

# Lower Bounds and Faster Algorithms for Equality Constraints

Peter Jonsson and Victor Lagerkvist

Department of Computer and Information Science, Linköping University, Linköping, Sweden  
 {peter.jonsson, victor.lagerkvist}@liu.se

## Abstract

We study the fine-grained complexity of NP-complete, infinite-domain constraint satisfaction problems (CSPs) parameterised by a set of first-order definable relations (with equality). Such CSPs are of central importance since they form a subclass of any infinite-domain CSP parameterised by a set of first-order definable relations. We prove that under the randomised exponential-time hypothesis it is not possible to find  $c > 1$  such that a CSP over an arbitrary finite equality language is solvable in  $O(c^n)$  time ( $n$  is the number of variables). Stronger lower bounds are possible for infinite equality languages where we rule out the existence of  $2^{o(n \log n)}$  time algorithms; a lower bound which also extends to *satisfiability modulo theories* solving for an *arbitrary* background theory. Despite these lower bounds we prove that for each  $c > 1$  there exists an NP-hard equality CSP solvable in  $O(c^n)$  time. Lower bounds like these immediately ask for closely matching upper bounds, and we prove that a CSP over a finite equality language is always solvable in  $O(c^n)$  time for a fixed  $c$ .

## 1 Introduction

The *constraint satisfaction problem* (CSP) is the problem of determining whether a set of constraints has at least one satisfying assignment. Depending on the set of allowed constraints  $\Gamma$ , called a *template* or a *constraint language*, it is possible to formulate many natural problems as  $\text{CSP}(\Gamma)$  problems. This is especially true if we allow templates over an infinite universe, which increases the expressive power of CSPs and e.g. makes it possible to formulate many problems from artificial intelligence [Bodirsky and Jonsson, 2017; Dylla *et al.*, 2017]. The complexity of CSPs have also been the subject of intense theoretical research: for each constraint language  $\Gamma$  over a finite domain  $\text{CSP}(\Gamma)$  is always either polynomial-time solvable or is NP-complete [Bulatov, 2017; Zhuk, 2017]. Infinite-domain CSPs are in general undecidable, but there exists a wealth of results when additional restrictions are imposed. Early examples include the CSP formulation of *Allen’s interval algebra* [Krokhin *et al.*, 2003], the *region connection calculus* [Renz and Nebel,

1999], CSPs over first-order definable relations with equality [Bodirsky and Kára, 2008] (equality CSPs), and *temporal CSPs* [Bodirsky and Kára, 2010]. More generally, it is common to consider *first-order reducts* of a fixed relational structure  $\mathcal{A}$ , i.e., languages that are first-order definable (with equality) over  $\mathcal{A}$ . Equality CSPs then correspond to  $\text{CSP}(\Gamma)$  when  $\Gamma$  is a first-order reduct of  $(\mathcal{A}; \emptyset)$  for some universe  $A$  (an *equality language*) while temporal CSPs correspond to  $\text{CSP}(\Gamma)$  when  $\Gamma$  is a first-order reduct of  $(\mathbb{Q}; <)$ . Equality CSPs have previously been intensively studied due to their fundamental importance for understanding more complex CSPs, since any classification of a larger relational structure  $\mathcal{A}$  necessarily also needs to include a classification of equality CSPs (an equality language  $\Gamma$  is a reduct of *any* countably infinite structure  $\mathcal{A}$ ). Let us also remark that CSPs in this setting are very similar to reasoning problems occurring in artificial intelligence, where one fixes a set of “base relations”  $\mathcal{A}$ , typically binary, and then consider a satisfiability problem where constraints are taken from e.g. the relation algebra generated by  $\mathcal{A}$ , or the set of all disjunctive clauses over  $\mathcal{A}$  [Dylla *et al.*, 2017]. A recent comparison may also be found in *satisfiability modulo theories* (SMT) where a background theory  $\mathcal{A}$  is fixed, and where one considers the satisfiability problem of first-order formulas (with equality) restricted to interpretations agreeing with  $\mathcal{A}$  [Biere *et al.*, 2009].

While theoretical CSP research has concentrated on classical complexity, complexity theory itself has partially shifted towards parameterised complexity and *fine-grained complexity*, which e.g. encompasses constructing improved exponential-time algorithms, and proving lower bounds with stronger assumptions than  $\text{P} \neq \text{NP}$  such as the (*strong*) *exponential-time hypothesis* (SETH) (see, e.g., [Cygan *et al.*, 2015]). In this paper we study the fine-grained complexity of NP-hard infinite-domain CSPs, with a particular focus on equality CSPs using the number of variables,  $n$ , as the complexity parameter. As remarked, equality CSPs constitute a natural starting point for questions of fine-grained complexity, since if we cannot even overcome this obstacle there is little hope of understanding fine-grained complexity questions for larger classes of CSPs. Assume, for example, that we prove that there exists an equality language  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is not solvable in  $O(f(n))$  time, for some function  $f$ . Then, regardless of which relational structure  $\mathcal{A}$  that we choose, we cannot hope to construct an algorithm with a run-

ning time of  $O(f(n))$  which is applicable to  $\text{CSP}(\Delta)$  for every first-order reduct  $\Delta$  of  $\mathcal{A}$ . Under this viewpoint it is therefore crucial to prove lower bounds for equality CSPs before moving on to construct faster exponential-time algorithms for broader classes of infinite-domain CSPs.

