ROBUST SATISFIABILITY FOR CSPS: HARDNESS AND ALGORITHMIC RESULTS

VÍCTOR DALMAU AND ANDREI KROKHIN

Abstract. An algorithm for a constraint satisfaction problem is called robust if it outputs an assignment satisfying at least a $(1-f(\epsilon))$-fraction of constraints for each $(1-\epsilon)$-satisfiable instance (i.e. such that at most a ϵ-fraction of constraints needs to be removed to make the instance satisfiable), where $f(\epsilon) \to 0$ as $\epsilon \to 0$. We establish an algebraic framework for analyzing constraint satisfaction problems admitting an efficient robust algorithm with functions f of a given growth rate. We use this framework to derive hardness results. We also describe three classes of problems admitting an efficient robust algorithm such that f is $O(1/\log(1/\epsilon))$, $O(\epsilon^{1/k})$ for some $k > 1$, and $O(\epsilon)$, respectively. Finally, we give a complete classification of robust satisfiability with a given f for the Boolean case.

1. Introduction

The constraint satisfaction problem (CSP) provides a framework in which it is possible to express, in a natural way, many combinatorial problems encountered in computer science and AI [17, 18, 25]. An instance of the CSP consists of a set of variables, a domain of values, and a set of constraints on combinations of values that can be taken by certain subsets of variables. The aim is then to find an assignment of values to the variables that satisfies the constraints (decision version) or that satisfies the maximum number of constraints (optimization version).

Since the CSP is NP-hard in full generality, a major line of research in CSP tries to identify the tractable cases of such problems (see [18, 19]), the primary motivation being the general picture rather than specific applications. The two main ingredients of a constraint are (a) variables to which it is applied and (b) relations specifying the allowed combinations of values or the costs for all combinations. Therefore, the main types of restrictions on CSP are (a) *structural* where the hypergraph formed by sets of variables appearing in individual constraints is restricted [28, 45], and (b) *language-based* where the constraint language Γ, i.e. the set of relations that can appear in constraints, is fixed (see, e.g. [10, 17, 18, 25]); the corresponding problem is denoted by CSP(Γ). The language-based direction is considerably more active than the structural one, and there are many partial language-based complexity classification results, e.g. [3, 4, 7, 9, 18, 23, 35, 36], but many central questions are still open.

The use of approximation algorithms is one of the most fruitful approaches to coping with NP-hard optimization problems. The CSP has always played an important role in the study of approximability. For example, the famous PCP theorem has an equivalent reformulation in terms of inapproximability of a certain CSP(Γ), see [1]; moreover, the recent combinatorial proof of this theorem [24] deals entirely with CSPs. The first optimal inapproximability results [32] by Håstad were about problems CSP(Γ), and they led to the study of a new hardness notion called approximation resistance [33], which, intuitively, means that a problem cannot be approximated better than by just picking a random assignment, even on almost satisfiable instances. Arguably, the most exciting development in approximability in the past five to six years is the work around the Unique Games Conjecture (UGC) of Khot, see survey [38]. The UGC states that it is NP-hard to tell almost satisfiable instances of CSP(Γ) from those where only a small fraction of constraints can be satisfied, where Γ is the constraint language consisting of all graphs of permutations over a large enough domain. This conjecture (if true) is known to imply optimal inapproximability results for many classical optimization problems [38]. Moreover, if the UGC is true then a simple algorithm based on semidefinite programming (SDP) provides the best possible approximation for all optimization problems CSP(Γ) [47], though the exact quality of this approximation is unknown. There is, however, no unanimity as to which way the UGC will be resolved [2]. A common theme in these results is the focus on *almost satisfiable* instances, i.e. those where a tiny fraction of constraints can be removed to make the remaining instance satisfiable. The approximability of CSPs restricted to such instances has been actively studied, see references in [38], also [15, 30, 31, 51]; this additional restriction may change the approximability of a problem. Most, but not all, algorithms used in this line of research are based on LP (linear programming) or SDP, and analytic methods are used to study them.

A polynomial-time algorithm for CSP(Γ) would, in general, treat all unsatisfiable instances the same. When can such an algorithm be made to also identify near-misses, i.e. almost satisfiable instances? There is a line of research aimed at identifying tractable optimization problems CSP(Γ), i.e. those where an optimal assignment can always be found in polynomial time [16], and this property is known to be quite restrictive [23, 35, 36]. The following natural notion of tractability, which is stronger than classical tractability of CSP(Γ), but much less restrictive than tractability of optimization version of CSP(Γ), was suggested in [51]. Call CSP(Γ) *robustly solvable* if there is a polynomial-time algorithm which, for every $\epsilon > 0$ and every $(1-\epsilon)$-satisfiable instance of CSP(Γ) (i.e. at most a ϵ-fraction of constraints can be removed to make the instance satisfiable), outputs a $(1-f(\epsilon))$-satisfying assignment (i.e. that fails to satisfy at most a $f(\epsilon)$-fraction of constraints) where f is a function such that $f(\epsilon) \to 0$ as $\epsilon \to 0$ and $f(0) = 0$. Note that the running time of the algorithm should not depend on ϵ. Thus, robust solvability combines, in a natural way, tractability and approximation for CSPs.

The main goal of this paper is to study robust algorithms for problems CSP(Γ). Two very recent papers [4, 41] study the same topic. In fact, some of our results, (hardness) Theorem 9 and (positive) Theorem 16, were announced simultaneously with [41] where Theorem 16 is proved independently in a different way. Our Theorem 9(1) describes problems CSP(Γ) that cannot have an efficient robust algorithm unless P = NP. Predicting this theorem, Guruswami and Zhou conjectured [31] that all other problems do admit an efficient robust algorithm, Theorem 16 was a partial confirmation of the conjecture. Soon after our Theorem 9 and Theorem 16 were announced, Barto and Kozik fully confirmed the conjecture in [4]. The function $f(\epsilon)$ in [4] is $O(\log\log(1/\epsilon)/\log(1/\epsilon))$ for the randomized algorithm and $O(\log\log(1/\epsilon)/\sqrt{\log(1/\epsilon)})$ for its derandomization, thus one can naturally ask which problems CSP(Γ) have efficient robust algorithms with better functions f.

Our results in this direction, Theorems 16, 17 and 18 contribute towards answering this question within a well known class of CSPs, CSPs of width 1. The last two theorems are obtained after, and influenced by, results from [4, 41].

Recent breakthroughs in the study of the complexity of CSP have been made possible by the introduction of the universal-algebraic approach (see [10, 17]), which extracts algebraic structure from a given constraint language Γ (via operations called *polymorphisms* of Γ) and uses it to analyse problem instances. More precisely, Γ is associated a finite universal algebra $\mathbb{A}$, whose operations are the polymorphisms of Γ, such that the complexity of CSP(Γ) (and some other important features) is determined solely by the properties of $\mathbb{A}$. This approach is usually used with the following pattern: a property is identified, often in terms of operations with specific identities, such that either $\mathbb{A}$ fails this property and then CSP(Γ) can simulate some simple problem(s) with undesirable attributes (e.g. intractable or not robustly solvable), or else $\mathbb{A}$ has the property, that often comes in several equivalent forms, which is then used to analyze problem instances and design required algorithms. Note that every single step in the above description usually requires non-trivial work. We adapt the universal-algebraic framework to study robust algorithms in Section 3. We hope that the algebraic approach will become just as fruitful for the study of robust satisfiability as it has been for the study of decision CSPs.

Establishing local consistency is one of the most natural algorithms for dealing with (decision) CSPs. The basic idea is to inspect a given instance locally, deriving new constraints according to the currently observed part of the instance, until no new constraints can be derived. Then either a contradiction is derived or else local consistency is established (which in general does not imply the existence of a solution). Under additional assumptions on Γ, the latter does imply the existence of a solution. These additional assumptions can often be expressed in terms of polymorphisms [11, 13, 20, 25]. There are many sorts of local consistency that have been studied in the literature, which use various rules for deriving new constraints. One nice way to formalize the fact that some form of local consistency correctly solves a CSP(Γ) is via homomorphism dualities, and we use this approach in the present paper (see Section 4). We use algebraic characterizations of some dualities to design robust approximation algorithms for CSP(Γ) in Section 5. For a given almost satisfiable instance, the algorithms seek to remove a small fraction of constraints to achieve some form of local consistency, thus obtaining an assignment satisfying the remaining constraints.

Finally, in Section 6, we use our results together with some earlier results to complete the picture of robust satisfiability in the Boolean (i.e. two-valued) case: for each Γ we describe the best possible function f, modulo complexity-theoretic assumptions.

2. Preliminaries

Let A be a finite set. A k-ary tuple $\mathbf{a} = (a_1, \dots, a_k)$ is any element of A^k. For $1 \leq i \leq k$, we shall use $\mathbf{a}_i$ to denote the ith element a_i of $\mathbf{a}$. A k-ary relation on A is a collection of k-ary tuples or, alternatively, a subset of A^k. We shall use $\rho(R)$ to denote the arity of relation R. For any $1 \leq i \leq k$, the *projection* of R to the ith coordinate $\mathrm{pr}_i(R) \subseteq A$ is defined as $\mathrm{pr}_i(R) = \{\mathbf{a}_i \mid \mathbf{a} \in R\}$.

An *instance* of the CSP is a triple $I = (V, A, \mathcal{C})$ with V a finite set of *variables*, A a finite set called *domain*, and $\mathcal{C}$ a finite list of *constraints*, where each constraint is a pair $C = (\mathbf{v}, R)$ where $\mathbf{v} = (v_1, \ldots, v_k)$ is a tuple of variables of length k, called the *scope* of C, and R an k-ary relation on D, called the *constraint relation* of C. The *arity* of a constraint C, $\rho(C)$, is defined to be arity of its constraint relation.

