Further Tractability Results for Fractional Hypertree Width

Wolfgang Fischl[1], Georg Gottlob[2], and Reinhard Pichler[3]

1 **TU Wien, Austria**
`wfischl@dbai.tuwien.ac.at`
2 **TU Wien, Austria & University of Oxford, UK**
`gottlob@dbai.tuwien.ac.at` & georg.gottlob@cs.ox.ac.uk
3 **TU Wien, Austria**
`pichler@dbai.tuwien.ac.at`

Abstract

The authors have recently shown that recognizing low fractional hypertree-width (fhw) is NP-complete in the general case and that the problem becomes tractable if the hypergraphs under consideration have degree and intersection width bounded by a constant, i.e., every vertex is contained in only constantly many different edges and the intersection of two edges contains only constantly many vertices. In this article, we show that bounded degree alone suffices to ensure tractability.

1998 ACM Subject Classification H.2.4 Systems - Query processing

Keywords and phrases Conjunctive queries; hypergraphs; query evaluation; tractability; acyclic queries; fractional hypertree width; decompositions

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Answering Conjunctive Queries (CQs) and solving Constraint Satisfaction Problems (CSPs) are two fundamental tasks in computer science. They are classical NP-complete problems [6]. Consequently, it has been a very active research area over several decades to identify tractable cases of these problems – with various hypergraph decomposition methods being the most powerful ones [14, 16]. To date, the biggest known tractable class of CQ answering and CSP solving is obtained by restricting the *fractional hypertree width* (*fhw*) of the underlying hypergraph by a constant. However, as we have recently shown [11], *fhw* has a drawback in that it is NP-complete to recognize low *fhw*. Formally, for *decomposition* $\in$ {HD, GHD, FHD} and $k \geq 1$, we have studied 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.

Here, *decomposition* $\in$ {HD, GHD, FHD} means that we are interested in the Check-problem for hypertree decompositions (HDs), generalized hypertree decompositions (GHDs), and fractional hypertree decompositions (FHDs), respectively. The Check(HD, k) problem is known to be tractable for any constant $k \geq 1$, while Check(GHD, k) was shown in [15] to be NP-complete for any $k \geq 3$. In [11], we have shown that NP-completeness for GHDs holds even for $k = 2$ and, more importantly, we have shown NP-completeness for FHDs for $k \geq 2$.

These NP-completeness results raise the question for meaningful tractable fragments of the Check-problem in case of GHDs and FHDs. In [11], we have shown several such tractability results centered around the notions of *bounded degree* and *bounded intersection width*, which are defined as follows:

▸ **Definition 1.** The *degree* d of a hypergraph $H = (V(H), E(H))$ 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 $\mathcal{C}$ of hypergraphs has bounded degree, if there exists $d \geq 1$, such that every hypergraph $H \in \mathcal{C}$ has degree $\leq d$.

▸ **Definition 2.** [11] 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.

We say that a class $\mathcal{C}$ of hypergraphs 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.

The following recent tractability results form the starting point of the current article:

▸ **Theorem 3.** *[11] For every hypergraph class $\mathcal{C}$ that has bounded degree, and for every constant $k \geq 1$, the* Check*(GHD, k) problem is tractable., i.e., given a hypergraph $H \in \mathcal{C}$, it is feasible in polynomial time to check ghw$(H) \leq k$ and, if this holds, to compute a GHD of width $\leq k$ of H.*

▸ **Theorem 4.** *[11] For every hypergraph class $\mathcal{C}$ that enjoys the BIP, and for every constant $k \geq 1$, the* Check*(GHD, k) problem is tractable, i.e., given a hypergraph $H \in \mathcal{C}$, it is feasible in polynomial time to check ghw$(H) \leq k$ and, if this holds, to compute a GHD of width $\leq k$ of H.*

▸ **Theorem 5.** *[11] For every hypergraph class $\mathcal{C}$ that has bounded degree and enjoys the BIP, and for every constant $k \geq 1$, the* Check*(FHD, k) problem is tractable, i.e., given a hypergraph $H \in \mathcal{C}$, it is feasible in polynomial time to check fhw$(H) \leq k$ and, if this holds, to compute an FHD of width $\leq k$ of H.*

In other words, while bounded degree alone and also the BIP alone suffices to make the Check(GHD, k) problem tractable, we had to impose both restrictions simultaneously to achieve tractability of the Check(FHD, k) problem. In this article, we strengthen the latter result by showing that bounded degree alone suffices to ensure tractability of Check(FHD, k). Formally, we thus get the following **main result** of this article:

▸ **Theorem 6.** *For every hypergraph class $\mathcal{C}$ that has bounded degree and for every constant $k \geq 1$, the* Check*(FHD, k) problem is tractable, i.e., given a hypergraph $H \in \mathcal{C}$, it is feasible in polynomial time to check fhw$(H) \leq k$ and, if this holds, to compute an FHD of width $\leq k$ of H.*

To prove this result, we proceed as follows. In Section 2, we recall some basic definitions and fix some notation. In Section 3, we make use of a classical result on fractional vertex covers to derive a constant (depending only on width k and degree d) upper bound on the support of fractional edge covers in case of bounded degree. In Section 4, we apply the notion of subedge functions (which was crucial for proving Theorems 3 and 4 in [11]) to FHDs. This will then allow us to prove our main result in Section 5. Finally, in Section 6, we conclude and outline some directions for future research.

2 Preliminaries

We refer the reader to our recent article [11] for most of the basic definitions needed also in this work. Below, we recall some crucial definitions and introduce some additional ones mainly to fix the terminology.

▸ **Definition 7.** We make use of the following notions related to egde-weight functions and vertex-weight functions for a hypergraph $H = (V(H), E(H))$.

- An edge-weight function γ for a hypergraph H assigns a weight $\gamma(e) \geq 0$ to each edge e of H. We say γ is a *fractional edge cover* of H if for each vertex $v \in V(H)$, $\Sigma_{v\in e}\gamma(e) \geq 1$. For an edge-weight function γ for hypergraph H, we denote by $\mathit{weight}(\gamma)$ its total weight, i.e. $\Sigma_{e\in E(H)}\gamma(e)$.
- A vertex-weight function w for a hypergraph H assigns a weight $w(v) \geq 0$ to each vertex v of H. We say w is a *fractional vertex cover* of H if for each edge $e \in E(H)$, $\Sigma_{v\in e}w(v) \geq 1$. For a vertex-weight function w for hypergraph H, we denote by $\mathit{weight}(w)$ its total weight, i.e. $\Sigma_{v\in V(H)}w(v)$.
- The fractional edge-cover number $\rho^*(H)$ is defined as the minimum $\mathit{weight}(\gamma)$ over all fractional edge covers of H. Likewise, the fractional vertex cover number $\tau^*(H)$ is defined as the minimum $\mathit{weight}(w)$ where w ranges over all fractional vertex covers of H.
- The ρ-support $\mathit{supp}_\rho(\gamma)$ (or simply, the support $\mathit{supp}(\gamma)$) of a hypergraph H under an edge-weight function γ is defined as $\mathit{supp}_\rho(\gamma) = \mathit{supp}(\gamma) = \{e \in E(H) \mid \gamma(e) > 0\}$.
- The τ-support $\mathit{supp}_\tau(w)$ of a hypergraph H under a vertex-weight function w is defined as $\mathit{supp}_\tau(w) = \{v \in V(H) \mid w(v) > 0\}$.
- A class $\mathcal{C}$ of hypergraphs has bounded ρ^*-support if there is a constant c such that for every hypergraph $H \in \mathcal{C}$, there exists an edge-weight function γ with $\mathit{weight}(\gamma) = \rho^*(H)$ and $\mathit{supp}_\rho(\gamma) \leq c$. The corresponding definition of bounded τ^*-support is analogous. ◇

We now recall the definition of FHDs and some related crucial notions:

▸ **Definition 8.** Let $H = (V(H), E(H))$ be a hypergraph and let $\gamma\colon E(H) \to [0,1]$ be an edge-weight function for H. For $v \in V(H)$, we write $\gamma(v)$ to denote the total weight that γ assigns to v. Moreover, we write by $B(\gamma)$ to denote the set of all vertices *covered* by γ, i.e.:

$$\gamma(v) = \sum_{e\in E(H), v\in e} \gamma(e).$$

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

▸ **Definition 9.** Let $H = (V(H), E(H))$ be a hypergraph. A *fractional hypertree decomposition* (FHD) of H is a tuple $\langle T, (B_u)_{u\in N(T)}, (\gamma)_{u\in N(T)} \rangle$, 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 $u \in N(T)$, γ_u is an edge-weight function $\gamma_u\colon E(H) \to [0,1]$ with $B_u \subseteq B(\gamma_u)$. ◇

The width of an FHD is the maximum weight of the functions γ_u, over all nodes u in T. Moreover, the fractional hypertree width of H (denoted $\mathit{fhw}(H)$) is the minimum width over all FHDs of H. Condition 2 is often called the "connectedness condition" and the set B_u is usually referred to as the "bag" at node u. Note that, in contrast to hypertree decompositions (HDs), the underlying tree T of an FHD does not need to be rooted. For the sake of uniformity, we assume that also the tree underlying an FHD is rooted with the understanding that the root is arbitrarily chosen. Finally, by slight abuse of notation, we shall write $u \in T$ short for $u \in N(T)$. Hence, FHDs will be referred to as $\langle T, (B_u)_{u \in T}, (\gamma_u)_{u \in T}\rangle$ or, simply as $\langle T, (B_u), (\gamma_u)\rangle$.

