Hypertree Width and Related Hypergraph Invariants

Isolde Adler * Georg Gottlob † Martin Grohe‡

5th May 2006

Abstract

We study the notion of hypertree width of hypergraphs. We prove that, up to a constant factor, hypertree width is the same as a number of other hypergraph invariants that resemble graph invariants such as bramble number, branch width, linkedness, and the minimum number of cops required to win Seymour and Thomas's robber and cops game.

1 Introduction

Tree width of graphs is a well studied notion, which plays an important role in structural graph theory and has many algorithmic applications. Various other graph invariants are known to be the same or within a constant factor of tree width, for example, the *bramble number* or *tangle number* of a graph [12, 13], the *branch width* [13], the *linkedness* [12], and the number of cops required to win the *robber and cops* game on the graph [14]. Several of these notions may be viewed as measures for the global connectivity of a graph. The various equivalent characterizations of tree width show that it is a natural and robust notion.

Formally, let us call two graph or hypergraph invariants I and J *equivalent* if they are within a constant factor of each other, that is, if there are constants $c, d > 0$ such that for all graphs or hypergraphs G we have $c \cdot I(G) \leq J(G) \leq d \cdot I(G)$.

Tree decompositions and tree width can be generalized to hypergraphs in a straightforward manner; the tree width of a hypergraph is equal to the tree width of its primal graph. Motivated by algorithmic problems from database theory and artificial intelligence, Gottlob, Leone, and Scarcello [6] introduced the *hypertree width* of a hypergraph. Hypertree width is based on the same tree decompositions as tree width, but the width is measured differently. Essentially, the hypertree width is the minimum number of hyperedges needed to cover all bags of a tree decomposition. The *bags* of a tree decomposition $(T, (B_t)_{t\in V(T)})$ of a hypergraph $H = (V, E)$ are the sets $B_t \subseteq V$ for the tree nodes t, and a bag B_t is *covered* by the hyperedges $e_1, \dots, e_k \in E$ if every vertex in B_t is contained in at least one of the edges e_i. Hypertree width is always bounded by one plus the tree width [1], but the tree width cannot be bounded in terms of the hypertree width (see Example 2).

Unfortunately, we have not yet given the full definition of hypertree width; the notion we have defined so far is called *generalized hypertree width*. Hypertree width is defined by adding a technical condition, the so-called *special condition*, that restricts the way bags can be covered by hyperedges. A *hypertree decomposition* consists of a tree decomposition together with an appropriate cover of the bags by edges. We will give the full technical definition in Section 3.3. The special condition is needed to prove that for fixed k, hypergraphs of hypertree width k can be recognized and hypertree decompositions of width k can be

*Abteilung für Mathematische Logik, Albert-Ludwigs-Universität Freiburg, Eckerstr. 1, 79104 Freiburg, Germany. Email: isolde.adler@math.uni-freiburg.de

†Institut für Informationssysteme, TU Wien, Favoritenstr. 9-11/184-2, A-1040 Wien, Austria. Email: gottlob@dbai.tuwien.ac.at

Georg Gottlob's work was supported by the Austrian Science Fund (FWF) under Project N. P17222-N04 'Complementary Approaches to Constraint Satisfaction'.

‡Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany. Email: grohe@informatik.hu-berlin.de

Martin Grohe acknowledges travel support from the Wolfgang Pauli Society, Vienna.

computed in polynomial time. As an application, Gottlob, Leone, and Scarcello [6] showed that constraint satisfaction problems whose instances have an underlying hypergraph of bounded hypertree width can be solved in polynomial time. Similarly, conjunctive database queries whose underlying hypergraph has bounded hypertree width can be evaluated in polynomial time. For more information on the algorithmic aspects of hypertree width, we refer the reader to the introductory survey [4].

Despite the technical special condition, hypertree width seems to be a quite natural notion. In [7], Gottlob, Leone, and Scarcello characterize hypertree width in terms of a game that is defined like Seymour and Thomas's robber and cops game on graphs, except that instead of cops occupying vertices, the robber now has to escape *marshals* occupying hyperedges. Unfortunately, this characterization also has a small technical defect: The marshals are required to play monotonically, that is, they have to shrink the escape space of the robber in each move. It can be shown that there are hypergraphs where this reduces the strength of the marshals, that is, k marshals have a winning strategy, but no monotone winning strategy [1].

In this paper, we prove that the invariants hypertree width, generalized hypertree width, minimum number of marshals with a monotone winning strategy, and minimum number of marshals with a winning strategy are all equivalent. More precisely, they are all within a constant factor of $(3+\varepsilon)$ from one another. Furthermore, we introduce invariants for measuring the global connectivity of a hypergraph that resemble bramble number, tangle number, branch width, and linkedness of a graph and prove them all to be equivalent to hypertree width. Our results show that hypertree width is a similarly robust hypergraph invariant as tree width is for graphs. Furthermore, they imply that the algorithmic applications of hypertree width can be extended to generalized hypertree width.

The basic technical ideas underlying our results go back to [12, 13], where similar results are proved for various graph invariants. However, in places the situation on hypergraphs differs considerably from that on graphs, and graph invariants that coincide differ in their hypergraph version. We provide many examples of hypergraphs showing this unexpected behaviour.

2 Preliminaries and definitions

A *hypergraph* is a pair $H = (V(H), E(H))$, consisting of a nonempty set $V(H)$ of *vertices*, and a set $E(H)$ of subsets of $V(H)$, the *hyperedges* of H. We only consider finite hypergraphs. *Graphs* are hypergraphs in which all hyperedges have two elements. An *isolated vertex* in a hypergraph is a vertex that is not contained in any hyperedge.

For technical convenience, throughout this paper we make the following assumptions:

Proviso 1. Hypergraphs have at least one vertex and no isolated vertices.