Note that we allow repetition of constraints in $\mathcal{C}$. Very often we will say that a constraint C belongs to instance I when, strictly speaking, we should be saying that appears in the constraint list $\mathcal{C}$ of I. Also, we might sometimes write $(v_1, \ldots, v_k, R)$ instead of $((v_1, \ldots, v_k), R)$. A finite set of relations Γ on a finite set A is called a *constraint language*. The problem CSP(Γ) consists of all instances of the CSP where all the constraint relations are from Γ. An assignment for I is a mapping $s : V \to A$. We say that s *satisfies* a constraint $(\mathbf{v}, R)$ if $s(\mathbf{v}) \in R$ (where s is applied component-wise). For $0 \leq \alpha \leq 1$, we say that assignment s α-*satisfies* I if it satisfies at least α-fraction of the constraints in I. In this case, we say that I is α-satisfiable.

The *decision problem* for CSP(Γ) asks whether an input instance I of CSP(Γ) has a solution, i.e., an assignment satisfying all constraints. The *optimization problem* for CSP(Γ) asks to find an assignment that satisfies the maximum number of constraints. The maximization problem is computationally intractable for the vast majority of constraint languages Γ motivating the study of *approximation* algorithms.

Let Γ be a constraint language and let ALG be an algorithm that receives as input an instance of CSP(Γ) and returns an assignment for its input. For real numbers $0 \leq \alpha, \beta \leq 1$ we say that ALG (α, β)-*approximates* CSP(Γ) if whenever it receives an α-satisfiable input instance I it returns an assignment that β-satisfies I.

Let $f : [0, 1] \to [0, \infty)$ be an *error* function with $f(\epsilon) \to 0$ as $\epsilon \to 0$ and $f(0) = 0$. If ALG $(1 - \epsilon, 1 - f(\epsilon))$-approximates CSP($\Gamma$) for every $\epsilon \geq 0$ then we say that ALG *robustly solves* CSP(Γ). Furthermore, if $f(\epsilon) = O(\epsilon^{1/k})$ for some $k \geq 1$ then we say that ALG *robustly solves* CSP(Γ) *with polynomial loss*.

If $\mathcal{G}$ is a finite Abelian group we denote by 3EQ-LIN($\mathcal{G}$) the constraint language over the base set of $\mathcal{G}$ that contains all linear equations over $\mathcal{G}$ with 3 variables. As a consequence of the following theorem by Håstad, 3EQ-LIN($\mathcal{G}$) is not robustly solvable if $\mathcal{G}$ has more than one element.

Theorem 1. [32] *If $\mathcal{G}$ is an Abelian group with $d > 1$ elements then for every $\epsilon > 0$ there is no polynomial-time algorithm that $(1 - \epsilon, 1/d + \epsilon)$-approximates* CSP(3EQ-LIN($\mathcal{G}$)) *unless* P = NP.

Local consistency is a powerful family of algorithms used in the decision problem for CSP(Γ). For fixed integers $0 \leq j \leq k$, the (j, k)-*consistency* algorithm derives constraints on j variables which can be deduced by looking at k variables at a time. The algorithm finishes after a polynomial number of steps. During this process, the algorithm might generate a contradiction, that is, a constraint with empty constraint relation meaning that the instance has no solution. Since CSP is NP-complete one cannot expect that the converse always holds. We say that CSP(Γ) has *width* (j, k) if an instance has a solution if and only if the (j, k)-consistency algorithm does not derive a contradiction. Finally, we say that CSP(Γ) has *width* j if it has width (j, k) for some $j \leq k$ and that CSP(Γ) has *bounded width* if it has width j for some $j \geq 0$. We shall give a precise, alternative characterization of bounded width CSPs in Section 2.1. The power of (j, k)-consistency is, by now, very well understood due to the results of Barto and Kozik [3], building upon [44] (see Theorem 6 below) and Bulatov [8].

Guruswami and Zhou conjectured the following connection between bounded width and approximation:

Conjecture 1. *(Guruswami, Zhou [31]) For every constraint language Γ, $\mathrm{CSP}(\Gamma)$ has bounded width if and only if it is robustly solvable.*

The 'only if' direction of the conjecture follows with just a little bit of work from known results. We prove it in Section 3. The 'if' part is much more difficult. In Section 5 we give a proof for the case of width 1 CPSs. This result has been obtained independently by Kun et al [41]. Later, Barto and Kozik presented a proof for all bounded width CSPs, settling the Guruswami-Zhou conjecture.

Theorem 2. [4] *For every constraint language Γ, if $\mathrm{CSP}(\Gamma)$ has bounded width then it is robustly solvable.*

In this paper, we are interested in a more fine-grained analysis of robust approximation that takes into consideration the quantitative dependence of f on ϵ (linear loss $O(\epsilon)$, quadratic loss $O(\epsilon^{1/2})$, etc.). To investigate it, we introduce the notation $\mathrm{CSP}(\Gamma) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma')$ as a shortand for: *for any error function f with $\lim_{\epsilon\to 0} f(\epsilon) = 0$, if some algorithm $(1-\epsilon, 1-f(\epsilon))$-approximates $\mathrm{CSP}(\Gamma')$ for every $\epsilon \geq 0$ then there is a polynomial-time algorithm that $(1-\epsilon, 1-O(f(\epsilon)))$-approximates $\mathrm{CSP}(\Gamma)$ for every $\epsilon \geq 0$.* Note that the relation $\leq_{\mathrm{RA}}$ is transitive.

We need a few concepts from propositional logic. A clause is Horn (respectively, dual Horn) if it contains at most one positive (respectively, one negative) literal. Let k-HORN (resp. k-DualHORN) be the constraint language over the Boolean domain that contains all Horn (dual Horn) clauses with at most k variables, and let 2-SAT be the constraint language over the Boolean domain containg all clauses with at most 2 literals. Let $\neq_2$ be the boolean relation $\{(0,1),(1,0)\}$.

The next theorem uses Khot's Unique Games (UG) conjecture [37]. This conjecture states that, for any $\epsilon \geq 0$, there is a large enough number $k = k(\epsilon)$ such that it NP-hard to tell ϵ-satisfiable from $(1-\epsilon)$-satisfiable instances of $\mathrm{CSP}(\Gamma_k)$, where Γ_k consists of all graphs of bijections on a k-element set.

Theorem 3. [51, 31, 15, 39] *Let $k \geq 1$. The following conditions hold:*

1. *There is a polynomial time algorithm that $(1-\epsilon, 1-O(1/\log(1/\epsilon)))$-approximates $\mathrm{CSP}(k\text{-HORN})$ for all $\epsilon \geq 0$.*
2. *If $k \geq 3$, there is no polynomial time algorithm that $(1-\epsilon, 1-o(1/\log(1/\epsilon)))$-approximates $\mathrm{CSP}(k\text{-HORN})$ for all $\epsilon \geq 0$ unless the UG conjecture is false.*
3. *There is a polynomial time algorithm that $(1-\epsilon, 1-O(\sqrt{\epsilon}))$-approximates $\mathrm{CSP}(2\text{-SAT})$ for all $\epsilon \geq 0$.*
4. *There is no polynomial time algorithm that $(1-\epsilon, 1-o(\sqrt{\epsilon}))$-approximates $\mathrm{CSP}(\{\neq_2\})$ for all $\epsilon \geq 0$ unless the UG conjecture is false.*

Conditions (1) and (2) obviously hold if we replace k-HORN by k-DualHORN.

For any instance $I = (V, A, \mathcal{C})$ of $\mathrm{CSP}(\Gamma)$, there is an equivalent canonical 0-1 integer program. It has variables $p_v(a)$ for every $v \in V$, $a \in A$, as well as variables $p_C(\mathbf{a})$ for every constraint $C = (\mathbf{v}, R)$ and every tuple $\mathbf{a} \in A^{\rho(R)}$. The interpretation of $p_v(a) = 1$ is that variable v is assigned value a; the interpretation of $p_C(\mathbf{a}) = 1$ is that $\mathbf{v}$ is assigned (component-wise) tuple $\mathbf{a}$. More formally, the program is the following:

maximize: $\displaystyle\frac{1}{|\mathcal{C}|}\sum_{C=(\mathbf{v},R)\in\mathcal{C}} p_C(R)$

subject to:

$$p_v(A) = 1 \qquad v \in V \quad (1)$$

$$p_C(A^{j-1}\times\{a\}\times A^{\rho(C)-j}) = p_{\mathbf{v}_j}(a) \quad C=(\mathbf{v},R)\in\mathcal{C}, 1\le j\le\rho(C), a\in A \quad (2)$$

where, for every $v \in V$ and $S \subseteq A$, $p_v(S)$ is a shorthand for $\sum_{a\in S} p_v(a)$ and for every C and every $T \subseteq A^{\rho(C)}$, $p_C(T)$ is a shorthand for $\sum_{\mathbf{a}\in T} p_C(\mathbf{a})$.

If we relax the previous program by allowing the variables to take values in the range $[0,1]$ instead of $\{0,1\}$, we obtain the *basic linear programming relaxation* for I, which we denote by $\mathrm{BLP}(I)$. As Γ is fixed, an optimal solution of $\mathrm{BLP}(I)$ can be computed in time polynomial in the representation size of I. Restriction (1) of $\mathrm{BLP}(I)$ expresses the fact that, for each $v \in V$, the quantities $p_v(a), a \in A$ form a discrete probability distribution on A. Also (1) and (2) together express the fact that, for each constraint $C = (\mathbf{v}, R)$, of arity k, the quantities $p_C(\mathbf{a}), \mathbf{a} \in A^k$ form a probability distribution on A^k and that the marginals of the p_C distribution are "consistent" with the p_v distributions.

2.1. **Algebra.** Most of the terminology introduced in this section is standard (see [12] for example). An n-ary *operation* on A f is a map from A^n to A.

Let us now define several types of operations that will be used in this paper.