In our tractability proof of Check(FHD, k), we will make heavy use of certain unions and intersections of sets of vertices. We thus introduce the following notation.

▸ **Definition 10.** Let S be a family of sets.

$\Cup S$ denotes the set-family obtained from S by adding to it all possible unions of an arbitrary number of sets from S. (Note that $|\Cup S| \leq 2^{|S|}$).

$\Cup_i S$ for an integer $i \geq 1$, denotes the set-family obtained from S by adding to it all possible unions of $\leq i$ sets from S. (Note that $|\Cup_i (S)| \leq |S|^{i+1}$).

$\Cap S$ denotes the set-family obtained from S by adding to it all possible intersections of an arbitrary number of sets from S. (Note that $|\Cap S| \leq 2^{|S|}$).

$\Cap_i S$ for an integer $i \geq 1$, denotes the set-family obtained from S by adding to it all possible intersections of $\leq i$ sets from S. (Note that $|\Cap_i S| \leq |S|^{i+1}$).

If S and S' are both families of sets, then $S \Cap S'$ denotes the *pointwise intersection* between S and S', i.e., $S \Cap S' = \{A \cap B \mid A \in S \text{ and } B \in S'\}$. ◇

We sometimes identify sets of edges with hypergraphs. If a set of edges E is used, where instead a hypergraph is expected, then we mean the hypergraph (V, E), where V is simply the union of all edges in E. Finally, for a set E of edges, it is convenient to write $\bigcup E$ (and $\bigcap E$, respectively) to denote the set of vertices obtained by taking the union (or the intersection, respectively) of the edges in E.

3 Bounded Support

In this section we show that, for every FHD $\mathcal{F}$ of width k of a hypergraph H of degree $\leq d$, there exists an FHD $\mathcal{F}'$ of width $\leq k$ of H satisfying the following important property: for every node u in the FHD $\mathcal{F}'$, the edge cover γ_u has support *supp*(γ) bounded by a constant that depends only on k and d.

To derive this result, we will exploit the well-known dualities $\rho^*(H) = \tau^*(H^d)$ and $\tau^*(H) = \rho^*(H^d)$, where H^d denotes the dual of H. To make optimal use of this, we make, for the moment, several assumptions. First of all, we will assume w.l.o.g. that (1) hypergraphs do not have isolated vertices and (2) hypergraphs do not have empty edges. In fact for hypergraphs with isolated vertices (empty edges), ρ^* (τ^*) would be undefined or at least not finite. Furthermore, we make the following temporary assumptions. Assume that (3) hypergraphs have never two vertices of the same edge-type (i.e., the two vertices occur in precisely the same edges) and (4) hypergraphs have never two edges of the same "vertex-type" (i.e., there do not exist duplicate edges).

Assumptions (1) – (4) can be safely made and, for the time being, we will make them here. Recall that we are ultimately interested in the computation of an FHD of width $\leq k$ for given k. As mentioned above, without assumption (1), the computation of an edge-weight function and, hence, of an FHD of width $\leq k$ makes no sense. Assumption (2) does not restrict the search for a specific FHD since we would never define an edge-weight function with non-zero weight on an empty edge. As far as assumption (3) is concerned, suppose that a hypergraph H has groups of multiple vertices of identical edge-type. Then it is sufficient to consider the reduced hypergraph H^- resulting from H by " fusing" each such group to a single vertex. Obviously $\rho^*(H) = \rho^*(H^-)$, and each edge-weight function for H^- can be extended in the obvious way to an edge-weight function of the same total weight to H. Finally, assumption (4) also can be made w.l.o.g., since we can again define a reduced hypergraph H^- resulting from H by retaining only one edge from each group of identical edges. Then every edge cover of H^- is an edge cover of H. Conversely, every edge cover of H can be turned into an edge cover of H^- by assigning to each edge e in H^- the sum of the weights of e and all edges identical to e in H.

Under our above assumptions (1) – (4), for every hypergraph H, the property $H^{dd} = H$ holds and there is an obvious one-to-one correspondence between the edges (vertices) of H and the vertices (edges) of H^d. Moreover, there is an obvious one-to-one correspondence between the fractional edge covers of H and the fractional vertex covers of H^d. In particular, if there is a fractional edge cover γ for H, then its corresponding "dual" γ^d assigns to each vertex v of H^d the same weight as to the edge in H that is represented by this vertex and vice versa.

Note that if we do not make assumptions (3) and (4), then there are hypergraphs H with $H^{dd} \neq H$. For instance, consider the hypergraph H_0 with $V(H_0) = \{a,b,c\}$ and $E(H_0) = \{\, e = \{a,b,c\}\,\}$, i.e., property (3) is violated. The hypergraph H_0^d has a unique vertex e and a unique hyperedge $\{e\}$. Hence, H_0^{dd} is (isomorphic to) the hypergraph with a unique vertex a and a unique hyperedge $\{a\}$, which is clearly different from the original hypergraph H_0.

To get an upper bound on the support $\mathit{supp}(\gamma)$ of a fractional edge cover of a hypergraph H, we make use of the following (dual) result for fractional vertex covers. This result is due to Zoltán Füredi [13], who extended earlier results by Chung et al. [8]. Below, we appropriately reformulate Füredi's result for our purposes:

▸ **Proposition 11** ([13], page 152, Proposition 5.11.(iii))**.** *For every hypergraph H of rank (i.e., maximal edge size) r, and every fractional vertex cover w for H satisfying $\mathit{weight}(w) = \tau^*(H)$, the property $|\mathit{supp}_\tau(w)| \leq r \times \tau^*(H)$ holds.*

By duality, exploiting the relationship $\rho^*(H) = \tau^*(H^d)$ and by recalling that the degree of H corresponds to the rank of H^d, we immediately get the following corollary:

▸ **Corollary 12.** *For every hypergraph H of degree d, and every fractional edge cover γ for H satisfying $\mathit{weight}(\gamma) = \rho^*(H)$, the property $|\mathit{supp}_\rho(\gamma)| \leq d \times \rho^*(H)$ holds.*

From now on, we no longer need to make the assumptions (3) and (4) above. In fact, Proposition 11 and Corollary 12 also hold for hypergraphs that do not fulfill these conditions as was pointed out above by our considerations on reduced hypergraphs H^-. Moreover, from now on, we exclusively concentrate on fractional *edge* covers. The excursion to fractional *vertex* covers was only needed to make use of Füredi's result reformulated in Proposition 11 above. Hence, in the sequel, we shall simply write $\mathit{supp}(\gamma)$ rather than $\mathit{supp}_\rho(\gamma)$, since no confusion with supp_τ can arise.

Proposition 11 and Corollary 12 state bounded support properties for the *optimal* weight functions τ^* and ρ^*. Note, however, that fractional hypertree decompositions $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ deviate from this setting in two ways: at decomposition node u in $\mathcal{F}$ (i) not every vertex of H is necessarily covered, i.e., we have $B_u \subseteq B(\gamma_u)$ but, of course, we do not necessarily have $V(H) \subseteq B(\gamma_u)$, and (ii) the edge-weight functions at single decomposition nodes are not required to be optimal in any sense with respect to whatever subhypergraph of H is considered. To deal with these issues, we state the following definition and derive a lemma.

▸ **Definition 13.** For a given hypergraph H and edge-weight function γ, we say that a vertex of $V(H)$ is "in" if the total weight $\gamma(v) \geq 1$, i.e., $v \in B(\gamma)$. Otherwise we say that v is "out". We shall write $OUT(\gamma)$ to denote the set $V(H) \setminus B(\gamma)$ of vertices which are "out".

