Constraint Solving via Fractional Edge Covers

Martin Grohe* Dániel Marx*

Abstract

Many important combinatorial problems can be modelled as constraint satisfaction problems, hence identifying polynomial-time solvable classes of constraint satisfaction problems received a lot of attention. In this paper, we are interested in structural properties that can make the problem tractable. So far, the largest structural class that is known to be polynomial-time solvable is the class of bounded hypertree width instances introduced by Gottlob et al. [20]. Here we identify a new class of polynomial-time solvable instances: those having bounded fractional edge cover number.

Combining hypertree width and fractional edge cover number, we then introduce the notion of fractional hypertree width. We prove that constraint satisfaction problems with bounded fractional hypertree width can be solved in polynomial time (provided that a the tree decomposition is given in the input). We also prove that certain parameterized constraint satisfaction, homomorphism, and embedding problems are fixed-parameter tractable on instances having bounded fractional hypertree width.

1. Introduction

Constraint satisfaction problems form a large class of combinatorial problems that contains many important "real-world" problems. An instance of a constraint satisfaction problem consists of a set V of variables, a domain D, and a set C of constraints. For example, the domain may be $\{0,1\}$, and the constraints may be the clauses of a 3-CNF-formula. The objective is to assign values in D to the variables in such a way that all constraints are satisfied. In general, constraint satisfaction problems are NP-hard; considerable efforts, both practical and theoretical, have been made to identify tractable classes of constraint satisfaction problems.

On the theoretical side, there are two main directions towards identifying polynomial-time solvable classes of constraint satisfaction problems. One is to restrict the *constraint language*, that is, the type of constraints that are allowed (see, for example, [5, 6, 7, 15, 24, 28]). Formally, the constraint language can be described as a set of relations on the domain. The other direction is to restrict the *structure* induced by the constraints on the variables (see, for example, [10, 11, 13, 16, 25]). The present work goes into this direction; our main contribution is the identification of a natural new class of structurally tractable constraint satisfaction problems.

The *hypergraph* of an instance (V, D, C) has V as its vertex set and for every constraint in C a hyperedge that consists of all variables occurring in the constraint. For a class $\mathcal{H}$ of hypergraphs, we let $\textsc{Csp}(\mathcal{H})$ be the class of all instances whose hypergraph is contained in $\mathcal{H}$. The central questions is for which classes $\mathcal{H}$ of hypergraphs the problem $\textsc{Csp}(\mathcal{H})$ is tractable. Most recently, this question has been studied in [8, 10]. It is worth pointing out that the corresponding question for the graphs (instead of hypergraphs) of instances, in which two variables are incident if they appear together in a constraint, has been completely answered in [22, 23] (under the complexity theoretic assumption FPT $\neq$ W[1]): For a class $\mathcal{G}$ of graphs, the corresponding problem $\textsc{Csp}(\mathcal{G})$ is in polynomial time if and only if $\mathcal{G}$ has bounded tree width. This can be generalized to $\textsc{Csp}(\mathcal{H})$ for classes $\mathcal{H}$ of hypergraphs of *bounded hyperedge size* (that is, classes $\mathcal{H}$ for which $\max\{|e| \mid \exists\, H = (V,E) \in \mathcal{H} :\ e \in E\}$ exists). It follows easily from the results of [22, 23] that for all classes $\mathcal{H}$ of bounded hyperedge size,

(1.1) $\textsc{Csp}(\mathcal{H}) \in$ PTIME $\iff$ $\mathcal{H}$ has bounded tree width

(under the assumption FPT $\neq$ W[1]).

It is known that (1) does not generalize to arbitrary classes $\mathcal{H}$ of hypergraphs (we will give a very simple counterexample in Section 2). The largest known family of classes of hypergraphs for which $\textsc{Csp}(\mathcal{H})$ is in PTIME consists of all classes of bounded *hypertree width* [20, 21, 19]. Hypertree width is a hypergraph invariant that generalizes acyclicity [4, 14, 30]. It is a very robust invariant; up to a constant factor it coincides with a number of other natural invariants that measure the global connectivity of a hypergraph [2]. On classes of bounded hyperedge size, bounded hypertree width coincides with bounded tree width, but in general it does not. It has been asked in [8, 10, 18, 22] whether there are classes $\mathcal{H}$ of unbounded hypertree width such that $\textsc{Csp}(\mathcal{H}) \in$ PTIME. We give an affirmative answer to this question.

*Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany.
{grohe,dmarx}@informatik.hu-berlin.de

Our key result states that $\text{CSP}(\mathcal{H}) \in$ PTIME for all classes $\mathcal{H}$ of bounded *fractional edge cover number*. A *fractional edge cover* of a hypergraph $H = (V, E)$ is a mapping $\psi : E \to [0, \infty)$ such that $\sum_{e \in E, v \in e} \psi(e) \geq 1$ for all $v \in V$. The number $\sum_{e \in E} \psi(e)$ is the *weight* of ψ. The *fractional edge cover number* $\rho^*(H)$ of H is the minimum of the weights of all fractional edge covers of H. It follows from standard linear programming results that this minimum exists and is rational. Furthermore, it is easy to construct classes $\mathcal{H}$ of hypergraphs that have bounded fractional edge cover number and unbounded hypertree width (see Example 8).

We then start a more systematic investigation of the interaction between fractional covers and hypertree width. We propose a new hypergraph invariant, the *fractional hypertree width*, which generalizes both the hypertree width and fractional edge cover number in a natural way. Fractional hypertree width is an interesting hybrid of the "continuous" fractional edge cover number and the "discrete" hypertree width. We show that it has similarly nice properties as hypertree width. In particular, we give an approximative game characterization of fractional hypertree width similar to the characterization of tree width by the "robber and cops" game [29]. Furthermore, we prove that for classes $\mathcal{H}$ of bounded fractional hypertree width, the problem $\text{CSP}(\mathcal{H})$ can be solved in polynomial time provided that a fractional hypertree decomposition of the underlying hypergraph is given together with the input instance. We leave open the question of whether for fixed k there is a polynomial-time algorithm that computes a fractional-hypertree decomposition of width k of a given hypergraph H of fractional hypertree width k.

We also discuss the problem of evaluating conjunctive database queries and the homomorphism problem for relational structures, which are both known to be equivalent to constraint satisfaction problems, and the embedding problem for relational structures. We show that all these problems, as well as constraint satisfaction problems parameterized by the number of variables, are *fixed parameter tractable* for instances of bounded fractional hypertree width.