Thus, among the class of NP-hard equality CSPs, how does the choice of  $\Gamma$  affect the fine-grained complexity of  $\text{CSP}(\Gamma)$ ? For example, it is known that  $\text{CSP}(\Gamma)$  is solvable in  $O^*(2^{n \cdot \log(\frac{0.792n}{\ln(n+1)}})})$  time when  $\Gamma$  is an arbitrary equality language [Jonsson and Lagerkvist, 2017] (the  $O^*$  notation is used to suppress polynomial factors). Concerning lower bounds it is known that no NP-complete equality  $\text{CSP}(\Gamma)$  problem is solvable in subexponential time without violating the ETH [Jonsson *et al.*, 2017b], and if  $\Gamma$  is the full first-order reduct of  $(A; \emptyset)$  then there cannot exist an  $O^*(c^n)$  time algorithm for  $\text{CSP}(\Gamma)$  for any constant  $c$  without violating the SETH [Jonsson and Lagerkvist, 2017]. Despite bounds like these, there are still large gaps in our understanding of fine-grained complexity of infinite-domain CSPs in general, and of equality CSPs in particular. For example, is it possible to find an equality language  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is NP-complete but solvable in  $O(c^n)$  time for a constant  $c > 1$ ? Is it possible to solve  $\text{CSP}(\Gamma)$  in  $O(c^n)$  time whenever  $\Gamma$  is a finite equality language, and in that case, does  $c$  depend on  $\Gamma$  or is it possible to find a uniform value? Furthermore, since no NP-complete equality CSP is solvable in subexponential time without violating the ETH, does there exist a  $c > 1$  such that no NP-complete equality CSP is solvable in  $O(c^n)$  time? After defining the necessary preliminaries (in Section 2) we in Section 3 begin to answer these questions by a careful study of lower bounds. First, we prove that under the *randomised ETH* for each  $c > 1$  there exists a finite equality language  $\Gamma_c$  such that  $\text{CSP}(\Gamma_c)$  is not solvable in  $O(c^n)$  time. Second, we showcase a striking difference between finite and infinite languages and prove the existence of an infinite equality language  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is not solvable in  $2^{o(n \log n)}$  time (under the ETH). In particular this lower bound rules out a uniform  $O(c^n)$  time algorithm,  $c > 1$ , applicable to arbitrary equality CSPs (which previously was only known to hold under the much stronger SETH). We also manage to lift this lower bound to SMT, where little is known about the fine-grained complexity, despite being a framework with a wide range of applications due to the success of efficient SAT solvers. We provide the first known lower bound under the ETH and show that *regardless* of the background theory it is not possible to solve the resulting SMT in  $2^{o(n \log n)}$  time without violating the ETH. Furthermore, we are able to prove this as a straightforward consequence of our general bounds for equality CSPs, indicating yet another advantage of studying fine-grained complexity in this setting. Third, we prove that for each constant  $c > 1$  there exists an NP-complete equality CSP which is solvable in  $O(c^n)$  time, and thus rule out the existence of an “easiest NP-complete equality CSP”. Such CSPs are known to exist for finite-domain CSPs [Jonsson *et al.*, 2017b] so we see a clear dividing line between finite and infinite-domain CSPs.

In light of these lower bounds, what is the best possible exponential-time algorithm for equality CSPs that we could

hope for? We tackle this question in Section 4 and construct an  $O^*(c^n)$  time algorithm for  $\text{CSP}(\Gamma)$  whenever  $\Gamma$  is a *finite* equality language, where  $c$  is a constant depending only on the arities of relations in  $\Gamma$ . Note that while the constant  $c$  likely can be improved, we have already established (under the randomised ETH) that it is not possible to find a uniform value. Similarly, it appears difficult to extend the algorithm to non-trivial classes of infinite equality languages since we have already proved that there is an infinite equality language that cannot be solved in  $2^{o(n \log n)}$  time (and the ETH). Here, it is also interesting to note that certain classes of infinite-domain CSPs do not admit an  $O(c^n)$  algorithm even if the template is finite. For instance, there is a finite temporal language whose CSP (under the randomised ETH) cannot be solved in  $2^{o(n \log n)}$  time [Jonsson and Lagerkvist, 2018].

These results paint a peculiar picture of the fine-grained complexity of equality CSPs (and *all* classes of infinite-domain CSPs over first-order reducts of relational structures). On the one hand, equality CSPs are incredibly hard to solve (no uniform  $O(c^n)$  time algorithm for finite languages under the randomised ETH, and no  $2^{o(n \log n)}$  time algorithm for infinite languages), but on the other hand one for any  $c > 1$ , say,  $c = 1.00001$ , can find an NP-hard equality CSP solvable in  $O(c^n)$  time. These conflicting messages indicate that a complete understanding of fine-grained complexity of equality CSPs is well out of reach, but we have simultaneously unravelled several interesting research directions. We discuss some of these in Section 5.

## 2 Preliminaries

A *relational structure* is a tuple  $(A; \sigma, I)$  where  $A$  is a set typically called a *domain*, or a *universe*,  $\sigma$  is a relational signature, and  $I$  is a function from  $\sigma$  to the set of all relations over  $A$  which assigns each relation symbol a corresponding relation over  $A$ . For simplicity, we will typically write a relational structure as  $(A; R_1, \dots, R_k)$  where each  $R_i$  is a relation over  $A$ , and will not make a sharp distinction between relations and their corresponding signatures. A set of relations  $\Gamma$  over  $A$  is a *first-order reduct* of a relational structure  $(A; R_1, \dots, R_k)$  if each  $R \in \Gamma$  is the set of models of a  $\sigma$ -formula (with equality) interpreted in  $(A; R_1, \dots, R_k)$ . In symbols, we write  $R(x_1, \dots, x_n) \equiv \varphi(x_1, \dots, x_n)$  if  $R$  is the set of models of the first-order formula  $\varphi(x_1, \dots, x_n)$  with respect to the free variables  $x_1, \dots, x_n$ .

### 2.1 The Constraint Satisfaction Problem

Let  $\Gamma$  be a set of finitary relations over some set  $A$  of values, occasionally called a *constraint language*. The *constraint satisfaction problem* over  $\Gamma$  ( $\text{CSP}(\Gamma)$ ) is defined as follows.

*Instance:* A set  $V$  of variables and a set  $C$  of *constraints* of the form  $R(x_1, \dots, x_k)$ , where  $k$  is the arity of  $R$ ,  $x_1, \dots, x_k \in V$  and  $R \in \Gamma$ .  
*Question:* Is there a function  $f: V \rightarrow A$  such that  $(f(x_1), \dots, f(x_k)) \in R$  for every  $R(x_1, \dots, x_k) \in C$ ?

Concerning representation, we take a simple approach and only consider the case when  $\Gamma$  is a first-order reduct of a re-

lational structure, and represent each relation  $R \in \Gamma$  by a first-order formula. However, the exact representation is only important if  $\Gamma$  is infinite, since any reasonable representation can be chosen and precomputed if  $\Gamma$  is finite.

## 2.2 Primitive Positive Definitions and Interpretations

Let  $\Gamma$  be a constraint language over a domain  $A$ . A  $k$ -ary relation  $R$  is said to have a *primitive positive definition* (pp-definition) over  $\Gamma$  if  $R(x_1, \dots, x_k) \equiv \exists y_1, \dots, y_{k'} : R_1(\mathbf{x}_1) \wedge \dots \wedge R_m(\mathbf{x}_m)$  where each  $R_i \in \Gamma \cup \{\text{Eq}_A\}$  and each  $\mathbf{x}_i$  is a tuple of variables over  $x_1, \dots, x_k, y_1, \dots, y_{k'}$  matching the arity of  $R_i$ . Here, and in the sequel,  $\text{Eq}_A$  is the equality relation  $\{(a, a) \mid a \in A\}$  over  $A$ . Thus,  $R$  is definable by a first-order formula consisting only of existential quantification and conjunction over positive atoms from  $\Gamma$  and equality constraints. If  $\Gamma$  is a constraint language we let  $\langle \Gamma \rangle$  be the smallest set of relations containing  $\Gamma$  closed under pp-definitions. Pp-definitions are typically only useful for comparing similar languages over the same domain, but can be generalised as follows.

