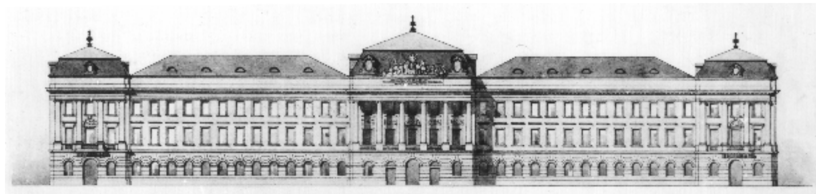
INFSYS
RESEARCH
REPORT



INSTITUT FÜR INFORMATIONSSYSTEME
ABTEILUNG WISSENSBASIERTE SYSTEME

STRUCTURAL PARAMETERIZATIONS OF LANGUAGE RESTRICTED CONSTRAINT SATISFACTION PROBLEMS

Simone Bova Stefan Szeider

INFSYS RESEARCH REPORT 1843-13-02
APRIL 2013

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT

INFSYS RESEARCH REPORT 1843-13-02, APRIL 2013

STRUCTURAL PARAMETERIZATIONS OF LANGUAGE RESTRICTED CONSTRAINT SATISFACTION PROBLEMS

Simone Bova Stefan Szeider

Abstract. We study the fixed-parameter tractability of the constraint satisfaction problem. We restrict constraint relations to languages in a family of NP-hard languages, classified by a purely combinatorial criterion that generalizes Boolean matrices with fixed row and column sum. For various natural and established structural parameterizations of the instances, we characterize the fixed-parameter tractable constraint languages in the family.

Acknowledgements: This work was supported by the ERC Starting Grant 239962 (COMPLEX REASON).

Publication Information: This work was presented at the Dagstuhl Seminar 12451 on "The Constraint Satisfaction Problem: Complexity and Approximability", November 4-9, 2012.

1 Introduction

The goal of a *constraint satisfaction problem* (CSP) is to assign values to variables subject to *constraints* on the values of certain groups of variables. In this general formulation, the CSP uniformly captures many algorithmic problems in various areas of computer science; in the parlance of database theory, the CSP is equivalent to model checking of *conjunctive queries* on *relational databases* [11]. As a consequence of its expressive power, the CSP is computationally hard, and a central research problem in computer science is to find *tractable restrictions* of the CSP.

In the framework of *polynomial-time tractability*, two main restriction patterns have been studied: *structural restrictions* and *language restrictions*. Roughly speaking, these patterns restrict the general problem to classes of instances where the constraints induce some nice mathematical structure on the variables and on the values. Very general tractable classes have been identified for each type of restriction singly taken [10, 2]. In particular, restrictions acting purely on the language side appear severe; for instance, the constraint language $\{(1,0,0),(0,1,0),(0,0,1)\}$, containing the size three identity matrix only, is already NP-hard by Schaefer's dichotomy [17].

With the purpose of contrasting this limitation, we study the CSP in the framework of parameterized complexity, looking for language restrictions that warrant *fixed-parameter tractability*, a relaxation of the notion of polynomial-time tractability [7]. Namely, we do not restrict instances to certain classes, but instead we attach to each instance a natural number k, the *parameter*, and the goal is to attain (if possible) fixed-parameter tractability, that is, a decision algorithm whose running time is bounded above by $f(k) \cdot n^{O(1)}$, where f is a computable function and n is the size of the input (the performance is interesting if k is assumed much smaller than n).

More precisely, in this paper we start exploring the interplay between *structural parameterizations* and *language restrictions* of the CSP, a novel but natural restriction pattern. Intuitively, a parameter can be viewed as a *structural parameter* if it is bounded above by the size of the "structural" part of the instance. In the aforementioned database interpretation of the CSP, the structural part of the instance is the query, as opposed to the "language" part of the instance formed by the database; note that the former is typically much smaller than the latter.

We parameterize instances by natural structural parameters (see Section 2 for the definitions): *query size*, *number of variables*, *number of constraints*, *treewidth of primal graph*, *treewidth of dual graph*, and *treewidth of incidence graph*. Observed that already the easiest parameter considered, query size, is in general not fixed-parameter tractable [15], we investigate the following question:

> *Which language restrictions of the* CSP *are fixed-parameter tractable under natural structural parameterizations?*

As the question clearly trivializes for polynomial-time tractable languages, we focus on a family of *hard* constraint languages. As usual, a constraint language is just a set of finite relations, which we conveniently display here as matrices; without loss of generality, we take matrices over nonnegative integers, and 0 as a distinguished value.

Each language in the family is attached a pair $(r, c) \in (\mathbb{N} \cup \{*\})^2$, defined as follows:

- for all relations in the language, r is an upper bound on the number of entries distinct from 0 in the rows ($r = *$, if such a bound does not exist);

- for all relations in the language and all columns in the relation, c is an upper bound on the number of rows having an entry distinct from 0 in the column ($c = *$, if such a bound does not exist).

For instance, the language containing all Boolean relations whose underlying matrix is an identity matrix would be classified by the pair $(1, 1)$. By ordering in the natural way the set of all pairs $(r, c) \in (\mathbb{N} \cup \{*\})^2$, we obtain a hierarchy of increasingly expressive, computationally hard constraint languages.

For every structural parameterization κ considered, we explicitly define the subset $S_\kappa \subseteq (\mathbb{N} \cup \{*\})^2$ such that:

> *The* CSP *with relations restricted to the language defined by the pair* $(r, c) \in (\mathbb{N} \cup \{*\})^2$*, is fixed-parameter tractable under the parameterization* κ *if and only if* $(r, c) \in S_\kappa$.

The backward direction requires FPT $\neq$ W[1], a standard hypothesis in parameterized complexity theory (see Theorem 1 in Section 3).

Related Work. We review previous work on restrictions of the CSP that is related and preparatory to the restriction pattern proposed in this paper.

Purely structural or language restrictions have been intensively investigated. Currently, the most general polynomial-time tractable structural restriction of the CSP is formed by classes of instances whose hypergraph has bounded fractional width, established by Marx and Grohe [10], and the classes of instances whose hypergraph has unbounded fractional width and bounded submodular width is the current obstacle towards the exact characterization of polynomial-time tractable structural restrictions [14]. In the special case of bounded arity, Grohe proved that the classes of instances whose hypergraph has bounded treewidth coincide with polynomial-time tractable classes [9]. As regards pure language restrictions of the CSP, it is conjectured that polynomial-time tractable classes are exactly those classes defined on constraint languages that admit a nontrivial polymorphism. Sufficiency is a difficult open problem, although very large classes of polynomial-time tractable constraint languages are known [5, 2]. Motivated by the limitations of the previous unilateral approaches, hybrid restrictions combine structural and language restrictions to attain further polynomial-time tractable classes, although in this area the theoretical framework and the complexity landscape are not as established and definite as in the area of pure restrictions [3].