2. Preliminaries

2.1. Hypergraphs. A *hypergraph* is a pair $H = (V(H), E(H))$, consisting of a set $V(H)$ of *vertices* and a set $E(H)$ of subsets of $V(H)$, the *hyperedges* of H. We always assume that hypergraphs have no isolated vertices, that is, for every $v \in V(H)$ there exists at least one $e \in E(H)$ such that $v \in e$.

For a hypergraph H and a set $X \subseteq V(H)$, the *subhypergraph of H 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 vertices of H is a *path* of H if it is a path of $\underline{H}$.

A *tree decomposition* of a hypergraph H is a tuple $(T, (B_t)_{t \in V(T)})$, where T is a tree and $(B_t)_{t \in V(T)}$ 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 a tree-decomposition $(T, (B_t)_{t \in V(T)})$ is $\max\{|B_t| \mid t \in V(t)\} - 1$. The *tree width* $\text{tw}(H)$ of a hypergraph H is the minimum of the widths of all tree-decompositions of H. It is easy to see that $\text{tw}(H) = \text{tw}(\underline{H})$ for all 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.

We say that a class $\mathcal{H}$ of hypergraphs is of *bounded tree width* if there is a k such that $\text{tw}(H) \leq k$ for all $H \in \mathcal{H}$. We use a similar terminology for other hypergraph invariants.

2.2. Constraint satisfaction problems. A *CSP instance* is a triple $I = (V, D, C)$, where V is a set of *variables*, D is a set called the *domain*, and C is a set of *constraints* of the form $\langle (v_1, \ldots, v_k), R \rangle$, where $k \geq 1$ and R is a k-ary relation on D. A *solution* to the instance I is an assignment $\alpha : V \to D$ such that for all constraints $\langle (v_1, \ldots, v_k), R \rangle$ in C we have $(\alpha(v_1), \ldots, \alpha(v_k)) \in R$.

Constraints are specified by explicitly enumerating all possible combinations of values for the variables, that is, all tuples in the relation R. Consequently, we define the *size* of a constraint $c = \langle (v_1, \ldots, v_k), R \rangle \in C$ to be the number $\|c\| = k + k \cdot |R|$. The *size* of an instance $I = (V, D, C)$ is the number $\|I\| = |V| + |D| + \sum_{c \in C} \|c\|$. Of course there is no need to store a constraint relation repeatedly if it occurs in several constraints, but this only changes the size by a polynomial factor.

Let us make a few remarks about this explicit representation of the constraints. There are important special cases of constraint satisfaction problems where the constraints are stored implicitly, which may make the representation exponentially more succinct. Examples are Boolean satisfiability, where the constraint relations are given implicitly by the clauses of a formula in conjunctive normal form, or systems of arithmetic (in)equalities, where the constraints are given

implicitly by the (in)equalities. However, our representation is the standard "generic" representation of constraint satisfaction problems in artificial inntelligence (see, for example, [12]). An important application where the constraints are always given in explicit form is the conjunctive query containment problem, which plays a crucial role in database query optimization. Kolaitis and Vardi [25] observed that it can be represented as a constraint satisfaction problem, and the constraint relations are given explicitly as part of one of the input queries. A related problem from database systems is the problem of evaluating conjunctive queries (cf. Theorem 17). Here the constraint relations represent the tables of a relational database, and again they are given in explictit form.

Observe that there is a polynomial-time algorithm deciding whether a given assignment for an instance is a solution.

The *hypergraph of* the CSP instance $I = (V, D, C)$ is the hypergraph H_I with vertex set V and a hyperedge $\{v_1, \ldots, v_k\}$ for all constraints $\langle (v_1, \ldots, v_k), R\rangle$ in C. For every class $\mathcal{H}$, we consider the following decision problem:

$\text{CSP}(\mathcal{H})$
Instance: A CSP instance I with $H_I \in \mathcal{H}$.
Problem: Decide if I has a solution.

If the class $\mathcal{H}$ is not polynomial-time decidable, we view this as a promise problem, that is, we assume that we are only given instances I with $H_I \in \mathcal{H}$, and we are only interested in algorithms that work correctly and efficiently on such instances.

We close this section with a simple example of a class of hypergraphs of unbounded tree width such that $\text{CSP}(\mathcal{H})$ is tractable.

Example 1. Let $\mathcal{H}$ be that class of all hypergraphs H that have a hyperedge that contains all vertices, that is, $V(H) \in E(H)$. Clearly, $\mathcal{H}$ has unbounded tree width, because the hypergraph $(V, \{V\})$ has tree width $|V| - 1$. We claim that $\text{CSP}(\mathcal{H}) \in$ PTIME.

To see this, let $I = (V, D, C)$ be an instance of $\text{CSP}(\mathcal{H})$. Let $\langle (v_1, \ldots, v_k), R\rangle$ be a constraint in C with $\{v_1, \ldots, v_k\} = V$. Such a constraint exists because $H_I \in \mathcal{H}$. Each tuple $\bar{d} = (d_1, \ldots, d_k) \in R$ completely specifies an assignment $\alpha_{\bar{d}}$ defined by $\alpha_{\bar{d}}(v_i) = d_i$ for $1 \leq i \leq k$. If for some i, j we have $v_i = v_j$, but $d_i \neq d_j$, we leave $\alpha_{\bar{d}}$ undefined.

Observe that I is satisfiable if and only if there is a tuple $\bar{d} \in R$ such that $\alpha_{\bar{d}}$ is (well-defined and) a solution for I. As $|R| \leq \|I\|$, this can be checked in polynomial time.

3. A Polynomial-time algorithm for CSPs with bounded fractional cover number

In this section we prove that if the hypergraph H_I of a CSP instance I has fractional edge cover number $\rho^*(H_I)$, then it can be decided in $\|I\|^{\rho^*(H_I)+O(1)}$ time whether I has a solution. Thus if $\mathcal{H}$ is a class of hypergraphs with bounded fractional edge cover number (that is, there is a constant r such that $\rho^*(H) \leq r$ for every $H \in \mathcal{H}$), then $\text{CSP}(\mathcal{H}) \in$ PTIME.

Actually, we prove a stronger result: A CSP instance I has at most $\|I\|^{\rho^*(H_I)}$ solutions and all the solutions can be enumerated in time $\|I\|^{\rho^*(H_I)+O(1)}$.

