Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 649–660
www.stacs-conf.org

TRACTABLE STRUCTURES FOR CONSTRAINT SATISFACTION WITH TRUTH TABLES

DÁNIEL MARX

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest H-1521, Hungary
E-mail address: dmarx@cs.bme.hu

Abstract. The way the graph structure of the constraints influences the complexity of constraint satisfaction problems (CSP) is well understood for bounded-arity constraints. The situation is less clear if there is no bound on the arities. In this case the answer depends also on how the constraints are represented in the input. We study this question for the truth table representation of constraints. We introduce a new hypergraph measure *adaptive width* and show that CSP with truth tables is polynomial-time solvable if restricted to a class of hypergraphs with bounded adaptive width. Conversely, assuming a conjecture on the complexity of binary CSP, there is no other polynomial-time solvable case.

1. Introduction

Constraint satisfaction is a general framework that includes many standard algorithmic problems such as satisfiability, graph coloring, database queries, etc. A constraint satisfaction problem (CSP) consists of a set V of variables, a domain D, and a set C of constraints, where each constraint is a relation on a subset of the variables. The task is to assign a value from D to each variable such that every constraint is satisfied (see Definition 1.4 for the formal definition). For example, 3SAT can be interpreted as a CSP problem where the domain is $D = \{0, 1\}$ and the constraints in C correspond to the clauses.

In general, solving constraint satisfaction problems is NP-hard if there are no additional restrictions on the instances. The main goal of the research on CSP is to identify tractable special cases of the general problem. The theoretical literature on the CSP investigates two main types of restrictions. The first type is to restrict the *constraint language,* that is, the type of constraints that are allowed. The second type is to restrict the *structure* induced by the constraints on the variables. The *hypergraph* of a CSP instance is defined to be a hypergraph on the variables of the instance such that for each constraint $c \in C$ there is a hyperedge E_c that contains all the variables that appear in c. If the hypergraph of the CSP instance has very simple structure, then the instance is easy to solve. For example, it is

Key words and phrases: computational complexity, constraint satisfaction, treewidth, adaptive width.

Research supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (Grant Number OTKA 67651).

STACS 2009
Symposium on Theoretical Aspects of Computer Science 2009 (Freiburg), pp. 649-660
http://drops.dagstuhl.de/opus/volltexte/2009/1807

well-known that a CSP instance I with hypergraph H can be solved in time $\|I\|^{O(\mathrm{tw}(H))}$ [5], where $\mathrm{tw}(H)$ denotes the treewidth of H and $\|I\|$ is the size of the representation of I in the input. Thus if we restrict the problem to instances where the treewidth of the hypergraph is bounded by some constant w, then the problem is polynomial-time solvable. The aim of this paper is to investigate whether there exists some other structural property of the hypergraph besides bounded treewidth that makes the problem tractable. Formally, for a class $\mathcal{H}$ of hypergraphs, let $\mathrm{CSP}(\mathcal{H})$ be the restriction of CSP where the hypergraph of the instance is assumed to be in $\mathcal{H}$. Our goal is to characterize the complexity of $\mathrm{CSP}(\mathcal{H})$ for every class $\mathcal{H}$.

We investigate two notions of tractability. $\mathrm{CSP}(\mathcal{H})$ is *polynomial-time solvable* if every instance of $\mathrm{CSP}(\mathcal{H})$ can be solved in time $(\|I\|)^{O(1)}$, where $\|I\|$ is the length of the representation of I in the input. The following notion interprets tractability in a less restrictive way: $\mathrm{CSP}(\mathcal{H})$ is *fixed-parameter tractable (FPT)* if there is a function f such that every instance I of $\mathrm{CSP}(\mathcal{H})$ can be solved in time $f(H)(\|I\|)^{O(1)}$, where H is the hypergraph of the instance. Equivalently, the factor $f(H)$ in the definition can be replaced with a factor $f(k)$ depending only on the number k of vertices of H: as the number of hypergraphs on k vertices is bounded by a function of k, the two definitions result in the same notion. The motivation behind the definition of fixed-parameter tractability is that in certain applications we expect the domain size to be much larger than the number of variables, hence a constant factor in the running time depending only on the number of variables (or on the hypergraph) is acceptable. For more background on fixed-parameter tractability, see [3, 4].

Bounded arities. If the constraints have bounded arity (i.e., edge size in $\mathcal{H}$ is bounded by a constant), then the complexity of $\mathrm{CSP}(\mathcal{H})$ is well understood:

Theorem 1.1 ([7]). *If $\mathcal{H}$ is a recursively enumerable class of hypergraphs with bounded edge size, then (assuming* FPT $\neq$ W[1]*) the following are equivalent:*

(1) $\mathrm{CSP}(\mathcal{H})$ *is polynomial-time solvable.*
(2) $\mathrm{CSP}(\mathcal{H})$ *is fixed-parameter tractable.*
(3) $\mathcal{H}$ *has bounded treewidth.*

The assumption FPT $\neq$ W[1] is a standard hypothesis of parameterized complexity. Thus in the bounded arity case bounded treewidth is the only property of the hypergraph that can make the problem polynomial-time solvable. Furthermore, the following sharpening of Theorem 1.1 shows that there is no algorithm whose running time is significantly better than the $\|I\|^{O(\mathrm{tw}(H))}$ bound of the treewidth based algorithm. The result is proved under the Exponential Time Hypothesis (ETH) [9]: it is assumed that there is no $2^{o(n)}$ time algorithm for n-variable 3SAT.

Theorem 1.2 ([11]). *If there is a computable function f and a recursively enumerable class $\mathcal{H}$ of hypergraphs with bounded edge size and unbounded treewidth such that the problem $\mathrm{CSP}(\mathcal{H})$ can be solved in time $f(H)\|I\|^{o(\mathrm{tw}(H)/\log \mathrm{tw}(H))}$ for instances I with hypergraph $H \in \mathcal{H}$, then ETH fails.*

This means that the treewidth-based algorithm is almost optimal: in the exponent only an $O(\log \mathrm{tw}(H))$ factor improvement is possible. It is conjectured in [11] that Theorem 1.2 can be made tight:

Conjecture 1.3 ([11]). *If $\mathcal{H}$ is a class of hypergraphs with bounded edge size, then there is no algorithm that solves $\mathrm{CSP}(\mathcal{H})$ in time $f(H)\|I\|^{o(\mathrm{tw}(H))}$ for instances I with hypergraph $H \in \mathcal{H}$, where f is an arbitrary computable function.*

Unbounded arities. The situation is less understood in the unbounded arity case, i.e., when there is no bound on the maximum edge size in $\mathcal{H}$. First, the complexity in the unbounded-arity case depends on how the constraints are represented. In the bounded-arity case, if each constraint contains at most r variables (r being a fixed constant), then every reasonable representation of a constraint has size $|D|^{O(r)}$. Therefore, the size of the different representations can differ only by a polynomial factor. On the other hand, if there is no bound on the arity, then there can be exponential difference between the size of succinct representations (e.g., formulas) and verbose representations (e.g., truth tables). The running time is expressed as a function of the input size, hence the complexity of the problem can depend on how the input is represented: longer representation means that it can be easier to obtain a polynomial-time algorithm.