**Definition 1.** Let  $A$  and  $B$  be two domains and let  $\Gamma$  and  $\Delta$  be two constraint languages over  $A$  and  $B$ , respectively. A primitive positive interpretation (pp-interpretation) of  $\Delta$  over  $\Gamma$  consists of a  $d$ -ary relation  $F \subseteq A^d$  and a surjective function  $f: F \rightarrow B$  such that  $F, f^{-1}(\text{Eq}_B) \in \langle \Gamma \rangle$  and  $f^{-1}(R) \in \langle \Gamma \rangle$  for every  $k$ -ary  $R \in \Delta$ , where  $f^{-1}(R)$  denotes the  $(k \cdot d)$ -ary relation  $\{(x_1^1, \dots, x_1^d, \dots, x_k^1, \dots, x_k^d) \in A^{k \cdot d} \mid (f(x_1^1, \dots, x_1^d), \dots, f(x_k^1, \dots, x_k^d)) \in R\}$ .

Hence, pp-interpretations are generalisations of pp-definitions, and can be used to obtain polynomial-time reductions between CSPs (cf. Theorem 5.5.6 in [Bodirsky, 2012]).

## 2.3 Equality Languages

We say that  $\Gamma$  is an *equality language* if each  $R \in \Gamma$  admits a first-order definition over a relational structure  $(A; \emptyset)$ , i.e. the empty structure. Recall here that the equality relation is always accessible in first-order logic. Without loss of generality we henceforth assume that  $A = \mathbb{N}$ , write  $\text{Eq}$  (or  $=$  in infix notation) for the equality relation over  $\mathbb{N}$ , and  $\text{R}_{\neq}$  or  $\neq$  (in infix notation) for the inequality relation  $\{(x, y) \in \mathbb{N}^2 \mid x \neq y\}$  over  $\mathbb{N}$ . The computational problem we consider is then  $\text{CSP}(\Gamma)$  when  $\Gamma$  is an equality language. This problem is easily seen to belong to NP for any finite language, and its classical complexity has been completely classified [Bodirsky and Kára, 2008].

**Theorem 1.** Let  $\Gamma$  be an equality language. Then either  $\text{CSP}(\Gamma)$  is (1) polynomial-time solvable or (2) there exists a finite  $\Delta \subseteq \Gamma$  such that  $\text{CSP}(\Delta)$  is NP-complete since  $\Delta$  pp-interprets every finite-domain relation.

**Example 1.** Let  $S = \{(x, x, y), (x, y, y) \mid x, y \in \mathbb{N}, x \neq y\}$ , and observe that  $S(x, y, z) \equiv (x = y \wedge y \neq z) \vee (x \neq y \wedge y = z)$ . Thus,  $\{S\}$  is an equality language, and it is known that  $\{S\}$  pp-interprets a language  $\Delta$  where  $\text{CSP}(\Delta)$  is NP-hard, which implies that  $\text{CSP}(\{S\})$  is NP-hard, too. For tractability, if we take  $\{\text{Eq}, \text{R}_{\neq}\}$  then  $\text{CSP}(\{\text{Eq}, \text{R}_{\neq}\})$  is well-known to be polynomial-time solvable. This can be proven

via Theorem 1, but  $\text{CSP}(\{\text{Eq}, \text{R}_{\neq}\})$  can also be solved by propagation methods.

## 2.4 Fine-Grained Complexity and the Exponential-Time Hypothesis

Assume that  $\text{CSP}(\Gamma)$  is NP-complete. How fast can we solve  $\text{CSP}(\Gamma)$ , and is it possible to prove stronger lower bounds than an expected superpolynomial running time (under  $\text{P} \neq \text{NP}$ )? Such questions, especially when the complexity parameter is the number of variables  $|V|$  or the number of constraints  $|C|$ , fall under the umbrella of *fine-grained complexity*. To prove non-trivial lower bounds for NP-complete problems we typically need stronger assumptions than  $\text{P} \neq \text{NP}$ . Say that  $\text{CSP}(\Gamma)$  is solvable in *subexponential time* if  $\text{CSP}(\Gamma)$  is solvable in  $O(2^{\varepsilon|V|})$  for each  $\varepsilon > 0$ . The conjecture that 3-SAT is not solvable in subexponential time is called the *exponential-time hypothesis* (ETH). There exists several stronger variants of the ETH. First, an algorithm  $A$  is said to be a  $2^{c|V|}$ -randomised algorithm if its running time is bounded by  $2^{c|V|} \cdot \text{poly}(\|I\|)$  and its error probability is at most  $1/3$  ( $\|I\|$  is the number of bits required to represent a CSP instance  $I$ ). For  $k, d \geq 1$  we then define (1)  $c_k = \inf\{c \mid \exists \text{ a deterministic } 2^{c|V|} \text{ algorithm for } k\text{-SAT}\}$  and (2)  $c_{d,k} = \inf\{c \mid \exists \text{ a } 2^{c|V|}\text{-randomised algorithm for } \text{CSP}(\Gamma_{d,k})\}$ , where  $\Gamma_{d,k}$  is the set of all relations over the set  $\{0, \dots, d-1\}$  of arity at most  $k$ . The *randomised exponential-time hypothesis* (r-ETH) is then the conjecture that  $c_{2,3} > 0$ , i.e., that 3-SAT is not solvable in subexponential time even with randomised algorithms, and the *strong exponential-time hypothesis* (SETH) is the conjecture that the limit of the sequence  $c_3, c_4, \dots$  is equal to 1.

## 3 Lower Bounds on the Complexity of Equality Constraints

In this section we investigate lower bounds for equality CSPs. As remarked in Section 1, such lower bounds are valuable since if it is possible to prove that  $\text{CSP}(\Gamma)$  is not solvable in  $O(f(|V|))$  time (for some function  $f$ ) then we cannot in general expect to solve  $\text{CSP}(\Delta)$  in  $O(f(|V|))$  time when  $\Delta$  is a first-order reduct of an *arbitrary* relational structure. Let us recapitulate two known lower bounds.