The following lemma allows us to extend the upper bound $k \times d$ on the support of a fractional edge cover γ of width k of a hypergraph of degree d to the fractional edge cover γ_u in every node u of an FHD of width k of a hypergraph of degree d.

▸ **Lemma 14.** *Let H be a hypergraph of degree d and let $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ be an FHD of H of width k. Then there exists an FHD $\mathcal{F}' = \langle T, (B_u), (\gamma'_u)\rangle$ of H of width $\leq k$ such that for every decomposition node u in T, we have $|supp(\gamma'_u)| \leq k \times d$ and, moreover, $\mathcal{F}$ and $\mathcal{F}'$ have exactly the same tree structure T and $B(\gamma'_u) = B(\gamma_u)$ for every u.*

Proof. Let H and $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ be as above. For each decomposition node u, consider the sub-hypergraph H_u of H where $V(H_u) = B(\gamma_u)$ and $E(H_u) = \{e \cap V(H_u) \,|\, e \in supp(\gamma_u)\} = \{e \cap B(\gamma_u) \,|\, e \in supp(\gamma_u)\}$. Note that one or more edges from $supp(\gamma_u)$ may give rise to a same edge e' of H_u, when deleting out-vertices from the support edges. We call all such edges the *originators* of e' and denote the set of all originator edges for e' by $orig(e')$.

Now let $\gamma_u^{\downarrow} : E(H_u) \rightarrow (0,1]$ be the edge-weight function which assigns each edge e' of H_u weight $\gamma_u^{\downarrow}(e') = \Sigma_{e \in orig(e')}\gamma_u(e)$, i.e., the sum of all weights by γ_u of its originators. Clearly, $\gamma_u^{\downarrow}$ is an edge cover of total weight at most k for H_u. Now take an optimal fractional edge cover γ_u^* for H_u. The total weight of this cannot be greater than k either. Hence, by Corollary 12, $supp(\gamma_u^*) \leq k \times d$. Now transform γ_u^* into an edge-weight function γ'_u of the entire hypergraph H by assigning for edge e' of H_u the entire weight of e' to only one of its originators, whilst assigning zero to all other originators. Clearly, the support of γ'_u is bounded by $k \times d$ and $B(\gamma'_u) = B(\gamma_u)$. B_u is thus covered by $B(\gamma'_u)$, and thus our FHD $\mathcal{F}' = \langle T, (B_u), (\gamma'_u)\rangle$ is now fully specified and has all requested properties. ◂

4 Subedge Functions

Towards a polynomial-time algorithms for deciding the CHECK(FHD, k) problem, the upper bound on the support of each edge-weight function γ_u will be of great help. However, we yet have to overcome the following obstacle: in the alternating algorithm in [14] for deciding the CHECK(HD, k) problem, we guess at every node u of the hypertree decomposition a set S_u of edges with $|S_u| \leq k$ such that the edges in S get weight 1 by γ_u and all other edges get weight 0. Hence, we get $B(\gamma_u) = \bigcup S_u$. From this, we determine the bag $B_u \subseteq B(\gamma_u)$ via the so-called *special condition* [14], which distinguishes HDs from GHDs. More specifically, let u' denote the parent of u in the hypertree decomposition and let C denote the vertices in the edges that have to be covered by some node in the subtree rooted at u. Then we may set $B_u = B(\gamma_u) \cap (B_{u'} \cup C)$.

In our case, when trying to construct a *fractional* hypertree decomposition of width $\leq k$ for a hypergraph with degree bounded by d, we know by Lemma 14 that we may restrict ourselves to edge-weight functions γ_u with $|supp(\gamma_u)| \leq k \times d$. Moreover, we can be sure that $B(\gamma_u) \subseteq \bigcup S$ with $S = supp(\gamma_u)$ holds. However, in contrast to the HD-setting studied in [14], $B(\gamma_u) = \bigcup S$ does in general *not* hold. Consequently, it is, of course, also unclear how to determine B_u. In this section, we provide a solution to both problems: how to determine $B(\gamma_u)$ and how to determine B_u for each node u in an FHD?

Towards solving the first problem, we establish a bound on the number of possible sets $B(\gamma)$ that can arise in a hypergraph when varying the weight function γ.

▸ **Definition 15.** Let $\mathit{INSET}(H)$ denote the set of all possible sets $B(\gamma)$ such that γ is an edge-weight function. (Recall that we refer to the vertices in $B(\gamma)$ as the vertices that are "in" and that, consequently, $B(\gamma)$ is now referred to as an "in"-set).

▸ **Definition 16.** An *intersection type* of a hypergraph $H = (V(H), E(H))$, short "type", is a set of edges of H, i.e., a subset of $E(H)$. For a hypergraph H, $\mathit{TYPES}(H) = 2^{E(H)}$ consists of all possible types of H.

For a type $t \in \mathit{TYPES}(H)$ define its class $\mathit{class}(t) = \bigcap_{e \in t} e$ as the intersection of all edges in t. The set of all classes of H is denoted by $\mathit{CLASSES}(H)$. For a class $K \in \mathit{CLASSES}(H)$ there may be more than one type t with $\mathit{class}(t) = K$, however there is only one maximal type, namely $\{e' \in E(H) \mid K \subseteq e'\}$; we denote by $\mathit{type}(K)$ this unique maximal type. ◇

Note that $\mathit{TYPES}(H)$ and $\mathit{CLASSES}(H)$ depend only on H and not on any edge-weight function. Moreover, every set $B(\gamma)$, for whatever edge-weight function, must be equal to the union of some classes of H. In fact, for any particular edge-weight function γ, the set $B(\gamma)$ consists of the union of all sets $\mathit{class}(t)$ for all types t that satisfy $\gamma(t) \geq 1$ where $\gamma(t) = \Sigma_{e \in t}\gamma(e)$. Finally, the inequality $|\mathit{CLASSES}(H)| \leq |\mathit{TYPES}(H)|$ clearly holds. We thus get the following lemma.

▸ **Lemma 17.** *Let H be a hypergraph. Then the following properties hold:*

1. *If γ is an edge-weight function, then $B(\gamma) \in \Cup \mathit{CLASSES}(H)$.*
2. *$\mathit{INSET}(H) \subseteq \Cup \mathit{CLASSES}(H)$.*
3. *$|\mathit{INSET}(H)| \leq 2^{|\mathit{CLASSES}(H)|} \leq 2^{|\mathit{TYPES}(H)|} \leq 2^{2^{|E(H)|}}$ and all three sets, INSET, $\mathit{CLASSES}$ and TYPES, can be computed from H in polynomial time if the cardinality of $E(H)$ is bounded by a constant.*

Clearly, the above inclusion $\mathit{INSET}(H) \subseteq \Cup \mathit{CLASSES}(H)$ only gives us an *exponential* upper bound $2^{2^{|E(H)|}}$ on the number of possible "in"-sets $B(\gamma_u)$ at any node u in an FHD. Hence, in a polynomial-time algorithm, we cannot afford to iterate through all of these candidates. However, by Lemma 14, we may assume w.l.o.g. that $|S_u| \leq k \times d$ with $S_u = \mathit{supp}(\gamma_u)$ holds for every edge-weight function γ_u of interest. Hence, there are only *polynomially* many sets $S_u \subseteq E(H)$ with $|S_u| \leq k \times d$ to be considered for the support $\mathit{supp}(\gamma_u)$. Moreover, for each S_u, there exist only *polynomially* many (namely at most $2^{2^{|S_u|}}$) possible "in"-sets $B(\gamma_u)$ with $\mathit{supp}(\gamma_u) = S_u$. Hence, with Lemma 17, the first problem stated above is essentially solved.

It remains to find a solution to the second problem stated above, i.e., how to determine B_u for each node u in an FHD of width k? We tackle this problem by adopting the idea of *subedge functions* as described in [15] and heavily used in [11] for deriving tractability results for the Check(GHD, k) problem. A subedge function takes as input a hypergraph $H = (V(H), E(H))$ and produces as output a set E' of subedges of the edges in $E(H)$, such that E' is then added to $E(H)$. Clearly, adding a set E' of subedges does not change the *fhw* of H (nor does it change the generalized hypertree width – that is why this idea was fruitfully applied to the Check(GHD, k) problem in [11]). Below, we shall define a whole family of subedge functions $h_{d,k}$, which, for fixed upper bounds d on the degree and k on the *fhw*, take a hypergraph H as input and return a polynomially bounded, polynomial-time computable set E' of subedges of $E(H)$. Adding these subedges to $E(H)$ will then allow us to define a *polynomial* upper bound on the set of all possible bags B_u at a given node u in an FHD of H.