The most well-studied representation of constraints is listing all the tuples that satisfy the constraint. For this representation, there are classes $\mathcal{H}$ with unbounded treewidth such that CSP restricted to this class is polynomial-time solvable. For example, classes with bounded *(generalized) hypertree width* [6], bounded *fractional edge cover number* [8], and bounded *fractional hypertree width* [8, 10] are such classes. However, no classification theorem similar to Theorem 1.1 is known for this version. More succinct representations were studied by Chen and Grohe [2]: constraints are represented by generalized DNF formulas and ordered binary decision diagrams (OBDD). The complexity of the problem with this representation was fully characterized: the complexity depends not on the treewidth of the hypergraph, but on the treewidth of the incidence structure.

Truth table representation. In this paper we study another natural representation: truth tables. A constraint of arity r is represented by having one bit for each possible r-tuple that can appear on the r variables of the constraint, and this bit determines whether this particular r-tuple satisfies the constraint or not. To increase the flexibility of the representation and make it more natural, we allow that the variables have different domains, i.e., each variable v has to be assigned a value from its domain $\mathrm{Dom}(v)$. Thus the size of the truth table of an r-ary constraint is proportional to the size of the direct product of the domains of the r variables. This representation is more verbose than listing satisfying tuples: the size of the representation is proportional to the number of possible tuples even if only few tuples satisfy the constraint. We define CSP as follows:

Definition 1.4. A *CSP instance* is a quadruple (V, D, Dom, C), where:

- V is a set of variables,
- D is a domain of values,
- $\mathrm{Dom} : V \to 2^D$ assigns a domain $\mathrm{Dom}(v) \subseteq D$ to each variable $v \in V$,
- C is a set of constraints. Each $c_i \in C$ is a pair $\langle s_i, R_i \rangle$, where:
  - $s_i = (u_{i,1}, \ldots, u_{i,m_i})$ is a tuple of variables (the *constraint scope*), and
  - R_i is a subset of $\prod_{j=1}^{m_i} \mathrm{Dom}(u_{i,j})$ (the *constraint relation*).

For each constraint $\langle s_i, R_i \rangle$ the tuples of R_i indicate the allowed combinations of values for the variables in s_i. The length m_i of the tuple s_i is called the *arity* of the constraint. A *solution* to a CSP instance is a function $f : V \to D$ such that $f(v) \in \mathrm{Dom}(v)$ for every $v \in V$ and for each constraint $\langle s_i, R_i \rangle$ with $s_i = \langle u_{i,1}, u_{i,2}, \ldots, u_{i,m_i} \rangle$, the tuple $\langle f(u_{i,1}), f(u_{i,2}), \ldots, f(u_{i,m_i}) \rangle$ is in R_i.

We denote by $\mathrm{CSP}_{\mathrm{tt}}$ the problem where each constraint $\langle s_i, R_i \rangle$ of arity m_i is represented by the truth table of the constraint relation R_i, that is, by a sequence of $\prod_{j=1}^{m_i} |\mathrm{Dom}(u_{i,j})|$

bits that describe that subset R_i of $\prod_{j=1}^{m_i} \mathrm{Dom}(u_{i,j})$. For a class $\mathcal{H}$, $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ is the restriction to instances with hypergraph in $\mathcal{H}$.

Results. The main result of the paper is a complete characterization of the complexity of $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ (assuming Conjecture 1.3). The complexity of the problem depends on a new hypergraph measure *adaptive width:*

Theorem 1.5 (Main)**.** *Assuming Conjecture 1.3, the following are equivalent:*

(1) $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ *is polynomial-time solvable.*
(2) $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ *is fixed-parameter tractable.*
(3) $\mathcal{H}$ *has bounded adaptive width.*

The assumption in Theorem 1.5 is nonstandard, so it is up to the reader to decide how strong this evidence is. However, the message of Theorem 1.5 is the following: a new tractable class for $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ would imply surprising new results for *binary* CSP. Thus it is not worth putting too much effort in further studying $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ with the hope of finding new tractable classes: as this would disprove Conjecture 1.3, such an effort would be better spent trying to disprove Conjecture 1.3 directly, by beating the $\|I\|^{O(\mathrm{tw}(H))}$ algorithm for binary CSP.

Listing the satisfying tuples is a more succinct representation of a constraint than a truth table. Thus if CSP is polynomial-time solvable or fixed-parameter tractable for some class $\mathcal{H}$ with the former representation, then this also holds for the latter representation as well. In particular, this means that by the results of [8, 10], $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ is polynomial-time solvable if $\mathcal{H}$ has bounded fractional hypertree width. This raises the question whether Theorem 1.5 gives any new tractable class $\mathcal{H}$. In other words, is there a class $\mathcal{H}$ having bounded adaptive width but unbounded fractional hypertree width? In Section 5, we answer this question by constructing such a class $\mathcal{H}$. This means that $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ is polynomial-time solvable, but if the constraints are represented by listing the satisfying tuples, then it is not even known whether the problem is FPT.

2. Width parameters

Treewidth and various variants are defined in this section. We follow the framework of width functions introduced by Adler [1]. A *tree decomposition* of a hypergraph H is a tuple $(T, (B_t)_{t\in V(T)})$, where T is a tree and $(B_t)_{t\in V(T)}$ is 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. Let $f : 2^{V(H)} \to \mathbb{R}^+$ be a function that assigns a real number to each subset of vertices. The *f-width* of a tree-decomposition $(T, (B_t)_{t\in V(T)})$ is $\max\{f(B_t) \mid t \in V(T)\}$. The *$f$-width* of a hypergraph H is the minimum of the f-widths of all its tree decompositions.