In the setting of fixed-parameter tractability, Samer and Szeider introduced a restriction pattern for the CSP which combines structural parameters and language parameters (with emphasis on the former type of parameters), and obtained a classification of all fixed-parameter tractable combinations of the considered parameters [16].

As mentioned, our restriction pattern demands a criterion to analyze the realm of hard constraint languages, where the universal algebraic toolkit developed in the area of language restrictions does not help (the criterion adopted, namely for a language to admit certain kinds of polymorphisms, is too coarse because all hard languages admit the same kinds of polymorphisms). Different and novel criteria, purely combinatorial as opposed to universal algebraic, are necessary. Our criterion generalizes Boolean relations whose underlying matrix has row sum equal to r and column sum equal to c, an established combinatorial object equivalent to semiregular bipartite graphs [1, 8]. In a similar veine, Marx [13] and Marx and Krokhin [12] use novel combinatorial criteria (called weak separability and flip separability) to refine the classification of constraint languages offered by the purely universal algebraic criteria, and characterize fixed-parameter tractable cases of a certain optimization problems related to the Boolean CSP.

2 Preliminaries

Treewidth. A *(simple) graph* is a pair (V, E), where V is the set of *vertices* and E is the set of *edges*; an edge is a two-element subset of V. In the following, all graphs are finite (that is, the set V is finite). A k*-clique* is a subset $K \subseteq V$ such that $|K| = k$ and $\{a, b\} \in E$ for every $\{a, b\} \subseteq K$. A *tree* is a connected acyclic graph (T, E); the elements in T are called *nodes*.

A *tree decomposition* of a graph $G = (V, E)$ is a pair $((T, E), \{H_t\}_{t \in T})$, where (T, E) is a tree, $H_t \subseteq V$ for all $t \in T$ (called the *bag* at t), and the following holds: for every vertex $a \in V$, the set $\{t \in T \mid a \in H_t\}$ is nonempty and connected in (T, E); for every edge $\{a, b\} \in E$, there exists a node $t \in T$ such that $\{a, b\} \subseteq H_t$. The *width* of the tree decomposition $((T, E), \{H_t\}_{t \in T})$ of G is $\max\{|H_t| \mid t \in T\} - 1$, and the *treewidth* of G, in symbols $\mathrm{tw}(G)$, is the minimum of the widths of all tree decompositions of G.

Parameterized Complexity. A *decision problem* is a set $Q \subseteq \Sigma^*$ of strings over Σ, where Σ is a finite nonempty alphabet; in the following, without loss of generality, $\Sigma \subseteq \mathbb{N}$. A string $x \in \Sigma^*$ is called an *instance* of the problem, and the *question* is whether $x \in Q$.

A *parameterization* of Σ^* is a polynomial-time computable mapping $\kappa \colon \Sigma^* \to \mathbb{N}$ that maps each string x to a *parameter* $\kappa(x)$. A *parameterized problem* (over Σ) is a pair (Q, κ) such that Q is a decision problem and κ is a parameterization of Σ^*.

A mapping g defined on Σ^* is *fixed-parameter tractable* (in short, an *fpt*-mapping) with respect to a parameterization κ of Σ^* if for some computable function $f \colon \mathbb{N} \to \mathbb{N}$, and some polynomial $p \colon \mathbb{N} \to \mathbb{N}$, there exists an algorithm that for every $x \in \Sigma^*$ computes $g(x)$ in time bounded above by $f(\kappa(x))p(|x|)$. A parameterized problem (over Σ) is *fixed-parameter tractable* (in short, an *fpt*-problem) if the characteristic function of Q is a fpt-mapping with respect to κ; FPT denotes the class of all fpt-problems.

Let (Q, κ) and (Q', κ') be parameterized problems. A *fixed-parameter tractable* reduction (or, an *fpt*-reduction) from (Q, κ) to (Q', κ') is a fpt-mapping g (with respect to κ), such that, for all $x \in \Sigma^*$, the following holds: (i) $x \in Q$ if and only if $g(x) \in Q'$; and, (ii) there is a computable function $f \colon \mathbb{N} \to \mathbb{N}$ such that $\kappa'(x) \leq f(\kappa(x))$. We write $(Q, \kappa) \leq^{\mathrm{fpt}} (Q', \kappa')$ if there exists a fpt-reduction from (Q, κ) and (Q', κ').

The class FPT is closed under fpt-reductions, that is, if $(Q, \kappa) \leq^{\mathrm{fpt}} (Q', \kappa')$ and $(Q', \kappa') \in$ FPT, then $(Q, \kappa) \in$ FPT. Let C be a class of parameterized problems closed under fpt-reductions. As in classical complexity theory, we say that a parameterized problem (Q, κ) is C*-hard under fpt-reductions* if every problem in C fpt-reduces to (Q, κ) and C*-complete under fpt-reductions* if, in addition, $(Q, \kappa) \in$ C.

A fundamental class of parameterized problems closed under fpt-reductions is W[1], which plays in parameterized complexity a role analogous to NP in classical complexity theory: If $(Q, \kappa) \leq^{\text{fpt}} (Q', \kappa')$ and (Q, κ) is W[1]-complete, then (Q', κ') is not in FPT unless W[1] $\subseteq$ FPT, which is conjectured false. Therefore, to establish that a parameterized problem (Q, κ) is unlikely in FPT, it is sufficient to give a fpt-reduction from a W[1]-complete problem to (Q, κ).

Constraint Satisfaction. In the following, A and m always denote a nonempty set and a positive integer, respectively. A *relation* R is a is a finite subset of the Cartesian power A^m, where A and m are respectively the *universe* and *arity* of of R.

A *constraint* is a pair $(R, (v_1, \ldots, v_m))$, also written $Rv_1 \ldots v_m$ in short, where R is an arity m relation, called the *(constraint) relation*, and $(v_1, \ldots, v_m) \in V^m$ is a tuple of m pairwise distinct *variables*, called the *(constraint) scope*. We liberally also think of the scope $(v_1, \ldots, v_m)$ as the set $\{v_1, \ldots, v_m\}$.