Towards this goal, we follow the approach taken in [11] when devising a polynomial-time decision procedure for the Check(GHD, k) problem in case of a slightly more general class than the hypergraphs of bounded degree, namely the class of hypegraphs that enjoy the *bounded multi-intersection width* (BMIP) defined as follows:

▸ **Definition 18.** [11] 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 $e_1, \dots, e_c$ of H. We say that a hypergraph H has the *i-bounded c-multi-intersection property (ic-BMIP)* if *c-miwidth(H)* $\leq i$ holds.

We say that a class $\mathcal{C}$ of hypergraphs has the *bounded multi-intersection property (BMIP)* if there exist constants c and i such that every hypergraph H in $\mathcal{C}$ has the ic-BMIP. ◇

Clearly, if a hypergraph H has degree bounded by d, then the intersection of any selection of $d+1$ edges yields the empty set. Hence, H with degree bounded by d satisfies the 0-bounded $(d+1)$-multi-intersection property, written as $0(d+1)$-BMIP.

The key idea of the polynomial-time decision procedure in [11] for the Check(GHD, k) problem in case of hypergraphs enjoying the BMIP is to reduce the Check(GHD, k) problem for hypergraph H to the Check(HD, k) problem for an appropriate hypergraph H'. This hypergraph H' is constructed from H by adding a polynomially big set $h(H)$ of subedges of H. This set $h(H)$ is obtained from $E(H)$ by inspecting an arbitrary GHD of width $\leq k$ of H and trying to eliminate all violations of the *special condition*. Recall that the special condition means that $B_u \subseteq \big(B(\lambda_u) \cap \bigcup_{p \in T_u} B_p\big)$ holds at every node u in the GHD, where T_u denotes the subtree rooted at u. As in [11], we use λ (rather than γ) to denote an *integral* edge-weight function $\lambda\colon E(H) \to \{0,1\}$. A special condition violation (SCV) occurs if, for some edge $e \in E(H)$, $\lambda_u(e) = 1$ holds and some variable $v \in e$ occurs in $\bigcup_{p \in T_u} B_p$ but is missing in B_u. The elimination of such SCVs proceeds in two steps: first, we turn the given GHD into a "bag-maximal" one, i.e.: we add all vertices of $B(\lambda_u) \setminus B_u$ to B_u as long as this does not lead to a violation of the connectedness condition. The second step is more tricky: the goal is to add to $E(H)$ an appropriate subedge $e' \subseteq e$ such that $v \notin e'$ holds and replacing e in λ_u by e' yields again a valid GHD. Now the crux of the polynomial-time decision procedure for the Check(GHD, k) problem is to find a (polynomial-time computable!) subedge function h such that adding all edges in $h(H)$ to $E(H)$ allows us to eliminate all possible SCVs in all possible GHDs of H of width $\leq k$.

We now show how the ideas of the polynomial-time decision procedure for GHDs can be carried over to the Check(FHD, k) problem for hypergraphs with degree bounded by some constant $d \geq 1$. There are mainly two issues when trying to adapt the construction from the GHD case. First, we need to define what bag-maximality means in the context of FHDs. This is actually easy. In analogy to the GHD setting, we say that an FHD $\mathcal{F}$ is *bag-maximal* if for each decomposition node u of $\mathcal{F}$, for every vertex $v \in B(\gamma_u) \setminus B_u$, adding v to B_u would violate the connectedness condition. Clearly, for every FHD $\langle T, (B_u), (\gamma_u)\rangle$, a bag-maximal FHD $\mathcal{F}^+ = \langle T, (B_u^+), (\gamma_u)\rangle$ can be generated by adding vertices from $B(\gamma_u) \setminus B_u$ to bags B_u as long as possible. We may thus assume w.l.o.g. that our FHD $\mathcal{F}$ is bag-maximal.

For the definition of an appropriate subedge function (denoted $h_{d,k}$ to indicate that $h_{d,k}$ depends on d and k), take a hypergraph H with degree bounded by $d \geq 1$ and let k denote an upper bound on the width of an FHD $\mathcal{F}$ of H. Now consider an arbitrary node u in $\mathcal{F}$ with edge-weight function γ_u, let $e \in supp(\gamma_u)$ and suppose that $e \cap B(\gamma_u) \nsubseteq B_u$, i.e., e contains at least one vertex v, which is contained in $B(\gamma_u) \setminus B_u$. Our subedge function $h_{d,k}$ will be constructed in such a way that $h_{d,k}(H)$ contains a subedge e' of e such that v is eliminated from e. We can then replace e in γ_u by e' to reduce the gap between B_u and $B(\gamma_u)$. By carrying out this operation for every edge $e \in supp(\gamma_u)$ and vertex $v \in e \cap (B(\gamma_u) \setminus B_u)$, we will ultimately be able to set B_u and $B(\gamma_u)$ equal.

As a first step towards this goal, we extend the notion of critical paths from the GHD setting in [11] to FHDs.

▸ **Definition 19.** Let $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ be an FHD of a hypergraph H. Moreover, let u be a node in $\mathcal{F}$ and let $e \in \mathit{supp}(\gamma_u)$ with $e \cap B(\gamma_u) \nsubseteq B_u$ holds.

By the connectedness condition, $\mathcal{F}$ contains a node that covers all of e. Let u' denote the node closest to u, such that u' covers e, i.e., $e \subseteq B_{u'}$. Then (analogously to [11]) we call the path from u to u' the *(extended) critical path* of (u, e) denoted as $\mathit{critp}^+(u, e)$.

By slight abuse of notation, we shall write $p \in \pi$ for $\pi = \mathit{critp}^+(u, e)$, to denote that p is a node on path π. ◇

The following lemma (which follows closely Lemma B.4 in [11]) allows us to characterize the subsets e' of e needed in the subedge function $h_{d,k}$.

▸ **Lemma 20.** *Let $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ be an FHD of a hypergraph H. Moreover, let u be a node in $\mathcal{F}$ and let $e \in \mathit{supp}(\gamma_u)$. Then the following equation holds:*

$$e \cap B_u = e \cap \bigcap_{p \in \mathit{critp}^+(u,e)} B(\gamma_p)$$

Proof. As in Definition 19, let u' denote the node closest to u, such that u' covers e, i.e., $e \subseteq B_{u'}$. Moreover, let $\pi = \mathit{critp}^+(u, e)$ denote the path connecting u with u'. The proof of the equation follows very closely the proof of Lemma B.4 in [11].

For the $\subseteq$-direction, assume $v \in (e \cap B_u)$. Given that v is also in $B_{u'}$, by the connectedness condition, v appears in all bags B_p of decomposition nodes p on the path π from u to u'. Hence, v appears in each $B(\gamma_p)$ and, therefore, we clearly have $v \in \big(e \cap \bigcap_{p \in \pi} B(\gamma_p)\big)$.

For the $\supseteq$-direction, assume to the contrary that there is a vertex $v \in \big(e \cap \bigcap_{p \in \pi} B(\gamma_p)\big)$, such that $v \notin (e \cap B_u)$. Then we could actually insert v into B_p for every node p along the path π (of course, if B_p already contains v, then we leave B_p unchanged) without violating the connectedness condition. In particular, in node u we would thus indeed add a new vertex v into B_u. This contradicts the assumption of bag-maximality. ◂

Our next goal is to find a subedge function $h_{d,k}$ which contains at least all subedges appearing on the right-hand side. To achieve this while abstracting from the knowledge of a particular decomposition and from the knowledge of particular edge-weight functions, we will make two bold over-approximations. First, instead of considering concrete critical paths, we will consider arbitrary finite sequences $\xi = \xi_1, \dots, \xi_{\max(\xi)}$ of groups of $\leq k \times d$ edges of H, where each such group represents a potential support $\mathit{supp}(\gamma_u)$ at some potential node u of a potential FHD of H. Clearly, each effective path $\mathit{critp}^+(u, e)$ for any possible combination of decomposition node u and edge e of any possible FHD $\mathcal{F}$ of H is among these sequences. The second over-approximation we make is that instead of considering particular edge-weight functions, we will simply consider (a superset of) all possible supports of $\leq k \times d$ atoms, and for each such support (a superset of) all "in"-sets that could possibly arise with this support. A support is simply given by a subset of $\leq k \times d$ edges of H. For each such support, by Lemma 17, there are in fact no more than $2^{2^{k \times d}}$ "in"-sets and these are determined by unions of classes from $\mathit{CLASSES}(H')$, where H' is the subhypergraph of H given by the support. To make this more formal, we give the following definition. Recall the notion of $\mathit{CLASSES}$ (which denotes the set of intersections of edges contained in some type; in our case, each type consists of at most $k \times d$ edges) from Definition 16.