For a hypergraph H and a set $X \subseteq V(H)$, the *subhypergraph induced by* X is the hypergraph $H[X] = (X, \{e \cap X \mid e \in E(H)\})$. We let $H \setminus X := H[V(H) \setminus X]$. The *primal graph* of a hypergraph H is the graph

$$\underline{H} = (V(H), \{\{v,w\} \mid v \neq w, \text{ there exists an } e \in E(H) \text{ such that } \{v,w\} \subseteq e\}).$$

A hypergraph H is *connected* if $\underline{H}$ is connected. A set $C \subseteq V(H)$ is *connected (in H)* if the induced subhypergraph $H[C]$ is connected, and a *connected component* of H is a maximal connected subset of $V(H)$. A sequence of nodes of $V(H)$ is a *path* of H if it is a path of $\underline{H}$.

Tree Decompositions and Hypertree Decompositions

In the following, let H be a hypergraph. A *tree decomposition* of H is a tuple (T, B), where T is a tree and $B = (B_t)_{t \in V(T)}$ a family of subsets of $V(H)$ such that for each $e \in E(H)$ there is a node $t \in V(T)$ such that $e \subseteq B_t$, and for each $v \in V(H)$ the set $\{t \in V(T) \mid v \in B_t\}$ is connected in T. The sets B_t are called the *bags* of the decomposition. The *width* of the decomposition (T, B) is $\max\{|B_t| \mid t \in T\} - 1$, and the *tree width* of H, denoted by $\mathrm{tw}(H)$, is the minimum of the widths of all tree decompositions of H.

It will be convenient for us to view the trees in tree decompositions as being rooted and directed from the root to the leaves. For a node t in a (rooted) tree $T = (V(T), E(T))$, we let T_t be the subtree rooted at t, that is, the induced subtree of T whose vertex set is the set of all vertices reachable from t.

A *generalized hypertree decomposition of* H is a triple (T,B,C), where (T,B) is a tree decomposition of H and $C=(C_t)_{t\in V(T)}$ is a family of subsets of $E(H)$ such that for every $t\in V(T)$ we have $B_t\subseteq\bigcup C_t$. Here $\bigcup C_t$ denotes the union of the sets (hyperedges) in C_t, that is, the set $\{v\in V(H)\mid \exists e\in C_t:\ v\in e\}$. The sets C_t are called the *guards* of the decomposition. The *width* of the decomposition (T,B,C) is $\max\{|C_t|\mid t\in V(T)\}$. The *generalized hypertree width* of H, denoted by $\mathrm{ghw}(H)$, is the minimum of the widths of the generalized hypertree decompositions of H.

A *hypertree decomposition* of H is a generalized hypertree decomposition (T,B,C) that satisfies the following *special condition*: $(\bigcup C_t)\cap\bigcup_{u\in V(T_t)}B_u\subseteq B_t$ for all $t\in V(T)$. Recall that T_t denotes the subtree of the T with root t. The *hypertree width* of H, denoted by $\mathrm{hw}(H)$, is the minimum of the widths of all hypertree decompositions of H.

Observe that $\mathrm{ghw}(H)\le\mathrm{hw}(H)$ and $\mathrm{ghw}(H)\le\mathrm{tw}(H)+1$. Actually, it has been shown in [5, 6] that the second inequality also holds for hypertree width instead of generalized hypertree width. Thus we have

$$\mathrm{ghw}(H)\le\mathrm{hw}(H)\le\mathrm{tw}(H)+1. \tag{1}$$

The following two examples show that both inequalities in (1) can be strict, and Example 4 shows a hypergraph H with $\mathrm{ghw}(H)=\mathrm{hw}(H)=\mathrm{tw}(H)+1$.

Example 2. Let H be the hypergraph obtained from a complete n-vertex graph K_n by adding one hyperedge that contains all vertices. Formally,

$$H=\Big(\{1,\dots,n\},\big\{\{i,j\}\bigm| 1\le i<j\le n\big\}\cup\big\{\{1,\dots,n\}\big\}\Big).$$

Then $\mathrm{ghw}(H)=\mathrm{hw}(H)=1$, because the one vertex tree with bag $\{1,\dots,n\}$ and guard $\big\{\{1,\dots,n\}\big\}$ is a hypertree decomposition of H of width 1.

However, we have $\mathrm{tw}(H)=n-1$, because every tree decomposition of H must have a bag that contains the hyperedge $\{1,\dots,n\}$. Let us remark that the *incidence graph* $\underline{\underline{H}}$ of H, defined by

$$\underline{\underline{H}}=\Big(V(H)\cup E(H),\big\{(v,e)\bigm| v\in V(H),e\in E(H),v\in e\big\}\Big)$$

has tree width n. The lower bound follows from the fact that $\underline{\underline{H}}$ contains a complete $(n+1)$-vertex graph as a minor.

Example 3 ([2])**.** Define

$$H=\Big(\{1,2,3,4,5,6,7,8,a,b\},\big\{\{1,8\},\{3,4\}\big\}\cup A\cup B\Big),$$

where $A=\big\{\{1,2,a\},\{4,5,a\},\{6,7,a\}\big\}$, and $B=\big\{\{2,3,b\},\{5,6,b\},\{7,8,b\}\big\}$ (see Figure 1). Then

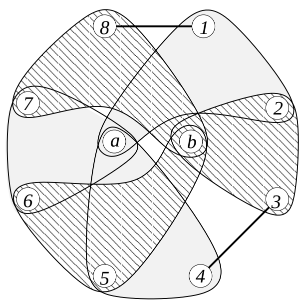

Figure 1: The hypergraph of Example 3

$\mathrm{ghw}(H)=2$ and $\mathrm{hw}(H)=3$. A generalized hypertree decomposition witnessing $\mathrm{ghw}(H)\le 2$ consists of a linear tree with four nodes, bags $\{1,2,7,8,a,b\},\{2,6,7,a,b\},\{2,5,6,a,b\},\{2,3,4,5,a,b\}$ and corresponding guards $\big\{\{1,2,a\},\{7,8,b\}\big\}$, $\big\{\{2,3,b\},\{6,7,a\}\big\}$, $\big\{\{1,2,a\},\{5,6,b\}\big\}$, $\big\{\{2,3,b\},\{4,5,a\}\big\}$. The special condition fails in the second node, so this is not a hypertree decomposition. To fix this we can add vertex 3 to the second and third bags, and hyperedge $\{2,3,b\}$ to the third guard. The resulting hypertree decomposition witnesses $\mathrm{hw}(H) \leq 3$. The proof that $\mathrm{hw}(H) \geq 3$ uses the game characterization of $\mathrm{hw}(H)$ by the (monotone) robbers and marshals game (see Theorem 9).