Definition 2.1. Let $s(B) = |B| - 1$. The *treewidth* of H is $\mathrm{tw}(H) := s\text{-width}(H)$.

A subset $E' \subseteq E(H)$ is an *edge cover* if $\bigcup E' = V(H)$. The *edge cover number* $\rho(H)$ is the size of the smallest edge cover (assuming H has no isolated vertices). For $X \subseteq V(H)$, let $\rho_H(X)$ be the size of the smallest set of edges covering X.

Definition 2.2. The *(generalized) hypertree width* of H is $\mathrm{hw}(H) := \rho_H\text{-width}(H)$.

We also consider the linear relaxations of edge covers: a function $\gamma : E(H) \to [0, 1]$ is a *fractional edge cover* of H if $\sum_{E:v\in E} \gamma(E) \geq 1$ for every $v \in V(H)$. The *fractional* *cover number* $\rho^*(H)$ of H is the minimum of $\sum_{E\in E(H)} \gamma(E)$ taken over all fractional edge covers of H. We define $\rho^*_H(X)$ analogously to $\rho_H(X)$: the requirement $\sum_{E:v\in E} \gamma(E) \geq 1$ is restricted to vertices of X.

Definition 2.3. The *fractional hypertree width* of H is $\mathrm{fhw}(H) := \rho^*_H\text{-width}(H)$.

The dual of covering is independence. A subset $X \subseteq V(H)$ is an *independent set* if $|X \cap E| \leq 1$ for every $E \in E(H)$. The *independence number* $\alpha(H)$ is the size of the largest independent set and $\alpha_H(X)$ is the size of the largest independent set that is a subset of X. A function $\phi : V(H) \to [0,1]$ is a *fractional independent set* of the hypergraph H if $\sum_{v\in E} \phi(v) \leq 1$ for every $E \in E(H)$. The *fractional independence number* $\alpha^*(H)$ of H is the maximum of $\sum_{v\in V(H)} \phi(v)$ taken over all fractional independent sets ϕ of H. It is well-known that $\alpha(H) \leq \alpha^*(H) = \rho^*(H) \leq \rho(H)$ for every hypergraph H. Thus α^*-width gives us the same notion as fractional hypertree width. The main new definition of the paper uses fractional independent sets, but in a different way. For a function $f : V(H) \to \mathbb{R}^+$, we define $f(X) = \sum_{v\in X} f(v)$ for $X \subseteq V(H)$ and define f-width accordingly.

Definition 2.4. The *adaptive width* $\mathrm{adw}(H)$ of a hypergraph H is the maximum of ϕ-width(H) taken over all fractional independent sets ϕ of H.

Currently, we do not have an efficient algorithm for computing adaptive width. Fortunately, the polynomial-time algorithm in Section 3 for instances with bounded adaptive width does not need to determine the adaptive width of the input, it is sufficient that the adaptive width is promised to be bounded. However, the hardness proof of Section 4 requires that the question $\mathrm{adw}(H) \geq w$ is decidable (proof is omitted).

Lemma 2.5. *There is an algorithm that, given hypergraph H and rational number w, decides if* $\mathrm{adw}(H) \geq w$. *If the answer is yes, then the algorithm returns a rational fractional independent set α such that the α-width of H is at least w.*

We finish the section with a combinatorial observation (the closed neighborhood of a vertex v is the union of all the edges containing v):

Lemma 2.6. *Given a tree decomposition of hypergraph H, it can be transformed in polynomial time (by removing vertices from some bags) into a tree decomposition of H satisfying the following property: if two adjacent vertices u and v have the same closed neighborhood, then u and v appear in exactly the same bags.*

Proof. Consider a tree decomposition of H. If u and v are two vertices that do not satisfy the requirements, then remove these vertices from those bags where only one of them appears (since u and v are neighbors, they appear together in some bag B_t, hence both vertices appear in at least one bag after the removals). The intersection of two subtrees is also a subtree, thus it remains true that u and v appear in a connected subset of the bags. We have to show that for every edge $E \in E(H)$, there is a bag B_t that fully contains E even after the removals. If $\{u, v\} \subseteq E$ or $E \cap \{u, v\} = \emptyset$, then this clearly follows from that fact that some bag fully contains E before the removals. Assume without loss of generality that $u \in E$ and $v \notin E$. We show that $E \cup \{v\}$ is fully contained in some bag B_t before the removals, hence (as $\{u, v\} \subseteq E \cup \{v\}$) B_t fully contains $E \cup \{v\}$ even after the removals. Since $u \in E$, edge E is in the closed neighborhood of v. Thus by assumption, E is also in the closed neighborhood of v, which means that $E \cup \{v\}$ is a clique in H. It is well known that every clique is fully contained in some bag of the tree decomposition (this follows from the fact that subtrees of a tree satisfy the Helly property), thus it follows that $E \cup \{v\} \subseteq B_t$ for some bag B_t.

Let us repeat these removals until there are no pairs u, v that violate the requirements; eventually we get a tree decomposition as required. Observe that the procedure terminates after a polynomial number of steps: vertices are only removed from the bags. ∎

3. Algorithm for bounded adaptive width

We prove that $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ is polynomial-time solvable if $\mathcal{H}$ has bounded adaptive width. Bounded adaptive width ensures that no matter what the distribution of the domain sizes in the input instance is, there is a decomposition where the variables in each bag have only a polynomial number of possible assignments. For such a decomposition, the instance can be solved by standard techniques.

Lemma 3.1. *There is an algorithm that, given an instance I of $\mathrm{CSP}_{\mathrm{tt}}$, an integer C, and a tree decomposition $(T, (B_t)_{t\in V(T)})$ of the hypergraph H of the instance such that $\prod_{v\in B_t} |\mathrm{Dom}(v)| \le C$ for every bag B_t, solves the instance I in time polynomial in $\|I\| \cdot C$.*

Proof. If $\prod_{v\in B_t} |\mathrm{Dom}(v)| \le C$, then there are at most C possible assignments on the variables in B_t. Using standard dynamic programming techniques, it is easy to check whether it is possible to select one assignment f_t for each bag B_t such that f_t satisfies the instance induced by the bag B_t and these assignments are compatible. For completeness, we briefly describe how this can be done by a reduction to binary CSP.