For algorithmic purposes, we think of a constraint $Rv_1 \ldots v_m$ as a $\{v_1, \ldots, v_m\}$-relation in the sense of database theory. An X*-relation* is a finite set of total maps with domain X; if R is an X-relation, we also write $R(X)$. The *size* of a suitable encoding of R is defined by $\|R\| = O(|X| \cdot |R|)$. [1] For every $Y \subseteq X$, the *projection* of the X-relation R onto Y, is the Y-relation defined by $\pi_Y R = \{f|_Y \mid f \in R\}$, which is computable in time $O(\|R\|)$. Also, for every X-relation R and every Y-relation S, the *join* of R and S, is the $X \cup Y$-relation defined by

$$R \bowtie S = \{f \colon X \cup Y \to A \mid f|_X \in R, f|_Y \in S\},$$

which is computable in time $O(\|R\| + \|S\| + \|R \bowtie S\|)$; compare [6, Appendix].

The *constraint satisfaction problem*, in symbols CSP, is defined as follows. The instance is (a suitable encoding of) a triple $I = (V, D, C)$, where V is a finite set of variables, D is a finite nonempty set (the *domain* of the instance), and C is a finite set of constraints with relations over D. The question is whether the instance is *satisfiable*, that is, whether there exists a mapping $f \colon V \to D$ such that $f|_X \in R(X)$ for all constraints $R(X) \in C$. In the following, n always denotes the *size* of the encoding of a CSP instance I, and $n = O(\sum_{R(X) \in C} \|R\|)$.

A *(constraint) language* is a set Γ containing relations over a universe A; in the following, $\Gamma^{01} = \{R \in \Gamma \mid R \text{ has universe } \{0, 1\}\}$ denotes the *Boolean* sublanguage of Γ. The problem CSP(Γ) is the problem of deciding, given a CSP instance $I = (V, D, C)$ with constraint relations in Γ, whether the instance is satisfiable.

Let $I = (V, D, C)$ be a CSP instance. The *primal graph* of I, in symbols $\text{prim}(I)$, is the graph whose vertices are the variables of the instance, and whose edges connect two variables if and only if they are contained together in a constraint scope. The *dual graph* of I, in symbols $\text{dual}(I)$, is the graph whose vertices are the constraints of the instance, and whose edges connect two constraints if and only if their scopes have nonempty intersection. The *incidence graph* of I, in symbols $\text{inc}(I)$, is the graph whose vertices are the variables and the constraints of the instance, and whose edges connect two vertices if and only if the two vertices are a variable and a constraint, and the variable is contained in the scope of the constraint.

[1] We adopt the standard random access machine model as computational model, and the uniform cost measure to analyze the running time of the algorithms.

Let $I = (V, D, C)$ be a CSP instance. We introduce the following measures on I:

$$\begin{aligned}
\mathrm{query}(I) &= \sum_{R(X)\in C} |X|;\\
\mathrm{var}(I) &= |V|;\\
\mathrm{constr}(I) &= |C|;\\
\mathrm{twprim}(I) &= \mathrm{tw}(\mathrm{prim}(I));\\
\mathrm{twdual}(I) &= \mathrm{tw}(\mathrm{dual}(I));\\
\mathrm{twinc}(I) &= \mathrm{tw}(\mathrm{inc}(I)).
\end{aligned}$$

The following inequalities are known [16]:

$$\mathrm{twinc}(I) \leq \mathrm{twprim}(I) + 1 \leq \mathrm{var}(I) \leq \mathrm{query}(I); \tag{1}$$

$$\mathrm{twinc}(I) \leq \mathrm{twdual}(I) + 1 \leq \mathrm{constr}(I) \leq \mathrm{query}(I). \tag{2}$$

We let,

$$\mathrm{PAR} = \{\mathrm{query}, \mathrm{var}, \mathrm{constr}, \mathrm{twprim}, \mathrm{twdual}, \mathrm{twinc}\}.$$

In view of (1) and (2), we define a partial order on PAR by letting, for all $p \in \mathrm{PAR}$: $\mathrm{twinc} \leq p$, $\mathrm{twprim} \leq \mathrm{var}$, $\mathrm{twdual} \leq \mathrm{constr}$, and $p \leq \mathrm{query}$.

3 Complexity Classification

For all $a \in A$ and $(a_1, \ldots, a_m) \in A^m$, the *row sum* of $(a_1, \ldots, a_m)$ with respect to a is the number of entries in $(a_1, \ldots, a_m)$ that are not equal to a; formally,

$$\mathrm{rowsum}(a, (a_1, \ldots, a_m)) = |\{i \in [m] \mid a_i \neq a\}|.$$

The row sum operation naturally extends to a relation $R \subseteq A^m$ by maximizing the row sum over all tuples in R,

$$\mathrm{rowsum}(a, R) = \max\{\mathrm{rowsum}(a, (a_1, \ldots, a_m)) \mid (a_1, \ldots, a_m) \in R\}.$$

We analogously define the *column sum* of a relation R with respect to a. For all $i \in [m]$, the *column sum* of R with respect to i and a is the number of tuples in R whose i-th entry is not equal to a; formally,

$$\mathrm{colsum}(a, i, R) = |\{(a_1, \ldots, a_m) \in R \mid a_i \neq a\}|.$$

The column sum extends to R by maximizing the column sums over all $i \in [m]$,

$$\mathrm{colsum}(a, R) = \max\{\mathrm{colsum}(a, i, R) \mid i \in [m]\}.$$

If $r, c \in \{0\} \cup \mathbb{N}$ are such that $\mathrm{rowsum}(a, R) \leq r$ and $\mathrm{colsum}(a, R) \leq c$, we say that R has row sum bounded by r and column sum bounded by c with respect to a.

Definition 1 (Constraint Language Γ_{rc})**.** *Let* $A = \{0\} \cup \mathbb{N}$ *and* $a = 0$*. For all* $r, c \in \mathbb{N} \cup \{*\}$*, the language* Γ_{rc} *over* A *is defined as follows:* $\Gamma_{**} = \{R \mid 1 \leq \mathrm{rowsum}(a, R), \mathrm{colsum}(a, R)\}$*;* $\Gamma_{r*} = \{R \mid \mathrm{rowsum}(a, R) \leq r\} \subseteq \Gamma_{**}$*;* $\Gamma_{*c} = \{R \mid \mathrm{colsum}(a, R) \leq c\} \subseteq \Gamma_{**}$*;* $\Gamma_{rc} = \Gamma_{r*} \cap \Gamma_{*c}$*.*

For all $p \in \mathrm{PAR}$, we define parameterized versions of the problem CSP(Γ):

Problem: CSP(p, Γ).