- An operation f is *idempotent* if it satisfies the identity $f(x,\ldots,x) = x$.
- An n-ary operation f on A is *totally symmetric* if $f(x_1,\ldots,x_n) = f(y_1,\ldots,y_n)$ whenever $\{x_1,\ldots,x_n\} = \{y_1,\ldots,y_n\}$. It follows from this condition that we can properly write $f(S)$ for every $S \subseteq A$.
- An n-ary ($n \ge 3$) operation is a *NU (near-unanimity)* operation if it satisfies the identities

$$f(y,x,\ldots,x,x) = f(x,y,\ldots,x,x) = \cdots = f(x,x,\ldots,x,y) = x$$

- A ternary NU operation is called a *majority* operation.
- A binary idempotent commutative associative operation is called a *semilattice* operation.
- A pair of semilattice operations on A is a pair of *lattice* operations if, in addition, they satisfy the absorption identities: $f(x,g(x,y)) = g(x,f(x,y)) = x$. In this case (A,f,g) is called a *lattice*.

It is standard practice to use infix notation for lattice operations, i.e., to write $x\sqcup y$ and $x\sqcap y$ for $f(x,y)$ and $g(x,y)$ respectively. A lattice is said to be *distributive* if it satisfies the identity $x\sqcap(y\sqcup z) = (x\sqcap y)\sqcup(x\sqcap z)$. Equivalently, a lattice is distributive if it can be represented by a family of subsets of a set with the operations interpreted as set-theoretic intersection and union (see [29]).

An operation f *preserves* (or is a *polymorphism* of) a k-ary relation R if for every n and (not necessarily distinct) tuples $(a_1^i,\ldots,a_k^i) \in R$, $1 \le i \le n$, the tuple

$$(f(a_1^1,\ldots,a_1^n),\ldots,f(a_k^1,\ldots,a_k^n))$$

belongs to A as well. Given a set Γ of relations on A, we denote by $\mathrm{Pol}(\Gamma)$ the set of all operations that preserve all relations in Γ. If $f \in \mathrm{Pol}(\Gamma)$ then Γ is said to be *invariant* under f. If R is a relation we might freely write $\mathrm{Pol}(R)$ to denote

$\mathrm{Pol}(\{R\})$. If every unary operation in $\mathrm{Pol}(\Gamma)$ is one-to-one then Γ is said to be a *core*.

The cornerstone of the use of algebra in the exploration of constraint satisfaction is a theorem proven by Geiger and also by Bodnarchuk et al. [6, 27]. In order to state it, we need to introduce some definitions. Let Γ be a finite set of relations on A and let $R \subseteq A^k$. Let eq_A (eq, if A is clear from the context) the relation $\{(a,a) \mid a \in A\}$. We say that R is pp-definable from Γ if there exists a (primitive positive) formula

$$\phi(x_1,\dots,x_k) \equiv \exists y_1,\dots,y_l \ \psi(x_1,\dots,x_k,y_1,\dots,y_l)$$

where ψ is a conjunction of atomic formulas with relations in Γ and eq_A such that for every $(a_1,\dots,a_k) \in A^k$

$$(a_1,\dots,a_k) \in R \text{ if and only if } \phi(a_1,\dots,a_k) \text{ holds.}$$

If ψ does not contain eq_A then we say that R is pp-definable from Γ *without equality*. Note that in the definition of primitive positive formulas we are slightly abusing notation by identifying a relation with its relation symbol.

A k-ary relation R is *irredundant* if for every two different coordinates $1 \leq i < j \leq k$, R contains a tuple $(a_1,\dots,a_k)$ with $a_i \neq a_j$.

Theorem 4. [6, 27] *Let Γ be a finite set of relations on A and let R be a relation on A. Then the following holds.*

(1) $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(R)$ *if and only if R is pp-definable from Γ.*
(2) *if R is irredundant and* $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(R)$ *then R pp-definable from Γ without equality.*

An *algebra* is an ordered pair $\mathbb{A} = (A, F)$ where A is a non-empty set, called the *universe* of $\mathbb{A}$, and F is a set of finitary operations on A, called the *basic operations* of $\mathbb{A}$. If Γ is a set of relations on A, the *algebra associated to* Γ is the algebra $(A, \mathrm{Pol}(\Gamma))$. Throughout the paper we use the same capital letters (with different font) to denote a structure and its universe.

The *term operations* of an algebra are the operations that can be built from its basic operations using composition and projections. The *full idempotent reduct* of an algebra $\mathbb{A}$ is the algebra with the same universe of $\mathbb{A}$ and whose basic operations are the idempotent term operations of $\mathbb{A}$. For the purposes of this paper it is only necessary to know that the full idempotent reduct of the algebra associated to Γ has as basic operations all the idempotent operations that preserve Γ.

There are some standard ways to assemble new algebras from those already at hand. The most standard ones are the formation of *subalgebras*, *direct products*, and *homomorphic images*, which are defined in a natural way. A class of algebras is a variety if it is closed under formation of homomorphic images (H), subalgebras (S) and direct products (P). The variety generated by $\mathbb{A}$ is denoted by $\mathcal{V}(\mathbb{A})$; it is known that $\mathcal{V}(\mathbb{A}) = \mathrm{HSP}(\mathbb{A})$, i.e. that every member $\mathbb{C}$ of the $\mathcal{V}(\mathbb{A})$ is obtained as a homomorphic image of a subalgebra of a power of $\mathbb{A}$; furthermore this power can be taken to be finite if $\mathbb{C}$ is finite.

A set Γ of finite relations on A is *compatible* with $\mathbb{A}$ if every relation in Γ is preserved by every basic operation in $\mathbb{A}$.

Tame Congruence Theory, developed by Hobby and McKenzie [34], is a powerful tool to analyze finite algebras. Every algebra can be assigned a subset of five *types* that correspond to different possible "local behaviours" of the algebra. The possible types are: (1) the *unary* type, (2) the *affine* type, (3) the *Boolean* type, (4) the *lattice* type, and (5) the *semilattice* type. We use Tame Congruence Theory as a black box to link existing results and we do not require a precise definition of types. A variety is said to *admit* a type if this type occurs in some finite algebra in the variety; otherwise, the variety *omits* the type.

The following result follows from [49] and [50] (see [42])

Theorem 5. *Let $\mathbb{A}$ be a finite idempotent algebra.*

(1) *If $\mathcal{V}(\mathbb{A})$ admits the unary or affine types then there exists an algebra $\mathbb{B}$ in* $\mathrm{HS}(\mathbb{A})$ *with more than one element and an Abelian group structure $\mathcal{G}$ on the base set, B, of $\mathbb{B}$ such that every relation in* $3\mathrm{EQ\text{-}LIN}(\mathcal{G})$ *is compatible with $\mathbb{B}$.*

(2) *If $\mathcal{V}(\mathbb{A})$ omits the unary and affine types, but admits the semilattice type then there exists an algebra $\mathbb{B}$ in* $\mathrm{HS}(\mathbb{A})$ *whose universe is* $\{0, 1\}$ *and such that every relation in* 3-HORN *is compatible with $\mathbb{B}$.*

It turns out that for every core constraint language Γ, $\mathrm{CSP}(\Gamma)$ has bounded width if and only if its associated algebra fails the first condition of Theorem 5. The class of bounded width problems has also several characterizations in terms of the presence of certain operations in $\mathrm{Pol}(\Gamma)$. The following theorem (obtained combining results from [3, 26, 44]) provides one of them.

Theorem 6. *Let Γ be a finite set of relations on A such that Γ is a core. Then the following are equivalent:*

(1) $\mathcal{V}(\mathbb{A})$ *omits the unary or affine types;*

(2) $\mathrm{Pol}(\Gamma)$ *contains a 3-ary idempotent operation f and a 4-ary idempotent operation g such that for all $a, b \in A$,*

$$f(a, a, b) = f(a, b, a) = f(b, a, a) = g(a, a, a, b) = \cdots = g(b, a, a, a);$$

(3) $\mathrm{CSP}(\Gamma)$ *has bounded width.*

3. Algebraic reductions

Let Γ be any finite set of operations on A. We start by observing that if Γ is not a core then we can easily define another constraint language Γ' on a smaller domain such that $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(\Gamma')$ behave identically with respect to approximation. Indeed, let e be any non-surjective unary operation in $\mathrm{Pol}(\Gamma)$ and define Γ' to be $\{e(R) \mid R \in \Gamma\}$ where $e(R) = \{e(\mathbf{a}) \mid \mathbf{a} \in R\}$ (recall that e is applied componentwise). Since e is not surjective, the domain of Γ' is a proper subset of A. For every instance I of $\mathrm{CSP}(\Gamma)$, one can construct an 'equivalent' instance I' of $\mathrm{CSP}(\Gamma')$. Define I' to be the instance with the same set of variables as I and that contains, for every constraint $(\mathbf{v}, R)$ in I the constraint $(\mathbf{v}, e(R))$. Every assignment for I can be transformed into an assignment for I' satisfying the same number of constraints by composing it with e and, conversely, every assignment for I' can be transformed into an assignment for I by composing it with e (this is because, as e preserves Γ, $e(e(R)) \subseteq R$ for every $R \in \Gamma$). Hence, one can use any polynomial-time algorithm that approximates $\mathrm{CSP}(\Gamma)$ to obtain a polynomial-time algorithm that approximates $\mathrm{CSP}(\Gamma')$ with the same error function and vice versa.

This implies that if we want to explore the robust approximation of constraint satisfaction problems we only need to consider those constraint languages that are cores.

The algebraic-based approach to robust approximation relies on the next theorem. It says that the algebraic structure of the set of operations that preserve a core Γ characterizes in a very tight way how its associated constraint satisfaction problem, $\mathrm{CSP}(\Gamma)$, behaves with respect to robust approximation.