Let us construct a binary CSP instance I' as follows. The set of variables of I' is $V(T)$, i.e., the bags of the tree decomposition. For $t \in V(T)$, let $b_t \le C$ be the number of assignments f to the variables in B_t such that $f(v) \in \mathrm{Dom}(v)$ for every $v \in B_t$; denote by $f_{t,i}$ the i-th such assignment on B_t ($1 \le i \le b_t$). The domain of I' is $D' = \{1, \dots, C\}$. For each edge $t't'' \in E(T)$, we introduce a constraint $c_{t',t''} = \langle (t', t''), R_{t',t''}\rangle$, where $(i, j) \in R_{t',t''}$ if and only if

- $f_{t',i}$ and $f_{t'',j}$ are compatible, i.e., $f_{t',i}(v) = f_{t'',j}(v)$ for every $v \in B_{t'} \cap B_{t''}$.
- $f_{t',i}$ satisfies every constraints of I whose scope is contained in $B_{t'}$.
- $f_{t'',j}$ satisfies every constraints of I whose scope is contained in $B_{t''}$.

It is easy to see that a solution of I' determines a solution of I. The size of I' is polynomial in C and $\|I\|$. Since the graph of I' is a tree, it can be solved in time $\|I'\|^{O(1)} = (\|I\|C)^{O(1)}$. ∎

Theorem 3.2. *If $\mathcal{H}$ has bounded adaptive width, then $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H}) \in$ PTIME.*

Proof. Let I be an instance of $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ with hypergraph H such that $\mathrm{adw}(H) \le c$. Let $N \le \|I\|$ be the size of the largest truth table in the input; we assume that $N > 1$, since the problem is trivial if $N = 1$. We show that it is possible to find in time $N^{O(c)}$ a tree decomposition $(T, B_{t\in V(T)})$ of the instance such that $\prod_{v\in B_t} |\mathrm{Dom}(v)| \le N^{O(c)}$ holds for every bag B_t. By Lemma 3.1, this means that the instance can be solved in time polynomial in $\|I\|$ and $N^{O(c)}$, i.e., the running time is $\|I\|^{O(c)}$.

Let $\phi(v) = \log_2 |\mathrm{Dom}(v)| / \log_2 N$. We claim that ϕ is a fractional independent set of H. If there is a constraint with $(v_{i_1}, v_{i_2}, \dots, v_{i_r})$ such that $\sum_{j=1}^{r} \phi(v_j) > 1$, then the size of the truth table describing the constraint is larger than N:

$$\prod_{j=1}^{r} |\mathrm{Dom}(i_j)| = \prod_{j=1}^{r} 2^{\phi(v_{i_j})\cdot \log_2 N} = 2^{\log_2 N \cdot \sum_{j=1}^{r} \phi(v_{i_j})} > 2^{\log_2 N} = N.$$

Define $\phi'(v) = \lceil \phi(v) \log_2 N \rceil$. Observe that $\phi(v) \geq 1/\log_2 N$, hence $\phi'(v) < 2\phi(v)\log_2 N$ (if $|\operatorname{Dom}(v)| = 1$, then the instance can be simplified). Let H' be the hypergraph that is obtained from H by replacing each vertex v with a set X_v of $\phi'(v)$ vertices; if an edge E contains some vertex v in H, then E contains every vertex of X_v in H'. We claim that H' has treewidth less than $2c\log_2 N$. Since $\operatorname{adw}(H) \leq c$, H has a tree decomposition $(T, B_{t\in V(T)})$ such that $\sum_{v\in B_t} \phi(v) \leq c$ holds for every bag B_t. Consider the analogous decomposition $(T, B'_{t\in V(T)})$ of H'; i.e., if a bag B_t contains a vertex v of H, then let bag B'_t contain every vertex of X_v. The size of a bag B'_t is $\sum_{v\in B_t} |X_v| = \sum_{v\in B_t} \phi'(v) \leq 2\log_2 N \cdot \sum_{v\in B_t} \phi(v) \leq 2c\log_2 N$, thus the treewidth of H' is indeed less than $2c\log_2 N$. Given a graph G with n vertices, it is possible to find a tree decomposition of width at most $4\operatorname{tw}(G)+1$ in time $2^{O(\operatorname{tw}(G))}n^{O(1)}$ (see e.g., [4, Prop. 11.14]). Thus we can a find a tree decomposition $(T, B''_{t\in V(T)})$ of width at most $8c\log_2 N$ for H' in time $2^{O(2c\log_2 N)}||H'||^{O(1)} = N^{O(c)}||H'||^{O(1)}$.

In H', every vertex of X_v is contained in the same set of edges. By Lemma 2.6, it can be assumed that each bag of $(T, B''_{t\in V(T)})$ contains either all or none of X_v. Define the tree decomposition $(T, B^*_{t\in V(T)})$ of H where bag B^*_t contains v if and only X_v is contained in B''_t. The ϕ-weight of a bag B^*_t can be bounded as

$$\sum_{v\in B^*_t} \phi(v) \leq \frac{1}{\log_2 N} \sum_{v\in B^*_t} \phi'(v) = \frac{1}{\log_2 N}|B''_t| \leq 8c.$$

Thus in the tree decomposition $(T, B^*_{t\in V(T)})$, the product of the domain sizes is

$$\prod_{v\in B^*_t} |\operatorname{Dom}(v)| = \prod_{v\in B^*_t} 2^{\phi(v)\cdot\log_2 N} = 2^{\log_2 N\cdot\sum_{v\in B^*_t}\phi(v)} \leq 2^{\log_2 N\cdot 8c} = N^{8c},$$

in each bag B^*_t, as required. ∎

4. Hardness result for unbounded adaptive width

We prove the main complexity result of the paper in this section.

Theorem 4.1. *Let $\mathcal{H}$ be a recursively enumerable class of hypergraphs with unbounded adaptive width. Assuming Conjecture 1.3,* $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ *is not FPT.*

Proof. Suppose that $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ can be solved in time $h_1(H)\|I\|^c$ for some constant c and computable function h_1. Let us fix an arbitrary computable enumeration of the hypergraphs in $\mathcal{H}$. For every $k \geq 1$, let H_k be the first hypergraph in this enumeration with $\operatorname{adw}(H_k) \geq k$. For each $k \geq 1$, let ϕ_k be the fractional independent set returned by the algorithm of Lemma 2.5 for the question '$\operatorname{adw}(H_k) \geq k$?'.