Instance: A pair (I, k), where I is an instance of CSP(Γ) and $p(I) \leq k \in \mathbb{N}$.

Parameter: k.

Question: Is I satisfiable?

Remark 1. Without loss of generality, the problem is formulated as a promise problem, whose instances (I, k) correspond to instances I of CSP(Γ) where $p(I) \leq k$. Indeed, by Bodlaender's algorithm [7, Theorem 11.12], the corresponding decision problem, which also involves checking whether the instance has the promised form, is equivalent to the promise version under fpt-reductions.

Remark 2. The choice $A = \{0\} \cup \mathbb{N}$ and $a = 0$ in Definition 1 is arbitrary, but without loss of generality. Namely, let Γ be any constraint language, let B be the universe of Γ, and let $b \in B$. If all relations in Γ have row sum bounded by some positive integer with respect to b (or column sum bounded by some positive integer with respect to b, or both, or none), then, for every $p \in \mathrm{PAR}$, the problem CSP(p, Γ) is polynomial-time equivalent to a problem of the form CSP(p, Γ_{rc}) for some $(r, c) \in (\mathbb{N} \cup \{*\})^2$.

Moreover, in Definition 1, we restrict without loss of generality to relations R satisfying $1 \leq \mathrm{rowsum}(a, R)$ and $1 \leq \mathrm{colsum}(a, R)$. Note that if $\mathrm{rowsum}(a, R) = 0$ or $\mathrm{colsum}(a, R) = 0$, then $R = \{(a, a, \ldots, a)\}$, and the language is therefore polynomial-time tractable.

Proposition 1. *For all* $p' \leq p$ *in* PAR *and all constraint languages* Γ*,*

$$\mathrm{CSP}(p, \Gamma) \leq^{\mathrm{fpt}} \mathrm{CSP}(p', \Gamma).$$

Proof. Immediate from (1) and (2). □

To state the main result in a compact fashion, we define a partial (in fact, lattice) order over the set $(\mathbb{N} \cup \{*\})^2$, by letting $(r, c) \leq (r', c')$ if and only if $i \leq i'$ and $j \leq j'$, where $1 < 2 < \cdots < *$. Note that $(2, *) \not\leq (r, c)$ if and only if $(r, c) \leq (1, *)$ or $(r, c) \leq (*, j)$ for all $j \in \mathbb{N}$.

Theorem 1 (Main Result)**.** *Let* $(r, c) \in (\mathbb{N} \cup \{*\})^2$*.*

1. *If* $p \in \{\mathrm{twinc}, \mathrm{twdual}\}$*, then* CSP$(p, \Gamma_{rc})$ *is in* FPT *if* $(r, c) \leq (1, *)$*, and* W[1]*-hard otherwise.*

2. *If* $p \in \{\mathrm{twprim}, \mathrm{var}, \mathrm{constr}, \mathrm{query}\}$*, then* CSP$(p, \Gamma_{rc})$ *is in* FPT *if* $(2, *) \not\leq (r, c)$*, and* W[1]*-hard otherwise.*

Proof. Exploiting the partial order over structural parameters and constraint languages, we reduce the entire classification to three positive results (in Section 4), in the form of fixed-parameter algorithms, and two negative results (in Section 5), in the form of W[1]-hardness proofs. We remark that Item 2 required us fairly different algorithmic techniques to settle the two tractable cases for $p = \mathrm{constr}$ (in Lemma 1 and Lemma 3, respectively). The details follow.

For tractability, Lemma 1 establishes fixed-parameter tractability of Γ_{rc} for all $(r, c) \leq (1, *)$ and all $p \in \mathrm{PAR}$, thus settling the FPT part of Item 1 (together with Proposition 1). Lemma 2 and Lemma 3 establish fixed-parameter tractability of Γ_{rc} for all $(r, c) \leq (*, j)$ with $j \in \mathbb{N}$, respectively if $p \geq \mathrm{twprim}$ and $p \geq \mathrm{constr}$, thus completing the FPT part of Item 2 (together with Proposition 1).

For intractability, Item 1 in Lemma 4 establishes W[1]-hardness of Γ_{rc} for all $(r, c) \geq (2, *)$ and all $p \in \mathrm{PAR}$, thus settling the hardness part of Item 2 (together with Proposition 1). Lemma 5 establishes W[1]-hardness of Γ_{rc} for all $(r, c) \geq (2, 1)$ and all $p \leq \mathrm{twdual}$, thus settling the hardness part of Item 1 (together with Proposition 1). □

Remark 3. Assuming FPT $\neq$ W[1] the conditions in the statement of Theorem 1 are also necessary for fixed-parameter tractability.

Since $\mathrm{CSP}(\mathrm{query}, \Gamma) \in$ W[1] for all languages Γ [15, Theorem 1], and it is folklore that $\mathrm{CSP}(\mathrm{constr}, \Gamma) \leq^{\mathrm{fpt}} \mathrm{CSP}(\mathrm{query}, \Gamma)$, the W[1]-hardness results for $p \in \{\mathrm{var}, \mathrm{constr}, \mathrm{query}\}$ in the statement of Theorem 1 are indeed W[1]-completeness results.

As an interesting special case, we derive the following complexity classification of Boolean languages.

Corollary 1 (Boolean Case)**.** *Let* $(r, c) \in (\mathbb{N} \cup \{*\})^2$.

1. *If* $p \in \{\mathrm{twinc}, \mathrm{twdual}, \mathrm{constr}\}$*, then* $\mathrm{CSP}(p, \Gamma^{01}_{rc})$ *is in* FPT *if* $(r, c) \leq (1, *)$*, and* W[1]*-hard otherwise.*

2. *If* $p \in \{\mathrm{twprim}, \mathrm{var}, \mathrm{query}\}$*, then* $\mathrm{CSP}(p, \Gamma^{01}_{rc}) \in$ FPT.

Proof. Item 1 follows along the lines of Theorem 1, with an appeal to Item 2 of Lemma 4 to settle W[1]-hardness of $\mathrm{CSP}(\mathrm{constr}, \Gamma^{01}_{rc})$ if $(r, c) \geq (2, *)$.