Theorem 7. *Let Γ be a finite set of relations on a finite set A such that Γ is a core. Let $\mathbb{A}$ denote the full idempotent reduct of the algebra associated to Γ. Let $\mathbb{C}$ be an algebra in $\mathcal{V}(\mathbb{A})$, and let Γ_0 be a finite set of relations invariant under the operations in $\mathbb{C}$. Then* $\mathrm{CSP}(\Gamma_0) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$ *whenever*

(1) $\mathrm{eq} \in \Gamma$ *or*
(2) $\mathbb{C} \in \mathrm{HS}(\mathbb{A})$ *and every relation in* Γ_0 *is irredundant.*

It seems plausible that, for every constraint language Γ, $\mathrm{CSP}(\Gamma \cup \{\mathrm{eq}\}) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$. If it is the case then Theorem 7 could be strengthened.

This section is devoted to the proof of Theorem 7. The proof is obtained via a chain of simple reductions. Both the proof structure and most of the arguments are fairly standard in the algebraic approach to CSP.

Lemma 1. *Let Γ be a finite set of relations on A and let R be a relation pp-definable from Γ without equality. Then* $\mathrm{CSP}(\Gamma \cup \{R\}) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$.

Proof. Let $\phi(x_1, \ldots, x_k)$ be a primitive positive formula defining R from Γ. Then, ϕ is of the form $\exists y_1, \ldots, y_l\ \psi(x_1, \ldots, x_k, y_1, \ldots, y_l)$ where ψ is the quantifier-free part of ϕ. The heart of the proof is the observation that ψ can be alternatively seen as an instance of $\mathrm{CSP}(\Gamma)$. More precisely, we define the instance associated to ψ, I_ψ, as the instance that has variables $x_1, \ldots, x_k, y_1, \ldots, y_l$ and contains for every atomic formula $S(v_1, \ldots, v_r)$ in ψ, the constraint $((v_1, \ldots, v_r), S)$. It follows that for any assignment $s : \{x_1, \ldots, x_k, y_1, \ldots, y_l\} \to A$, s is a solution of I_ψ if and only if $\psi(s(x_1), \ldots, s(x_k), s(y_1), \ldots, s(y_l))$ holds.

Let K be the number of atomic formulas in ψ. Assume that there is a polynomial-time algorithm ALG that $(1-\epsilon, 1-f(\epsilon))$-aproximates $\mathrm{CSP}(\Gamma)$ for all $\epsilon \geq 0$. We shall give a polynomial-time algorithm that $(1-\epsilon, 1-Kf(\epsilon))$-approximates $\mathrm{CSP}(\Gamma \cup \{R\})$ for all $\epsilon \geq 0$.

Let I be an instance of $\mathrm{CSP}(\Gamma \cup \{R\})$. Our algorithm starts by constructing in polynomial time an instance I' of $\mathrm{CSP}(\Gamma)$ 'equivalent' to I.

Initially place in instance I' K copies of every constraint in I whose constraint relation belongs to Γ. Then, for every constraint C of the form $((v_1, \ldots, v_k), R)$ (that is, whose constraint relation is R) in I, do the following: rename the variables of ψ such that for every $1 \leq i \leq k$, x_i becomes v_i and every y_j, $(1 \leq j \leq l)$ becomes a different fresh (i.e., not used in I') variable. We refer to this new formula (which is obviously logically equivalent to ψ) as ψ_C. Add to I' all constraints in the instance associated to ψ_C.

Then, run algorithm ALG with input I'. In polynomial time ALG will stop and report an assignment t'. Output the assignment t obtained by projecting t' to the variables of I.

Let us determine the quality of t. Assume that there is an assignment s for I that $(1-\epsilon)$-satisfies I. We claim that s can be extended to an assignment for I' that $(1-\epsilon)$-satisfies I'. Notice that every variable occurring in I' but not in I has been introduced when replacing a constraint C of the form $((v_1, \ldots, v_k), R)$ by the constraints in I_{ψ_C}. If s satisfies C then we can extend it over the fresh variables of I_{ψ_C} in such a way that all constraints in I_{ψ_C} are satisfied. If, otherwise, s does not satisfy C then just extend it over the fresh variables of I_{ψ_C} arbitrarily. Proceeding in this way for every such constraint, we produce a complete assignment for I' that we call s'. Since every constraint unsatisfied by s gives rise to at most K constraints unsatisfied by s' and the total number of constraints in I' is K times the total number of constraints in I, it follows that s' $(1-\epsilon)$-satisfies I' as we claimed.

The assignment returned by ALG, t', is guaranteed to $(1-f(\epsilon))$-satisfy I'. Every constraint unsatisfied by t' gives rise to at most one constraint unsatisfied by t. Since the total number of constraints in I' is at most K times the total number of constraints of I, it follows that assignment t $(1-Kf(\epsilon))$-satisfies I. ∎

As a byproduct of Lemma 1 we can state the following strengthened version of the hardness results in [32, 31, 39], involving only irredundant relations, which will be useful in our proofs.

Theorem 8. *(Hardness results of Theorems and 1 and 3, restated)*

1. *Let $\mathcal{G}$ be an Abelian group with more than one element and let Γ be the set of all irredundant relations in* 3EQ-LIN($\mathcal{G}$). *There is no polynomial-time algorithm that robustly solves* CSP(Γ) *unless* P = NP.
2. *Let Γ be the set containing relations $\{0\}$, $\{1\}$, and $\{(x,y,z) \mid x \wedge y \rightarrow z\}$. There is no polynomial-time algorithm that $(1-\epsilon, 1-o(1/\log(1/\epsilon)))$-aproximates* CSP($\Gamma$) *for all $\epsilon \geq 0$ unless the UG conjecture is false.*
3. *There is no polynomial-time algorithm that $(1-\epsilon, 1-o(\sqrt{\epsilon}))$-aproximates* CSP($\{\neq_2\}$)) *for all $\epsilon \geq 0$ unless the UG conjecture is false.*

Proof. (1) Follows from the fact that eq is pp-definable without equality from irredundant relations in Γ (for example with $\exists u, v\ (x+u+v=0) \wedge (y+u+v=0)$) and Theorem 1. (2) Follows from Theorem 3(2) and the well-known fact that one can pp-define without equality any Horn clause using the relations in Γ. (3) This is merely Theorem 3(4) which we restate here for convenience. ∎

If $\mu : B \rightarrow C$ is a surjective map and R is a k-ary relation on C, $\mu^{-1}(R)$ is defined to be the k-ary relation $\{\mathbf{b} \in B^k \mid \mu(\mathbf{b}) \in R\}$.

Lemma 2. *Let Γ_0 be a finite set of relations on C, let $\mu : B \rightarrow C$ be a surjective map, and let $\Gamma_1 = \{\mu^{-1}(R) \mid R \in \Gamma_0\}$. Then* $\mathrm{CSP}(\Gamma_0) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_1)$

Proof. This is straightforward. Let I_0 be any instance of CSP(Γ_0) with variables V and let I_1 be an instance of CSP(Γ_1) obtained by replacing every constraint relation $R \in \Gamma_0$ by $\mu^{-1}(R)$. Every assignment $s_1 : V \rightarrow B$ for I_1 can be transformed into an assignment s_0 for I_0 satisfying the same number of constraints by composing it with μ. Similarly, any assignment $s_0 : V \rightarrow C$ for I_0 can be transformed into an assignment s_1 for I_1 by setting $s_1(v)$ to be an arbitrary element in $\mu^{-1}(s_0(v))$ for every $v \in V$. It follows easily that one can use any polynomial-time algorithm that approximates CSP(Γ_1) to obtain a polynomial-time algorithm that approximates CSP(Γ_0) with the same error function. ∎

If R is a k-ary relation on A^m (not on A) then the *coordinatization* of R, coord(R), is the $(k \times m)$-ary relation on A

$$\mathrm{coord}(R) = \{((a_1^1, \ldots, a_1^m, \ldots, a_k^1, \ldots, a_k^m) \mid ((a_1^1, \ldots, a_1^m), \ldots, (a_k^1, \ldots, a_k^m)) \in R\}$$

Lemma 3. *Let Γ_1 be a finite set of relations in A^m and let $\Gamma_2 = \{\mathrm{coord}(R) \mid R \in \Gamma_1\}$. Then* $\mathrm{CSP}(\Gamma_1) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_2)$.

Proof. This is straightforward. Let I_1 be an instance of Γ_1 and let I_2 be an instance of Γ_2 defined in the following way. For every variable v of I_1, I_2 contains m variables $v^1, \dots, v^m$. Also, for every constraint $((v_1, \dots, v_k), R)$ in I_1, I_2 contains the constraint $((v_1^1, \dots, v_1^m, \dots, v_k^1, \dots, v_k^m), \mathrm{coord}(R))$. Every assignment s_2 of I_2 can be transformed into an assignment s_1 of I_1 satisfying the same number of constraints by just setting $s_1(v) = (s_2(v^1), \dots, s_2(v^m))$ for every variable v in I_1. Similarly any assignment s_2 of I_2 can be transformed (by reversing the previous transformation) to an assignment s_1 of I_1 satisfying again the same number of constraints. It follows easily that one can use any polynomial-time algorithm that approximates $\mathrm{CSP}(\Gamma_2)$ to obtain a polynomial-time algorithm that approximates $\mathrm{CSP}(\Gamma_1)$ with the same error function. ∎

The operators $\mu^{-1}(R)$ and $\mathrm{coord}(R)$ interact very nicely with the algebraic constructions in a variety. In particular, the following lemma follows directly from the definitions.

Lemma 4. *Let $\mathbb{A}$, $\mathbb{B}$ be algebras and let Γ be a finite set of relations on B compatible with $\mathbb{B}$. Then:*

(1) *If $\mathbb{A}$ is homomorphic to $\mathbb{B}$ via the surjective mapping $\mu : A \to B$ then $\mu^{-1}(\Gamma)$ is compatible with $\mathbb{A}$.*
(2) *If $\mathbb{B}$ is a subalgebra of $\mathbb{A}$ then Γ is compatible with $\mathbb{A}$.*
(3) *If $\mathbb{B} = \mathbb{A}^m$ then $\mathrm{coord}(\Gamma)$ is compatible with $\mathbb{A}$.*

Lemma 5. *Let Γ be a finite set of relations on $A = \{a_1, \dots, a_n\}$ such that Γ is a core. Then $\mathrm{CSP}(\Gamma \cup \{\{a_i\} \mid 1 \leq i \leq n\}) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$.*

Proof. We denote by $\mathrm{Aut}(\Gamma)$ the set of all unary operations in $\mathrm{Pol}(\Gamma)$ that are one-to-one. Every member of $\mathrm{Aut}(\Gamma)$ is said to be an *automorphism* of Γ.