**Theorem 2.** (1) If  $\text{CSP}(\Gamma)$  for an equality language  $\Gamma$  is NP-hard then it is not solvable in subexponential time unless the ETH is false [Jonsson et al., 2017b, Thm. 9] and (2) if  $\Gamma$  is the full first-order reduct of  $(\mathbb{N}; \emptyset)$  then  $\text{CSP}(\Gamma)$  is not solvable in  $O(c^{|V|})$  time for any  $c > 1$  unless the SETH is false. (Theorem 19 in [Jonsson and Lagerkvist, 2017].)

### 3.1 Finite Versus Infinite Equality Languages

We begin by proving that for every  $c > 1$  there exists a finite equality language  $\Gamma_c$  such that  $\text{CSP}(\Gamma_c)$  is not solvable in  $O(2^{c|V|})$  time without contradicting the r-ETH. We first require the following result [Traxler, 2008, Thm. 1].

**Theorem 3.** If r-ETH holds, then there exists a universal constant  $\alpha > 0$  such that for all  $d \geq 3$ ,  $\alpha \cdot \log(d) \leq c_{d,2}$ .

**Theorem 4.** *For every  $c > 1$ , there exists a finite equality language  $\Gamma_c$  such that  $\text{CSP}(\Gamma_c)$  cannot be solved in  $O(2^{c \cdot |V|})$  (randomised) time unless the r-ETH is false.*

*Proof.* For  $1 \leq a, b \leq d$  define  $R_{d,a,b}(c_1, \dots, c_d, x, y) \equiv \bigvee_{i=1}^d x = c_i \wedge \bigvee_{i=1}^d y = c_i \wedge (x \neq c_a \vee y \neq c_b)$ . For arbitrary  $d$  then define the finite equality language  $\Theta_d = \{\neq\} \cup \{R_{d,a,b} \mid 1 \leq a, b \leq d\}$ . We present a polynomial-time reduction from  $\text{CSP}(\Gamma_{d,2})$  to  $\text{CSP}(\Theta_d)$  only introducing a constant number of fresh variables. Let  $(V, C)$  be an instance of  $\text{CSP}(\Gamma_{d,2})$ . Introduce  $d$  fresh variables  $c_1, \dots, c_d$  together with constraints  $\{c_i \neq c_j \mid 1 \leq i < j \leq d\}$ . For each  $R(x, y) \in C$ , add the constraints  $R_{d,a,b}(c_1, \dots, c_d, x, y)$  for every  $1 \leq a, b \leq d$  such that  $(a, b) \notin R$ . The resulting instance  $(V \cup \{c_1, \dots, c_d\}, C')$  can be constructed in polynomial time, and is clearly satisfiable if and only if  $(V, C)$  is satisfiable. Furthermore,  $d$  is fixed so only a constant number of fresh variables are introduced. By Theorem 3,  $\text{CSP}(\Theta_d)$  cannot be solved in  $2^{(c_{d,2}-\epsilon) \cdot |V|}$  time for any  $\epsilon > 0$  unless r-ETH is false, and the result follows by choosing  $d$  such that  $c_{d,2} \geq c$ . We know that  $\alpha \cdot \log(d) \leq c_{d,2}$  so it is sufficient to choose a  $d$  such that  $\alpha \cdot \log(d) \geq c$ , e.g.  $d = 2^{\lceil \frac{c}{\alpha} \rceil}$ .  $\square$

Thus, assuming the r-ETH, there cannot exist an algorithm solving  $\text{CSP}(\Gamma)$  in  $O(c^{|V|})$  time for every finite equality language  $\Gamma$ . This can be strengthened even further for infinite equality languages, and we will show the existence of  $\Gamma$  such that  $\text{CSP}(\Gamma)$  is not solvable in  $O(2^{o(|V| \log |V|)})$  time without contradicting the ETH. In contrast, the second statement of Theorem 2 is only valid under the much stronger SETH, and only if  $\Gamma$  consists of all first-order definable relations over  $(\mathbb{N}; \emptyset)$ . For this lower bound we provide a reduction from the  $k \times k$  INDEPENDENT SET problem: given a graph  $G$  over the vertex set  $\{1, \dots, k\} \times \{1, \dots, k\}$  (where  $k$  is part of the input), is there an independent set of size  $k$  in  $G$  with exactly one element from each row? The following lower bound is known under the ETH [Lokshtanov *et al.*, 2018].

**Theorem 5.**  *$k \times k$  INDEPENDENT SET is not solvable in  $2^{o(k \log k)}$  time unless the ETH is false.*

For  $n \geq 1$  define  $R_n(y, x_1, \dots, x_n) \equiv y = x_1 \vee y = x_2 \vee \dots \vee y = x_n$ , and let  $R(x, y, z, w) \equiv x \neq y \vee z \neq w$ . Let  $\Gamma_{\text{inf}}$  be the infinite equality language  $\{\neq, R, R_1, R_2, \dots\}$ .

**Theorem 6.**  *$\text{CSP}(\Gamma_{\text{inf}})$  cannot be solved in  $2^{o(|V| \log |V|)}$  time unless the ETH is false.*

*Proof.* To prove the result, we present a polynomial-time reduction from  $k \times k$  INDEPENDENT SET to  $\text{CSP}(\Gamma_{\text{inf}})$  such that the resulting  $\text{CSP}(\Gamma_{\text{inf}})$  instance only contains  $2k$  variables. Let  $G = (V, E)$  denote an arbitrary graph where  $V = \{1, \dots, k\} \times \{1, \dots, k\}$ . We then begin by introducing  $k$  variables  $a_1, \dots, a_k$  together with the constraints  $a_i \neq a_j, 1 \leq i < j \leq k$ . Second, for each row  $1 \leq i \leq k$  in  $G$ , introduce a variable  $x_i$  and the constraint  $R_k(x_i, a_1, \dots, a_k)$ . This constraint ensures that  $x_i$  equals one of the variables  $a_1, \dots, a_k$ . Third, for each edge  $e = ((a, b), (c, d)) \in E$ , introduce the constraint  $R(x_a, a_b, x_c, a_d)$ . This constraint guarantees that both endpoints of an edge are not put into the independent set simultaneously.  $\square$

Hence, we cannot even hope to solve  $\text{CSP}(\Gamma)$  in  $O(c^{|V|})$  time for any  $c$  when  $\Gamma$  is allowed to be infinite. Furthermore, since an equality CSP is always solvable in  $2^{O(|V| \log |V|)}$  time [Jonsson and Lagerkvist, 2017], the bound in Theorem 6 is asymptotically tight. Last, let us consider a problem which is related to equality CSPs, for which we rather effortlessly can obtain lower bounds by reducing from  $\text{CSP}(\Gamma_{\text{inf}})$ . *Satisfiability modulo theories* is a decision problem for logical formulas with respect to combinations of background theories expressed in classical first-order logic with equality. Let  $\text{SMT}(\mathcal{T})$  be the problem of determining whether a first-order formula (with respect to a background theory  $\mathcal{T}$ ) is satisfiable, and let  $\text{SMT}_{\forall}(\mathcal{T})$  be the subproblem where universal quantifiers are not allowed. We can then readily prove a matching lower bound valid for any background theory  $\mathcal{T}$ .