▸ **Definition 21.** Let H be a hypergraph and let $\xi = \xi_1, \dots, \xi_{\max(\xi)}$ be an arbitrary sequence of groups of $\leq k \times d$ edges of H. For $i \in \{1, \dots, \max(\xi)\}$, by slight abuse of notation, we overload the notion of $\mathit{CLASSES}$ from Definition 16 as follows: we write $\mathit{CLASSES}(\xi_i)$ to denote the set $\mathit{CLASSES}(H^i_\xi)$, where H^i_ξ is the subhypergraph of H whose edges are the $\le k \times d$ edges of ξ_i and whose vertices are precisely all vertices occurring in these edges.

Let π be an (extended) critical path of length r of some FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ of H, i.e., π is of the form $\pi = u_1, \dots, u_r$. Suppose that each edge-weight function γ_u in $\mathcal{F}$ has $k \times d$-bounded support. Then we denote by ξ^π the sequence of length r where, for $1 \le i \le r$, $\xi_i = \mathit{supp}(\gamma_{u_i})$. ◇

Our goal is to compute a set of subedges of the edges in $E(H)$ such that all sets of the form $e \cap \bigcap_{p \in \pi} B(\gamma_p)$ with $\pi = \mathit{critp}^+(u, e)$ from Lemma 20 are contained. We may use that each γ_p has $k \times d$-bounded support. By Lemma 17, we know that every possible "in"-set $B(\gamma_p)$ is contained in $\Cup \mathit{CLASSES}(H)$, i.e., every possible $B(\gamma_p)$ along a critical path π can be represented as the union of classes (where each class is in turn the intersection of some edges selected from $\mathit{supp}(\gamma_p)$). Hence, to obtain $e \cap \bigcap_{p \in \pi} B(\gamma_p)$, we need to compute the intersection of all unions of classes along a critical path π.

Recall that we generalize the support of edge-weight functions γ_p along a concrete critical path π in a concrete FHD $\mathcal{F}$ of H to sequences $\xi = \xi_1, \dots, \xi_{\max(\xi)}$ as introduced above, where each ξ_i is an arbitrary set of $\le k \times d$ edges from H. As the crucial data structure to compute the desired intersections of unions of classes, we now define the *intersection forest* $\mathit{IF}(\xi)$. This data structure will give us a systematic way to convert the intersections of unions of classes for all possible sequences ξ into a union of intersections. Intuitively, each branch (starting at a root) in $\mathit{IF}(\xi)$ represents a possible transversal of the family $\{\mathit{CLASSES}(\xi_i)\}_{1 \le i \le \max(\xi)}$ for some sequence $\xi = \xi_1, \dots, \xi_{\max(\xi)}$, i.e., a transversal selects one class from $\mathit{CLASSES}(\xi_i)$ for each $i \in \{1, \dots, \max(\xi)\}$. On every branch, we will then compute the intersection of the classes selected along this branch. Since each class is in turn an intersection of edges (namely the edges contained in some type), every branch in $\mathit{IF}(\xi)$ therefore simply yields an intersection of edges from H.

Construction of the intersection forest $\mathit{IF}(\xi)$.
We define $\mathit{IF}(\xi)$ as a rooted forest such that each of its nodes v is labeled by

- a subset $\mathit{set}(v) \subseteq V(H)$,
- a set of levels $\mathit{levels}(v)$,
- a set $\mathit{edges}(v)$ of edges of $E(H)$ such that $\mathit{set}(v) \subseteq \bigcap \mathit{edges}(v) = \bigcap_{e \in \mathit{edges}(v)} e$,
  in other words: $\mathit{set}(v)$ is a *class* and $\mathit{edges}(v)$ is its (maximal) *type*, see Definition 16,
- and a mark $\mathit{mark}(v) \in \{\mathit{ok}, \mathit{fail}\}$.

Initialization of $\mathit{IF}(\xi)$. For every non-empty class $K \in \mathit{CLASSES}(\xi_1)$, the intersection forest $\mathit{IF}(\xi)$ contains a root vertex v where

- $\mathit{set}(v) = K$,
- $\mathit{levels}(v) = \{1\}$,
- $\mathit{edges}(v) = \{e \in E(H) \mid K \subseteq e\}$, and
- $\mathit{mark}(v) = \mathit{ok}$.

Further expansion of $\mathit{IF}(\xi)$. Each tree in $\mathit{IF}(\xi)$ is further expanded and updated by the following inductive procedure. Let v be a node of $\mathit{IF}(\xi)$ with $\max(\mathit{levels}(v)) = i < \max(\xi)$ and $\mathit{mark}(v) = \mathit{ok}$. Then we distinguish between three cases:

1. *Dead End.* If for each class K of $\mathit{CLASSES}(\xi_{i+1})$, $\mathit{set}(v) \cap K = \emptyset$, then v has no children, and its mark is set to $\mathit{mark}(v) = \mathit{fail}$. Intuitively this is a dead end as it cannot be continued to yield a non-empty intersection of a transversal of the family $\{\mathit{CLASSES}(\xi_i)\}_{1 \le i \le \max(\xi)}$.
2. *Passing.* For each class K of $\mathit{CLASSES}(\xi_{i+1})$ fulfilling $\mathit{set}(v) \cap K = \mathit{set}(v)$, insert $i+1$ into $\mathit{levels}(v)$. Intuitively, this makes sure the same value $\mathit{set}(v)$ is never repeated on a branch,

and, as a consequence, every child node must have a strictly smaller $set()$-component and, thus, at least one more edge in its $edges()$-label than its parent (see also Fact 1 in Lemma 22 below).

3. *Expand.* For each class K of $CLASSES(\xi_{i+1})$ fulfilling $set(v) \cap K \subsetneq set(v)$, create a child v' of v, and let $set(v') = set(v) \cap K$, $levels(v') = \{i+1\}$, $edges(v') = \{e \in E(H) \mid set(v') \subseteq e\}$, and $mark(v) = ok$. Note that we thus clearly have $set(v') \subsetneq set(v)$ and $edges(v) \subsetneq edges(v')$.

For $1 \le i \le \max(\xi)$, let $iflevel_i(\xi)$ denote the set of all nodes v of $IF(\xi)$ such that $i \in level(v)$ and $mark(v) = ok$. Denote by $FRINGE_i(\xi)$ the collection of all sets $set(v)$ where $v \in iflevel_i(\xi)$. Finally, let the *fringe of* ξ be defined as $FRINGE(\xi) = FRINGE_{\max(\xi)}(\xi)$. Note that, given that by definition of $IF(\xi)$ there cannot be any *fail* node on level $\max(\xi)$, $FRINGE(\xi)$ coincides with the set of all *set*-labels at level $\max(\xi)$.

We now establish some easy facts about $IF(\xi)$.

▸ **Lemma 22.** *Let H be a hypergraph and let $\xi = \xi_1, \ldots, \xi_{\max(\xi)}$ be an arbitrary sequence of groups of $\le k \times d$ edges of H. Then the intersection forest $IF(\xi)$ according to the above construction has the following properties:*

Fact 1. *If node v' is a child of node v in $IF(\xi)$, then $edges(v')$ must contain at least one new edge in addition to the edges already present in $edges(v)$.*

Fact 2. *The depth of $IF(\xi)$ is at most $d-1$.*

Fact 3. *Let $c = 2^{k \times d}$. Then $IF(\xi)$ has no more than c^{d+1} nodes and $|FRINGE(\xi)| \le c^d = 2^{d^2 \times k}$.*

Proof. The facts stated above can be seen as follows.

Fact 1. Note that we can only create a child node through an *expand* operation. But this requires that $set(v') = set(v) \cap K \subsetneq set(v)$. Given that K is the intersection of all edges of $type(K)$, for $set(v')$ to shrink, $type(K)$ must contain at least one new edge not yet contained in $edges(v)$, and this edge is therefore included into $edges(v')$.

Fact 2. This follows from Fact 1 and the fact that, given that H is of degree d, at most d edges have a non-empty intersection.

Fact 3. For each sequence ξ as above, for whatever ξ_i, the inequality $|CLASSES(\xi_i)| \le c$ holds (cf. Lemma 17). Hence, Fact 3 follows from the depth $d-1$ established in Fact 2 and the fact that we have at most c such trees, each with branching not larger than c.