Let ϕ be the (quanitfier-free) pp-formula with free variables $x_1, \dots, x_n$ defined as

$$\phi = \bigwedge_{S \in \Gamma, (a_{i_1}, \dots, a_{i_{\rho(S)}}) \in S} S(x_1, \dots, x_{i_{\rho(S)}})$$

The structure of the solutions of ϕ is easy to understand. In particular, for every $(b_1, \dots, b_n) \in A^n$, $\phi(b_1, \dots, b_n)$ holds if and only if the mapping $e : A \to A$ sending a_i to b_i for every $1 \leq i \leq n$ belongs to $\mathrm{Pol}(\Gamma)$. Furthermore, since Γ is a core, the later condition is equivalent to the fact that e is an automorphism of Γ.

Hence, the n-ary relation $R = \{(e(a_1), \dots, e(a_n)) \mid e \in \mathrm{Aut}(\Gamma)\}$ is pp-definable from Γ (without equality) via ϕ. Now, for every $1 \leq i \leq n$ consider the binary relation eq_i defined by the primitive positive formula

$$\begin{aligned} \exists x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \quad & R(x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_n) \wedge \\ & R(x_1, \dots, x_{i-1}, z, x_{i+1}, \dots, x_n) \end{aligned}$$

It follows from the definition of eq_i that $\{(a_i, a_i)\} \subseteq \mathrm{eq}_i \subseteq \mathrm{eq}$.

Let $\Gamma' = \Gamma \cup \{R\} \cup \{\mathrm{eq}_i \mid 1 \leq i \leq n\}$. It follows from Lemma 1 that $\mathrm{CSP}(\Gamma') \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$. In what remains we shall show that $\mathrm{CSP}(\Gamma \cup \{\{a_i\} \mid 1 \leq i \leq n\}) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma')$ completing the proof.

Assume that there is a polynomial-time algorithm ALG that $(1-\epsilon, 1-f(\epsilon))$-approximates $\mathrm{CSP}(\Gamma')$ for all $\epsilon \geq 0$. We shall show how we can use ALG to obtain a polynomial-time algorithm that $(1-\epsilon, 1-2f(\epsilon))$-approximates $\mathrm{CSP}(\Gamma \cup \{\{a_i\} \mid 1 \leq$

$i \leq n\})$ for all ϵ in $[0, \beta]$ for a fixed β. This immediately gives, for some $K > 0$, a $(1 - \epsilon, 1 - Kf(\epsilon))$-approximation for all $\epsilon \geq 0$.

Let I be an instance of $\mathrm{CSP}(\Gamma \cup \{\{a_i\} \mid 1 \leq i \leq n\})$ and let m be the number of its constraints. Our algorithm starts by constructing in polynomial time an instance I' of $\mathrm{CSP}(\Gamma')$ in the following way. The set of variables of I' contains all the variables of I in addition to n new variables $v_1, \ldots, v_n$. Instance I' has $2m$ constraints which are constructed in the following way:

(a) Place in I' every constraint in I whose constraint relation is not in $\{\{a_i\} \mid 1 \leq i \leq n\}$.
(b) For every constraint in I of the form $(v, \{a_i\})$ $1 \leq i \leq n$, place in I' the constraint $((v, v_i), \mathrm{eq}_i)$.
(c) Place m copies of the constraint $((v_1, \ldots, v_n), R)$ in I'.

Assume that there is an assignment s that $(1 - \epsilon)$-satisfies I. Let s' be the assignment for I' that acts as s on the variables in I and that sets $s'(v_i) = a_i$ for every $1 \leq i \leq n$, and let C be any constraint in I'. If C is added in step (a) then we know that is satisfied by s' whenever it is satisfied by s. If C is added in step (b) then C is of the form $((v, v_i), \mathrm{eq}_i)$, $1 \leq i \leq n$. As $(a_i, a_i) \in \mathrm{eq}_i$ we have that C is satisfied by s' whenever $(v, \{a_i\})$ is satisfied by s. Finally, constraint $((v_1, \ldots, v_n), R)$ is always satisfied by s' as the identity map is always an automorphism.

We conclude that s' falsifies the same total number of constraints as s. It follows that s' is $(1 - \epsilon)$-satisfiable as the total number of constraints in I' is larger than that in I.

Now, run algorithm ALG with input I'. In polynomial time ALG will stop and report an assignment t' that satisfies a $(1 - f(\epsilon))$-fraction of the constraints in I'. By requiring ϵ to be small enough we can guarantee that $(1 - f(\epsilon)) > 1/2$ which implies that t' must necessarily satisfy constraint $R(v_1, \ldots, v_n)$. Consider the mapping $e : A \to A$ defined as $e(a_i) = t'(v_i)$. It follows that $(e(a_1), \ldots, e(a_n)) \in R$ and hence e is an automorphism of Γ. It follows that e^{-1} is also an automorphism of Γ and, by Theorem 4, of Γ' as well. Consequently, the assignment t defined as $t(v) = e^{-1}(t'(v))$ satisfies exactly the same constraints in I' as t'. Additionally, $t(v_i) = a_i$ holds for every $1 \leq i \leq n$. Output the assignment obtained by projecting t to the variables of I. We shall prove that t (and hence its projection to the variables of I) $(1 - 2f(\epsilon))$-satisfies I.

Let C be any constraint in I. If the constraint relation of C is not in $\{\{a_i\} \mid 1 \leq i \leq n\}$ then C must also appear in I'. Otherwise C is of the form $(v, \{a_i\})$, $1 \leq i \leq n$. In this case, as $\mathrm{eq}_i \subseteq \mathrm{eq}$ and $t(v_i) = a_i$ it follows that if t satisfies $((v, v_i), \mathrm{eq}_i)$ then t must satisfy $(v, \{a_i\})$ as well. It follows that the total number of clauses falsified by t (in I) is not larger than the number of clauses falsified by t (in I'). Since the total number of constraints in I' is twice the total number of constraints in I we conclude that t satisfies at least a $(1 - 2f(\epsilon))$-fraction of the constraints in I. ■

We are finally in a position to prove Theorem 7.

Proof. (of Theorem 7)

(1) Since $\mathbb{C} \in \mathcal{V}(\mathbb{A})$, there exist a power, $\mathbb{A}^m$, of $\mathbb{A}$, and a subalgebra, $\mathbb{B}$, of $\mathbb{A}^m$ such that there is an surjective homomorphism μ from $\mathbb{B}$ to $\mathbb{C}$. Let $\Gamma_1 = \{\mu^{-1}(R) \mid R \in \Gamma_0\}$ and let $\Gamma_2 = \{\mathrm{coord}(R) \mid R \in \Gamma_1\}$. By Lemma 2, $\mathrm{CSP}(\Gamma_0) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_1)$ and, by Lemma 3, $\mathrm{CSP}(\Gamma_1) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_2)$. Furthemore, by Lemma 4, Γ_2 is compatible with $\mathbb{A}$.

Let $\Gamma_3 = \Gamma \cup \{\{a\} \mid a \in A\}$ and let g be any operation preserving Γ_3 (and hence preserving Γ as well). It follows from the fact that g preserves $\{a\}$ for every $a \in A$ that g should be idempotent. It follows that g belongs to the full idempotent reduct of Γ, that is, $\mathbb{A}$. Since Γ_2 is compatible with $\mathbb{A}$ it follows that g preserves Γ_2. We have just seen that $\mathrm{Pol}(\Gamma_3) \subseteq \mathrm{Pol}(\Gamma_2)$. It follows by Theorem 4(1) and Lemma 1 that $\mathrm{CSP}(\Gamma_2) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_3)$. Finally, Lemma 5 guarantees that $\mathrm{CSP}(\Gamma_3) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$.

(2) After inspecting the previous argument one realizes that the condition $\mathrm{eq} \in \Gamma$ is only required when applying Theorem 4(1) to prove $\mathrm{CSP}(\Gamma_2) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_3)$. This can be overcome by noticing that since $\mathbb{C} \in \mathrm{HS}(\mathbb{A})$ we can assume $m = 1$ and, hence, $\Gamma_2 = \Gamma_1$. Observe also that if R is an irredundant relation in Γ_0 then $\mu^{-1}(R)$ must necessarily be irredundant as well. Then, in this case $\mathrm{CSP}(\Gamma_2) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma_3)$ follows from Theorem 4(2) and Lemma 1. ■

Combining Theorem 7 with Theorems 1 and 3 we obtain the following hardness results.

Theorem 9. *Let Γ be a finite set of relations on A such that Γ is a core and let $\mathbb{A}$ be the algebra associated to Γ. Then:*

(1) *If $\mathcal{V}(\mathbb{A})$ admits the unary or affine types then* $\mathrm{CSP}(\Gamma)$ *is not robustly solvable unless* $\mathrm{P} = \mathrm{NP}$.

(2) *If $\mathcal{V}(\mathbb{A})$ admits the semilattice type then* $\mathrm{CSP}(\Gamma)$ *is not robustly solvable with polynomial loss unless the UG conjecture is false.*

Proof. It follows from ([34],Chapter 5) that if $\mathcal{V}(\mathbb{A})$ satisfies one of the conditions of items (1) or (2) then so does its full idempotent reduct.

In case (1), it follows from Theorems 5 and 7 that there exists an abelian group $\mathcal{G}$ with more than one element such that $\mathrm{CSP}(\Gamma_1) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$ where Γ_1 is the set of irredundant relations in 3EQ-LIN($\mathcal{G}$). Item (1) follows by composition of $\leq_{\mathrm{RA}}$ and Theorem 8(1).