**Lemma 1.**  *$\text{SMT}_{\forall}(\emptyset)$  cannot be solved in  $2^{o(|V| \log |V|)}$  time unless the ETH is false.*

*Proof.* We present a polynomial-time reduction from  $\text{CSP}(\Gamma_{\text{inf}})$  which does not introduce any fresh variables. Let  $(V, C)$  be an instance of  $\text{CSP}(\Gamma_{\text{inf}})$ , where  $V = \{x_1, \dots, x_k\}$  and  $C = \{c_1, \dots, c_p\}$ . Define  $F = \exists x_1 \dots \exists x_k: F_1 \wedge \dots \wedge F_p$  where  $F_i = (\neg(x = y))$  if  $c_i = x \neq y$ ,  $F_i = (y = x_1 \vee y = x_2 \vee \dots \vee y = x_n)$  if  $c_i = R_n(y, x_1, \dots, x_n)$ , and  $F_i = (\neg(x = y) \vee \neg(z = w))$  if  $c_i = S(x, y, z, w)$ . It is obvious that  $F$  is true if and only if  $(V, C)$  has a solution, that  $F$  can easily be constructed in polynomial time, and that  $F$  contains as many variables as there are variables in  $V$ . The result then follows from Theorem 6.  $\square$

To exemplify this, we consider the well-known *unit two variable per inequality* (UTVPI) class of constraints, i.e.,  $\text{SMT}_{\forall}(\text{UTVPI})$  where UTVPI for each integer  $b$  and coefficients  $c_1, c_2 \in \{-1, 1\}$  contains  $c_1 \cdot x + c_2 \cdot y \geq b$ . The UTVPI class has many applications in, for instance, abstract interpretation, spatial databases, and theorem proving (cf. Schutt and Stuckey [2010] and the references therein.) It is known [Seshia *et al.*, 2007] that  $\text{SMT}_{\forall}(\text{UTVPI})$  can be solved in  $2^{O(|V| \log d)}$  time where  $d = 2|V|(b_{\max} + 1) + 1$  and  $b_{\max}$  is the maximum over the absolute values of constant terms in the constraints. Using Lemma 1 we can prove that this algorithm is close to optimal.

**Theorem 7.**  *$\text{SMT}$  over UTVPI constraints,  $\text{SMT}_{\forall}(\text{UTVPI})$ , cannot be solved in  $2^{o(n \log d)}$  time unless the ETH is false.*

*Proof.* Assume there is an algorithm  $A$  that solves  $\text{SMT}_{\forall}(\text{UTVPI})$  in  $2^{o(|V| \log d)}$  time. The formulas constructed in Lemma 6 are  $\text{SMT}_{\forall}(\text{UTVPI})$  formulas (degenerated ones, though, since they do not contain UTVPI constraints). Thus,  $b_{\max}$  for this class  $X$  of formulas is 0, implying that  $A$  can solve  $\text{SMT}_{\forall}(\text{UTVPI})$  restricted to  $X$  in  $2^{o(n \log n)}$  time, contradicting Lemma 1.  $\square$

### 3.2 No Easiest NP-Hard Infinite-Domain CSP

Our lower bounds suggest that equality CSPs are rather different from finite-domain CSPs when viewed under the lens of fine-grained complexity. In this section we prove yet another differentiating factor. For each finite  $A$  it is known that there exists  $\Gamma_A$  such that  $\text{CSP}(\Gamma_A)$  is NP-complete, but if an

NP-complete  $\text{CSP}(\Delta)^1$  over  $A$  is solvable in  $O(c^{|V|})$  time, then  $\text{CSP}(\Gamma_A)$  is solvable in  $O(c^{|V|})$  time, too [Jonsson *et al.*, 2017b]. More generally, if  $\mathcal{G}$  is a set of constraint languages over  $A$ , say that  $\text{CSP}(\Gamma)$  for  $\Gamma \in \mathcal{G}$  is the *easiest CSP problem in  $\mathcal{G}$*  if  $\text{CSP}(\Gamma)$  is solvable in  $O^*(c^{|V|})$  time whenever  $\text{CSP}(\Delta)$  for  $\Delta \in \mathcal{G}$  is solvable in  $O^*(c^{|V|})$  time.

Contrary to the finite-domain case we will prove that there does *not* exist an easiest NP-complete equality CSP, unless the ETH is false. To prove this we for every  $c > 1$  show the existence of an equality language  $\Gamma_c$  such that  $\text{CSP}(\Gamma_c)$  is NP-complete but solvable in  $O(c^{|V|})$  time. First, recall from Example 1 that the ternary relation  $S = \{(x, x, y), (x, y, y) \mid x, y \in \mathbb{N}, x \neq y\}$  has an NP-complete CSP. We will show how  $S$  can be extended with additional arguments in order to decrease the time complexity of the resulting CSP. If  $\mathbf{v} = (v_1, \dots, v_k)$  and  $\mathbf{w} = (w_1, \dots, w_k)$  are two  $k$ -ary tuples of variables,  $x$  a variable, and  $R$  is a binary relation, then we write  $R(x, \mathbf{v})$  for  $\bigwedge_{1 \leq i \leq k} R(x, v_i)$ ,  $R(\mathbf{v}, \mathbf{w})$  for  $\bigwedge_{1 \leq i, j \leq k} R(v_i, w_j)$ , and  $R(\mathbf{v})$  for  $\bigwedge_{1 \leq i, j \leq k, i \neq j} R(v_i, v_j)$ . For each  $k \geq 1$  now define  $S^k(x, y, z, \mathbf{v}, \mathbf{w})$  as  $\bigwedge_{s \in \{x, y, z\}, t \in \{\mathbf{v}, \mathbf{w}\}} R_{\neq}(s, t) \wedge R_{\neq}(\mathbf{v}, \mathbf{w}) \wedge (x = y \wedge y \neq z \wedge \text{Eq}(\mathbf{v}) \wedge R_{\neq}(\mathbf{w})) \vee (x \neq y \wedge y = z \wedge R_{\neq}(\mathbf{v}) \wedge \text{Eq}(\mathbf{w}))$  where  $\mathbf{v} = (v_1, \dots, v_k)$  and  $\mathbf{w} = (w_1, \dots, w_k)$  are two distinct  $k$ -ary tuples of variables. The problem  $\text{CSP}(\{S^k\})$  is clearly NP-complete since  $S \in \{S^k\}$ , and we will now prove that the fine-grained complexity of  $\text{CSP}(\{S^k\})$  decreases, in the following sense.

