- Case (1): $\hat{u}_1$ is on the path from u_A to $\hat{u}_2$. Then, $\hat{u}_1$ has to cover the following (sets of) vertices:
  - $S \cup \{z_1, z_2\}$, since $\hat{u}_1$ is on the path from u_A to u'_A.
  - a_1, since $\hat{u}_1$ covers $e_{\min \ominus 1}$,
  - $a'_{\min}$, since $\hat{u}_1$ covers $e_{\min}$.

  By applying Lemma 3.2, we may conclude that the set $S \cup \{z_1, z_2, a_1, a'_{\min}\}$ cannot be covered by a fractional edge cover of weight ≤ 2.
- Case (2): There is some $\hat{u}$ on the path from $\hat{u}_1$ to $\hat{u}_2$, such that $\hat{u} \neq \hat{u}_1$ and u_A is closest to $\hat{u}$ among all nodes on $\pi(\hat{u}_1, \hat{u}_N)$. Then $\hat{u}$ has to cover the following (sets of) vertices:
  - $S \cup \{z_1, z_2\}$, since $\hat{u}$ is on the path from u_A to u'_A,
  - a_1, since $\hat{u}$ is on the path from u_A to $\hat{u}_1$,
  - $a'_{\min}$, since $\hat{u}$ is on the path from $\hat{u}_1$ to $\hat{u}_2$.

  As in Case (1) above, $S \cup \{z_1, z_2, a_1, a'_{\min}\}$ cannot be covered by a fractional edge cover of weight ≤ 2 due to Lemma 3.2.

It remains to consider the "inner" nodes $\hat{u}_i$ with $2 \leq i \leq N-1$. Each such $\hat{u}_i$ has to cover $S \cup \{z_1, z_2\}$ since all these nodes are on the path from u_A to u'_A by Claim E. Now suppose that $\hat{u}_i$ covers $e_p = A'_p \cup \overline{A_p}$ for some $p \in \{e_{\min}, \ldots, e_{\max \ominus 1}\}$. By Lemma 3.3, covering all of the vertices $A'_p \cup \overline{A_p} \cup S \cup \{z_1, z_2\}$ by a fractional edge cover of weight ≤ 2 requires that we put total weight 1 on the edges $e^{k,0}_p$ and total weight 1 on the edges $e^{k,1}_p$ with $k \in \{1, 2, 3\}$. However, then it is impossible to cover also $e_{p'}$ for some p' with $p' \neq p$. This concludes the proof of Claim F. □

Proof of Claim I. For each j, there is a node $u_{(s,j)}$ between $u_{(s,1)}$ and $u_{(s+1,1)}$, such that $B_{u_{(s,j)}} \supseteq A'_{(s,j)} \cup \overline{A_{(s,j)}} \cup S \cup \{z_1, z_2\}$. Now let $p = (s, j)$. Observe that, by the definition of FHDs, we have that $B_{u_p} \subseteq B(\gamma_{u_p})$ and, by $X_s \subseteq B_{u_p}$, also $X_s \subseteq B(\gamma_{u_p})$ holds. By Lemma 3.3, the only way to cover B_{u_p} with weight ≤ 2 is by using edges $e^{k,0}_p$ and $e^{k,1}_p$ with $k \in \{1, 2, 3\}$. Therefore, $\gamma_{u_p}(e^{k,0}_p) > 0$ for some k. Now suppose $L^k_p = x_l$. By Lemma 3.2, we also have that $\gamma_{u_p}(e^{k,1}_p) > 0$ and therefore the weight on y'_l is less than 1, which means that $y'_l \notin B(\gamma_{u_p})$ and consequently $y'_l \notin X_s$. Since this implies that $y_l \in X_s$, we have that $\sigma(x_l) = 1$. Conversely, suppose $L^k_p = \neg x_l$. Since $\gamma_{u_p}(e^{k,0}_p) > 0$, the weight on y_l is less than 1, which means that $y_l \notin B(\gamma_{u_p})$ and consequently $y_l \notin X_s$. Hence, we have $\sigma(x_l) = 0$. In either case, L^k_p is satisfied by σ and therefore, σ satisfies φ. □