In case (2) let Γ_2 be the constraint language containing $\{0\}$, $\{1\}$, and $\{(x, y, z) \mid x \wedge y \to z\}$. Since all relations in Γ_2 are irredundant it follows from Theorems 5 and 7 that $\mathrm{CSP}(\Gamma_2) \leq_{\mathrm{RA}} \mathrm{CSP}(\Gamma)$. Item (2) follows by composition of $\leq_{\mathrm{RA}}$ and Theorem 8(2). ■

Item (1) of Theorem 9 is the 'easy' direction of the Guruswami-Zhou conjecture. Combining it with with Theorems 2 and 6 one obtains the full proof.

4. Dualities

In this section we present a combinatorial view on CSPs and local consistency algorithms, in the form of dualities, and link it with polymorphisms. The combinatorial description of dualities is not used in approximation algorithms in Section 5, but it helps to place our results into a uniform perspective. We refer the reader to survey [11] for more information about dualities.

A *(finite relational) structure* is a tuple $\mathbf{A} = (A, R_1, \ldots, R_m)$ where A, the *universe* of $\mathbf{A}$, is a non-empty set, and for each $1 \leq i \leq m$, R_i is a relation on A. Let $\mathbf{I} = (V, S_1, \ldots, S_m)$ and $\mathbf{A} = (A, R_1, \ldots, R_m)$ be *similar* structures, meaning that they have the same number of relations and that $\rho(R_i) = \rho(S_i)$ for every $1 \leq i \leq m$. A map $f : V \to A$ is a homomorphism from $\mathbf{I}$ to $\mathbf{A}$ if $f(S_i) \subseteq R_i$ for every $1 \leq i \leq m$, where for every relation R we have

$$f(R) = \{f(\mathbf{a}) \mid \mathbf{a} \in R\}.$$

We write $\mathbf{I} \to \mathbf{A}$ if there is a homomorphism from $\mathbf{I}$ to $\mathbf{A}$ and $\mathbf{I} \not\to \mathbf{A}$ otherwise. Two structures $\mathbf{A}$ and $\mathbf{A}'$ are said to be *homomorphically equivalent* if $\mathbf{A} \to \mathbf{A}'$ and $\mathbf{A}' \to \mathbf{A}$.

The constraint satisfaction problem can be rephrased in terms of homomorphisms as follows: If $\Gamma = \{R_1, \ldots, R_m\}$ is a finite set of relations on A and $I = (V, A, \mathcal{C})$ is an instance of CSP(Γ), let $\mathbf{A}$ be the structure $(A, R_1, \ldots, R_m)$ and let $\mathbf{I} = (V, S_1, \ldots, S_m)$ be the structure with universe V and where for every $1 \leq i \leq m$, S_i contains the scopes of all constraints in $\mathcal{C}$ whose constraint relation is R_i. It is easy to verify that any assignment $s : V \to A$ satisfies all constraints in $\mathcal{C}$ if and only if s is a homomorphism from $\mathbf{I}$ to $\mathbf{A}$.

A set $\mathcal{O}$ of structures is called an *obstruction set* for $\mathbf{A}$ if for any structure $\mathbf{I}$ similar to $\mathbf{A}$, $\mathbf{I} \to \mathbf{A}$ if and only if $\mathbf{O} \not\to \mathbf{A}$ for every $\mathbf{O} \in \mathcal{O}$.

In graph theory, the treewidth of a graph is a natural number that measures how much the graph resembles a tree. This measure, as many others, is lifted in a natural way to structures.

For $0 \leq j \leq k$, a structure $\mathbf{I} = (V, S_1, \ldots, S_m)$ is said to have treewidth at most (j, k) if there is a tree T, called a *tree-decomposition* of $\mathbf{I}$, such that

(1) the nodes of T are subsets of V of size at most k,
(2) adjacent nodes can share at most j elements,
(3) nodes containing any given element form a subtree, and
(4) for any tuple in any relation in $\mathbf{I}$, there is a node in T containing all elements from that tuple.

If T is a path then it is called a *path-decomposition* of $\mathbf{I}$ and $\mathbf{I}$ is said to have *pathwidth* at most (j, k).

Definition 1. *A finite set Γ of relations on A is said to have (j, k)-*treewidth duality *if the structure (A, Γ) has an obstruction set consisting only of structures of treewidth at most (j, k). We say that Γ has j-*treewidth duality *if it has (j, k)-treewidth duality for some $k \geq j$ and that Γ has bounded treewidth duality if it has j-treeduality for some $j \geq 0$.*

The following result establishes a fundamental connection between width and dualities.

Theorem 10. [25, 40] *Let $0 \leq j \leq k$ and let Γ be a finite set of relations on A. The following conditions are equivalent:*

(1) CSP(Γ) *has width (j, k);*
(2) Γ *has (j, k)-treewidth duality.*

Besides bounded treewidth duality many other types of dualites have been explored in the study of CSP. We shall present some of them that are particularly relevant to the present work.

4.1. **Tree duality.** As in [46], the *incidence multigraph* of a structure $\mathbf{I} = (V, S_1, \ldots, S_m)$, denoted Inc($\mathbf{I}$), is defined as the bipartite multigraph with parts V and Block($\mathbf{I}$), where Block($\mathbf{I}$) consists of all pairs $(S_i, \mathbf{v})$ such that $1 \leq i \leq m$ and $\mathbf{v} \in S_i$, and with edges $e_{v,i,Z}$ joining $v \in V$ to $Z = (S, (v_1, \ldots, v_r)) \in$ Block($\mathbf{A}$) when $v_i = v$. Then $\mathbf{I}$ is said to be a *tree* if its incidence multigraph is a tree (in particular, it has no multiple edges). For a tree $\mathbf{I}$, we say that an element of V is a *leaf* if it is incident to at most one block in Inc($\mathbf{I}$). A block of $\mathbf{I}$ (i.e., a member of Block($\mathbf{I}$)) is said to be *pendant* if it is incident to at most one non-leaf element, and it is said to be *non-pendant* otherwise. For example, any block with a unary relation is always pendant. If **I** has just one binary relation, i.e. is a digarph, then **I** is tree in the above sense if and only if it is an oriented tree in the usual sense of graph theory.

We shall say that a structure has *tree duality* if it has an obstruction set consisting of tree structures. See [11] for examples of structures with tree duality.

Let (A, Γ) be a structure. Let A_P be the set of all non-empty subsets of A. If R is a r-ary relation on A then we define relation, R_P, as the r-ary relation on A_P

$$R_P = \{(\mathrm{pr}_1 S, \ldots, \mathrm{pr}_r S) \mid \emptyset \neq S \subseteq R\}$$

It follows from the definition of R_P that for every $(S_1, \ldots, S_r), (T_1, \ldots, T_r) \in (A_P)^r$

$$\{(S_1, \ldots, S_r), (T_1, \ldots, T_r)\} \subseteq R_P \quad \Rightarrow \quad (S_1 \cup T_1, \ldots, S_r \cup T_r) \in R_P \tag{3}$$

Let Γ_P be the constraint language on A_P defined as $\Gamma_P = \{R_P \mid R \in \Gamma\}$.

Theorem 11. [25] *Let Γ be a finite set of relations on A. The following conditions are equivalent:*

(1) *(A, Γ) has tree duality;*
(2) *(A, Γ) had 1-treewidth duality;*
(3) *(A, Γ) is homomorphically equivalent to (A_P, Γ_P);*
(4) *Γ totally symmetric polymorphisms of all arities.*
(5) CSP(Γ) *has width* 1.

Width 1 problems are closely related with arc-consistency, one of main types of local consistency [22].

4.2. **Pathwidth Duality.** By replacing "treewidth" with "pathwidth" throughout Definition 1 one obtains the corresponding notions of *pathwidth dualities*. Bounded path duality was introduced in [20] as a tool to study CSPs solvable in non-deterministic logarithmic space. See [11, 13, 14, 20] for examples of structures with this duality. The following theorem, due to Larose and Tesson [42], gives a general necessary condition for bounded pathwidth duality.

Theorem 12. *Let Γ be a finite set of relations on A such that Γ is a core and let $\mathbb{A}$ be the algebra associated to Γ. If $\mathcal{V}(\mathbb{A})$ admits the unary, affine, or semilattice types then Γ does not have bounded pathwidth duality.*

The comparison of Theorems 9 and 12 hints at a link between pathwidth duality and robust approximation with polynomial loss as both properties share the same forbidden typesets (and the same basic forbidden structures, 3EQ-LIN($\mathcal{G}$) and 3-HORN). In view of this, it seems reasonable to investigate whether one can robustly solve with polynomial loss a CSP(Γ) whenever its constraint language, Γ, has pathwith duality. Some sufficient conditions for bounded pathwidth duality are known, we shall now present the two most general ones (to the best of our knowledge).

It was shown in [21] that a finite set Γ of relations on A has bounded pathwidth duality whenever it is preserved by a majority operation, and this result has been recently generalized in [5] to an NU operation of any arity. In a different direction, [13] characterizes those finite sets Γ of relations that possess an obstruction set consisting on trees of bounded pathwidth. In Section 5 we show that, for every such Γ, CSP(Γ) is robustly solvable with polynomial loss. In what follows we shall describe in detail some of the results in [13].