**Theorem 8.** *Let  $c > 1$ . Then there exists a  $k$  such that  $\text{CSP}(\{S^k\})$  is solvable in  $O^*(c^{|V|})$  time.*

*Proof.* We will present an algorithm for  $\text{CSP}(\{S^k\})$  which runs in  $O^*(2^{\frac{n}{k}})$  time. The claim then follows when choosing sufficiently large  $k \geq \frac{1}{\log c}$ . Thus, choose  $k \geq 1$  and let  $(V, C)$  be an instance of  $\text{CSP}(\{S^k\})$ , where  $|V| = n$ . Say that a set of inequality constraints  $L$  is *consistent* if  $L$ , viewed as an instance of  $\text{CSP}(\{R_{\neq}\})$ , is satisfiable, and *inconsistent* otherwise. The consistency of a set of inequality constraints can be determined in polynomial time since  $\text{CSP}(\{R_{\neq}\})$  is in P (from Example 1). Consider the following algorithm for  $\text{CSP}(\{S^k\})$ , where the set  $L$  is used to keep track of inequality constraints induced by the constraints in the instance.

1. Let  $L = \emptyset$ .
2. If  $L$  is inconsistent, return no.
3. If  $L$  is consistent and  $C = \emptyset$ , return yes.
4. Pick a constraint  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i) \in C$  where  $\mathbf{v}_i = (v_i^1, \dots, v_i^k)$  and  $\mathbf{w}_i = (w_i^1, \dots, w_i^k)$ .
5. Return no if:
  - (a)  $|\{x_i, y_i, z_i\}| = 1$ ,
  - (b)  $\{x_i, y_i, z_i\} \cap \{v_i^1, \dots, v_i^k, w_i^1, \dots, w_i^k\} \neq \emptyset$ ,
  - (c)  $\{v_i^1, \dots, v_i^k\} \cap \{w_i^1, \dots, w_i^k\} \neq \emptyset$ , or if
  - (d)  $|\{v_i^1, \dots, v_i^k\}| < k$  and  $|\{w_i^1, \dots, w_i^k\}| < k$ .

<sup>1</sup>For technical reasons  $\Delta$  contains all unary relations over  $A$ .

6. If  $|\{v_i^1, \dots, v_i^k\}| < k$  and  $|\{w_i^1, \dots, w_i^k\}| = k$  then we identify  $y_i$  with  $x_i$ ,  $v_i^1$  with every variable in  $\{v_i^1, \dots, v_i^k\} \setminus \{v_i^1\}$ , remove  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i)$ , add  $R_{\neq}(\mathbf{w}_i)$ ,  $R_{\neq}(x_i, \mathbf{v}_i)$ ,  $R_{\neq}(x_i, \mathbf{w}_i)$ ,  $R_{\neq}(z_i, \mathbf{v}_i)$ ,  $R_{\neq}(z_i, \mathbf{w}_i)$ ,  $R_{\neq}(x_i, z_i)$  to  $L$ , and jump to step (2).
7. The case  $|\{v_i^1, \dots, v_i^k\}| = k$ ,  $|\{w_i^1, \dots, w_i^k\}| < k$ , is handled analogously.
8. If none of the above cases apply we proceed as follows.
  - (a) If  $|\{x_i, y_i, z_i\}| = 2$  then no branching is necessary, and depending on whether  $x_i = y_i$  or  $x_i \neq y_i$  we jump to step (b) or step (c) below.
  - (b) Identify  $y_i$  with  $x_i$ ,  $v_i^j$  ( $2 \leq j \leq k$ ) with  $v_i^1$ , add  $R_{\neq}(\mathbf{w}_i)$ ,  $R_{\neq}(x_i, \mathbf{v}_i)$ ,  $R_{\neq}(x_i, \mathbf{w}_i)$ ,  $R_{\neq}(z_i, \mathbf{v}_i)$ ,  $R_{\neq}(z_i, \mathbf{w}_i)$ , and  $R_{\neq}(x_i, z_i)$  to  $L$ , remove  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i)$ , and jump to step (2).
  - (c) Identify  $z_i$  with  $y_i$ ,  $w_i^j$  ( $2 \leq j \leq k$ ) with  $w_i^1$ , add  $R_{\neq}(\mathbf{v}_i)$ ,  $R_{\neq}(x_i, \mathbf{v}_i)$ ,  $R_{\neq}(x_i, \mathbf{w}_i)$ ,  $R_{\neq}(y_i, \mathbf{v}_i)$ ,  $R_{\neq}(y_i, \mathbf{w}_i)$ , and  $R_{\neq}(x_i, y_i)$  to  $L$ , remove  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i)$ , and jump to step (2).
9. Answer yes if any of the two recursive branches return yes, and otherwise no.

For correctness, the algorithm branches on a constraint  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i) \in C$ , and either identifies  $x_i$  with  $y_i$ , or  $y_i$  with  $z_i$ , and in the process identifies variables and introduces inequality constraints according to the definition of  $S^k$ . Furthermore, the algorithm answers 'yes' if and only if it for each constraint  $S^k(x_i, y_i, z_i, \mathbf{v}_i, \mathbf{w}_i) \in C$  is possible to identify  $x_i$  with  $y_i$ , or  $y_i$  with  $z_i$ , in a non-contradictory way, and answers 'no' if and only if this is not possible. Concerning time complexity, note first that all variables in  $\mathbf{v}_i$  and  $\mathbf{w}_i$  are distinct, once step (8) is reached. This follows from the tests undertaken in step 5 where we systematically verify that  $\{w_i^1, \dots, w_i^k\}$  and  $\{v_i^1, \dots, v_i^k\}$  are disjoint and that  $|\{w_i^1, \dots, w_i^k\}| = |\{v_i^1, \dots, v_i^k\}| = k$ . Furthermore, if (8)(b) or (8)(c) is reached then  $|\{x_i, y_i, z_i\}| = 3$ , as otherwise the current instance is unsatisfiable ( $|\{x_i, y_i, z_i\}| = 1$ ) or no branching was required ( $|\{x_i, y_i, z_i\}| = 2$ ). Thus, in each branch we eliminate  $k$  variables via variable identification, which implies that the time complexity is bounded by the recurrence  $T(n) = 2T(n - k) + \text{poly}(\|I\|)$ . Thus, the total running time is  $O^*(2^{\frac{n}{k}})$ , which solves  $\text{CSP}(\{S^k\})$  in  $O^*(c^n)$  time for sufficiently large  $k$ .  $\square$

We immediately obtain the following corollary.