Example 4. Let $n \geq 3$, and let H be the graph obtained from K_n by duplicating each edge and then subdividing all edges once (that is, by replacing each edge by two parallel paths of length 2). Figure 2 shows the graph H for $n = 3$. Then $\mathrm{tw}(H) = n-1$. It is not hard to see that $\mathrm{ghw}(H) = n$, essentially

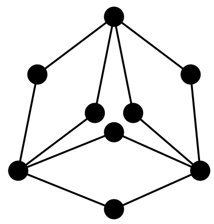

Figure 2: The graph obtained from K_3 as described in Example 4

because every tree decomposition of H must have a bag that contains all n vertices of the original K_n. By Equation (1) it follows that $\mathrm{hw}(H) = n$ as well.

3 Characterizations of hypertree width

3.1 From separators and hyperlinkedness ...

Let H be a hypergraph, $M \subseteq E(H)$ and $C \subseteq V(H)$. C is *M-big*, if it intersects more than half of the edges of M, that is,

$$\big|\{e \in M \mid e \cap C \neq \emptyset\}\big| > \frac{|M|}{2}.$$

Note that if $S \subseteq E(H)$, then $H \setminus \bigcup S$ has at most one M-big connected component.

Let $k \geq 0$ be an integer. A set $M \subseteq E(H)$ is *k-hyperlinked*, if for any set $S \subseteq E(H)$ with $|S| < k$, $H \setminus \bigcup S$ has an M-big component. Note that if M is k-hyperlinked, then M is also $(k-1)$-hyperlinked. The largest k for which H contains a k-hyperlinked set is called *hyperlinkedness of H*, $\mathrm{hlink}(H)$. Hyperlinkedness is an adaptation of the linkedness of a graph [12] to our setting.

A set $S \subseteq E(H)$ is a *balanced separator* for a set $M \subseteq E(H)$ if $H \setminus \bigcup S$ has no M-big connected component. Observe that $\mathrm{hlink}(H) \leq k$ if and only if every $M \subseteq E(H)$ has a balanced separator of size at most k.

Example 5. Let $H := \big(\{1,2,3,4,5\}, \{\{1,2\},\{1,3,5\},\{1,4\},\{2,5\},\{4,5\}\}\big)$ (cf. Figure 3).

- Let $M_1 := E(H)$. Then $S := \{\{1,3,5\}\}$ is a balanced separator of size 1 for M_1.
- $M_2 := \{\{1,2\},\{1,3,5\},\{2,5\}\}$ has no balanced separator of size 1.
- $\mathrm{hlink}(H) = 2$.

3.2 ... to brambles ...

Next, we adapt the bramble number of a graph [12] to our hypergraph context.

Let H be a hypergraph. Sets $X_1, X_2 \subseteq V(H)$ *touch* if $X_1 \cap X_2 \neq \emptyset$ or there exists an $e \in E(H)$ such that $e \cap X_1 \neq \emptyset$ and $e \cap X_2 \neq \emptyset$. A *bramble of H* is a set B of pairwise touching connected subsets of $V(H)$. The *hyperorder of a bramble B* is the least integer k such that there exists a set $S \subseteq E(H)$ with $|S| = k$ and $\bigcup S \cap X \neq \emptyset$ for all $X \in B$. The *hyperbramble number* hbramble-no(H) of H is the maximum of the hyperorders of all brambles of H.

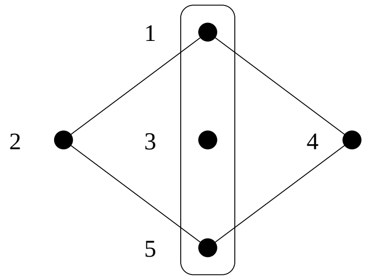

Figure 3: The hypergraph of Example 5

Lemma 6. *For every hypergraph H,*

$$\text{hlink}(H) \leq \text{hbramble-no}(H).$$

Proof. Let $\text{hlink}(H) = k$. Every k-hyperlinked set M generates a bramble of hyperorder at least k: Set $B := \{C \mid C \text{ the } M\text{-big component of } H \setminus \bigcup S \text{ for some } S \subseteq E(H),\ |S| < k\}$. Obviously, any two elements of B touch, and no set of fewer than k hyperedges intersects every element of B. Hence B is a bramble of order at least k. □

We cannot obtain equality here, as the following example shows:

Example 7. $\text{hlink}(K_5) \leq 2 < 3 \leq \text{hbramble-no}(K_5)$, where K_5 denotes the 5-clique.

To see that $3 \leq \text{hbramble-no}(K_5)$, note that $B := \big\{\{v\} \bigm| v \in V(K_5)\big\}$ is a bramble of hyperorder 3.

Towards $\text{hlink}(K_5) \leq 2$, let $M \subseteq E(K_5)$. For each $v \in V(K_5)$ let $S_v \subseteq E(K_5)$ be a set of two edges such that $\bigcup S_v = V(K_5) \setminus \{v\}$. It is sufficient to show that there is a vertex $v \in V(K_5)$ such that S_v is a balanced separator for M. To prove this, let $m_v := |\{e \in M \mid v \in e\}|$ for $v \in V(K_5)$. Then

$$|M| = \frac{1}{2}\Big(\sum_{v \in V(K_5)} m_v\Big).$$

Now suppose none of the S_v is a balanced separator for M. Then all $v \in V(K_5)$ satisfy $m_v > \frac{|M|}{2}$. Then

$$|M| = \frac{1}{2}\Big(\sum_{v \in V(K_5)} m_v\Big) \geq \frac{1}{2}\Big(5\frac{|M|}{2}\Big) = \frac{5}{4}|M|,$$

a contradiction. Hence there is a vertex $v \in V(K_5)$ s. t. S_v is a balanced separator for M.

3.3 ... and via marshals and hypertree width ...