This concludes the proof of the lemma. ◂

Recall that we are studying the set of arbitrary sequences $\xi = \xi_1, \ldots, \xi_{\max(\xi)}$ of groups of $\le k \times d$ edges of H because they give us a superset of possible critical paths $\pi = critp^+(u,e)$ in possible FHDs of H, such that each group ξ_i of edges corresponds to the support of the edge-weight function γ_i at the i-th node on path π. The following lemma establishes that the intersection forests (and, in particular, the notion of $FRINGE(\xi)$) introduced above indeed give us a tool to generate a superset of the set of all possible sets $\bigcap_{p\in\pi} B(\gamma_p)$ in all possible FHDs of H of width $\le k$. Recall from Lemma 20 that these sets $\bigcap_{p\in\pi} B(\gamma_p)$ are precisely what we need to characterize all possible subedges (of edges in H) of the form $e \cap B_u$, where e is an arbitrary edge in H and B_u is a possible bag in a possible (bag-maximal) FHD of H of width $\le k$.

▸ **Lemma 23.** *Let H be a hypergraph of degree $d \ge 1$ and let $\mathcal{F}$ be an FHD of H of width $\le k$. Consider an extended critical path π of FHD $\mathcal{F}$, together with its associated sequence ξ^π introduced in Definition 21. We claim that the following relationship holds:*

$$\bigcap_{p\in\pi} B(\gamma_p) \ \in\ \Cup FRINGE(\xi^\pi)$$

Proof. Let $\pi = p_1, \ldots, p_r$ with $r = \max(\pi)$ and let $\gamma_1, \ldots, \gamma_r$ denote the edge-weight functions along this sequence of nodes. For $i \in \{1, \ldots, r\}$, let π_i denote the initial fragment $p_1, \ldots, p_i$ of π. We proceed by induction on $i \in \{1, \ldots, r\}$, i.e., we show that, for every $i \in \{1, \ldots, r\}$, the following relationship holds:

$$\bigcap_{p \in \pi_i} B(\gamma_p) \;\in\; \Cup \mathit{FRINGE}_i(\xi^\pi)$$

Basis Step. The base case $i = 1$ follows from statement $\mathit{INSET}(H) \subseteq \Cup\, \mathit{CLASSES}(H)$ of Lemma 17(2). In fact, $B(\gamma_1)$ is (by the definition of INSET) an element of $\mathit{INSET}(\mathit{supp}(\gamma_1))$ which is thus contained in $\Cup \mathit{CLASSES}(\mathit{supp}(\gamma_1))$ which is identical to $\Cup\, \mathit{FRINGE}_1(\xi^\pi)$ by the above initialization of $\mathit{IF}(\xi)$.

Inductive Step. Assume for some $i < r$ that $\bigcap_{p \in \pi_i} B(\gamma_p) \;\in\; \Cup \mathit{FRINGE}_i(\xi^\pi)$ holds. We show the desired relationship also holds for $i+1$. Clearly, $\bigcap_{p \in \pi_{i+1}} B(\gamma_p)$ is equal to $(\bigcap_{p \in \pi_i} B(\gamma_p)) \cap B(\gamma_{i+1})$. From this, by using the inductive assumption, together with the fact that $B(\gamma_{i+1}) \in \mathit{INSET}(\xi^\pi_{i+1})$, and the inclusion $\mathit{INSET}(\xi^\pi_{i+1}) \subseteq \Cup \mathit{CLASSES}(\xi^\pi_{i+1})$, which holds by Lemma 17, we obtain:

$$\bigcap_{p \in \pi_{i+1}} B(\gamma_p) \;\in\; \mathit{FRINGE}_i(\xi^\pi) \Cap \Cup \mathit{CLASSES}(\xi^\pi_{i+1}).$$

By using the distributivity of $\Cap$ over $\Cup$, we get:

$$\bigcap_{p \in \pi_{i+1}} B(\gamma_p) \;\in\; \Cup(\mathit{FRINGE}_i(\xi^\pi) \Cap \mathit{CLASSES}(\xi^\pi_{i+1})).$$

However, by the construction of $\mathit{IF}(\xi^\pi)$, $\mathit{FRINGE}_i(\xi^\pi) \Cap \mathit{CLASSES}(\xi^\pi_{i+1})$ is just the same as $\mathit{FRINGE}_{i+1}(\xi^\pi)$. In fact, the *Passing* and *Expand* rules make precisely these intersections when producing level $i+1$ of $\mathit{IF}(\xi^\pi)$. Therefore, we obtain that

$$\bigcap_{p \in \pi_{i+1}} B(\gamma_p) \;\in\; \Cup \mathit{FRINGE}_{i+1}(\xi^\pi)$$

indeed holds, which settles the inductive step. ◂

The desired subedge function $h_{d,k}$ therefore looks as follows:

▸ **Lemma 24.** *Let H be a hypergraph of degree $d \geq 1$ and let $k \geq 1$. Let the subedge function $h_{d,k}$ be defined as*

$$h_{d,k}(H) = E(H) \Cap (\Cup_{2^{d^2 \times k}} \Cap_d E(H))$$

Then (for fixed constants d and k), the size of $h_{d,k}(H)$ is polynomially bounded and $h_{d,k}(H)$ can be computed in polynomial time. Moreover $h_{d,k}(H)$ contains all subedges $e \cap B_u$ of all $e \in E(H)$ for all possible bags B_u of whatever bag-maximal FHD of width $\leq k$ of H.

Proof. Observe that for whatever sequence ξ, each element of $\mathit{FRINGE}(\xi)$ is, by construction, the intersection of at most d edges. Moreover, recall that by Fact 3 of Lemma 22, $|\mathit{FRINGE}(\xi)| \leq c^d = 2^{d^2 \times k}$ holds. Therefore, for all possible sequences ξ, we have

$$\Cup \mathit{FRINGE}(\xi) \subseteq \Cup_{2^{d^2 \times k}} \Cap_d E(H).$$

Given that d and k are constants, the set $\Cup_{2^{d^2 \times k}} \Cap_d E(H)$ is of polynomial size and is clearly computable in polynomial time from H. By Lemma 20 together with Lemma 23, it is then also clear that the subedge function $h_{d,k}(H)$ contains all subedges $e \cap B_u$ for all possible bags of whatever bag-maximal FHD of width $\leq k$ of H. ◂

5 Deciding the Check Problem for Hypergraphs of Bounded Degree

With the subedge function $h_{d,k}$ at hand, we have a powerful tool that will allow us to devise a polynomial-time decision procedure for the $\textsc{Check}(\text{HD}, k)$ problem. Towards this goal, we first observe that adding the edges in $h_{d,k}(H)$ to a hypergraph H allows us to restrict ourselves to FHDs of a very peculiar form. This form is captured by the following definition, which is applicable to FHDs and (G)HDs alike.

Recall that if S is a set of sets then $\bigcup S$ denotes the union of all sets in S, i.e., $\bigcup S = \bigcup_{e \in S} e$. Below, we shall apply this notation, in particular, to settings where S denotes the support (i.e., a set of edges) of some edge-weight function.

▸ **Definition 25.** An FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ of a hypergraph H is *strict* if for every decomposition node u in T, the equality $B_u = B(\gamma_u) = \bigcup \mathit{supp}(\gamma_u)$ holds.

Likewise, let $\mathcal{H} = \langle T, (B_u), (\lambda_u) \rangle$ be an HD or a GHD of a hypergraph H. We call $\mathcal{H}$ *strict* if for every decomposition node u in T, the equality $B_u = B(\lambda_u) = \bigcup \mathit{supp}(\lambda_u)$ holds.

Note that every strict HD or GHD is actually a *query decomposition* as introduced in [7], where the width of this query decomposition is equal to the cardinality of the largest support. Moreover, every strict FHD $\mathcal{F}$ (and likewise, every strict GHD) trivially fulfills the special condition. Hence, every strict FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ can be naturally transformed into an HD $\mathcal{H} = \langle T, (B_u), (\lambda_u) \rangle$ by leaving the tree structure and the bags B_u unchanged and by defining λ_u as the characteristic function of $\mathit{supp}(\gamma_u)$, i.e., $\lambda_u(e) = 1$ if $e \in \mathit{supp}(\gamma_u)$ and $\lambda_u(e) = 0$ otherwise.

Below we show that, in case of bounded degree, we can transform every FHD of width $\leq k$ into a strict FHD of width $\leq k$.