The proof relies on a combinatorial lemma known as Shearer's Lemma. We use Shearer's lemma to bound the number of solutions of a CSP instance; our argument resembles an argument that Friedgut and Kahn [17] used to bound the number of subhypergraphs of a certain isomorphism type in a hypergraph. The second author recently applied similar ideas in a completely different algorithmic context [26].

Lemma 2 (Shearer's Lemma [9]). *Let $H = (V, E)$ be a hypergraph, and let $A_1, A_2, \ldots, A_p$ be (not necessarily distinct) subsets of V such that each $v \in V$ is contained in at least q of the A_i's. Denote by E_i the edge set of the induced hypergraph $H[A_i]$. Then*

$$|E| \leq \prod_{i=1}^{p} |E_i|^{1/q}.$$

Note that we admit empty hyperedges. In particular, if $e \cap A_i = \emptyset$ for some $e \in E$ and $i \leq p$, then $\emptyset \in E_i$. Lemma 2 is easy to see in the special case when $q = 1$ and $\{A_1, \ldots, A_p\}$ is a partition of V. The proof of the general case is based on an entropy argument.

Lemma 3. *A CSP instance I has at most $\|I\|^{\rho^*(H_I)}$ solutions.*

Proof: Consider a CSP instance $I = (V, D, C)$ with $V = \{v_1, \ldots, v_n\}$. Let $\hat{H}$ be a hypergraph over $V \times D$ where the edges correspond to the solutions of the instance: for each solution $\alpha : V \to D$ there is an edge $\{(v_1, \alpha(v_1)), \ldots, (v_n, \alpha(v_n))\}$ in $\hat{H}$. We will bound the number of edges in $\hat{H}$ using Shearer's Lemma. Let ψ be a fractional edge cover of H_I with $\sum_{e \in E} \psi(e) = \rho^*(H_I)$; it follows from the standard results of linear programming that such a ψ exists with rational values. Denote by q the least common denominator of the values $\psi(e)$. Let $p = \rho^*(H_I) \cdot q$, and let $A_1, \ldots, A_p$ be a sequence of edges such that edge $e \in E$ appears exactly $\psi(e) \cdot q$ times. From the definition of the fractional edge cover, it follows that each vertex $v \in V$ appears in at least q of the A_i's. Define $\hat{A}_i = A_i \times D$, these sets cover every vertex of $\hat{H}$ at least q times. Hence by

Shearer's Lemma, the number of edges of $\hat{H}$ can be bounded by

$$\prod_{i=1}^{p} |\hat{E}_i|^{1/q},$$

where $\hat{E}_i$ is the edge set of the subhypergraph $\hat{H}[\hat{A}_i]$ induced by $\hat{A}_i$. The hypergraph $\hat{H}[\hat{A}_i]$ describes how the solutions look like if we consider only the variables in A_i: each edge of $\hat{H}[\hat{A}_i]$ describes a possible combination of values that a solution can have on the variables in A_i. More precisely, assume that A_i corresponds to some constraint $\langle(v'_1, \ldots, v'_k), R\rangle$. The hypergraph $\hat{H}[\hat{A}_i]$ contains the edge $\{(v'_1, d_1), \ldots, (v'_k, d_k)\}$ if and only if there is a solution α with $\alpha(v'_i) = d_i$. This means that there can be at most $|R| \leq \|I\|$ edges in $\hat{E}_i$. Hence the number of edges of $\hat{H}$ and hence the number of solutions can be bounded by

$$\prod_{i=1}^{p} \|I\|^{1/q} = \|I\|^{p/q} = \|I\|^{\rho^*(H_I)}. \qquad \square$$

We would like to turn the upper bound of Lemma 3 into an algorithm enumerating all the solutions, but the proof of Shearer's Lemma is not algorithmic. However, a very simple algorithm can enumerate the solutions, and Lemma 3 can be used to bound the running time of this algorithm. We need the following definition:

Definition 4. Let $I = (V, D, C)$ be a CSP instance and let $V' \subseteq V$ be a nonempty subset of variables. The *CSP instance $I[V']$ induced by V'* is $I' = (V', D, C')$, where C' is defined the following way: For each constraint $c = \langle(v_1, \ldots, v_k), R\rangle$ having at least one variable in V', there is a corresponding constraint c' in C'. Suppose that $v_{i_1}, \ldots, v_{i_\ell}$ are the variables among $v_1, \ldots, v_k$ that are in V'. Then the constraint c' is defined as $\langle(v_{i_1}, \ldots, v_{i_\ell}), R'\rangle$, where the relation R' is the projection of R to the components $i_1, \ldots, i_\ell$, that is, R' contains an ℓ-tuple $(d'_1, \ldots, d'_\ell) \in D^\ell$ if and only if there is a k-tuple $(d_1, \ldots, d_k) \in R$ such that $d'_j = d_{i_j}$ for $1 \leq j \leq \ell$.

Thus an assignment α on V' satisfies $I[V']$ if for each constraint c of I, there is an assignment extending α that satisfies c (however, it is not necessariliy true that there is an assignment extending α that satisfies every constraint of I simultaneously).

Theorem 5. *The solutions of a CSP instance I can be enumerated in time $\|I\|^{\rho^*(H_I)+O(1)}$.*

Proof: If $V = \{v_1, \ldots, v_n\}$ is an arbitrary ordering of the variables of I, then let V_i be the subset $\{v_1, \ldots, v_i\}$. For $i = 1, 2, \ldots, n$, the algorithm creates a list L_i containing the solutions of $I[V_i]$. Since $I[V_n] = I$, the list L_n is exactly what we want.

For $i = 1$, the instance $I[V_i]$ has at most $|D|$ solutions, hence the list L_i is easy to construct. Notice that a solution of $I[V_{i+1}]$ induces a solution of $I[V_i]$. Therefore, the list L_{i+1} can be constructed by considering the solutions in L_i, extending them to the variable v_{i+1} in all the $|D|$ possible ways, and checking whether this assignment is a solution of $I[V_{i+1}]$. Clearly, this can be done in $|L_i| \cdot |D| \cdot \|I[V_{i+1}]\|^{O(1)} = |L_i| \cdot \|I\|^{O(1)}$ time. Repeating this procedure for $i = 1, 2, \ldots, n-1$, the list L_n can be constructed.