The next hypergraph invariant we shall consider is based on a search game similar to *robber and cops* game due to Seymour and Thomas [14], which characterizes tree width. The *Robber and Marshals Game* on a hypergraph H is played by a *robber* against k *marshals* (the marshals act coordinated, so this is a two player game). The marshals move on the hyperedges of H, trying to catch the robber. They are not bound to move along paths of the hypergraph (we can imagine they use helicopters to move from one hyperedge to another). The robber moves on the vertices of H, and he can only move along paths (of the primal graph $\underline{H}$). He sees where the marshals intend to move to and quickly tries to escape running arbitrarily fast along paths of H, not being allowed to run through a vertex that is occupied by a marshal before *and* after the flight. The marshals' objective is to capture the robber by occupying a hyperedge that contains the vertex where the robber is. The robber tries to elude capture.

Formally, the game $\text{RM}(H,k)$ (the *robber and k marshals game on H*) is played by two players, the *robber* and the *marshals*. A *position* of the game is a pair (v,M), where $v \in V(H)$ and $M \subseteq E(H)$ with $|M| \leq k$. To start a play of the game, the robber picks an arbitrary v_0, and the initial position is $(v_0,\emptyset)$. In each round, the players move from the current position (v,M) to a new position (v',M') as follows: The marshals select M', and then the robber selects v' such that there is a path from v to v' in the hypergraph

$$H \setminus (\bigcup M \cap \bigcup M').$$

If a position (v,M) with $v \in M$ is reached, the play ends and the marshals win. If the play continues forever, the robber wins.

Winning strategies for the players are defined in the natural way. A winning strategy for the robber is *monotone* if in each play following this strategy and each move of this play, say, from position (v,M) to position (v',M'), the connected component of v' in $H \setminus M'$ is a subset of the connected component of v in $H \setminus M$. Intuitively, a strategy for the marshals is monotone if the escape space of the robber never increases during a play.

The *marshal width* $\text{mw}(H)$ of H is the least k such that the marshals have a winning strategy for the game $\text{RM}(H,k)$. The *monotone marshal width* $\text{mon-mw}(H)$ of H is the least k such that the marshals have a monotone winning strategy for the game $\text{RM}(H,k)$.

Example 8. The hypergraph H from Example 5 (Figure 3) satisfies $\text{mw}(H) = \text{mon-mw}(H) = 2$.

Obviously, for all hypergraphs H we have

$$\text{mw}(H) \leq \text{mon-mw}(H). \tag{2}$$

It has been shown in [1, Theorem 4.1] that for every k there is a hypergraph H such that $\text{mw}(H) \leq \text{mon-mw}(H) - k$. (The proof actually shows that $\text{ghw}(H) \leq \text{mon-mw}(H) - k$.) The monotone marshal width turns out to be precisely the hypertree width:

Theorem 9 ([7])**.** *For every Hypergraph H,*

$$\text{hw}(H) = \text{mon-mw}(H).$$

The next lemma relates marshal width to hyperbramble number and hence hyperlinkedness.

Lemma 10. *For every Hypergraph H,*

$$\text{hbramble-no}(H) \leq \text{mw}(H).$$

Proof. Let B be a bramble in H of hyperorder $k = \text{hbramble-no}(H)$. We show that the robber has a winning strategy for the game $\text{RM}(H,k-1)$: Observe first that for every possible position M of $k-1$ marshals on H there is an $X \in B$ such that $X \cap \bigcup M = \emptyset$. The robber's strategy is to always be on a vertex $v \in X$ for some $X \in B$ that does not intersect the hyperedges currently occupied by the marshals.

The robber starts the game on an arbitrary vertex of an arbitrary $X \in B$. Suppose now that the game is in a position (v,M), where $v \in X$ for some $X \in B$ with $X \cap \bigcup M = \emptyset$. Suppose that in the next round of the game the marshals move to M'. Let $X' \in B$ such that $X' \cap \bigcup M' = \emptyset$. Then $(X \cup X') \cap (\bigcup M \cap \bigcup M') = \emptyset$, and since the sets X and X' are connected and they touch, the robber can move along vertices from $X \cup X'$ from v to a vertex $v' \in X'$.

Hence the robber can elude capture forever and thus he wins the game. □

3.4 ... back to separators

Let H be a hypergraph and $X \subseteq V(H)$. For every connected component C of $H \setminus X$ we let

$$\partial C = \{v \in X \mid \text{there is a hyperedge } e \in E(H) \text{ with } v \in e \text{ and } e \cap C \neq \emptyset\}.$$

Lemma 11. *Let $k \geq 1$, H a hypergraph with $\text{hlink}(H) \leq k$, and $M \subseteq E(H)$ with $|M| \leq 2k+1$.*

Then there exists a set $N \subseteq E(H)$ with $M \subseteq N$ and $|N| \leq 3k+1$ such that for all components C' of $H \setminus (\bigcup N)$ there exists a subset $M' \subseteq N$ with $|M'| \leq 2k$ and $\partial C' \subseteq \bigcup M'$.

Proof. Let S be a balanced separator for M with $|S| \leq k$ and $N := M \cup S$. Then $M \subseteq N$ and $|N| \leq 3k+1$. Let C' be a component of $H \setminus (\bigcup N)$. Then $C' \subseteq C$ for a component C of $H \setminus (\bigcup S)$. Since S is a balanced separator for M, C is not M-big, that is,

$$|\{e \in M \mid e \cap C \neq \emptyset\}| \leq \frac{|M|}{2} = \frac{2k+1}{2}.$$

Let $M' := S \cup \{e \in M \mid e \cap C \neq \emptyset\}$, then $M' \subseteq N$ and $|M'| \leq k+k$. Furthermore,

$$\partial C' \subseteq \bigcup \{e \in N \mid e \cap C \neq \emptyset\} \subseteq \bigcup S \cup \bigcup \{e \in M \mid e \cap C \neq \emptyset\} = \bigcup M'. \qquad \square$$

Lemma 12. *For every hypergraph H,*

$$\text{mon-mw}(H) \leq 3 \cdot \text{hlink}(H) + 1.$$

Proof. Let $\text{hlink}(H) = k$. We describe a monotone winning strategy for the marshals in the game $\text{RM}(H, 3k+1)$. We claim that from an arbitrary position (v, M) of the game with $|M| \leq 2k$, in two moves the marshals can force the robber to a position (v', M') where $|M'| \leq 2k$ and the connected component C' of v' in $H \setminus \bigcup M'$ is *strictly* contained in the connected component C of v in $H \setminus \bigcup M$. Clearly, this implies the existence of the desired monotone winning strategy, because the initial position $(v, \emptyset)$ satisfies $|\emptyset| \leq 2k$.