We call a operation f of arity $k \cdot m \cdot n$ on A *k-layered m-block symmetric* if it satisfies the following condition:

$$\begin{aligned} & f(\overbrace{x_{11}^{(1)},\ldots,x_{1m}^{(1)}}^{S_1^{(1)}},\ldots,\overbrace{x_{n1}^{(1)},\ldots,x_{nm}^{(1)}}^{S_n^{(1)}},\ldots,\overbrace{x_{11}^{(k)},\ldots,x_{1m}^{(k)}}^{S_1^{(k)}},\ldots,\overbrace{x_{n1}^{(k)},\ldots,x_{nm}^{(k)}}^{S_n^{(k)}}) = \\ & = f(\underbrace{y_{11}^{(1)},\ldots,y_{1m}^{(1)}}_{T_1^{(1)}},\ldots,\underbrace{y_{n1}^{(1)},\ldots,y_{nm}^{(1)}}_{T_n^{(1)}},\ldots,\underbrace{y_{11}^{(k)},\ldots,y_{1m}^{(k)}}_{T_1^{(k)}},\ldots,\underbrace{y_{n1}^{(k)},\ldots,y_{nm}^{(k)}}_{T_n^{(k)}}) \end{aligned}$$

whenever $\{S_1^{(l)},\ldots,S_n^{(l)}\} = \{T_1^{(l)},\ldots,T_n^{(l)}\}$ for each "level" l where, for all i, $S_i^{(l)} = \{x_{i1}^{(l)},\ldots,x_{im}^{(l)}\}$ and $T_i^{(l)} = \{y_{i1}^{(l)},\ldots,y_{im}^{(l)}\}$. This allows us to write such an operation as $f(\mathcal{S}_1,\ldots,\mathcal{S}_k)$, where $\mathcal{S}_i = \{S_1^{(i)},\ldots,S_n^{(i)}\}$ for all i.

Let us call a sequence $\mathcal{S}_1,\ldots,\mathcal{S}_k$ *nested* if either $k=1$ or, for each $1 \le j < k$, every set in $\mathcal{S}_{j+1}$ is a subset of every set in $\mathcal{S}_j$. We say that a k-layered m-block symmetric operation f is a *k-layered m-ABS* operation if the following *absorption* property holds: for any $1 \le i \le k$ and for any nested sequence $\mathcal{S}_1,\ldots,\mathcal{S}_k$ we have

$$f(\mathcal{S}_1,\ldots,\mathcal{S}_i,\ldots,\mathcal{S}_k) = f(\mathcal{S}_1,\ldots,\mathcal{S}_i',\ldots,\mathcal{S}_k)$$

where $\mathcal{S}_i'$ is any subset of $\mathcal{S}_i$ obtained by removing any element (i.e., a subset of A) in $\mathcal{S}_i$ that entirely contains some other element in $\mathcal{S}_i$.

Example 1. *Let $A = \{0,1\}^k$. In this example we will think of elements of A as k-columns of Boolean values. Consider the operation f on A such that*

$$f(x_{11}^{(1)},\ldots,x_{1k}^{(1)},\ldots,x_{n1}^{(1)},\ldots,x_{nm}^{(1)},\ldots,x_{11}^{(k)},\ldots,x_{1m}^{(k)},\ldots,x_{n1}^{(k)},\ldots,x_{nm}^{(k)}) =$$

$$\left(\begin{array}{r} (\bigvee_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(1)}[1]) \wedge (\bigwedge_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(2)}[1]) \wedge \ldots \wedge (\bigwedge_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(k)}[1]) \\ (\bigvee_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(2)}[2]) \wedge \ldots \wedge (\bigwedge_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(k)}[2]) \\ \vdots \\ (\bigvee_{i=1}^n \bigwedge_{j=1}^m x_{ij}^{(k)}[k]) \end{array}\right)$$

where $x_{ij}^{(w)}[l]$ denotes the l-th component of variable $x_{ij}^{(w)}$.

It can be directly verified that f is a k-layered m-ABS operation.

The following theorem follows from [13]:

Theorem 13. *Let Γ be a finite set of relations on A. Then the following are equivalent:*

(1) *(A,Γ) has an obstruction set consisting on trees of bounded pathwidth;*
(2) *there exists some $k \ge 1$ such that for every $m,n \ge 1$, Γ is invariant under a mkn-ary k-layered m-ABS operation.*

4.3. **Caterpillar and jellyfish dualities.** The particular case of 1-layered ABS operations gives rise to a well-understood type of dualities called caterpillar duality. In graph theory, a caterpillar is a tree which becomes a path after all its leaves are removed. Following [43], we say that a tree is a *caterpillar* if each of its blocks is incident to at most two non-leaf elements, and each element is incident to at most two non-pendant blocks. Informally, a caterpillar has a body consisting of a chain of elements $v_1,\ldots,v_{n+1}$ with blocks $B_1,\ldots,B_n$ where B_i is incident to v_i and v_{i+1} $(i = 1,\ldots,n)$, and legs of two types: (i) pendant blocks incident to exactly one of the elements $v_1, \ldots, v_{n+1}$, together with some leaf elements incident to such blocks, and (ii) leaf elements incident to exactly one of the blocks $B_1, \ldots, B_n$. Examples of structures with caterpillar duality can be found in [14].

Theorem 14. [14] *Let Γ be a finite set of relations on A. Then the following are equivalent:*

(1) (A, Γ) *has caterpillar duality;*
(2) (A, Γ) *is homomorphically equivalent to a structure with lattice polymorphisms;*
(3) *for every $m, n \geq 1$, Γ is invariant under a mn-ary 1-layered m-ABS operation.*

Note that robust satisfiability for structures with lattice polymorphisms was studied in [41], where a robust algorithm with linear loss for the corresponding CSPs is presented. By Lemma 6 below, this result extends to all structures with caterpillar duality. We will further extend this result to a subclass of structures covered in Theorem 13.

We say that a non-leaf $a \in A$ of a tree structure $\mathbf{A}$ is extreme if it is incident to at most one non-pendant block (i.e., it has at most one other non-leaf at distance two from it) in $Inc(\mathbf{A})$, and we say that a pendant block is extreme if either it is the only block of $\mathbf{A}$ or else it is adjacent to a non-leaf, and this (unique) non-leaf is extreme. Finally, we say that an element is *terminal* if it is isolated (i.e., does not appear in any relation in $\mathbf{A}$) or it appears in an extreme pendant block. We say that a tree structure $\mathbf{A}$ is a *jellyfish* if it is a one-element structure with empty relations or it is obtained from one tuple (in one relation) $\mathbf{a}$, called the body of the jellyfish, and a family of caterpillars by identifying one terminal element of each caterpillar with some element in the tuple $\mathbf{a}$. It is not hard to see that a jellyfish structure is a tree of bounded pathwidth. A structure has jellyfish duality if it has an obstruction set consisting of jellyfish structures. Examples of structures with jellyfish duality can be found in [14]. It can be checked using results of [13, 14] that each structure with jellyfish duality has an $2mn$-ary 2-layered m-ABS polymorphism for all m, n.

Theorem 15. [14] *Let Γ be a finite set of relations on A. Then the following are equivalent:*

(1) (A, Γ) *has jellyfish duality;*
(2) (A, Γ) *is homomorphically equivalent to a structure (A', Γ') with polymorphism $x \sqcup (y \sqcap z)$ for some distributive lattice $(A', \sqcup, \sqcap)$.*

5. Positive Approximation Results

In this section we show that each width 1 CSP(Γ) admits a robust $(1-\epsilon, 1-O(1/\log(1/\epsilon)))$-approximation algorithm and describe two subclasses of width 1 CSPs where the approximation guarantee can be improved to $O(\epsilon^{1/k})$ with $k > 1$ and $O(\epsilon)$, respectively.

Lemma 6. *If structures $\mathbf{A} = (A, \Gamma)$ and $\mathbf{A}' = (A', \Gamma')$ are homomorphically equivalent then* CSP(Γ) $\leq_{\rm RA}$ CSP(Γ') *and* CSP(Γ') $\leq_{\rm RA}$ CSP(Γ).

Proof. Since the two structures are homomorphically equivalent, the relations in Γ and Γ' are in one-to-one correspondence. If I is an instance of CSP(Γ), one can construct an equivalent instance of CSP(Γ') by simply replacing each constraint relation in I by the corresponding relation from Γ'. If s is an assignment for I

and f is a homomorphism from $\mathbf{A}$ to $\mathbf{A}'$ then it is easy to check that $f \circ s$ is an assignment for I' that satisfies all constraints satisfied by s. It follows that, for any ϵ, I is $(1-\epsilon)$ satisfiable if and only if I' is $(1-\epsilon)$ satisfiable, and one can easily switch between assignments for I and I' by using homomorphisms. ■

Theorem 16. *Let Γ be a finite set of relations on A such that* $\mathrm{CSP}(\Gamma)$ *has width* 1*. There is a polynomial-time algorithm that* $(1-\epsilon, 1-O(1/\log(1/\epsilon)))$*-approximates* $\mathrm{CSP}(\Gamma)$ *for every* $\epsilon \geq 0$.

Proof. By Theorem 11 and Lemma 6, it is enough to prove the theorem for Γ_P. Now, fix an arbitrary order $\{a_1, \ldots, a_k\}$ on A and rename A_P by replacing every element S of A_P by its indicator k-ary tuple, namely, the tuple $(b_1, \ldots, b_k) \in \{0,1\}^k$ such that $b_i = 1$ if $a_i \in S$ and $b_i = 0$ otherwise. Let Γ_C be the finite set of relations on $\{0,1\}$ defined as $\Gamma_C = \{\mathrm{coord}(R_P) \mid R_P \in \Gamma_P\}$ where $\mathrm{coord}(\cdot)$ is the coordinazation operator introduced in Section 3. It follows from (3) that Γ_C is preserved by the disjunction operation $\vee : \{0,1\}^2 \to \{0,1\}$. It is well known that every boolean relation invariant under $\vee$ can be expressed as a conjunction of dual Horn clauses. It follows from Lemma 1 and Theorem 3 that there is a polynomial-time algorithm that $(1-\epsilon, 1-O(1/\log(1/\epsilon)))$-approximates $\mathrm{CSP}(\Gamma_C)$ for every $\epsilon \geq 0$. The result then follows from Lemma 3. ■

Which structures admit an efficient robust algorithm with polynomial loss? As mentioned in Section 4 the properties of robust approximation with polynomial loss and of pathwidth duality share the same forbidden typesets. It seems natural then to try to prove that every constraint language Γ with bounded pathwidth duality gives rise to a constraint satisfaction problem, $\mathrm{CSP}(\Gamma)$ that is robusly solvable with polynomial loss. The next theorem gives a partial result in this direction. However, current understanding suggests that it is quite feasible that the theorem covers all constraint languages Γ with tree duality such that $\mathrm{CSP}(\Gamma)$ is robustly solvable with polynomial loss.