Constructing the graph class $\mathcal{G}$. For each $k \geq 1$, we construct a graph G_k based on H_k and ϕ_k. Let q_k be the least common denominator of the rational values $\phi_k(v)$ for $v \in V(H_k)$. The graph G_k has a clique K_v of size $q_k \cdot \phi(v)$ for each $v \in V(H_k)$ and if u and v are neighbors in H_k, then every vertex of K_u is connected to every vertex of K_v. Let $\mathcal{G} = \{G_k \mid k \geq 1\}$.

We claim that $\operatorname{tw}(G_k) \geq q_k k - 1$. Suppose for contradiction that G_k has a tree decomposition $(T, (B_t)_{t\in V(T)})$ of width less than $q_k k - 1$, i.e., the size of every bag is smaller than $q_k k$. By Lemma 2.6, it can be assumed that for every $v \in V(H_k)$ and bag B_t of the decomposition, either B_t fully contains K_v or disjoint from it. Let us construct a tree decomposition $(T, (B'_t)_{t \in V(T)})$ of H_k such that B'_t contains v if and only if B_t fully contains K_v. It is easy to see that this is a tree decomposition of H_k: for every $E \in E(H_k)$, the set $\bigcup_{v \in E} K_v$ is a clique in G_k, hence there is a bag B_t containing $\bigcup_{v \in E} K_v$, i.e, B'_t contains E. Furthermore, $\phi_k(B'_t) < k$ for every bag B'_t: if $\phi_k(B'_t) \geq k$, then $|\bigcup_{v \in E} K_v| \geq q_k k$, contradicting the assumption that every bag B_t has size strictly less than $q_k k$. This would contradict the assumption $\mathrm{adw}(H_k) \geq k$, thus $\mathrm{tw}(G_k) \geq q_k k - 1$.

Simulating G_k by H_k. We present an algorithm for $\mathrm{CSP}(\mathcal{G})$ violating Conjecture 1.3. We show how a binary $\mathrm{CSP}(\mathcal{G})$ instance I_1 with graph G_k can be reduced to a $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ instance I_2 with hypergraph $H_k \in \mathcal{H}$. Then I_2 can be solved with the assumed algorithm for $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$. Let $G \in \mathcal{G}$ be the graph of the CSP instance I_1. By enumerating the hypergraphs in $\mathcal{H}$, we can find the first value k such that $G = G_k$. We construct a $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ instance I_2 with hypergraph H_k where every variable $v \in V(H_k)$ simulates the variables in K_v.

The domain $\mathrm{Dom}(v)$ of v is $D^{|K_v|}$, i.e., $\mathrm{Dom}(v)$ is the set of $|K_v|$-tuples of D. For every $v \in V(H_k)$, there is a natural bijection between the elements of $\mathrm{Dom}(v)$ and the $|D|^{|K_v|}$ possible assignments $f : K_v \to D$. For each edge $E = (v_1, \dots, v_r) \in E(H_k)$, we add a constraint $c_E = \langle (v_1, \dots, v_r), R_E \rangle$ to I_2 as follows. Let $(x_1, \dots, x_r) \in \prod_{i=1}^r \mathrm{Dom}(v_i)$. For $1 \leq i \leq r$, let g_i be the assignment of K_{v_i} corresponding to $x_i \in \mathrm{Dom}(v_i)$. These r assignments together define an assignment $g : \bigcup_{i=1}^r K_{v_i} \to D$ on the union of their domains. We define the relation R_E such that $(x_1, \dots, x_r)$ is a member of R_E if and only if the corresponding assignment g satisfies every constraint of I_1 whose scope is contained in $\bigcup_{i=1}^r K_{v_i}$.

Assume that I_1 has a solution $f_1 : V(G_k) \to D$. For every $v \in V(H_k)$, define $f_2(v)$ to be the member of $\mathrm{Dom}(v)$ corresponding to the assignment f_1 restricted to K_v. Now f_2 is a solution of I_2: for every edge E of H_k, assignment f_1 restricted to $\bigcup_{v \in E} K_v$ clearly satisfies every constraint of I_1 whose scope is in $\bigcup_{v \in E} K_v$.

Assume now that I_2 has a solution $f_2 : V_2 \to D_2$. For every $v \in V(H_k)$, there is an assignment $f_v : K_v \to D$ corresponding to the value $f_2(v)$. These assignments together define an assignment $f_1 : V(G_k) \to D$. We claim that f_1 is a solution of I_1. Let $c = \langle (u', v'), R \rangle$ be an arbitrary constraint of I_1. Assume that $u' \in K_u$ and $v' \in K_v$ for some $u, v \in V(H_k)$. Since $u'v' \in E(G_k)$, there is an edge $E \in E(H_k)$ with $u, v \in E$. The definition of c_E in I_2 ensures that f_1 restricted to $K_u \cup K_v$ satisfies every constraint of I_1 whose scope is contained in $K_u \cup K_v$; in particular, f_1 satisfies constraint c.

Running time. Assume that an instance I_1 of $\mathrm{CSP}(\mathcal{G})$ is solved by first reducing it to an instance I_2 as above and then applying the algorithm for $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$. Let us determine the running time of this algorithm. The first step of the algorithm is to enumerate the hypergraphs in $\mathcal{H}$ until the correct value of k is found. The time required by this step depends only on the graph $G \in \mathcal{G}$; denote it by $h_2(G)$. Let us determine the time required to construct instance I_2 and the size of the representation of I_2. As defined above, for each constraint c_E in I_2, we have to enumerate every tuple $(x_1, \dots, x_r) \in \prod_{i=1}^r \mathrm{Dom}(v)$ and check whether the corresponding assignment g is a solution of the instance $I_1[\bigcup_{i=1}^r K_{v_i}]$. Checking a vector $(x_1, \dots, x_r)$ can be done in time polynomial in $\|I_1\|$. Moreover,

$$\left| \prod_{i=1}^r \mathrm{Dom}(v) \right| = \prod_{i=1}^r |D|^{|K_v|} = \prod_{i=1}^r |D|^{q_k \cdot \phi_k(v)} = |D|^{q_k \sum_{i=1}^r \phi_k(v)} \leq |D|^{q_k},$$

since ϕ_k is a fractional independent set and $\{x_1, \dots, x_r\}$ is an edge of H_k. Every other step is polynomial in $\|I_1\|$, hence the reduction can be done in time $h_2(G)\|I_1\|^{O(q_k)}$, which is also a bound on $\|I_2\|$. Thus the algorithm for $\mathrm{CSP}_{\mathrm{tt}}(\mathcal{H})$ requires $h_1(H_k)(h_2(G)\|I_1\|)^{O(q_k c)}$ time, yielding a total time of $h_3(G)\|I_1\|^{O(q_k c)}$ for some computable function h_3.