The total running time of the algorithm can be bounded by $\sum_{i=1}^{n-1} |L_i| \cdot \|I\|^{O(1)}$. Observe that $\rho^*(H_{I[V_i]}) \leq \rho^*(H_I)$: $H_{I[V_i]}$ is the subhypergraph of H_I induced by V_i, thus any fractional cover of the hypergraph of I gives a fractional cover of $I[V_i]$. Therefore, by Lemma 3, $|L_i| \leq \|I\|^{\rho^*(H_I)}$, and it follows that the total running time is $\|I\|^{\rho^*(H_I)+O(1)}$. $\square$

We note that the algorithm of Theorem 5 does not actually need a fractional edge cover: the fact that the hypergraph has small fractional edge cover number is used only in proving the time bound of the algorithm.

Corollary 6. *Let $\mathcal{H}$ be a class of hypergraphs of bounded fractional edge cover number. Then $\textsc{Csp}(\mathcal{H})$ is in polynomial time.*

4. Fractional hypertree decompositions

Let H be a hypergraph. A *generalized hypertree decomposition of H* is a triple $(T, (B_t)_{t\in V(T)}, (C_t)_{t\in V(T)})$, where $(T, (B_t)_{t\in V(T)})$ is a tree decomposition of H and $(C_t)_{t\in V(T)}$ 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 $\{v \in V(H) \mid \exists e \in C_t : v \in e\}$. We call the sets B_t the *bags* of the decomposition and the sets C_t the *guards*. The *width* of $(T, (B_t)_{t\in V(T)}, (C_t)_{t\in V(T)})$ is $\max\{|C_t| \mid t \in V(T)\}$. The *generalized hypertree width* $\mathrm{ghw}(H)$ of H is the minimum of the widths of the generalized hypertree decompositions of H.

For the sake of completeness, let us mention that a *hypertree decomposition* of H is a generalized hypertree decomposition $(T, (B_t)_{t\in V(T)}, (C_t)_{t\in V(T)})$ that satisfies the following additional *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* $\mathrm{hw}(H)$ of H is the minimum of the widths of all hypertree decompositions of H. It has been proved in [2] that $\mathrm{ghw}(H) \leq \mathrm{hw}(H) \leq 3 \cdot \mathrm{ghw}(H) + 1$. This means that for our purposes, hypertree width and generalized hypertree width are equivalent. For simplicity, we will only work with generalized hypertree width.

Observe that for every hypergraph H we have $\text{ghw}(H) \leq \text{tw}(H) + 1$. Furthermore, if H is a hypergraph with $V(H) \in E(H)$ we have $\text{ghw}(H) = 1$ and $\text{tw}(H) = |V(H)| - 1$.

We now give an approximate characterization of (generalized) hypertree width by a game that is a variant of the *robber and cops* game [29], which characterizes tree width: In the *robber and marshals game on* H [21], a robber plays against k marshals. The marshals move on the hyperedges of H, trying to catch the robber. In each move, some of the marshals fly in helicopters to new hyperedges. The robber moves on the vertices of H. She sees where the marshals will be landing 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 land a marshal via helicopter on a hyperedge containing the vertex occupied by the robber. The robber tries to elude capture. The *marshal width* $\text{mw}(H)$ of a hypergraph H is the least number k of marshals that have a winning strategy in the robber and marshals game played on H (see [1] or [21] for a formal definition).

It is easy to see that $\text{mw}(H) \leq \text{ghw}(H)$ for every hypergraph H. To win the game on a hypertree of generalized hypertree width k, the marshals always occupy guards of a decomposition and eventually capture the robber at a leaf of the tree. Conversely, it can be proved that $\text{ghw}(H) \leq 3 \cdot \text{mw}(H) + 1$.

Observe that for every hypergraph H, the generalised hypertree width is less than or equal to the (integral) edge cover number of H. The following two examples show that hypertree width and fractional edge cover number are incomparable.

Example 7. Consider the class of all graphs that only have disjoint edges. The tree width and hypertree width of this class is 1, the fractional edge cover number is unbounded.

Example 8. For $n \geq 1$, let H_n be the following hypergraph: H_n has a vertex v_S for every subset S of $\{1, \ldots, 2n\}$ of cardinality n. Furthermore, for every $i \in \{1, \ldots, 2n\}$ the hypergraph H_n has a hyperedge $e_i = \{v_S \mid i \in S\}$.

Observe that the fractional edge cover number $\rho^*(H_n)$ is at most 2, because the mapping ψ that assigns $1/n$ to every hyperedge e_i is a fractional edge cover of weight 2. Actually, it is easy to see that $\rho^*(H_n) = 2$.

We claim that the hypertree width of H_n is n. It is easy to see that H_n has a hypertree decomposition of width n (with a two node tree). Thus $\text{ghw}(H_n) \leq n$. To see that $\text{ghw}(H_n) > n - 1$, we argue that the robber has a winning strategy against $(n-1)$ marshals in the robber and marshals game. Consider a position of the game where the marshals occupy edges $e_{j_1}, \ldots, e_{j_{n-1}}$ and the robber occupies a vertex v_S for a set S with $S \cap \{j_1, \ldots, j_{n-1}\} = \emptyset$. Suppose that in the next round of the game the marshals move to the edges $e_{k_1}, \ldots, e_{k_{n-1}}$. The robber moves along the edge e_i to a vertex v_R for a set $R \subseteq \{1, \ldots, 2n\} \setminus \{k_1, \ldots, k_{n-1}\}$ of cardinality n that contains i. If she plays this way, she can never be captured.

For a hypergraph H and a mapping $\gamma : E(H) \to [0, \infty)$, we let

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

We may think of $B(\gamma)$ as the set of all vertices "blocked" by γ. Furthermore, we let $\text{weight}(\gamma) = \sum_{e \in E} \gamma(e)$.

Definition 9. Let H be a hypergraph. A *fractional hypertree decomposition of* H is a triple $(T, (B_t)_{t \in V(T)}, (\gamma_t)_{t \in V(T)})$, where $(T, (B_t)_{t \in V(T)})$ is a tree decomposition of H and $(\gamma_t)_{t \in V(T)}$ is a family of mappings from $E(H)$ to $[0, \infty)$ such that for every $t \in V(T)$ we have $B_t \subseteq B(\gamma_t)$.