▸ **Lemma 26.** *Assume a hypergraph $H = (V(H), E(H))$ of degree $\leq d$ has an FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ of width $\leq k$. Suppose that $H^\#$ is obtained from H by adding the edges in $h_{d,k}(H)$, i.e., $H^\# = (V(H^\#), E(H^\#))$ with $V(H^\#) = V(H)$ and $E(H^\#) = E(H) \cup h_{d,k}(H)$.*

Then $H^\#$ admits a strict *FHD $\mathcal{F}^\# = \left\langle T, (B_u), (\gamma_u^\#) \right\rangle$ of width $\leq k$ that has $(k \times d)$-bounded support.*

Proof. Let $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ be an arbitrary FHD of H of width $\leq k$. W.l.o.g., assume that $\mathcal{F}$ is bag-maximal. Of course, $\mathcal{F}$ is also an FHD of $H^\#$ of width $\leq k$. Let u be a node in $\mathcal{F}$ and let $e \in \mathit{supp}(\gamma_u)$. Suppose that $e \cap B(\gamma_u) \notin B_u$. Then we modify γ_u as follows: by Lemma 24, $E(H) \cup h_{d,k}(H)$ is guaranteed to contain the subedge $e' = e \cap B_u$ of e. Then we "replace" e in γ_u by e', i.e., we set $\gamma_u(e') := \gamma_u(e') + \gamma_u(e)$ and $\gamma_u(e) := 0$.

The FHD $\mathcal{F}^\# = \left\langle T, (B_u), (\gamma_u^\#) \right\rangle$ is obtained by exhaustive application of this transformation step. Clearly, such a transformation step never increases the support. Moreover, the resulting FHD $\mathcal{F}^\#$ is strict, since in every node u the transformation eliminates all edges $e \in \mathit{supp}(\gamma_u)$ with $e \cap B(\gamma_u) \notin B_u$. ◂

Our strategy to devise a polynomial-time decision procedure for the $\textsc{Check}(\text{FHD}, k)$ problem is to reduce it to the $\textsc{Check}(\text{HD}, k)$ problem and then adapt the algorithm from [14]. Note however, that the algorithm from [14] requires the HDs to be in a certain *normal form*. We thus have to make sure that also in the FHD-setting, we can always achieve an analogous normal form. We thus recall below the *fractional normal form* (FNF) introduced in [11]:

▸ **Definition 27.** [11] An FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u) \rangle$ of a hypergraph H is in *fractional normal form (FNF)* if for each node $r \in T$, and for each child s of r, the following conditions hold:

1. there is *exactly one* $[B_r]$-component C_r such that the equality $V(T_s) = C_r \cup (B_r \cap B_s)$ holds;
2. $B_s \cap C_r \neq \emptyset$, where C_r is the $[B_r]$-component satisfying Condition 1;
3. $B(\gamma_s) \cap B_r \subseteq B_s$.

In condition 1 above, a $[B_r]$-component denotes a set of vertices which is maximal connected in the subhypergraph of H induced by $V(H) \setminus B_r$. Moreover, we write $V(T_s)$ to denote the set of all vertices occurring in some bag in the subtree of T rooted at s. ◇

An HD $\mathcal{H} = \langle T, (B_u), (\lambda_u)\rangle$ can be considered as a special case of an FHD where the edge-weight functions λ_u only assign weights 0 or 1 to each edge and where the so-called *special condition* holds. When applied to HDs, the fractional normal form recalled above coincides with the normal form defined in [14]. Indeed, the transformation of an arbitrary FHD into FNF given in [11] follows closely the transformation into normal form given in [14].

We now strengthen Lemma 26 in that also FNF can be guaranteed.

▸ **Lemma 28.** *Assume a hypergraph* $H = (V(H), E(H))$ *of degree* $\leq d$ *has an FHD* $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ *of width* $\leq k$*. Suppose that* $H^{\#}$ *is obtained from* H *by adding the edges in* $h_{d,k}(H)$*, i.e.,* $H^{\#} = (V(H^{\#}), E(H^{\#}))$ *with* $V(H^{\#}) = V(H)$ *and* $E(H^{\#}) = E(H) \cup h_{d,k}(H)$*.*

Then $H^{\#}$ *admits a* strict *FHD* $\mathcal{F}^{\#} = \left\langle T, (B_u), (\gamma^{\#}_u)\right\rangle$ in fractional normal form *of width* $\leq k$ *that has* $(k \times d)$*-bounded support.*

Proof. By Lemma 14 there exists an FHD $\mathcal{F}_1$ of H whose supports are all bounded by $k \times d$. Without changing the supports, we can transform this FHD into a bag-maximal one, and we thus assume w.l.o.g. that $\mathcal{F}_1$ is bag-maximal.

Now transform $\mathcal{F}_1$ into an FHD $\mathcal{F}_2$ of width $k \times d$ *in FNF*, by proceeding according to the proof of Theorem C.1 in [11], which, in turn follows closely the transformation in the proof of Theorem 5.4 in [14]. Note that this transformation preserves the support bound of $k \times d$. In fact, the component split made for ensuring condition 1 of FNF can never lead to larger supports, given that the sets $B_{new(v,C_i)}$ become smaller. Ensuring condition 2 results in eliminating nodes from the tree, so nothing bad can happen. Observe that condition 3, which is $B(\gamma_s) \cap B_r \subseteq B_s$ for a child node s of decomposition node r, is initially satisfied, because the initial FHD $\mathcal{F}_1$ is bag-maximal. Observe further that the splitting of a node (subtree) into several nodes (subtrees) performed to achieve condition 1 of FNF does not destroy the validity of condition 3.

Finally transform the FHD $\mathcal{F}_2$ via Lemma 26 into a strict FHD $\mathcal{F}^{\#} = \left\langle (T, (B_u), (\gamma^{\#}_u)\right\rangle$ of $H^{\#}$ of width $\leq k$ and with $(k \times d)$-bounded support. Observe that this strict FHD $\mathcal{F}^{\#}$ is still in FNF. To see this, first note that the tree structure T of the decomposition and all bags (B_u) remain exactly the same. Moreover, for whatever set $S \subseteq V(H)$, H and $H^{\#}$ have exactly the same $[S]$-components. This can be seen by recalling from [14] that two vertices v_1, v_2 in hypergraph H are $[S]$-adjacent if they are adjacent in the subhypergraph of H induced by $V(H) \setminus S$. Hence, $[S]$-adjacency remains unaltered when adding subedges.

Given that conditions 1 and 2 of FNF are only formulated in terms of B_i-bags and B_i-components – and all such bags and components are the same for $\mathcal{F}$ and $\mathcal{F}^{\#}$ – they remain valid. Condition 3, which requires that $B(\gamma^{\#}_s) \cap B_r \subseteq B_s$ for child node s of r, is now trivially satisfied, because $\mathcal{F}^{\#}$ is *strict* and, therefore, even $B(\gamma^{\#}_s) = B_s$ holds. ◂

The following theorem finally establishes the close connection between the CHECK(FHD, k) and CHECK(HD, k) problems for hypergraphs H of degree bounded by some constant $d \geq 1$. Recall that the edge-weight functions λ_u in an HD only assign values 0 or 1 to edges. As in

[14], it is convenient to identify λ_u with a set S_u of edges, namely the edges in $E(H)$ that are assigned value 1. In other words, $S_u = \mathit{supp}(\lambda_u)$. Moreover, as mentioned in Section 2, we can identify a set of edges S_u with the hypergraph whose set of vertices is $\bigcup S_u$ and whose set of edges is S_u. For given edge-weight function λ_u with $S_u = \mathit{supp}(\lambda_u)$, we shall write H_{λ_u} to denote this hypergraph.

▸ **Theorem 29.** *Let H be a hypergraph whose degree is bounded by $d \geq 1$ and define $H^\#$ as above, i.e., $H^\# = (V(H^\#), E(H^\#))$ with $E(H^\#) = E(H) \cup h_{d,k}(H)$. Then the following statements are equivalent:*

1. *$fhw(H) \leq k$.*
2. *$H^\#$ admits a strict hypertree decomposition (thus a query decomposition) $\mathcal{H} = \langle T, (B_u), (\lambda_u)\rangle$ of width $\leq k \times d$ in normal form such that for each decomposition node u of $\mathcal{H}$, $\rho^*(H_{\lambda_u}) \leq k$ holds (i.e., H_{λ_u} has a fractional edge cover of weight $\leq k$).*

Proof. 1 ⇒ 2 follows immediately from Lemma 28 by defining λ_u for each decomposition node u as the the characteristic function of $\mathit{supp}(\gamma_u)$, i.e., $\lambda_u(e) = 1$ if $e \in \mathit{supp}(\gamma_u)$ and $\lambda_u(e) = 0$ otherwise. Clearly, replacing the edge-weight function γ_u by the *integral* edge-weight function λ_u preserves the normal form.