Item 2 follows because, if $p \in \{\mathrm{twprim}, \mathrm{var}, \mathrm{query}\}$, then $\mathrm{CSP}(p, \Gamma^{01}) \in$ FPT for any Boolean language Γ^{01} [16, Corollary 2]. □

4 Tractability

Lemma 1. *If* $(r, c) \leq (1, *)$*, then* $\mathrm{CSP}(\mathrm{twinc}, \Gamma_{rc}) \in$ FPT.

Proof. We prove that $\mathrm{CSP}(\mathrm{twinc}, \Gamma_{rc}) \leq^{\mathrm{fpt}} \mathrm{CSP}(\mathrm{twinc}, \Gamma^{01}_{rc})$.

Given an instance $I = (V, D, C)$ of $\mathrm{CSP}(\Gamma_{rc})$, we compute a CSP instance $I' = (V, \{0, 1\}, C')$, as follows. For all $R(X), S(Y) \in C$ and all $z \in X \cap Y$, enforce the condition

$$\{f(z) \mid f \in R\} = \{g(z) \mid g \in S\}; \tag{3}$$

intuitively, the condition warrants that all constraints on a certain variable agree on the "possible" values of that variable. This is polynomial-time computable by looping over all variables $z \in V$ and then looping over all pairs of constraints $R(X), S(Y) \in C$ such that $z \in X \cap Y$, redefining $R \coloneqq \pi_X(R \bowtie S)$ and $S \coloneqq \pi_Y(R \bowtie S)$ at each iteration, until condition (3) is established on all constraints sharing variable z; note that the inner loop iterates at most $|D| \cdot |C|$ times, because each iteration shrinks the domain of variable z in at least one constraint. Now let $C' = \{R'(X) \mid R(X) \in C\}$, where

$$R'(X) = \{f' \colon X \to \{0,1\} \mid f \in R(X)\} \tag{4}$$

and $f'(x) = 0$ if and only if $f(x) = 0$ for all $x \in X$; intuitively, we replace all "possible" values of each variable, distinct from 0, by the "representative" value 1.

Note that I' is an instance of CSP(Γ^{01}_{rc}).

Claim 1. I is satisfiable if and only if I' is satisfiable.

We prove Claim 1. Clearly if $f \colon V \to D$ satisfies I, then the map $f' \colon V \to \{0,1\}$ such that $f'(x) = 0$ if and only if $f(x) = 0$ satisfies I'. Conversely, assume that the map $f' \colon V \to \{0,1\}$ satisfies I'. Define the map $f \colon V \to D$ as follows, for all $x \in V$. If $f'(x) = 0$, then $f(x) = 0$. Otherwise, settle $f(x) = d \in D$ where d is an arbitrarily chosen element in $\{f(x) \mid f \in R(X)\}$, for an arbitrarily chosen constraint $R(X) \in C$ such that $x \in X$. Note that such a d exists by (4) since I' is satisfied with $f'(x) = 1$. We check that f satisfies I. Let $R(X) \in C$. Since $R'(X)$ has row sum bounded by $r = 1$, there are exactly two cases. If f' satisfies $R'(X)$ by $f'|_X = (x \mapsto 0) \in R'$ then $(x \mapsto 0) \in R$ by (4) and $f|_X = (x \mapsto 0)$ by construction. Otherwise, f' satisfies $R'(X)$ by $f'|_X \neq (x \mapsto 0) \in R'$, say $f'(x) = 1$ for $x \in X$ and $f'(y) = 0$ for $y \in X \setminus \{x\}$. In this case, (3) implies that the map $f|_X$ is in R.

Note that the mapping $I \mapsto I'$ is computable in polynomial time, and $\mathrm{twinc}(I) = \mathrm{twinc}(I')$, thus by the previous claim the mapping $(I, k) \mapsto (I', k)$ is a fpt-reduction of CSP$(\mathrm{twinc}, \Gamma_{rc})$ to CSP$(\mathrm{twinc}, \Gamma^{01}_{rc})$.

Claim 2. CSP$(\mathrm{twinc}, \Gamma^{01}_{rc}) \in$ FPT.

We prove Claim 2. We give a fpt-reduction to the problem of deciding, given a monadic second-order sentence φ and a graph G, whether the formula is true in the graph. The problem, parameterized by $\mathrm{tw}(G)$ and the size of φ, is known in FPT [4].

Given an instance I of CSP$(\mathrm{twinc}, \Gamma^{01}_{rc})$, we compute an instance (G, φ) of the above problem as follows. For all $R(X) \in C$ and $x \in X$, enforce the condition

$$\{f(x) \mid f \in R\} \neq \{0\}, \tag{5}$$

that is, $\{f(x) \mid f \in R\} = \{0,1\}$ if nonempty. This is polynomial-time computable by looping over all $R(X) \in C$ and $x \in X$ and redefining $R \coloneqq \pi_{X \setminus \{x\}} R$ if (5) does not hold.

We now complete the definition of (G, φ), as follows: The graph G is the incidence graph of I, with edge relation E, together with two subsets $C', C'' \subseteq C$ such that C' contains exactly the constraint vertices whose constraint relation contains the tuple $(0, 0, \dots, 0)$, and $C'' = C \setminus C'$. The monadic second-order formula is $\varphi = \exists X \forall x(\varphi' \wedge \varphi'')$ where

$$\varphi'(x) = C'x \to (\exists^{=1} y(y \in X \wedge Exy) \vee \forall z(Exz \to z \notin X)),$$
$$\varphi''(x) = C''x \to \exists^{=1} y(y \in X \wedge Exy),$$

and $\exists^{=1} x\psi(x)$ abbreviates $\exists x(\psi(x) \wedge \forall y(\psi(y) \to y = x))$. The map that sends I to (G, φ) is the required fpt-reduction, which settles the claim.

The lemma is settled. □

Lemma 2. *If* $(r, c) \leq (*, j)$*, then* $\text{CSP}(\text{twprim}, \Gamma_{rc}) \in \text{FPT}$.

Proof. Let Γ be any language. Consider the problem of deciding, given a a pair $(I, k_1 + k_2)$, where $I = (V, D, C)$ is an instance of CSP(Γ), $k_1 \geq \text{twprim}(I)$, and $k_2 \geq \max\{|R| \mid R \in C\}$, whether I is satisfiable. The problem, parameterized by $\kappa(I, k_1 + k_2) = k_1 + k_2$, is in FPT [16, Corollary 3]; we show that CSP$(\text{twprim}, \Gamma_{rc})$ fpt-reduces to it.