To prove the claim, let (v, M) be a position of the game with $|M| \leq 2k$ and let C be the connected component of v in $H \setminus \bigcup M$. Let $e \in E(H)$ be a hyperedge with $e \cap C \neq \emptyset$. We apply Lemma 11 to $M \cup \{e\}$. We obtain a set $N \subseteq E(H)$. Let C' be the connected component of v in $H \setminus \bigcup N$. Then $C' \subseteq C \setminus e \subset C$. In the first move, the marshals move from M to N. The robber can only move to some vertex w of C'. In position (w, N), the marshals move to a set $M' \subseteq N$ with $|M'| \leq 2k$ and $\partial C' \subseteq \bigcup M'$. Again, the robber can only move to some v' in C'. This proves the claim. $\square$

3.5 From brambles to tangles ...

Let H be a hypergraph. Recall that a bramble in H is a set of connected subsets of $V(H)$ any two of which touch. A *tangle* in H is a bramble A such that the touching condition holds even for triples of elements, that is, for all $X_1, X_2, X_3 \in A$ either $X_1 \cap X_2 \cap X_3 = \emptyset$ or there exists an $e \in E(H)$ with $e \cap X_i \neq \emptyset$ for $i \in \{1,2,3\}$. As for brambles, the *hyperorder of a tangle* A is the least integer k such that there exists a set $S \subseteq E(H)$ with $|S| = k$ and $\bigcup S \cap X \neq \emptyset$ for all $X \in A$. The *hypertangle number* of a hypergraph H, denoted by $\text{htangle-no}(H)$, is the maximum of the hyperorders of all tangles in H.

Example 13. The triangle K_3 satisfies $1 = \text{htangle-no}(K_3) < \text{hbramble-no}(K_3) = 2$.

Lemma 14. *For every hypergraph H,*

$$\text{htangle-no}(H) \leq \text{hbramble-no}(H) \leq 3 \cdot \text{htangle-no}(H).$$

Proof. The first inequality holds because every tangle is a bramble. For the second inequality, let B be a bramble in H of hyperorder $k = \text{hbramble-no}(H)$. We show that H has a tangle A of hyperorder at least $k/3$. Let $S \subseteq E(H)$ with $|S| < k/3$. Then there is an $X \in B$ such that $\bigcup S \cap X = \emptyset$, and all $X' \in B$ with $\bigcup S \cap X' = \emptyset$ are subsets of the same connected component C_S of $H \setminus \bigcup S$, because any two elements of B touch. We let

$$A := \left\{ C_S \;\middle|\; S \subseteq E(H),\ |S| < \frac{k}{3} \right\}.$$

We claim that A is a tangle of hyperorder $\geq \frac{k}{3}$. Clearly, all elements of A are connected, and for each $S \subseteq E(H)$ with $|S| < k/3$ there is a tangle element — C_S — that has an empty intersection with $\bigcup S$. It remains to verify the touching condition. Let $C_1, C_2, C_3 \in A$ and let $S_1, S_2, S_3 \subseteq E(H)$ such that $|S_i| < k/3$ and $C_i = C_{S_i}$. Let $S = S_1 \cup S_2 \cup S_3$. Then $|S| < k$. Thus there exists an $X \in B$ such that $X \cap \bigcup S = \emptyset$. Since each of the C_i contains an $X_i \in B$ and since X touches X_i, we must have $X \subseteq C_i$. Thus $X \subseteq C_1 \cap C_2 \cap C_3$. $\square$

3.6 ... and via branch decompositions back to hypertree decompositions

A tree T is *subcubic* if every vertex has degree at most 3. A *branch decomposition* of a hypergraph H is a pair (T,τ), where T is a subcubic tree, and τ is a bijection from $E(H)$ to the set of leaves of T. For every edge $f=\{t,u\}\in V(T)$ we let $L_{f,t}$ and $L_{f,u}$ be the leaf-sets of the two subtrees into which the tree is divided if the edge f is removed. (Obviously, $L_{f,u}$ is the leaf-set of the subtree that contains u.) We let $D_{f,t}$ be the set of vertices in edges contained in $\tau^{-1}(L_{f,t})$, that is,

$$D_{f,t}=\bigcup\tau^{-1}(L_{f,t})=\big\{v\in e\;\big|\;e\in E(H)\text{ with }\tau(e)\in L_{f,t}\big\}$$

and define $D_{f,u}$ analogously. The *hyperorder* of f is the minumum number of hyperedges of H needed to cover $D_{f,t}\cap D_{f,u}$, that is,

$$\text{hyperorder}(f)=\min\big\{|S|\;\big|\;S\subseteq E(H)\text{ such that }D_{f,u}\cap D_{f,t}\subseteq\bigcup S\big\}.$$

The *hyperwidth* of the branch decomposition (T,τ) is the maximum of the hyperorders of all edges $f\in E(T)$, or if T has no edges, it is 0. The *hyperbranch width* of H, denoted by hbranch-width(H), is the minimum of the hyperwidths of all branch decompositions of H.

Observe that hbranch-width$(H)=0$ if and only if the edges of H are pairwise disjoint. To simplify the following statements, let us call a hypergraph whose edges are pairwise disjoint *trivial*.

Lemma 15. *For every nontrivial hypergraph H,*

$$\text{htangle-no}(H)\le\text{hbranch-width}(H).$$

Proof. Let (T,τ) be a branch decomposition of H of width $k=\text{hbranch-width}(H)\ge1$. For every edge $f=\{t,u\}\in E(T)$, let $S_f\subseteq E(H)$ such that $|S_f|\le k$ and $D_{f,u}\cap D_{f,t}\subseteq\bigcup S_f$. Let $C_{f,u}=D_{f,u}\setminus\bigcup S_f$ and $C_{f,t}=D_{f,t}\setminus\bigcup S_f$.