We show that $\|I_1\|^{O(q_k c)}$ is $\|I_1\|^{o(\mathrm{tw}(G_k))}$, violating Conjecture 1.3. Let $s(w)$ be the smallest k such that $\mathrm{tw}(G_k)$ is greater than w (as $\mathrm{tw}(G_k) \geq q_k k - 1$, this is well defined). Observe that $s(w)$ is nondecreasing and unbounded. We have

$$\|I_1\|^{O(q_k c)} \leq \|I_1\|^{O(c(\mathrm{tw}(G_k)+1)/k)} \leq \|I_1\|^{O(c(\mathrm{tw}(G)+1)/s(\mathrm{tw}(G)))} = \|I_1\|^{o(\mathrm{tw}(G))}.$$

Thus the total running time is $h_3(G)\|I_1\|^{o(\mathrm{tw}(G))}$, violating Conjecture 1.3. ∎

5. Relation of bounded fractional hypertree width and bounded adaptive width

We show that the class of sets of hypergraphs with bounded adaptive width strictly includes the class of sets with bounded fractional hypertree width. First, fractional hypertree width is an upper bound for adaptive width.

Proposition 5.1. *For every hypergraph H,* $\mathrm{adw}(H) \leq \mathrm{fhw}(H)$.

Proof. Let $(T, B_{t\in V(T)})$ be a tree decomposition of H whose ρ^*_H-width is $\mathrm{fhw}(H)$. If ϕ is a fractional independent set, then $\phi(B_t) \leq \rho^*_H(B_t) \leq \mathrm{fhw}(H)$ for every bag B_t of the decomposition, i.e., ϕ-width$(H) \leq \mathrm{fhw}(H)$. This is true for every fractional independent set ϕ, hence $\mathrm{adw}(H) \leq \mathrm{fhw}(H)$. ∎

This implies that if a set of hypergraphs has bounded fractional hypertree width, then it has bounded adaptive width as well. The converse is not true: the main result of this section is a set of hypergraphs with bounded adaptive width (Corollary 5.11) that has unbounded fractional hypertree width (Corollary 5.8).

Definition 5.2. The hypergraph $H(d, c)$ has $2^{d+1} - 1$ vertices $v_{i,j}$ $(0 \leq i \leq d,\ 0 \leq j < 2^i)$ and the following edges:

- For every $0 \leq k < 2^d$, there is a *large edge* E_k of size $d+1$ that contains $v_{i,\lfloor k/2^{d-i}\rfloor}$ for every $0 \leq i \leq d$.
- For every i, j_1, j_2 with $|j_1 - j_2| \leq c$, there is a *small edge* $\{v_{i,j_1}, v_{i,j_2}\}$.

We say that vertex $v_{i,j}$ is on *level* i. We define $\chi(v_{i,j}) = j2^{d-i}$. The set $\mathcal{H}_c$ contains every hypergraph $H(d, c)$ for $d \geq 1$.

Definition 5.3. If $v_{i,j}$ and $v_{i',j'}$ are covered by the same large edge E_k and $i \leq i'$, then $v_{i,j}$ is an *ancestor* of $v_{i',j'}$; and $v_{i',j'}$ is a *descendant* of $v_{i,j}$.

Proposition 5.4. *If $v_{i,j}$ is an ancestor of $v_{i',j'}$, then $\chi(v_{i,j}) \leq \chi(v_{i',j'}) < \chi(v_{i,j}) + 2^{d-i}$.*

Proof. The ancestor of $v_{i',j'}$ on level i is $v_{i,\lfloor j'/2^{i'-i}\rfloor}$. Therefore,

$$\chi(v_{i,j}) = \lfloor j'/2^{i'-i}\rfloor \cdot 2^{d-i} \leq j'/2^{d-i'} = \chi(v_{i',j'})$$

and

$$\chi(v_{i,j}) = \lfloor j'/2^{i'-i}\rfloor \cdot 2^{d-i} > (j'/2^{i'-i} - 1) \cdot 2^{d-i} = j' \cdot 2^{d-i'} - 2^{d-i} = \chi(v_{i',j'}) - 2^{d-i}.$$

∎

5.1. Lower bound on fractional hypertree width

Fractional hypertree width has various other characterizations that are equivalent up to a constant factor [8]. Here we use the characterization by balanced separators to prove a lower bound on the fractional hypertree width of $H(d,c)$.

For a function $\gamma : E(H) \to \mathbb{R}^+$, we define $\mathrm{weight}(\gamma) := \sum_{E \in E(H)} \gamma(E)$. For a set $W \subseteq V(H)$, we let $\mathrm{weight}(\gamma|W) = \sum_{e \in E_W} \gamma(e)$, where E_W is the set of all edges intersecting W. For $\lambda > 0$, a set $S \subseteq V(H)$ is a *λ-balanced separator for γ* if $\mathrm{weight}(\gamma|C) \le \lambda \cdot \mathrm{weight}(\gamma)$ for every component C of $H \setminus S$.

Theorem 5.5 ([8]). *Let H be a hypergraph and $\gamma : E(H) \to \mathbb{R}^+$. There is a $\frac{1}{2}$-balanced separator S for γ such that $\rho^*_H(S) \le \mathrm{fhw}(H)$.*

Theorem 5.5 can be generalized to λ-separators with arbitrary $\lambda > 0$ (proof is omitted):

Corollary 5.6. *Let H be a hypergraph and $\gamma : E(H) \to \mathbb{R}^+$. For every $\lambda > 0$, there is a λ-balanced separator S for γ such that $\rho^*_H(S) \le 2\,\mathrm{fhw}(H)/\lambda$.*

Proposition 5.7. *For every $c \ge 5$ and $d > 2\log_2 c$, $\mathrm{fhw}(H(d,c)) \ge \sqrt{d}/(2c)$.*

Proof. Let γ be a weight function on the edges that assigns 1 to each large edge and 0 to the small edges. We show that every $\frac{1}{2c}$-balanced separator of $H(d,c)$ for γ has fractional cover number at least $\sqrt{d}/2$. By Corollary 5.6, $\mathrm{fhw}(H(d,c)) \ge \sqrt{d}/(8c)$ follows.