We call the sets B_t the *bags* of the decomposition and the mappings γ_t the *(fractional) guards*.

The *width* of $(T, (B_t)_{t \in V(T)}, (\gamma_t)_{t \in V(T)})$ is $\max\{\text{weight}(\gamma_t) \mid t \in V(T)\}$. The *fractional hypertree width* $\text{fhw}(H)$ of H is the minimum of the widths of the fractional hypertree decompositions of H.

It is easy to see that the minimum of the widths of all fractional hypertree decompositions of a hypergraph H always exists and is rational. This follows from the fact that, up to an obvious equivalence, there are only finitely many tree decompositions of a hypergraph.

Clearly, for every hypergraph H we have

$$\text{fhw}(H) \leq \rho^*(H) \quad \text{and} \quad \text{fhw}(H) \leq \text{ghw}(H).$$

Examples 7 and 8 above show that there are families of hypergraphs of bounded fractional hypertree width, but unbounded fractional edge cover number and unbounded generalized hypertree width.

It is also worth pointing out that for every hypergraph H,

$$\text{fhw}(H) = 1 \iff \text{ghw}(H) = 1.$$

To see this, note that if $\gamma : E(H) \to [0, \infty)$ is a mapping with $\text{weight}(\gamma) = 1$ and $B \subseteq B(\gamma)$, then $B \subseteq e$ for all $e \in E(H)$ with $\gamma(e) > 0$. Thus instead using γ as a guard in a fractional hypertree decomposition, we may use the integral guard $\{e\}$ for any $e \in E(H)$ with $\gamma(e) > 0$. Let us remark that $\text{ghw}(H) = 1$ if and only if H is acyclic [20].

4.1. The robber and army game. As robbers are getting ever more clever, it takes more and more powerful security forces to capture them. In the *robber and army game on a hypergraph* H, a robber plays against a general commanding an army of r battalions of soldiers. The general may distribute his soldiers arbitrarily on the hyperedges.

However, a vertex of the hypergraph is only blocked if the number of soldiers on all hyperedges that contain this vertex adds up to the strength of at least one battalion. The game is then played like the robber and marshals game.

Definition 10. Let H be a hypergraph and r a nonnegative real. The game $\mathrm{RA}(H, r)$ (the *robber and army game on H with r battalions* is played by two players, the *robber* and the *general*. A *position* of the game is a pair (γ, v), where $v \in V(H)$ and $\gamma : E(H) \to [0, \infty)$ with $\mathrm{weight}(\gamma) \leq r$. To start a play of the game, the robber picks an arbitrary v_0, and the initial position is $(0, v_0)$, where 0 denote the constant zero mapping.

In each round, the players move from the current position (γ, v) to a new position (γ', v') as follows: The general selects γ', and then the robber selects v' such that there is a path from v to v' in the hypergraph $H \setminus \big(B(\gamma) \cap B(\gamma')\big)$.

If a position (γ, v) with $v \in B(\gamma)$ is reached, the play ends and the general wins. If the play continues forever, the robber wins.

The *army width* $\mathrm{aw}(H)$ of H is the least r such that the general has winning strategy for the game $\mathrm{RA}(H, r)$.

Again, it is easy to see that $\mathrm{aw}(H)$ is well-defined and rational.

Theorem 11. *For every hypergraph H,*

$$\mathrm{aw}(H) \leq \mathrm{fhw}(H) \leq 3 \cdot \mathrm{aw}(H) + 2.$$

The rest of this subsection is devoted to a proof of this theorem. The proof is similar to the proof of the corresponding result for the robber and marshal game and generalized hypertree width in [2], which in turn is based on ideas from [27, 29].

Let H be a hypergraph and $\gamma, \sigma : E(H) \to [0, \infty)$. For a set $W \subseteq V(H)$, we let

$$\mathrm{weight}(\gamma|W) = \sum_{\substack{e \in E(H) \\ e \cap W \neq \emptyset}} \gamma(e).$$

A mapping $\sigma : E(H) \to [0, \infty)$ is a *balanced separator for* γ if for every connected component R of $H \setminus B(\sigma)$,

$$\mathrm{weight}(\gamma|R) \leq \frac{\mathrm{weight}(\gamma)}{2}.$$

Lemma 12. *Let H be a hypergraph and $r = \mathrm{aw}(H)$. Then every $\gamma : E(H) \to [0, \infty)$ has a balanced separator of weight r.*

Proof: Suppose for contradiction that $\gamma : E(H) \to [0, \infty)$ has no balanced separator of weight r. We claim that the robber has a winning strategy for the game $\mathrm{RA}(H, r)$. The robber simply maintains the invariant that in every position (σ, v) of the game, v is contained in the connected component R of $H \setminus B(\sigma)$ with $\mathrm{weight}(\gamma|R) > \mathrm{weight}(\gamma)/2$.

To see that this is possible, let (σ, v) be such a position. Suppose that the general moves from σ to σ', and let R' be the connected component of $H \setminus B(\sigma')$ with $\mathrm{weight}(\gamma|R') > \mathrm{weight}(\gamma)/2$. Then there must be some $e \in E(H)$ such that $e \cap R \neq \emptyset$ and $e \cap R' \neq \emptyset$, because otherwise we had

$$\begin{aligned}\mathrm{weight}(\gamma) &= \mathrm{weight}(\gamma)/2 + \mathrm{weight}(\gamma)/2 \\ &< \mathrm{weight}(\gamma|R) + \mathrm{weight}(\gamma|R') \leq \mathrm{weight}(\gamma),\end{aligned}$$

which is impossible. Thus the robber can move from R to R' via the edge e. □

Let H be a hypergraph and H' an induced subhypergraph of H. Then the *restriction* of a mapping $\gamma : E(H) \to [0, \infty)$ to H' is the mapping $\gamma' : E(H') \to [0, \infty)$ defined by

$$\gamma'(e') = \sum_{\substack{e \in E(H) \\ e \cap V(H') = e'}} \gamma(e).$$

Conversely, the *canonical extension* of a mapping $\gamma' : E(H') \to [0, \infty)$ to H is the mapping $\gamma : E(H) \to [0, \infty)$ defined by

$$\gamma(e) = \frac{\gamma'(e \cap V(H'))}{|\{e_1 \in E(H) \mid e_1 \cap V(H') = e \cap V(H)\}|}.$$

Note that in both cases, we have $\mathrm{weight}(\gamma) = \mathrm{weight}(\gamma')$ and $B(\gamma') = B(\gamma) \cap V(H')$.