Let (I, k) be an instance of CSP$(\text{twprim}, \Gamma_{rc})$. If $r = \max\{|X| \mid R(X) \in C\}$ is the maximum arity attained by a constraint in I, then

$$|R| \leq \sum_{i=1}^{r-1} \left(\binom{r}{i} \cdot j^{r-i} \right)$$

for every constraint $R \in C$. Since $r \leq \text{twprim}(I) + 1$ (simply because each constraint induces a clique in $\text{prim}(I)$), we have that $k_2 \leq g(k)$ for some computable function $g \colon \mathbb{N} \to \mathbb{N}$. Therefore the mapping $(I, k) \mapsto (I, k + g(k))$ is a fpt-reduction from CSP$(\text{twprim}, \Gamma_{rc})$ to the above problem, and the lemma is settled. □

Lemma 3. *If* $(r, c) \leq (*, j)$*, then* $\text{CSP}(\text{constr}, \Gamma_{rc}) \in \text{FPT}$.

Proof. Let $j \in \mathbb{N}$, $(r, c) \leq (*, j)$, and $I = (V, D, C)$ be an instance of CSP(Γ_{rc}).

Let E be the set of all two-element subsets of C, so that $|E| = \binom{|C|}{2}$, and let $L(E)$ be the set of all maps $g \colon E \to \{0, 1\}$, so that $|L(E)| = 2^{|E|}$. The algorithm loops over $L(E)$, and rejects if the loop terminates. For each $g \in L(E)$, a corresponding simplified version of the instance I is computed by the following steps:

1. For all $\{R(X), S(Y)\} \subseteq C$ and $T(Z) \in C$, if $g(\{R(X), S(Y)\}) = 0$, then redefine $T \coloneqq \pi_{Z \setminus (X \cap Y)}(T \bowtie U)$, where $U = \{x \mapsto 0\}$ is the $(X \cap Y)$-relation containing the constant 0 mapping, $x \mapsto 0$.

2. For all $\{R(X), S(Y)\} \subseteq C'$, if $g(\{R(X), S(Y)\}) = 1$, then redefine $R \coloneqq R \bowtie U'$ and $S \coloneqq S \bowtie U''$, where $U' = (\pi_{X \cap Y} R) \setminus \{x \mapsto 0\}$, and $U'' = (\pi_{X \cap Y} S) \setminus \{x \mapsto 0\}$.

Step 1 is feasible in polynomial-time, by computing for at most $\binom{|C|}{2}|C|$ times at most one join and at most one projection of relations of size at most $\max\{\|R\| \mid R \in C\}$. Similarly, Step 2 is feasible in polynomial-time.

Now, the simplified instance returned by Step 1 and Step 2 is checked for satisfiability, as follows. In polynomial-time, compute all maximal connected components of the dual graph H of the (simplified) instance.

Loop over all maximal connected components K of H (clearly, $|K| \leq |C|$). If $|K| = 1$, that is $K = \{X\}$ for some constraint $R(X)$, then loop over the next map in $L(E)$ if R is empty. This is polynomial-time computable. Otherwise, if $|K| \geq 2$, then compute the join of all constraints whose scope is in K, namely

$$\bowtie_{X \in K} R(X), \tag{6}$$

and loop over the next map in $L(E)$ if $\bowtie_{X \in K} R(X)$ is empty. If $\bowtie_{X \in K} R(X)$ is nonempty for all maximal connected components K of H, accept.

We claim that (6) is computable in polynomial-time. First note that each constraint in the simplified instance returned by Step 1 and Step 2, and thus each $R(X)$ with $X \in K$, has column sum bounded by j. Moreover, if $R(X), S(Y) \in C$ with $X, Y \in K$, then by Step 2 we have that $\pi_{X \cap Y}(R \bowtie S)$ does not contain the constant 0 map.

Claim 3. If $R(X)$ and $S(Y)$ have column sum bounded by j and $\pi_{X \cap Y}(R \bowtie S)$ does not contain the constant 0 map, then $|R \bowtie S| \leq j|X \cap Y|$.

We prove Claim 3. Suppose $|R \bowtie S| > j|X \cap Y|$ for a contradiction. Note that both $\pi_{X \cap Y} R$ and $\pi_{X \cap Y} S$ have column sum bounded by j. Therefore, $|\pi_{X \cap Y} R|, |\pi_{X \cap Y} S| \leq j|X \cap Y|$. Then, by the pigeonhole principle, there exist at least $j+1$ maps $f_1, \dots, f_{j+1}$ in R such that $|\pi_{X \cap Y}\{f_1, \dots, f_{j+1}\}| = 1$, or there exist at least $j+1$ maps $g_1, \dots, g_{j+1}$ in S such that $|\pi_{X \cap Y}\{g_1, \dots, g_{j+1}\}| = 1$. In the first case, by hypothesis, the map $f \in \pi_{X \cap Y}\{f_1, \dots, f_{j+1}\}$ is not the constant 0, so that there exists $x \in X$ such that $|\{f \in R \mid f(x) \neq 0\}| \geq j+1$, therefore $\mathrm{colsum}(0, R(X)) \geq j+1$, a contradiction. The second case is similar.

By the claim $R \bowtie S$ is computable in time $O(3j|V|) = O(|V|)$, so that $\bowtie_{X \in K} R(X)$ is computable in time $O(|C| \cdot |V|)$. Thus, the algorithm runs in time $f(|C|)n^{O(1)}$. Correctness is clear, and the lemma is settled. □

5 Intractability

Lemma 4. *Let* $(r, c) \geq (2, *)$.

1. $\mathrm{CSP}(\mathrm{query}, \Gamma_{rc})$ *is* W[1]*-hard.*
2. $\mathrm{CSP}(\mathrm{constr}, \Gamma^{01}_{rc})$ *is* W[1]*-hard.*

Proof. The parameterized problem CLIQUE is the problem of deciding, given in input a graph G and $k \in \mathbb{N}$, whether G contains a k-clique. The parameterization is $\kappa(G, k) = k$.