**Corollary 1.** *Let  $\mathcal{A} = (A; R_1, \dots, R_k)$  be a relational structure over a countably infinite  $A$ , such that  $\text{CSP}(\Gamma)$  for a first-order reduct of  $\mathcal{A}$  is NP-complete if and only if  $\Gamma$  pp-interprets 3-SAT. Let  $\mathcal{G} = \{\Gamma \mid \Gamma \text{ is a first-order reduct of } \mathcal{A} \text{ and } \text{CSP}(\Gamma) \text{ is NP-complete}\}$ . If  $\mathcal{G}$  has an easiest CSP problem then the ETH is false.*

*Proof.* For each  $c > 1$  there exists a constraint language  $\Gamma_c \in \mathcal{G}$  such that  $\text{CSP}(\Gamma_c)$  is NP-complete and solvable in  $O^*(c^{|V|})$  time (Theorem 8). If  $\mathcal{G}$  has an easiest NP-complete problem  $\text{CSP}(\Gamma)$  then (1)  $\text{CSP}(\Gamma)$  pp-interprets 3-SAT, and (2)  $\text{CSP}(\Gamma)$  is solvable in  $O^*(c^{|V|})$  time for each  $c > 1$ . Thus,

CSP( $\Gamma$ ) is solvable in subexponential time, but this violates the ETH by Corollary 10 in [Jonsson *et al.*, 2017b].  $\square$

This captures the NP-hard cases of the CSP dichotomy conjecture over finitely bounded homogeneous structures [Barto and Pinsker, 2016].

#### 4 Upper Bounds for Equality CSPs

The lower bounds established in Section 3 suggest that we cannot construct an  $O(c^{|V|})$  time algorithm ( $c > 1$ ) which is applicable to arbitrary equality languages. However, if we fix a finite equality language  $\Gamma$ , this still leaves the possibility of constructing an  $O(c^{|V|})$  time algorithm for a constant  $c$  depending on  $\Gamma$ . We manage to construct a novel algorithm for CSP( $\Gamma$ ), where  $\Gamma$  is a finite equality language with maximum arity  $\alpha$ , with a running time of  $O^*\left(\left(\frac{\alpha(\alpha-1)}{2}\right)^{|V|}\right)$ . Thus, the algorithm runs in  $O^*(c^{|V|})$  time for a constant  $c$  depending on  $\Gamma$ , which is a significant improvement over the algorithm proposed by [Jonsson and Lagerkvist, 2017] which solves CSP( $\Gamma$ ) in  $O^*(2^{|V| \cdot \log\left(\frac{0.792|V|}{\ln(|V|+1)}\right)})$  time.

**Theorem 9.** *Let  $\Gamma$  be a finite equality language and let  $\alpha = \max\{ar(R) \mid R \in \Gamma\}$ . Then, CSP( $\Gamma$ ) can be solved in  $O^*\left(\left(\frac{\alpha(\alpha-1)}{2}\right)^{|V|}\right)$  time.*

*Proof.* Consider the following algorithm  $A(I)$  for an instance  $I$  of CSP( $\Gamma$ ).

1. Let  $I = (V, C)$  and let  $V = \{x_1, \dots, x_n\}$ .
2. Define  $s: V \rightarrow \{1, \dots, n\}$  such that  $s(x_i) = i$ .
3. If  $s$  is a solution to  $I$ , then return 'yes'.
4. If  $s$  is not a solution to  $I$  and  $|V| = 1$ , then return 'no'.
5. Arbitrarily choose a constraint  $R(x_{i_1}, \dots, x_{i_p})$  that is not satisfied by  $s$ .
6. For each  $1 \leq j < k \leq p$ , let  $I_{j,k}$  denote the instance obtained by identifying  $x_{i_j}$  with  $x_{i_k}$  in  $I$ .
  - (a) If  $A(I_{j,k}) = \text{'yes'}$  then return 'yes'.
7. Return 'no'.

We begin by proving correctness by induction over  $|V| = n$ . If  $n = 1$ , then the tests in steps (3) and (4) provide the correct answer. Assume the algorithm is correct when  $n = m > 1$ . Let  $I = (V, C)$  be an instance where  $|V| = n = m + 1$ . First, assume that  $I$  has an injective solution. Then it is readily verified that  $f: \{1, \dots, |V|\}$  defined as  $f(x_i) = i$  for each  $x_i \in V = \{x_1, \dots, x_{|V|}\}$ , is a solution to  $I$  as well (in technical terms this follows from the well-known fact that the automorphisms of  $\Gamma$  is the full symmetric group [Bodirsky and Kára, 2008]). Hence, the algorithm answers 'yes' via step (3). Otherwise  $I$  does not have an injective solution and at least one constraint  $c = R(x_{i_1}, \dots, x_{i_p}) \in C$  is not satisfied by the function  $s$ . This implies that (at least) two variables in  $\{x_{i_1}, \dots, x_{i_p}\}$  must be assigned the same value. This is systematically tested in step (6), and the correctness follows from the inductive hypothesis.

Concerning the time complexity, it is bounded from above by the recurrence  $T(n) = \frac{\alpha(\alpha-1)}{2} \cdot T(n-1) + \text{poly}(|I|)$

since  $i_p \leq \alpha$  for each possible choice of constraint  $R(x_{i_1}, \dots, x_{i_p})$ . Thus,  $T(n) \in O^*\left(\left(\frac{\alpha(\alpha-1)}{2}\right)^n\right)$ , and we get the desired bound on the time complexity.  $\square$

#### 5 Concluding Remarks

We have studied fine-grained complexity of infinite-domain equality CSPs, and have proven that this class of problems differ from finite-domain CSPs in almost every way conceivable. Despite the disarray of this complexity landscape, it is possible to outline several concrete future research directions. First, since we know that all finite equality languages can be solved in  $O(c^{|V|})$  time and that there exists infinite equality languages not solvable in  $O(c^{|V|})$  time for any  $c > 1$ , is it possible to prove a dichotomy separating the problems solvable in  $O(c^{|V|})$  time to those that are not?