Suppose that S is a $\frac{1}{2c}$-balanced separator of $H(d,c)$ for γ. Observe that on level $d/2$, there are at least c vertices: $2^{d/2} \ge c$. We claim that there is a $d/2 \le i \le d$ for which there is no $0 \le a_i \le 2^i - c$ such that $v_{i,j} \in S$ for every $a_i \le j < a_i + c$. Suppose that there is such an a_i for every $d/2 \le i \le d$. Let $b_i = a_i + c - 1$. It follows from the definition of a_i that $v_{i,a_i}, v_{i,b_i} \in S$ for every $d/2 \le i \le d$. We claim that the set $X = \{v_{i,a_i}, v_{i,b_i} : d/2 \le i \le d\}$ contains an independent set of size at least $\sqrt{d}/2$, contradicting the assumption that the fractional cover number $\rho^*(S)$ is less than $\sqrt{d}/2$ (recall that $\alpha_H(S) \le \rho^*_H(S)$ holds). First we show that if a large edge E_k covers v_{i,a_i} and $v_{i',a_{i'}}$ then v_{i,b_i} and $v_{i',b_{i'}}$ are independent. Assume without loss of generality that $i < i'$. By Prop. 5.4, $|\chi(v_{i,a_i}) - \chi(v_{i',a_{i'}})| < 2^{d-i}$. Since $\chi(v_{i,b_i}) = \chi(v_{i,a_i}) + (c-1)2^{d-i}$ and $\chi(v_{i',b_{i'}}) = \chi(v_{i',a_{i'}}) + (c-1)2^{d-i'}$,

$$\begin{aligned}|\chi(v_{i,b_i}) - \chi(v_{i',b_{i'}})| &> (c-1)2^{d-i} - (c-1)2^{d-i'} - 2^{d-i} \\ &\ge (c-1)2^{d-i} - \frac{c-1}{2} \cdot 2^{d-i} - 2^{d-i} = (c/2 - 3/2)2^{d-i} \ge 2^{d-i},\end{aligned}$$

if $c \ge 5$. Therefore, v_{i,b_i} and $v_{i',b_{i'}}$ are independent (Prop. 5.4). Similarly, if a large edge E_k covers both $v_{b_j,j}$ and $v_{b_{j'},j'}$ then $v_{a_j,j}$ and $v_{a_{j'},j'}$ are independent.

If X can be covered with weight less than $\sqrt{d}/2$, then there is an edge that covers at least $|X|/(\sqrt{d}/2) = 2\sqrt{d}$ vertices of X. Denote by $Y \subseteq X$ this set of vertices, and let $Y_a = \{v_{i,a_i} \in Y : d/2 \le i \le d\}$ and $Y_b = \{v_{i,b_i} \in Y : d/2 \le i \le d\}$. Now either $|Y_a| \ge \sqrt{d}$ or $|Y_b| \ge \sqrt{d}$. For each vertex v_{i,a_i}, we call the vertex v_{i,b_i} the *pair* of v_{i,a_i} and vice versa. If $|Y_a| \ge \sqrt{d}$, then we have seen that the pairs of the vertices in Y_a form an independent set of size $|Y_a|$, thus X cannot be covered with weight less than $\sqrt{d}$. Similarly, if $|Y_b| \ge \sqrt{d}$, then the pairs of the vertices in Y_b give an independent set of size $\sqrt{d}$. This contradicts the assumption that S can be covered with weight strictly less than $\sqrt{d}/2$.

Thus there is a $d/2 \le i \le d$ such that for every $0 \le j \le 2^i - c$, at least one of $v_{i,j}, \ldots, v_{i,j+c-1}$ is not in S. It is not difficult to see that the set C_i of vertices on level i not in S is connected and intersects more than $1/(2c)$ fraction of the large edges. Thus $\text{weight}(\gamma|C) > \text{weight}(\gamma)/2c$ for the component C of $H(d,c)\setminus S$ containing C_i, contradicting the assumption that S is a $\frac{1}{2c}$-balanced separator for γ. ∎

Corollary 5.8. *$\mathcal{H}_c$ has unbounded fractional hypertree width for every $c \ge 5$.*

5.2. Upper bound on adaptive width

We use the following lemma to give an upper bound for f-width (proof is omitted):

Lemma 5.9. *Let H be a hypergraph, $0 < \lambda < 1, w > 0$ constants, and $f : 2^{V(H)} \to \mathbb{R}^+$ a function such that $f(X) \le f(Y)$ for every $X \subseteq Y$ and $f(X \cup Y) \le f(X) + f(Y)$ for arbitrary X, Y. Assume that for every subset $W \subseteq V(H)$ there is a subset $S \subseteq V(H)$ with $f(S) \le w$ such that every component C of $H \setminus S$ has $f(C \cap W) \le \lambda f(W)$. Then the f-width of H is at most $2w/(1-\lambda) + w$.*

To obtain the upper bound on adaptive width, we have to show that the required separator S exists for every fractional independent set. We say that a set S is *closed* if the set S contains every ancestor of every vertex of S. For future use, we show that even a closed separator exists for $H(d,c)$.

Lemma 5.10. *Let ϕ be a fractional independent set of $H(d,c)$ and let W be a subset of vertices. Then there is a closed set S with $\phi(S) \le 4c(c+1)+5$ such that for every component C of $H(d,c) \setminus S$ we have $\phi(C \cap W) \le 3\phi(W)/4$.*

Proof. Let $M(a,b)$ be the set of vertices $v_{i,j}$ with $a \le \chi(v_{i,j}) < b$. Let x and y be integers such that $\phi(M(x, x+y) \cap W) \ge \phi(W)/4$ and y is as small as possible. Let $d_0 = d - \lceil \log_2 y \rceil$; clearly, we have $y \le 2^{d-d_0} \le 2y$. Let $A(t) := \{v_{i,j} : \chi(v_{i,j}) \ge t \text{ and } i \ge d_0\}$. Denote by $S(t)$ the set of those vertices v that have a descendant $v_{i,j}$ with $\chi(v_{i,j}) < t$ such that $v_{i,j}$ has a neighbor in $A(t)$. We show that $\phi(S(t_1)) \le 2c(c+1)+1$ for some $x - y < t_1 \le y$.