Proof of Theorem 11: Let H be a hypergraph. To prove that $\mathrm{aw}(H) \leq \mathrm{fhw}(H)$, let $(T, (B_t)_{t \in V(T)}, (\gamma_t)_{t \in V(T)})$ be a fractional hypertree decomposition of H. We claim that the general has a winning strategy for $\mathrm{RA}(H, r)$. Let $(0, v_0)$ be the initial position. The general plays in such a way that all subsequent positions are of the form (γ_t, v) such that $v \in B_u$ for some $u \in V(T_t)$. Intuitively, this means that the robber is trapped in the subtree below t. Furthermore, in each move the general reduces the height of t. He starts by selecting γ_{t_0} for the root t_0 of T. Suppose the game is in a position (γ_t, v) such that $v \in B_u$ for some $u \in V(T_t)$. If $u = t$, then the robber has lost the game. So let us assume that $u \neq t$. Then there is a child t' of t such that $u \in V(T_{t'})$. The general moves to $\gamma_{t'}$. Suppose the robber escapes to a v' that is not contained in $B_{u'}$ for any $u' \in T_{t'}$. Then there is a path from v to v' in $H \setminus (B(\gamma_t) \cap B(\gamma_{t'}))$ and hence in $H \setminus (B_t \cap B_{t'})$. However, it follows easily from the fact that $(T, (B_t)_{t \in T})$ is a tree decomposition of H that every path from a bag in $T_{t'}$ to a bag in $T \setminus T_{t'}$ must intersect $B_t \cap B_{t'}$. This proves that $\mathrm{aw}(H) \leq \mathrm{fhw}(H)$.

For the second inequality, we shall prove the following stronger claim:

Claim: Let H be a hypergraph and $r = \mathrm{aw}(H)$. Furthermore, let $\gamma : E(H) \to [0, \infty)$ such that $\mathrm{weight}(\gamma) \leq 2r + 2$. Then there exists a fractional hypertree decomposition of H of width at most $3r + 2$ such that $B(\gamma)$ is contained in the bag of the root of this decomposition.

Note that for $\gamma = 0$, the claim yields the desired fractional hypertree decomposition of H.

Proof of the claim: The proof is by induction on the cardinality of $V(H) \setminus B(\gamma)$.

By Lemma 12, there is a balanced separator of weight at most r for γ in H. Let σ be such a separator, and define $\chi : E(H) \to [0, \infty)$ by $\chi(e) = \gamma(e) + \sigma(e)$. Then $\mathrm{weight}(\chi) \leq 3r + 2$, and $B(\gamma) \subseteq B(\chi)$.

If $V(H) = B(\chi)$ (this is the induction bases), then the 1-node decomposition with bag $V(H)$ and guard χ is a fractional hypertree decomposition of H of width at most $3r + 2$.

Otherwise, let $R_1, \ldots, R_m$ be the connected components of $H \setminus B(\chi)$. Note that we cannot exclude the case $m = 1$ and $R_1 = V(H) \setminus B(\gamma)$.

For $1 \leq i \leq m$, let e_i be an edge of H such that $e_i \cap R_i \neq \emptyset$, and let S_i be the unique connected component of $H \setminus B(\sigma)$ with $R_i \subseteq S_i$. Note that $\mathrm{weight}(\gamma|S_i) \leq r + 1$, because σ is a balanced separator for γ. Let $\chi_i : E(H) \to [0, \infty)$ be defined by

$$\chi_i(e) = \begin{cases} 1 & \text{if } e = e_i, \\ \sigma(e) + \gamma(e) & \text{if } e \neq e_i \text{ and } S_i \cap e \neq \emptyset, \\ \sigma(e) & \text{otherwise.} \end{cases}$$

Then

$$\mathrm{weight}(\chi_i) \leq 1 + \mathrm{weight}(\sigma) + \mathrm{weight}(\gamma|S_i) \leq 2r + 2.$$

Let $H_i = H[R_i \cup B(\chi_i)]$ and observe that

$$V(H_i) \setminus B(\chi_i) \subseteq R_i \setminus e \subset R_i \subseteq V(H) \setminus B(\gamma).$$

Thus the induction hypothesis is applicable to H_i and the restriction of χ_i to H_i. It yields a fractional hypertree decomposition $(T^i, (B^i_t)_{t \in V(T^i)}, (\gamma^i_t)_{t \in V(T^i)})$ of H_i of weight at most $3r + 2$ such that $B(\chi_i) \cap V(H_i)$ is contained in the bag $B^i_{t^i_0}$ of the root t^i_0 of T^i.

Let T be the disjoint union of $T^1, \ldots, T^m$ together with at a new root t_0 that has edges to the roots t^i_0 of the T^i. Let $B_{t_0} = B(\chi)$ and $B_t = B^i_t$ for all $t \in V(T^i)$. Moreover, let $\gamma_{t_0} = \chi$, and let γ_t be the canonical extension of γ^i_t to H for all $t \in V(T^i)$.

It remains to prove that $(T, (B_t)_{t \in V(T)}, (\gamma_t)_{t \in V(T)})$ is a fractional hypertree decomposition of H of width at most $3r + 2$. Let us first verify that $(T, (B_t)_{t \in V(T)})$ is a tree decomposition.

- Let $v \in V(H)$. To see that $\{v \in V(T) \mid v \in B_t\}$ is connected in T, observe that $\{t \in V(T^i) \mid v \in B_{t_i}\}$ is connected (maybe empty) for all i. If $v \in V(H_i)$ for exactly one i, this already shows that $\{v \in V(T) \mid v \in B_t\}$ is connected. Otherwise, $v \in B(\chi) = B_{t_0}$ and hence $v \in B(\chi_i) \subseteq B_{t^i_0}$ for all i such that $v \in V(H_i)$. Again this shows that $\{v \in V(T) \mid v \in B_t\}$ is connected.

- Let $e \in E(H)$. Either $e \subseteq B(\chi) = B_{t_0}$, or there is exactly one i such that $e \subseteq R_i \cup B(\chi_i)$. In the latter case, $e \subseteq B_t$ for some $t \in V(T^i)$.