The reduction from CLIQUE given in [15, Theorem 1] basically proves the first statement. Assume $(r, c) \geq (2, *)$. Let (G, k) with $G = (V, E)$ be an instance of CLIQUE. Without loss of generality, $V \subseteq \mathbb{N}$. Let $I = (\{v_1, \ldots, v_k\}, V, C)$ be the CSP instance defined by $C = \{E'v_iv_j \mid i < j \text{ in } [k]\}$ where $E' = \{(a,b),(b,a) \mid \{a,b\} \in E\}$. Note that $\text{rowsum}(0, E') \leq 2$, just because E' is binary, thus I is an instance of CSP(Γ_{rc}).

Clearly, if G has a k-clique, then I is satisfiable. Conversely, any mapping of the variables in V satisfying I is injective (because the graph is loopless), thus if I is satisfiable, then G has a k-clique. Moreover, $\text{query}(I) \leq 2 \cdot \binom{k}{2} = f(k)$, so that the mapping $(G,k) \mapsto (I, f(k))$ is indeed a fpt-reduction from CLIQUE to CSP$(\text{query}, \Gamma_{rc})$. Since CLIQUE is W[1]-complete, the first part of the lemma is settled.

For the second statement, let (G, k) be an instance of CLIQUE, with $G = ([n], E)$. Intuitively, we rephrase the reduction given in Item 1 by encoding the vertices of G in unary. The details, similar to [16, Theorem 8], follow.

Let $R_n = \{b_1, \ldots, b_n\} \subseteq \{0,1\}^n$ such that $b_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with $(b_i)_j = 1$ if and only if $j = i$. If $b_i, b_j \in R_n$, we let $b_ib_j \in \{0,1\}^{2n}$ denote the $2n$-tuple such that $(b_ib_j)_l = 1$ if and only if $l \in \{i, n+j\}$. We then let $E_n = \{b_ib_j, b_jb_i \mid \{i,j\} \in E\} \subseteq \{0,1\}^{2n}$. The instance $I = (V, \{0,1\}, C)$ of CSP(Γ^{01}_{rc}) is defined as follows: $V = \{v_{i1}, \ldots, v_{in} \mid i \in [k]\}$; C contains constraints $R_nv_{i1} \ldots v_{in}$ for all $i \in [k]$, and $E_nv_{i1} \ldots v_{in}v_{j1} \ldots v_{jn}$ for all $i < j$ in $[k]$. It is easy to check that the mapping $(G,k) \mapsto (I,k)$ is the required reduction. □

Lemma 5. *If* $(r,c) \geq (2,1)$*, then* CSP$(\text{twdual}, \Gamma^{01}_{rc})$ *is* W[1]*-hard.*

Proof. A *k-partite graph* is a graph $G = (V_1 \cup \cdots \cup V_k, E)$ such that the family $\{V_1, \ldots, V_k\}$ partitions the vertices of G into k independent sets. The problem PARTITIONED-CLIQUE is the problem of deciding, given as input a k-partite graph G and $k \in \mathbb{N}$, whether G contains a k-clique. The parameterization is $\kappa(G,k) = k$. PARTITIONED-CLIQUE is W[1]-complete; we give a reduction PARTITIONED-CLIQUE $\leq^{\text{fpt}}$ CSP$(\text{twdual}, \Gamma^{01}_{rc})$.

Given a k-partite graph G, we construct an instance $I = (V \cup V' \cup E', \{0,1\}, C)$ of CSP(Γ^{01}_{rc}), where $V' = \{a' \mid a \in V\}$, $E' = \{(a,b),(b,a) \mid \{a,b\} \in E\}$, and C is specified as follows. The constraint relations are (copies of) the following relations R_m and S_{2m}, defined for all $m \leq |G|^2$:

- $R_m \subseteq \{0,1\}^m$ such that $\text{rowsum}(0, R_m) = \text{colsum}(0, R_m) = 1$;
- $S_{2m} = \{b_1, \ldots, b_m\} \subseteq \{0,1\}^{2m}$ such that $\text{rowsum}(0, S_{2m}) = 2$ and $b_{ii} = b_{i(m+i)} = 1$ for all $i \in [m]$; for instance, $S_4 = \{(1,0,1,0),(0,1,0,1)\}$.

Note that $\text{colsum}(0, S_{2m}) = 1$, so I is an instance of CSP(Γ^{01}_{21}).

The constraints in C are defined as follows. For all $i \in [k]$ and $a \in V_i$, if $V_i = \{a_1, \ldots, a_r\}$, then C contains the constraints

$$\psi_i = R_r a'_1 \ldots a'_r, \text{ and} \quad (7)$$

$$\psi_{ia} = R_2 aa'. \quad (8)$$

Intuitively, a mapping $f \colon V \cup V' \cup E' \to \{0,1\}$ satisfies constraints of the form (7)-(8) if and only if, for every block V_i of the k-partition of the vertices of G, it holds that $f(a) = 0$ for exactly one vertex $a \in V_i$ (in this case, say that f *selects* a in V_i).

For all $i < j$ in $[k]$, $a \in V_i$, and $b \in V_j$, if $\{(a, b_1), \ldots, (a, b_s)\} = E' \cap (\{a\} \times V_j)$, $\{(b, a_1), \ldots, (b, a_t)\} = E' \cap (\{b\} \times V_i)$, and $\{\{c_1, d_1\}, \ldots, \{c_u, d_u\}\} = E \cap (V_i \cup V_j)$, then C contains the constraints

$$\varphi_{ija} = R_{s+1}a(a, b_1) \ldots (a, b_s), \tag{9}$$

$$\varphi_{ijb} = R_{t+1}b(b, a_1) \ldots (b, a_t), \text{ and} \tag{10}$$

$$\varphi_{ij} = S_{2u}(c_1, d_1) \ldots (c_u, d_u)(d_1, c_1) \ldots (d_u, c_u). \tag{11}$$

Intuitively, a mapping $f \colon V \cup V' \cup E' \to \{0, 1\}$ satisfies all constraints of the form (9)-(11) if and only if, if f selects a in V_i and b in V_j, then the edge $\{a, b\}$ is in E.

It is then straightforward to check that G has a k-clique if and only if I is satisfiable.

Claim 4. $\mathrm{twdual}(I) \leq (k-1)(1 + \binom{k}{2}) \coloneqq g(k)$.