Suppose for contradiction that H has a tangle A of hyperorder at least $k+1$. Then for every edge $f=\{t,u\}\in E(T)$ there is an $X_f\in A$ such that $\bigcup S_f\cap X_f=\emptyset$. Then either $X_f\subseteq C_{f,u}$ or $X_f\subseteq C_{f,t}$. If $X_f\subseteq C_{f,u}$, we let $\vec{f}=(t,u)$; otherwise we let $\vec{f}=(u,t)$. This orientation of f does not depend on the particular choice of the set $X_f\in A$, because all elements of A touch and hence must be on the same side. We orient all edges $f\in E(T)$ in this way. We claim that in the resulting oriented tree, no vertex has outdegree 0. Clearly, this leads to a contradiction and hence proves the lemma. To prove the claim, suppose for contradiction that $t\in V(T)$ has outdegree 0.

Case 1: t is a leaf. Let $e=\tau^{-1}(t)$, and let $\vec{f}=(u,t)$ be the edge directed towards t. Then $X_f\subseteq e$. Since the hyperorder of the tangle A is at least $k+1\ge2$, there must be a $Y\in A$ such that $Y\cap e=\emptyset$. Since Y and X_f touch, there is an $e'\in E(H)$ such that $e'\cap X_f\neq\emptyset$ and $e'\cap Y\neq\emptyset$. Since $e'\in\tau^{-1}(L_{f,u})$ and $e\in\tau^{-1}(L_{f,t})$, we have $e\cap e'\subseteq\bigcup S_f$ and hence $X_f\cap e'\subseteq\bigcup S_f$. Thus $X_f\cap\bigcup S_f\neq\emptyset$, which is a contradiction.

Case 2: t has degree 2. This case is treated similarly to the more complicated case 3.

Case 3: t has degree 3. Let $\vec{f}_1=(u_1,t)$, $\vec{f}_2=(u_2,t)$, and $\vec{f}_3=(u_3,t)$ be the three oriented edges incident with t. For $i\in\{1,2,3\}$ let $X_i=X_{f_i}$. Since A is a tangle and $X_1,X_2,X_3\in A$, either $X_1\cap X_2\cap X_3\neq\emptyset$ or there is an edge $e\in E(H)$ such that $e\cap X_i\neq\emptyset$ for $i\in\{1,2,3\}$.

Observe that

$$\begin{aligned}X_1&\subseteq C_{f_1,t}\subseteq(D_{f_2,u_2}\cup D_{f_3,u_3})\setminus\bigcup S_{f_1},\\X_2&\subseteq C_{f_2,t}\subseteq(D_{f_1,u_1}\cup D_{f_3,u_3})\setminus\bigcup S_{f_2},\\X_3&\subseteq C_{f_3,t}\subseteq(D_{f_1,u_1}\cup D_{f_2,u_2})\setminus\bigcup S_{f_3}.\end{aligned}$$

Suppose that $v\in X_1\cap X_2\cap X_3$. Then v is contained in at least two of the sets $D_{f_1,u_1},D_{f_2,u_2},D_{f_2,u_2}$ and in none of the sets $\bigcup S_{f_1},\bigcup S_{f_2},\bigcup S_{f_3}$. Say, $v\in D_{f_1,u_1}\cap D_{f_2,u_2}$. But now observe that

$$D_{f_1,u_1}\cap D_{f_2,u_2}\subseteq D_{f_1,u_1}\cap D_{f_1,t}\subseteq\bigcup S_{f_1}.$$

This contradicts the fact that $v \notin \bigcup S_{f_1}$. Hence $X_1 \cap X_2 \cap X_3 = \emptyset$. Suppose next that there is an edge $e \in E(H)$ such that $e \cap X_i = \emptyset$ for $i \in \{1,2,3\}$. Since $L_{f_1,u_1} \cup L_{f_2,u_2} \cup L_{f_3,u_3}$ is the set of all leaves of T, we have $\tau(e) \in L_{f_i,u_i}$ for some $i \in \{1,2,3\}$. Without loss of generality we assume that $\tau(e) \in L_{f_1,u_1}$. But then $e \cap X_1 = \emptyset$, which again is a contradiction. Thus there is no edge $e \in E(H)$ such that $e \cap X_i = \emptyset$ for $i \in \{1,2,3\}$. Overall, we have reached a contradiction. □

Lemma 16. *For every hypergraph H,*

$$\text{hbranch-width}(H) \leq \text{ghw}(H).$$

Proof. Let (T,B,C) be a generalized hypertree decomposition of H. We first transform this decomposition into a new decomposition (T',B',C') and define a bijection τ from $E(H)$ to the leaves of T' such that for every $e \in E(H)$ we have $B_{\tau(e)} = e$ and $C_{\tau(e)} = \{e\}$. To achieve this, for every edge $e \in E(H)$ we pick a vertex $t_e \in V(T)$ such that $e \subseteq B_{t_e}$. We define the tree T' by attaching a new leaf ℓ_e to t_e for every edge $e \in E(H)$. If there are other leaves in T' than the newly created leaves ℓ_e, we delete them, and if the deletion creates new leaves, we delete them as well, until the leaves ℓ_e are the only leaves of T'. For the interior vertices t of T' we let $B'_t = B_t$ and $C'_t = C_t$. For the leaves, we let $B'_{\ell_e} = e, C'_{\ell_e} = \{e\}$. We define the bijection τ by $\tau(e) = \ell_e$. It is easy to see that (T',B',C') and τ have the desired properties.

In a second step, we turn T' into a subcubic tree T'' that has the same leaves as T' by repeatedly splitting nodes of degree greater than 3. For example, if t has neighbours $u_1,\dots,u_k$, where $k \geq 4$, we replace t by nodes t_1 and t_2, connect these two nodes by an edge, and attach $u_1,\dots,u_{\lfloor k/2 \rfloor}$ to t_1 and $u_{\lfloor k/2 \rfloor+1},\dots,u_k$ to t_2. We define B'' and C'' on T'' by letting the split vertices keep their bags and guards. That is, if we split t into t_1 and t_2, we let $B''_{t_1} = B''_{t_2} = B'_t$ and $C''_{t_1} = C''_{t_2} = C'_t$.