It remains to prove that $B_t \subseteq B(\gamma_t)$ for all $t \in T$. For the root, we have $B_{t_0} = B(\gamma_{t_0})$. For $t \in V(T^i)$, we have $B_t \subseteq B(\gamma^i_t) = B(\gamma_t) \cap V(H^i) \subseteq B(\gamma_t)$. Finally, note that $\mathrm{weight}(\gamma_t) \leq 3r + 2$ for all $t \in V(T)$. This completes the proof of the claim. □

Remark 13. With respect to the difference between hypertree decompositions and generalized hypertree decompositions, it is worth observing that the fractional tree decomposition $(T, (B_t)_{t \in V(T)}, (\gamma_t)_{t \in V(T)})$ of width at most $3r + 2$ constructed in the proof of the theorem satisfies the following special condition: $B(\gamma_t) \cap \bigcup_{u \in V(T_t)} B_u \subseteq B_t$ for all $t \in V(T)$.

This implies that a hypergraph of fractional hypertree width at most r has a fractional hypertree decomposition of width at most $3r + 2$ that satisfies the special condition.

4.2. Finding decompositions. We currently do not know whether for any fixed $r > 1$ there is a polynomial-time algorithm that, given a hypergraph H of fractional hypertree width at most r, computes a fractional hypertree decomposition of H of width r or at least of width $f(r)$ for some function f. Similar to hypertree width, one way of obtaining such an algorithm would be through the army and robber game characterization. The idea would be to inductively compute the set of all positions of the game from which the general wins in $0, 1, \ldots$ rounds. The problem is that, as opposed to the robber and marshals game, there is no polynomial bound on the number of positions. Let us state a conjecture which would imply such a polynomial bound for a sufficiently large set of positions and hence for every $r > 1$ a polynomial-time algorithm that, given a hypergraph H of fractional hypertree width at most r, computes a fractional hypertree decomposition of H of width $3r + 2$.

Let H be a hypergraph and $r \geq 1$. Let us call a set $B \subseteq V(H)$ *maximal r-covered* if there is a $\gamma : E(H) \to [0, \infty)$ with $\mathrm{weight}(\gamma) \leq r$ and $B \subseteq B(\gamma)$, but there is no $B' \supset B$ for which such a γ exists. Note that a covering γ for a maximal r-covered set can easily be computed in polynomial time by linear programming. In the robber and army game,

we may assume without loss of generality that the general always plays mappings γ such that $B(\gamma)$ is maximal r-covered.

Conjecture 14. *Let $r \geq 1$. Then for every hypergraph H, there are at most $|V(H) + E(H)|^{O(\text{fhw}(H))}$ maximal r-covered sets. Furthermore, there is a polynomial-time algorithm that enumerates all these sets.*

4.3. Algorithmic applications. Recall that H_I denotes the hypergraph of a CSP instance I.

Theorem 15. *Let $r \geq 1$. Then there is a polynomial-time algorithm that, given a CSP instance I and a fractional hypertree decomposition D of H_I of width at most r, decides if I is satisfiable (and computes a solution if it is).*

Proof: Given a fractional hypertree decomposition $(T, (B_t)_{t\in V(T)}, (\gamma_t)_{t\in V(T)})$ of an instance I, define $V_t := \bigcup_{t\in V(T_t)} B_t$. For each $t \in V(T)$, our algorithm constructs the list L_t of those solutions of $I[B_t]$ that can be extended to a solution of $I[V_t]$ (recall that $I[B_t]$ denotes the instance induced by B_t, see Definition 4.) Clearly, I has a solution if and only if L_{t_0} is not empty for the root t_0 of the tree decomposition.

The algorithm proceeds in a bottom-up manner: when constructing the list L_t, we assume that for every child t' of t, the lists $L_{t'}$ are already available.

If t is a leaf node, then $V_t = B_t$, and L_t is simply the list of all solutions of $I[B_t]$. By the definition of fractional hypertree decompositions, the hypergraph of $I[B_t]$ has fractional edge cover number at most r, hence by Theorem 5, L_t can be determined in time $\|I\|^{r+O(1)}$.

Assume now that t has children $t_1, \ldots, t_k$. It is easy to see that a solution α of $I[B_t]$ can be extended to $I[V_t]$ if and only if for each $1 \leq i \leq k$, there is a solution α_i of $I[V_{t_i}]$ that is compatible with α (that is, α and α_i assign the same values to the variables in $B_t \cap V_{t_i}$). Therefore, L_t can be determined by first enumerating every solution α of $I[B_t]$ using the algorithm of Theorem 5, and then for each i, going through the list L_{t_i} and checking whether it contains an assignment compatible with α. Since the number of solutions of $I[B_t]$ and the size of each L_{t_i} is $\|I\|^{r+O(1)}$, the algorithm spends $\|I\|^{O(r)}$ time at each node, hence the total running time is polynomial for each fixed r. Using standard bookkeeping techniques, it is not difficult to extend the algorithm such that it actually returns a solution if exists. □

In the remainder of this section, we sketch further algorithmic applications of fractional hypertree decompositions. Details will appear in the full version of this paper, but we believe that for a reader familiar with the concepts and techniques our brief outline should be sufficient.

To prove that $\text{CSP}(\mathcal{H})$ for classes $\mathcal{H}$ of hypergraphs of bounded fractional hypertree width is in polynomial time, we probably need a polynomial algorithm that, for fixed r, computes an at least approximately optimal fractional hypertree decompositions for hypergraphs of fractional hypertree width at most r. However, if we are only interested in the *parameterized complexity* of our constraint satisfaction problems, we do not need a decomposition algorithm. A *parameterization* of a problem is a function κ (usually assumed to be polynomial-time computable) that maps instances of the problem to the positive integers. A parameterized problem is *fixed parameter tractable* if there is an algorithm that, given an instance I, solves it in time $f(\kappa(I)) \cdot p(\|I\|)$ for some computable function f and polynomial $p(X)$. This definition reflects the idea that if the parameter is small, then the dependence of the running time on the parameter can be disregarded.

We consider the following parameterization of $\text{CSP}(\mathcal{H})$:

p-$\text{CSP}(\mathcal{H})$	
Instance:	A CSP instance $I = (V, D, C)$ with $H_I \in \mathcal{H}$.
Parameter:	$\|V\|$, the number of variables of I.
Problem:	Decide if I has a solution.