To see 2 ⇒ 1, assume 2 holds with query decomposition $\mathcal{H} = \langle T, (B_u), (\lambda_u)\rangle$ and assume further that, for each decomposition node u of $\mathcal{H}$, there exists an edge cover γ'_u for H_{λ_u} of width $\leq k$. In particular, we thus have $B(\lambda_u) = B(\gamma'_u) = B_u$. Similarly to the proof of Lemma 14, we can transform each edge cover γ'_u of the induced subhypergraph H_{λ_u} of $H^\#$ into an edge-weight function γ_u of H by moving the weights $\gamma'_u(e')$ of each edge e' in H_{λ_u} to one of its “originator edges” e in H (i.e., an edge e in H with $e' \subseteq e$). By replacing λ_u with γ_u, we obtain an FHD $\mathcal{F} = \langle T, (B_u), (\gamma_u)\rangle$ of width $\leq k$ of H. ◂

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

▸ **Theorem 30.** *For every hypergraph class $\mathcal{C}$ that has bounded degree, and for every constant $k \geq 1$, the* Check*(FHD, k) problem is tractable, i.e., given a hypergraph $H \in \mathcal{C}$, it is feasible in polynomial time to check $\mathit{fhw}(H) \leq k$ and, if this holds, to compute an FHD of width k of H.*

Proof. By Theorem 29, it is sufficient to look for a strict hypertree decomposition (thus a query decomposition) $\mathcal{H} = \langle T, (B_u), (\lambda_u)\rangle$ of $H^\#$ of width $\leq d \times k$ such that for each decomposition node u of $\mathcal{H}$, $\rho^*(H_{\lambda_u}) \leq k$ holds. This is achieved by modifying the alternating algorithm k-decomp from [14] by inserting the following two checks at runtime at each decomposition node u:

- if u has a parent r, then $\bigcup S_u \subseteq B(\lambda_r) \cup \mathit{treecomp}(u)$ with $S_u = \mathit{supp}(\lambda_u)$. This makes sure that $B_u = \bigcup S_u$, i.e., the decomposition is strict.
- $\rho^*(H_{\lambda_u}) \leq k$.

The so modified algorithm clearly runs in ALogSpace = Ptime ◂

6 Conclusion

In this article, we have shown that recognizing low fractional hypertree width becomes tractable if the hypergraphs under consideration have bounded degree. This strengthens a previous result in [11], which required both bounded degree and bounded intersection width to ensure tractability. Along the way to our tractability result, we have derived several interesting properties of FHDs of hypergraphs of bounded degree. Above all, by defining an appropriate subedge function, we have established a surprising relationship between fractional hypertree decompositions and query decompositions.

Many interesting questions for future research remain, which can be divided into two main categories: further improvements of the computation of GHDs and FHDs on the one hand, and actually putting to work hypergraph decomposition methods in query answering on the other hand. As far as the computation of GHDs and FHDs is concerned, the following questions deserve further attention:

- What about d-sparse hypergraphs, i.e., hypergraphs with average degree d? From Füredi's precise formulation of his result it seems that for such hypergraphs, Check(FHD, k) could still be tractable. Of course, to avoid padding techniques, we should actually define sparseness via the reduced hypergraph H^{-} introduced in Section 3 rather than via H itself.
- What is the parameterized complexity of the Check(FHD, k) problem parameterized by d. We conjecture that it is hard for some suitable class in the W-hierarchy of fixed-parameter intractable problems.
- Most importantly, we want to answer the open question if bounded intersection width alone (without bounded degree) suffices to make the Check(FHD, k) problem tractable.
- Our tractability result for the Check(FHD, k) problem is heavily based on the upper bound on the support of fractional edge covers in case of bounded degree. This upper bound is a consequence of the corresponding upper bound on fractional vertex covers in case of bounded rank [13]. It would be nice to have an intuitive *direct* proof (i.,e., not relying on other deep theorems as is the case in the proof by Füredi) of the upper bound on the support of fractional edge covers in case of bounded degree.

For the actual use of decompositions in query answering, we want to go beyond their obvious application to CSPs and CQs. Indeed, decomposition techniques have recently also been applied to Datalog [4] – so far, restricted to tree decompositions, though. We would like to extend this work in two directions: first, the more powerful HDs, GHDs, and FHDs should also be used. Second, we are targeting query languages beyond "classical" Datalog: in the context of ontology-mediated data access, various subclasses of Datalog$^{\pm}$ [5] have been proposed, which are (restricted forms of) tuple-generated dependencies intensively studied for defining schema mappings [18] in the area of data exchange [10]. One such subclass of Datalog$^{\pm}$ with particularly good computational properties is warded Datalog$^{\pm}$, for which a powerful engine has been developed recently [2, 3]. This engine (called the Vadalog system) aims at data wrangling [12] and data acquisition [17] tasks at web scale. Hence, efficient query evaluation is of utmost importance. Of course, this need for high performance also applies to planned extensions of the Vadalog system such as dealing with inconsistent data (a notoriously difficult task, see e.g., [1] for an overview of complexity results) and handling preferences at web scale [9].

References

1 Sebastian Arming, Reinhard Pichler, and Emanuel Sallinger. Complexity of repair checking and consistent query answering. In *Proceedings of ICDT 2016*, volume 48 of *LIPIcs*. SD-LZI, 2016.

2 Luigi Bellomarini, Georg Gottlob, Andreas Pieris, and Emanuel Sallinger. Swift logic for big data and knowledge graphs. In *Proceedings of IJCAI 2017*, pages 2–10. ijcai.org, 2017.

3 Luigi Bellomarini, Emanuel Sallinger, and Georg Gottlob. The vadalog system: Datalog-based reasoning for knowledge graphs. *PVLDB*, 11(9):975–987, 2018.

4 Bernhard Bliem, Marius Moldovan, Michael Morak, and Stefan Woltran. The impact of treewidth on ASP grounding and solving. In *Proceedings of IJCAI 2017*, pages 852–858. ijcai.org, 2017.

5 Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013. URL: `https://doi.org/10.1613/jair.3873`, [doi:10.1613/jair.3873]-path.

6 Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of STOC 1977*, pages 77–90. ACM, 1977.

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

8 Fan R. K. Chung, Zoltan Fueredi, MR Garey, and Ronald L. Graham. On the fractional covering number of hypergraphs. *SIAM journal on discrete mathematics*, 1(1):45–49, 1988.

9 Theresa Csar, Martin Lackner, Reinhard Pichler, and Emanuel Sallinger. Winner determination in huge elections with mapreduce. In *Proceedings of AAAI 2017*, pages 451–458. AAAI Press, 2017.

10 Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005. URL: `https://doi.org/10.1016/j.tcs.2004.10.033`, [doi:10.1016/j.tcs.2004.10.033]-path.

11 Wolfgang Fischl, Georg Gottlob, and Reinhard Pichler. General and fractional hypertree decompositions: Hard and easy cases. *CoRR*, abs/1611.01090, 2016. URL: `http://arxiv.org/abs/1611.01090`.

12 Tim Furche, Georg Gottlob, Bernd Neumayr, and Emanuel Sallinger. Data wrangling for big data: Towards a lingua franca for data wrangling. In *Proceedings of AMW 2016*, volume 1644 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

13 Zoltán Füredi. Matchings and covers in hypergraphs. *Graphs and Combinatorics*, 4(1):115–206, 1988.

14 Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002. URL: `http://dx.doi.org/10.1006/jcss.2001.1809`, [doi:10.1006/jcss.2001.1809]-path.

15 Georg Gottlob, Zoltán Miklós, and Thomas Schwentick. Generalized hypertree decompositions: NP-hardness and tractable variants. *J. ACM*, 56(6):30:1–30:32, September 2009. URL: `http://doi.acm.org/10.1145/1568318.1568320`, [doi:10.1145/1568318.1568320]-path.

16 Martin Grohe and Dániel Marx. Constraint solving via fractional edge covers. *ACM Trans. Algorithms*, 11(1):4:1–4:20, 2014.

17 Christopher Michels, Ruslan R. Fayzrakhmanov, Michael Ley, Emanuel Sallinger, and Ralf Schenkel. Oxpath-based data acquisition for dblp. In *Proceedings of JCDL 2017*, pages 319–320. IEEE CS, 2017.

18 Emanuel Sallinger. Reasoning about schema mappings. In *Data Exchange, Information, and Streams*, volume 5 of *Dagstuhl Follow-Ups*, pages 97–127. SD-LZI, 2013.