We obtain a generalized hypertree decomposition (T'',B'',C'') of H and a bijection τ from $E(H)$ to the leaves of T'' such that T'' is a subcubic tree, and for every $e \in E(H)$ we have $B_{\tau(e)} = e$ and $C_{\tau(e)} = \{e\}$. Then (T'',τ) is a branch decomposition of H. We claim that the hyperwidth of this decomposition is at most k. To see this, let $f = \{t,u\}$ be an edge of T''. It is a fundamental property of tree decompositions that $B_t \cap B_u$ separates the vertices in the bags of the two parts of the tree obtained by removing the edge f. Thus in particular, $D_{f,t} \cap D_{f,u} \subseteq B_t \cap B_u \subseteq \bigcup C_t$. Hence the hyperorder of e is at most $|C_t| \leq k$. □

Lemma 17. *For every nontrivial hypergraph H,*

$$\text{ghw}(H) \leq 2 \cdot \text{hbranch-width}(H).$$

Proof. Let (T,τ) be a branch decomposition of H of width $k = \text{hbranch-width}(H)$. For every edge $f = \{t,u\} \in E(T)$, let $S_f \subseteq E(H)$ such that $|S_f| \leq k$ and $D_{f,t} \cap D_{f,u} \subseteq \bigcup S_f$. We define a generalized hypertree decomposition (T,B,C) as follows: For an interior vertex $t \in V(T)$, let $e_1 = \{u_1,t\}, e_2 = \{u_2,t\} \in E(T)$ be two of the edges incident with t. We let

$$\begin{aligned} B_t &= \big(D_{e_1,u_1} \cap D_{e_1,t}\big) \cup \big(D_{e_2,u_2} \cap D_{e_2,t}\big), \\ C_t &= S_{f_1} \cup S_{f_2}. \end{aligned}$$

For a leaf ℓ with $\tau^{-1}(\ell) = e$, we let $B_\ell = e$ and $C_\ell = \{e\}$. Let us first argue that (T,B) is a tree-decomposition of H: For every edge $e \in E(H)$ we have $e \subseteq B_{\tau(e)}$. For a vertex $v \in V(H)$, consider the set $B^{-1}(v) = \{t \in V(T) \mid v \in B_t\}$. An interior vertex $t \in V(T)$ belongs to this set, if at least two of the (at most three) components of $T \setminus \{t\}$ have a leaf ℓ such that $v \in \tau^{-1}(\ell)$. A leaf ℓ belongs to $B^{-1}(v)$ if $v \in \tau^{-1}(\ell)$. Thus $B^{-1}(v)$ is the union of all paths connecting leaves ℓ with $v \in \tau^{-1}(\ell)$. Clearly, this set is connected. Thus (T,B) is a tree-decomposition of H.

It follows immediately from the definition of the guards C_t that $B_t \subseteq \bigcup C_t$ for all $t \in V(T)$, thus (T,B,C) is a generalized hypetree decomposition. Since $|S_f| \leq k$ for all $f \in E(T)$, the width of this decomposition is at most $2k$. □

The following example shows that the inequalities in the previous two lemmas are tight:

Example 18.

1. For the triangle K_3 we have $\text{hbranch-width}(K_3) = 1$ and $\text{ghw}(K_3) = 2$. Thus $\text{ghw}(K_3) = 2 \cdot \text{hbranch-width}(K_3)$.
2. For the 4-cycle C_4 we have $\text{ghw}(C_4) = \text{hbranch-width}(C_4) = 2$.

3.7 Putting things together

Theorem 19. *Let H be a hypergraph. Then:*

1. $\text{hlink}(H) \leq \text{hbramble-no}(H) \leq \text{mw}(H) \leq \text{ghw}(H) \leq \text{hw}(H) = \text{mon-mw}(H) \leq 3 \cdot \text{hlink}(H) + 1$.
2. $\text{htangle-no}(H) \leq \text{hbramble-no}(H) \leq 3 \cdot \text{htangle-no}(H)$.
3. *If H is nontrivial (that is, its edges are not pairwise disjoint), then*

$$\text{htangle-no}(H) \leq \text{hbranch-width}(H) \leq \text{ghw}(H) \leq 2 \cdot \text{hbranch-width}(H).$$

(If H is trivial, then $\text{hbranch-width}(H) = 0$*, and the other invariants take the value* 1*.)*

In particular, all these hypergraph invariants are equivalent.

The various examples we gave show that for almost all of the inequalities there are hypergraphs for which they are strict. Two inequalities that are still conceivable to be equalities are

$$\text{hbramble-no}(H) \leq \text{mw}(H) \quad \text{and} \quad \text{htangle-no}(H) \leq \text{hbranch-width}(H).$$

Let us remark that for the corresponding graph invariants, both inequalities are actually equalities. But that does not mean too much, because it is also the case for several other inequalities of which we know that they can be strict for hypergraphs.

But even though for most of the inequalities we know that they can be strict, in almost no cases we know whether our bounds are tight. For example, it is an open question whether the inequality $\text{hw}(H) \leq 3 \cdot \text{ghw}(H) + 1$ can be improved.

4 Concluding remarks

4.1 Branch decompositions and submodularity

Robertson and Seymour [13] and later Oum and Seymour [10, 11] laid out a general theory of branch decompositions of which branch decompositions of graphs are just one instance. For a finite set E and a function $\kappa : 2^E \to \mathbb{R}$ that is *symmetric*, that is, $\kappa(X) = \kappa(E \setminus X)$ for all $X \subseteq E$, a *branch decomposition* of (E, κ) is defined to be a pair (T, τ), where T is a subcubic tree and τ is a bijection from E to the set of leaves of T. Again, by $L_{f,t}$ and $L_{f,u}$ we denote the leaf-sets of the two parts into which the tree is divided if the edge $f = \{t, u\} \in E(T)$ is removed. We define the *weight* of f to be $\kappa(\tau^{-1}(L_{f,t}))$. Note that this is equal to $\kappa(\tau^{-1}(L_{f,u}))$, because $\tau^{-1}(L_{f,u}) = E \setminus \tau^{-1}(L_{f,u})$ and κ is symmetric. The *width* of the decomposition (T, τ) is the maximum of the weights of all edges, and the *branch width* of (E, κ) is the minimum of the widths of all branch decompositions of (E, κ).