More generally, which infinite-domain CSPs are solvable in  $O(c^{|V|})$  time? Is this strictly a property of finite first-order reducts (disregarding trivial examples of infinite language which can be constructed by adding an infinite number of relations which do not affect the complexity)? An interesting continuation is the class of *temporal CSPs*, i.e., CSPs over first-order reduct of  $(\mathbb{Q}; <)$ . Temporal languages are well-behaved from a model theoretic viewpoint ( $\omega$ -categorical), admit a dichotomy between P and NP-complete, and are always solvable in  $O^*(2^{|V| \log |V|})$  time, so one would expect similarities between equality CSPs and temporal CSPs when it comes to fine-grained complexity. Thus, which temporal CSPs are solvable in  $O(c^{|V|})$  time? Despite the aforementioned similarities there are still large differences to equality CSPs. For example, there exists a finite first-order reduct  $\Gamma$  of  $(\mathbb{Q}; <)$  such that CSP( $\Gamma$ ) is not solvable in  $2^{o(|V| \log |V|)}$  time without violating the r-ETH [Jonsson and Lagerkvist, 2018].

Last, we have seen that the class of NP-complete equality CSPs does not admit an "easiest problem" without violating the ETH, contrary to satisfiability problems [Jonsson *et al.*, 2017a] and finite-domain CSPs [Jonsson *et al.*, 2017b]. The existence of easiest problems of this form can be explained by so-called *weak bases*. We brush aside the technical definition and simply remark that a weak base is a constraint language  $\Gamma$  implying that CSP( $\Gamma$ ) is the easiest CSP problem with respect to the set of constraint languages that can pp-define (and be pp-defined by)  $\Gamma$ . A weak base  $\Gamma$  such that  $\langle \Gamma \rangle$  is the full first-order reduct E of  $(\mathbb{N}; \emptyset)$  would therefore be a great surprise since it, in the light of Theorem 8, would contradict the ETH. However, it is in fact possible to *unconditionally* prove that E does not have a weak base, using techniques from *partial clone theory*. In general, both positive and negative examples of infinite-domain weak bases exist [Romov, 2018], but the resulting CSPs are not of practical interest. Thus, does there exist  $\langle \Gamma \rangle$  over a countably infinite domain admitting a weak base, and where CSP( $\Gamma$ ) is NP-hard?

#### Acknowledgements

The authors are partially supported by the Swedish Research Council (VR) under grant 2017- 04112. In addition, Victor Lagerkvist is partially supported by the Swedish Research Council (VR) under grant 2019-03690.

## References

- [Barto and Pinsker, 2016] L. Barto and M. Pinsker. The algebraic dichotomy conjecture for infinite domain constraint satisfaction problems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2016)*, pages 615–622, New York, NY, USA, 2016. ACM.
- [Biere *et al.*, 2009] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [Bodirsky and Jonsson, 2017] M. Bodirsky and P. Jonsson. A model-theoretic view on qualitative constraint reasoning. *Journal of Artificial Intelligence Research*, 58:339–385, 2017.
- [Bodirsky and Kára, 2008] M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 43(2):136–158, 2008.
- [Bodirsky and Kára, 2010] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):9:1–9:41, 2010.
- [Bodirsky, 2012] M. Bodirsky. Complexity classification in infinite-domain constraint satisfaction. Mémoire d’habilitation à diriger des recherches, Université Diderot – Paris 7. Available at arXiv:1201.0856, 2012.
- [Bulatov, 2017] A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS-2017)*. IEEE Computer Society, 2017.
- [Cygan *et al.*, 2015] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition, 2015.
- [Dylla *et al.*, 2017] F. Dylla, J. H. Lee, T. Mossakowski, T. Schneider, A. Van Delden, J. Van De Ven, and D. Wolter. A survey of qualitative spatial and temporal calculi: Algebraic and computational properties. *ACM Computing Surveys*, 50(1):7:1–7:39, April 2017.
- [Jonsson and Lagerkvist, 2017] P. Jonsson and V. Lagerkvist. An initial study of time complexity in infinite-domain constraint satisfaction. *Artificial Intelligence*, 245:115–133, 2017.
- [Jonsson and Lagerkvist, 2018] P. Jonsson and V. Lagerkvist. Why are CSPs based on partition schemes computationally hard? In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS-2018)*, pages 43:1–43:15, 2018.
- [Jonsson *et al.*, 2017a] P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Strong partial clones and the time complexity of SAT problems. *Journal of Computer and System Sciences*, 84:52 – 78, 2017.
- [Jonsson *et al.*, 2017b] P. Jonsson, V. Lagerkvist, and B. Roy. Time complexity of constraint satisfaction via universal algebra. In *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS-2017)*, pages 17:1–17:15, 2017.
- [Krokhin *et al.*, 2003] A. Krokhin, P. Jeavons, and P. Jonsson. Reasoning about temporal relations: The tractable subalgebras of Allen’s interval algebra. *Journal of the ACM*, 50(5):591–640, September 2003.
- [Lokshtanov *et al.*, 2018] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. *SIAM Journal on Computing*, 47(3):675–702, 2018.
- [Renz and Nebel, 1999] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artificial Intelligence*, 108(1-2):69–123, 1999.
- [Romov, 2018] B. A. Romov. Endpoints of associated intervals for local clones on an infinite set. *Algebra universalis*, 79(4):82, Oct 2018.
- [Schutt and Stuckey, 2010] A. Schutt and P. J. Stuckey. Incremental satisfiability and implication for UTVPI constraints. *INFORMS Journal on Computing*, 22(4):514–527, 2010.
- [Seshia *et al.*, 2007] S. A. Seshia, K. Subramani, and R. E. Bryant. On solving boolean combinations of UTVPI constraints. *Journal on Satisfiability, Boolean Modeling and Computation*, 3(1-2):67–90, 2007.
- [Traxler, 2008] P. Traxler. The time complexity of constraint satisfaction. In *Proceeding of the Third International Workshop on Parameterized and Exact Computation (IWPEC-2008)*, pages 190–201, 2008.
- [Zhuk, 2017] D. Zhuk. The proof of CSP dichotomy conjecture. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS-2017)*. IEEE Computer Society, 2017.