Let $S_1(t)$ be those vertices of $S(t)$ that are on level less than d_0 and let $S_2(t)$ be those vertices that are on level at least d_0. First we bound $\phi(S_1(t))$. Observe that every $v \in S_1(t)$ has a descendant $v_{i,j}$ with $\chi(v_{i,j}) \ge t - c2^{d-d_0}$: if descendant $v_{i,j}$ has a neighbor $u \in A(t)$, then either $v_{i,j}$ and u are connected by a large edge (in this case u is also a descendant of v) or $v_{i,j}$ and u are connected by a small edge (in this case $\chi(v_{i,j}) \ge t - c2^{d-i} \ge t - c2^{d-d_0}$). Let X be the set of vertices $v_{d_0,j}$ with $t - c2^{d_0} \le \chi(v_{d_0,j}) < t$, we have $|X| \le c$. By the observation above, every $v \in S_1(t)$ has a descendant in X. The vertices in X and the ancestors of X can be covered by $|X| \le c$ large edges. Thus $\phi(S_1(t)) \le c$, as $S_1(t)$ can be covered with at most c large edges and $\phi(E_k) \le 1$ for every large edge E_k.

We show that $\phi(S_2(t))$ is small on average. We claim that

$$\sum_{t=x-y+1}^{x} \phi(S_2(t)) \le c \sum_{t=\max\{0,(x-y-c2^{d-d_0})\}}^{\min\{2^d, x+2^{d-d_0}-1\}} \phi(E_t) \le c(c+1)2^{d-d_0} + y \le 2c(c+1)y + y.$$

holds, implying that $\phi(S_2(t)) \le 2c(c+1)+1$ for at least one t. To see the first inequality, observe that $v_{i,j}$ with $i \ge d_0$ is in $S_2(t)$ only if $t - c2^{d-i} \le \chi(v_{i,j}) < t$. Thus such a vertex contributes to the first sum for at most $c2^{d-i}$ values of t. However, if $v_{i,j}$ contributes at all to the first sum, then it contributes to the second sum for exactly 2^{d-i} values of t, as every large edge containing $v_{i,j}$ is counted. Thus $v_{i,j}$ contributes $\phi(v_{i,j})$ at most c times more to the first sum than to the second, which is taken care by the factor c before the second sum.

Similarly, we can show that there is a value $x+y \le t_2 < x+2y$ such that $\phi(S_2(t_2)) \le 2c(c+1)+1$. Denote by $T(t_1,t_2)$ the vertices of $M(t_1,t_2)$ on level less than d_0. We claim that $\phi(T(t_1,t_2)) \le 3$. First, $T(t_1,t_2)$ can contain at most 3 vertices on each level: if $v_{i,j}, v_{i,j'} \in T(t_1,t_2)$ and $j' \ge j+3$, then $|\chi(v_{i,j})-\chi(v_{i,j'})| \ge 3\cdot 2^{d-i} > 3\cdot 2^{d-d_0} \ge 3y \ge t_2-t_1$, contradicting the assumption on the χ-values. Every $v_{i,j} \in T(t_1,t_2)$ has a descendant $v_{i',j'} \in T(t_1,t_2)$ for every $i < i' < d_0$, namely $v_{i',j'}$ with $j' = j2^{i'-i}$. Thus by covering the at most 3 vertices of $T(t_1,t_2)$ on level d_0-1 by at most 3 large edges, we can cover $T(t_1,t_2)$, and $\phi(T(t_1,t_2)) \le 3$ follows.

Define $S := S(t_1)\cup S(t_2)\cup T(t_1,t_2)$. Clearly, $\phi(S) \le 2(2c(c+1)+1)+3 = 4c(c+1)+5$. We show that S separates $M(t_1,t_2)$ from the rest of the vertices. Suppose that $v_{i,j}, v_{i',j'} \notin S$ are adjacent vertices such that $v_{i,j} \in M(t_1,t_2)$ and $v_{i',j'} \notin M(t_1,t_2)$. We have $i \ge d_0$ (otherwise $v_{i,j} \in T(t_1,t_2)$), hence $v_{i,j} \in A(t_1)$. If $\chi(v_{i',j'}) < t_1$, then $v_{i',j'} \in S(t_1) \subseteq S$, a contradiction. Moreover, if $\chi(v_{i',j'}) > t_2$, then $i' \ge d_0$ as $v_{i,j}$ and $v_{i',j'}$ are not neighbors if $\chi(v_{i,j}) < \chi(v_{i',j'})$ and $i > i'$. Thus $v_{i',j'} \in A(t_2)$ and $v_{i,j} \in S(t_2) \subseteq S$, a contradiction.

By the definition of x and y, we have $\phi(M(t_1,t_2)\cap W) \ge \phi(M(x,x+y)\cap W) \ge \phi(W)/4$. To complete the proof that $\phi(W\cap C) \le 3\phi(W)/4$ for every component C of $H(d,c)\setminus S$, we show that $\phi(M(t_1,t_2)\cap W) \le 3\phi(W)/4$: as we have seen that every such component C is fully contained in either $M(t_1,t_2)$ or $V\setminus M(t_1,t_2)$, this means that no component C can have $\phi(W\cap C) > 3\phi(W)/4$. Since $x-t_1 < y$, the minimality of y implies $\phi(M(t_1,x)\cap W) \le \phi(W)/4$. Similarly, it follows from $t_2-(x+y) < y$ that $\phi(M(x+y,t_2)\cap W) \le \phi(W)/4$. Now $\phi(M(t_1,t_2)\cap W) = \phi(M(t_1,x)\cap W)+\phi(M(x,x+y)\cap W)+\phi(M(x+y,t_2)\cap W) \le \frac{3}{4}\phi(W)$. ∎

By Lemma 5.10, the requirements of Lemma 5.9 hold for $H(d,c)$ with $w := 4c(c+1)+5$ and $\lambda := 3/4$, hence $\mathrm{adw}(H(d,c)) \le 9w = 36c(c+1)+45$.

Corollary 5.11. *The class $\mathcal{H}_c$ has bounded adaptive width for every fixed $c \ge 1$.*

References

[1] I. Adler. *Width functions for hypertree decompositions.* PhD thesis, 2006.

[2] H. Chen and M. Grohe. Constraint satisfaction problems with succinctly specified relations, 2006.

[3] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[4] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, Berlin, 2006.

[5] E. C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proc. of AAAI-90*, 4–9, Boston, MA, 1990.

[6] G. Gottlob, F. Scarcello, and M. Sideri. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artificial Intelligence*, 138(1-2):55–86, 2002.

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

[8] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *SODA '06*, 289–298, 2006.

[9] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.

[10] D. Marx. Approximating fractional hypertree width. In *SODA '09,* 902–911, 2009.

[11] D. Marx. Can you beat treewidth? In *FOCS '07*, 169–179, 2007.