Corollary 16. *For every class $\mathcal{H}$ of hypergraphs of bounded fractional hypertree width, p-$\text{CSP}(\mathcal{H})$ is fixed parameter tractable.*

Fixed-parameter tractability is only relevant in situations where the parameter can be expected to be small, and it is very dubious to expect the number of variables in a constraint satisfaction problem to be small. However, there is an equivalent problem where this assumption is realistic, and that is the evaluation of conjunctive (database) queries. Database queries can usually be assumed to be small, certainly if compared to the size of the database, and the number of variables of the corresponding constraint satisfaction is bounded by the size of the query. The size of the domain, on the other hand, gets very large, because the domain consists of all database entries. However, when evaluating a database query, one is usually not interested in solving the decision problem ("Does there exist a tuple satisfying the query?"), but in the enumeration problem ("Compute all tuples satisfying the query."). As the number of tuples satisfying a query is not polynomially bounded in terms of the size of the database, it is reasonable to measure the running time of an enumeration algorithm in terms of the input size plus the output size. We refer to the corresponding notion of fixed-parameter tractability as *output fixed-parameter tractability*.

Theorem 17. *Let $\mathcal{H}$ be a class of hypergraphs of bounded fractional hypertree width. Then the evaluation problem for conjunctive queries whose underlying hypergraph is in $\mathcal{H}$ is output fixed-parameter tractable.*

This is proved very similarly to Theorem 15 and Corollary 16.

It is has been observed by Feder and Vardi [15] that constraint satisfaction problems can be described as homomorphism problems for relational structures. A *homomorphism* from a structure A to a structure B is a mapping from the domain of A to the domain of B that preserves membership in all relations. With each structure A we can associate a hypergraph H_A whose vertices are the elements of the domain of A and whose hyperedges are all sets $\{a_1, \ldots, a_k\}$ such that $(a_1, \ldots, a_k)$ is a tuple in some relation of A. For every class $\mathcal{H}$ of hypergraphs, we consider the following problem:

HOM($\mathcal{H}$)	
Instance:	Structure A with $H_A \in \mathcal{H}$ and structure B.
Problem:	Decide if there is a homomorphism from A to B.

We may also consider the parameterized version *p*-HOM($\mathcal{H}$), where the parameter is the number of elements of A. Then HOM($\mathcal{H}$) and CSP($\mathcal{H}$) are essentially the same problem, and all results we obtained for (*p*-)CSP($\mathcal{H}$) also apply to (*p*-)HOM($\mathcal{H}$).

An *embedding* is a homomorphism that is one-to-one. Note that there is an embedding from a structure A to a structure B if and only if B has a substructure isomorphic to A. Analogously to HOM($\mathcal{H}$) and *p*-HOM($\mathcal{H}$) we define the embedding problems EMB($\mathcal{H}$) and *p*-EMB($\mathcal{H}$). Note that EMB($\mathcal{P}$) is NP-complete even for the class $\mathcal{P}$ of all paths, because the Hamiltonian path problem is a special case. However, for the parameterized version we obtain the following result:

Theorem 18. *Let $\mathcal{H}$ be a class of hypergraphs of bounded fractional hypertree width. Then p-EMB($\mathcal{H}$) is fixed parameter tractable.*

The proof combines our previous results with Alon, Yuster, and Zwick's [3] color coding technique.

5. Conclusions

In this paper we have considered structural properties that can make a constraint satisfaction problem polynomial-time solvable. Previously, bounded hypertree width was the most general such property. Answering an open question raised in [8, 10, 18, 22], we have identified a new class of polynomial-time solvable CSP instances: instances having bounded fractional edge cover number. This result suggests the definition of fractional hypertree width, which is always at most as large as the hypertree width (and in some cases much smaller). It turns out that CSP is polynomial-time solvable for instances having bounded fractional hypertree width, if the hypertree decomposition is given together with the instance. It remains an important open question whether there is a polynomial-time algorithm that determines (or approximates) the fractional hypertree width and constructs a corresponding decomposition. We have provided an approximate characterization of fractional hypertree width using the so-called robber and army game, and we have laid out how it might help to design an algorithm for approximating the fractional hypertree width.

Another open question is whether there are polynomial-time solvable families of CSP instances having unbounded fractional hypertree width. If the answer is no, then it might be possible to prove this using parameterized complexity similarly to [22, 23].

References

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

[2] I. Adler, G. Gottlob, and M. Grohe. Hypertree-width and related hypergraph invariants. In *Proceedings of the 3rd European Conference on Combinatorics, Graph Theory, and Applications*, 2005. To appear.

[3] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42:844–856, 1995.

[4] C. Berge. *Graphs and Hypergraphs*. North Holland, 1976.

[5] A. A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 649–658, 2002.

[6] A. A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science*, pages 321–330, 2003.

[7] A. A. Bulatov, A. A. Krokhin, and P. Jeavons. The complexity of maximal constraint languages. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 667–674, 2001.

[8] H. Chen and V. Dalmau. Beyond hypertree width: Decomposition methods without decompositions. In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming*, 2005. To appear.

[9] F. Chung, P. Frank, R. Graham, and J. Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Series A*, 43:23–37, 1986.

[10] 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.

[11] V. Dalmau, Ph. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In

P. Van Hentenryck, editor, *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, volume 2470 of *Lecture Notes in Computer Science*, pages 310–326. Springer-Verlag, 2002.

[12] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[13] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, 38:353–366, 1989.

[14] R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM*, 30(3):514–550, 1983.

[15] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1998.

[16] E.C Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.

[17] E. Friedgut and J. Kahn. On the number of copies of a hypergraph in another. *Israel Journal of Mathematics*, 105:251–256, 1998.

[18] G. Gottlob, M. Grohe, N. Musliu, M. Samer, and F. Scarcello. Hypertree decompositions: Structure, algorithms, and applications. In *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science*, 2005. To appear.

[19] G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 394–399. Morgan Kaufmann, 1999.

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

[21] 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.

[22] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 552–561, 2003.

[23] M. Grohe, T. Schwentick, and L. Segoufin. When is the evaluation of conjunctive queries tractable. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 657–666, 2001.

[24] P. Jeavons, D. A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.

[25] Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 205–213, 1998.

[26] D. Marx. The closest substring problem with small distances. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2005. To appear.

[27] 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.

[28] T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[29] 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.

[30] M. Yannakakis. Algorithms for acyclic database schemes. In *7th International Conference on Very Large Data Bases*, pages 82–94, 1981.