We prove Claim 4. First note that $\mathrm{twinc}(I) \leq 1 + \binom{k}{2}$. Indeed, it is easy to check that, for every graph J with vertex set U and every subgraph J' of J induced by some $U' \subseteq U$, it holds that $\mathrm{tw}(J) \leq \mathrm{tw}(J') + |U \setminus U'|$. Applying the inequality with $U = V \cup C$ and $U' = V \cup \{\psi \in C \mid \psi \text{ not of form (11)}\}$, we have $\mathrm{tw}(J') = 1$ and $|U \setminus U'| = \binom{k}{2}$. Next note that each variable in $V \cup V'$ is contained in the scope of at most $k - 1$ constraints in C. In fact, if $a' \in V'$, then a' is contained in the two constraints in (7)-(8); if $(a, b) \in E$, then (a, b) is contained in one constraint of the form (11), and in either the constraint in (9) or the constraint in (10); if $a \in V$, then a is contained in either at most $k - 1$ constraints of the form (9) or in at most $k - 1$ constraints of the form (10). Therefore, given a tree decomposition of $\mathrm{inc}(I)$ of width at most $1 + \binom{k}{2}$, a tree decomposition of $\mathrm{dual}(I)$ of width at most $g(k) = (k-1)(1 + \binom{k}{2})$ is defined by replacing, in each bag, each variable contained in the bag by the at most $k-1$ constraints whose scopes contain the variable.

The mapping $(G, k) \mapsto (I, g(k))$ is computable in polynomial time, therefore it gives an fpt-reduction from PARTITIONED-CLIQUE to CSP$(\mathrm{twdual}, \Gamma^{01}_{rc})$. □

6 Conclusion

We proposed a novel restriction pattern for the CSP by combining *structural parameterizations* and *language restrictions*. For every structural parameterization $p \in \mathrm{PAR}$ and constraint language Γ, we were interested in finding a criterion such that CSP$(p, \Gamma) \in$ FPT if and only if Γ satisfies the criterion.

We introduced a countable family of *infinite* constraint languages, of the form Γ_{rc} with $(r, c) \in (\mathbb{N} \cup \{*\})^2$, and we proved a classification result (Theorem 1) of the following form: For every $p \in \mathrm{PAR}$, CSP$(p, \Gamma_{rc}) \in$ FPT if and only if $(r, c) \in S_p$, where $S_p \subseteq (\mathbb{N} \cup \{*\})^2$ is explicitly defined.

The classification identifies nontrivial fixed-parameter tractable cases; a notable example is the language Γ_{rc} with $(r, c) \leq (*, j)$ for any $j \in \mathbb{N}$, which has the Boolean language of identity matrices as a special case, is in FPT parameterized by the number of constraints (Lemma 3). Moreover, for an arbitrary language Γ, a careful inspection of the proofs indicates that the hardness of CSP(p, Γ_{rc}) arises, respectively, from $r \geq 2$ if $p \in \{\text{twdual}, \text{twinc}\}$, and from $c = *$ if $p \in \{\text{query}, \text{var}, \text{constr}, \text{twprim}\}$.

We remark that there are examples of (even infinite and Boolean) languages of the form $\Gamma \subseteq \Gamma_{rc}$ such that the CSP$(p, \Gamma) \in$ FPT, but $(r, c) \notin S_p$; so the classification does not characterize fixed-parameter tractable languages under the structural parameterizations addressed. However, it provides directions for future refinements; we briefly discuss its consequences with respect to *finite* constraint languages.

Clearly, CSP$(p, \Gamma) \in$ FPT for every $\Gamma \subseteq \Gamma_{rc}$ with $(r, c) \in S_p$. Moreover, if Γ is a finite language, then $(r, c) \in \mathbb{N}^2$. Hence, by the classification, CSP(p, Γ) is W[1]-hard only if $p \in \{\text{twdual}, \text{twinc}\}$ and $(r, c) \geq (2, 1)$. Therefore, to refine the classification in the finite case, natural candidates are finite languages $\Gamma \subseteq \Gamma_{21}$ having row sum equal to 2, that is, containing at least one relation R such that $\text{rowsum}(0, R) = 2$.

References

[1] Brualdi, R.A.: Matrices of zeroes and ones with fixed row and column sum vectors. Linear Algebra Appl. 33, 159–231 (1980)

[2] Bulatov, A.A., Jeavons, P., Krokhin, A.A.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. 34(3), 720–742 (2005)

[3] Cohen, D.A., Cooper, M.C., Creed, P., Marx, D., Salamon, A.Z.: The tractability of CSP classes defined by forbidden patterns. J. Artif. Intell. Res. 45, 47–78 (2012)

[4] Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inform. Comput. 85(1), 12–75 (1990)

[5] Feder, T., Vardi, M.Y.: The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. SIAM J. Comput. 28(1), 57–104 (1999)

[6] Flum, J., Frick, M., Grohe, M.: Query evaluation via tree-decompositions. J. ACM 49(6), 716-752 (2002)

[7] Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Heidelberg (2006)

[8] da Fonseca, C.M., Mamede, R.: On $(0, 1)$-matrices with prescribed row and column sum vectors. Discrete Math. 309(8), 2519–2527 (2009)

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

[10] Grohe, M., Marx, D.: Constraint solving via fractional edge covers. In: Proc. 17th ACM-SIAM Symposium on Discrete Algorithms (SODA'06), pp. 289–298. ACM Press, New York (2006)

[11] Kolaitis, P.G., Vardi, M.Y.: Conjunctive-query containment and constraint satisfaction. J. Comput. Syst. Sci. 61(2), 302–332 (2000)

[12] Krokhin, A.A., Marx, D.: On the hardness of losing weight. ACM T. Algorithms 8(2), 19 (2012)

[13] Marx, D.: Parameterized complexity of constraint satisfaction problems. Computat. Complex. 14(2), 153–183 (2005)

[14] Marx, D.: Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In: Proc. 42nd ACM Symposium on Theory of Computing (STOC'10), pp. 735–744. ACM Press, New York (2010)

[15] Papadimitriou, C.H., Yannakakis, M.: On the complexity of database queries. J. Comput. Syst. Sci. 58(3), 407–427 (1999)

[16] Samer, M., Szeider, S.: Constraint satisfaction with bounded treewidth revisited. In: Proc. 12th Internationial Conference on Principles and Practice of Constraint Programming (CP'06). LNCS, vol. 4204, pp 499–513. Springer, Heidelberg (2006)

[17] Schaefer, T.J.: The complexity of satisfiability problems. In: Proc. 10th Symposium on Theory of Computing (STOC'78), pp. 216–226. ACM Press, New York (1978)