For the standard branch width of a hypergraph H, we let $E = E(H)$ and, for $X \subseteq E$,

$$\kappa(X) = \Big|\bigcup X \cap \bigcup (E \setminus X)\Big|.$$

Then the branch width of H is the branch width of (E, κ). For the hyperbranch width of H, we also let $E = E(H)$, but we define κ by letting

$$\kappa(X) = \min\Big\{|S| \;\Big|\; S \subseteq E \text{ such that } \bigcup X \cap \bigcup (E \setminus X) \subseteq \bigcup S\Big\} \tag{3}$$

(for $X \subseteq E$). Other invariants that can be described as branch decompositions are the *rank width of graphs* and the *branch width of matroids*.

Branch width is particularly well-behaved if the function κ is *submodular*, that is, for all $X, Y \subseteq E$,

$$\kappa(X) + \kappa(Y) \geq \kappa(X \cup Y) + \kappa(X \cap Y).$$

Using a general minimization algorithm for submodular functions [9], Oum and Seymour [11] proved that if κ is submodular and computable in polynomial time, then there is an algorithm that, given $k \in \mathbb{N}$ and

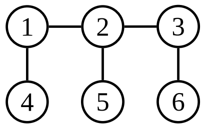

Figure 4: The hypergraph of Example 20

an instance (E,κ) of branch width at most k, computes a branch decomposition of (E,κ) of width at most $3k+1$ in time $f(k)\cdot p(n)$ for some computable function f and polynomial p. Here n denotes the size of the instance.

The following example shows that the κ for hyperbranch width is not submodular:

Example 20. Let H be the hypergraph

$$\Big(\{1,\ldots,6\},\big\{\{1,2\},\{2,3\},\{1,4\},\{2,5\},\{3,6\}\big\}\Big)$$

(cf. Figure 4). Let $\kappa : E(H) \to \mathbb{N}$ be defined as in (3). Then for $X = \big\{\{1,2\},\{2,5\}\big\}$ and $Y = \big\{\{2,3\},\{2,5\}\big\}$ we have $\kappa(X) = 1$, $\kappa(Y) = 1$, $\kappa(X\cup Y) = 2$, and $\kappa(X\cap Y) = 1$. Thus

$$\kappa(X) + \kappa(Y) = 2 < 3 = \kappa(X\cup Y) + \kappa(X\cap Y),$$

and therefore κ is not submodular.

Thus the general theory developed for branch decompositions of submodular functions does not apply to hyperbranch width. It is an interesting open question whether there is an alternative definition of a function κ that is submodular, but still yields a branch width equivalent to hypertree width. Conversely, the fact that our κ for hyperbranch width is not submodular, but still has an interesting theory, indicates that the submodularity condition may not be necessary for a reasonable notion of branch width.

4.2 Fractional hypertree width

The main motivation for the introduction of hypertree width was that several important algorithmic problems (constraint satisfaction problems from artificial intelligence, conjunctive query containment and conjunctive query evaluation from database theory) can be solved in polynomial time on instances whose "underlying hypergraph" has bounded hypertree width. Our results imply that these algorithmic properties also hold for instances whose hypergraph has bounded generalized hypertree width. This was the original motivation for our work, and when we wrote the first version of this paper, classes of instances whose hypergraph has bounded generalized hypertree width were the largest known "structurally defined" classes for which these problems are tractable. We refer the reader to [3] for a discussion of structurally defined tractable constraint satisfaction problems.

In the meantime, even larger structurally defined classes of tractable instances have been discovered [8]. They are based on a relaxation of generalized hypertree width called *fractional hypertree width*. A *fractional hypertree decomposition* of a hypergraph is defined as a hypertree decomposition, except that the guards can be divided and distributed over many edges. Formally, a fractional hypertree decomposition of a hypergraph H is a triple (T,B,γ), where (T,B) is a tree decomposition and $\gamma = (\gamma_t)_{t\in V(T)}$ is a family of functions from $E(H)$ to the nonnegative reals such that for each $t \in V(T)$ and each $v \in B_t$ we have $\sum_{e\in E(H), v\in e} \gamma_t(e) \geq 1$. The width of the decomposition is $\max_{t\in V(T)} \sum_{e\in E(H)} \gamma_t(e)$. It is known that hypertree width and fractional hypertree width are not equivalent [8], as a matter of fact, there is a family of hypergraphs of fractional hypertree width 2 and unbounded hypertree width.

Nevertheless, our theory also applies to fractional hypertree width. It is not hard to see that most of the invariants considered here and the results proved here have a "fractional version", which can usually be proved by the same techniques as those used here.

References

[1] I. Adler. Marshals, monotone marshals, and hypertree-width. *Journal of Graph Theory*, 47:275–296, 2004.

[2] I. Adler. A note on monotonicity cost on hypergraphs. *Unpublished note*, 2005.

[3] D. Cohen, P. Jeavons, and M. Gyssens. A unified theory of structural tractability for constraint satisfaction and spread cut decomposition. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005. To appear.

[4] G. Gottlob, M. Grohe, N. Musliu, M. Samer, and F. Scarcello. Hypertree decompositions: Structure, algorithms, and applications. In D. Kratsch, editor, *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 3787 of *Lecture Notes in Computer Science*, pages 1–15. Springer Verlag, 2005.

[5] G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

[6] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64:579–627, 2002.

[7] G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: Game theoretic and logical characterizations of hypertree width. *Journal of Computer and System Sciences*, 66:775–808, 2003.

[8] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *Proceedings of the of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2006. To appear.

[9] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

[10] S.-I. Oum. Approximating rank-width and clique-width quickly. In D. Kratsch, editor, *WG2005*, volume 3787 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2005.

[11] S.-I. Oum and P.D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*. To appear.

[12] B. Reed. Tree width and tangles: A new connectivity measure and some applications. In R.A. Bailey, editor, *Surveys in Combinatorics*, volume 241 of *LMS Lecture Note Series*, pages 87–162. Cambridge University Press, 1997.

[13] N. Robertson and P.D. Seymour. Graph minors X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52:153–190, 1991.

[14] P.D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58:22–33, 1993.