Theorem 17. *If Γ and $k \geq 1$ satisfy condition (2) from Theorem 13 then there is a polynomial-time algorithm that* $(1-\epsilon, 1-O(\epsilon^{1/k}))$*-approximates* $\mathrm{CSP}(\Gamma)$ *for every* $\epsilon \geq 0$.

Proof. Let $I = (V, A, \mathcal{C})$ be any instance of $\mathrm{CSP}(\Gamma)$ and assume that I is $(1-\epsilon')$-satisfiable for some $\epsilon' \geq 0$. Start by solving the LP relaxation of I, $\mathrm{BLP}(I)$, determining an optimal solution. Let $1-\epsilon$ the value of the goal function achieved by the optimal solution. Since $\mathrm{BLP}(I)$ is a relaxation of the integer canonical program for I it follows that $\epsilon \leq \epsilon'$. For every constraint $C = (\mathbf{v}, R) \in \mathcal{C}$ we shall use ϵ_C to denote $1 - p_C(R)$.

Let $H > 1$ be a constant. To prove the present theorem we could fix straight away H to be, say, 2, but it will be handy later, when proving Theorem 18 to be able to reuse the analysis with a different value for H. Let L be the maximum arity of any relation in Γ, let $J = L2^{|A|} + 1$, let $b = \epsilon/J$, and let $z = J(Hb)^{1/k}$.

For every $\theta = \{1, \ldots, \lfloor z^{-1} \rfloor\}$ and every $0 \leq i \leq k$ define M_θ^i as

$$M_\theta^i = \begin{cases} 0 & \text{if } i = 0 \\ b(J\theta)^i & \text{otherwise} \end{cases}$$

We shall obtain a solution by applying the following randomized rounding algorithm to the optimal solution of the LP:

(1) Choose $\theta \in \{1, \ldots, \lfloor z^{-1} \rfloor\}$ uniformly at random.

(2) For every v and every $0 \le i \le k$ define $\mathcal{S}_v^i = \{S \subseteq A \mid p_v(S) \ge 1 - M_\theta^i\}$. Since $M_\theta^i \le M_{z^{-1}}^k = H^{-1} < 1$ it follows that $\emptyset \notin \mathcal{S}_v^i$.

(3) Let f be an mkn-ary k-layered m-ABS operation that preserves Γ with $m = L|A|$ and $n = L(2^{|A|} - 1)$. Output the assignment $t : V \to A$ defined by $t(v) = f(\min \mathcal{S}_v^1, \dots, \min \mathcal{S}_v^k))$ where $\min \mathcal{S}_v^i$ contains all those sets in $\mathcal{S}_v^i$ that are minimal with respect to inclusion. By the properties of f we see that t is well-defined.

We shall prove for each constraint $C \in \mathcal{C}$ that the probability that C is falsified by assignment t is at most $D\epsilon^{1/k}(1 + \epsilon_C/\epsilon)$ where $D = 2kHJ^{2-1/k}$. It follows from linearity of expectation that the expected fraction of constraints falsified by t is at most $D\epsilon^{1/k}(1 + \mathrm{avg}\{\epsilon_C\}/\epsilon) = 2D\epsilon^{1/k}$. Note that as z^{-1} depends on ϵ it can be, in principle, very large. To overcome it we observe that we can safely replace any value of $\epsilon \le (4|\mathcal{C}|D)^{-k}$ with $(4|\mathcal{C}|D)^{-k}$ as the fraction of falsified constraints, $2D\epsilon^{1/k}$, would be at most $1/(2|\mathcal{C}|)$, meaning that, indeed, all constraints are satisfied. Hence, we can assume that z^{-1} is bounded by a polynomial in the input size. In consequence, we can even make the algorithm deterministic (besides polynomial-time) by trying all choices for θ and selecting the one producing the best assignment.

Let $C = ((v_1, \dots, v_r), R)$ be a constraint in $\mathcal{C}$. We shall see that the probability that C is falsified by t is at most $D\epsilon^{1/k}(1 + \epsilon_C/\epsilon)$ completing the proof. This will follow from Lemmas 7 and 9 below.

Definition 2. *A choice of θ is* good *for C if*

$$M_\theta^1 \ge \epsilon_C \tag{4}$$

and for every variable v in the scope of C, every $1 \le i \le k$, and every $S \subseteq A$ the two following conditions hold:

$$M_\theta^1 \notin [1 - p_v(S) - \epsilon_C, 1 - p_v(S)) \tag{5}$$

$$\theta + 1 \ne \lceil (1 - p_v(S))^{1/i} b^{-1/i} J^{-1} \rceil \tag{6}$$

Lemma 7. *The probability that θ is not good for C is at most $D\epsilon^{1/k}(1 + \epsilon_C/\epsilon)$.*

Proof. We shall see how many out of the $\lfloor z^{-1} \rfloor$ choices for θ falsify each one of the conditions of definition 2. The number of values for θ that falsify (4) is $\lfloor \frac{\epsilon_C}{Jb} \rfloor$. For every v in the scope of C, every $1 \le i \le k$ and every $S \subseteq A$, there is only one choice that falsifies (6) and $1 + \lfloor \frac{\epsilon_C}{Jb} \rfloor$ choices that falsify (5). Hence the total number of choices that make θ not good is at most

$$2Lk2^{|A|} + (Lk2^{|A|} + 1)\frac{\epsilon_C}{Jb} \le 2kJ\,(1 + \epsilon_C/\epsilon)$$

The bound of the lemma is obtained dividing it by z^{-1}. ■

The following lemma will be useful.

Lemma 8. *Assume that θ is good for C. Then for every variable v in the scope, every $1 \le i \le k$, and every $S \subseteq A$: if $\mathrm{pr}_v(S) \ge 1 - \epsilon_C - M_\theta^i - (J-1)M_\theta^{i-1}$ then $p_v(S) \ge 1 - M_\theta^i$.*

Proof. Case $i = 1$ follows from (5) and $M_\theta^0 = 0$. Assume now that $i > 1$. Condition (6) can be rewritten as $(1 - p_v(S))^{1/i} b^{-1/i} J^{-1} \notin (\theta, \theta + 1]$ which again can be rewriten as $p_v(S) \notin [1 - b(J(\theta+1))^i, 1 - b(J\theta)^i) = [1 - M_{\theta+1}^i, 1 - M_\theta^i)$. Now, assume that $p_v(S) \ge 1 - \epsilon_C - M_\theta^i - (J-1)M_\theta^{i-1}$. We have that $M_\theta^{i-1} \ge M_\theta^1 \ge \epsilon_C$

where the second inequality is by (4). It follows that $p_v(S) \geq 1 - M^i_\theta - JM^{i-1}_\theta \geq 1 - M^i_{\theta+1}$ where the second inequality follows from $M^i_{\theta+1} = b(J(\theta+1))^i \geq bJ^i(\theta^i + \theta^{i-1}) = M^i_\theta + JM^{i-1}_\theta$. Since $p_v(S) \notin [1 - M^i_{\theta+1}, 1 - M^i_\theta)$ it follows that $p_v(S) \geq 1 - M^i_\theta$. ■

Lemma 9. *It θ is good for C then C is satisfied by t.*

Proof. For every $0 \leq i \leq k-1$ and every $v \in V$ define $G^i_v = \cap_{S \in \mathcal{S}^i_v} S$. Let $1 \leq i \leq k$, let $1 \leq j, l \leq r$ and let $S \in \mathcal{S}^i_{v_j}$. We shall prove that:

$$\mathrm{pr}_l(R \cap (G^{i-1}_{v_1} \times \cdots \times G^{i-1}_{v_{j-1}} \times (S \cap G^{i-1}_{v_j}) \times G^{i-1}_{v_{j+1}} \times \cdots \times G^{i-1}_{v_r})) \in \mathcal{S}^i_{v_l} \qquad (7)$$

Relation $R' = R \cap (G^{i-1}_{v_1} \times \cdots \times G^{i-1}_{v_{j-1}} \times (S \cap G^{i-1}_{v_j}) \times G^{i-1}_{v_{j+1}} \times \cdots \times G^{i-1}_{v_r})$ can be written down as the intersection of R, $(A^{j-1} \times \ S \times A^{r-j})$, and all relations of the form $(A^{s-1} \times S' \times A^{r-s})$ where $1 \leq s \leq r$ and $S' \in \mathcal{S}^{i-1}_{v_s}$. By condition (2) of BLP(I) it follows that $p_C(A^{j-1} \times S \times A^{r-j}) = p_{v_j}(S) \geq 1 - M^i_\theta$. Similarly, we have $p_C(A^{s-1} \times S' \times A^{r-s}) \geq 1 - M^{i-1}_\theta$ for every $1 \leq s \leq r$ and $S' \in \mathcal{S}^{i-1}_{v_s}$. It follows from the union bound that $p_C(R') \geq 1 - \epsilon_C - M^i_\theta - r2^{|A|}M^{i-1}_\theta$. This quantity is, by Lemma 8, at least $1 - M^i_\theta$. It follows by consistency of marginals that $p_{v_l}(\mathrm{pr}_l R') \geq 1 - M^i_\theta$ and hence that $\mathrm{pr}_l R' \in \mathcal{S}^i_{v_l}$.

We are ready to show that that $(t(v_1), \ldots, t(v_r)) \in R$. The rest of the proof follows that of Lemma 22 in [13]. We shall build a matrix N as follows. Recall that $m = L|A|$. For every $1 \leq j \leq r$, every $1 \leq i \leq k$, and every set $S \in \mathcal{S}^i_{v_j}$ construct a $(m \times r)$-matrix $N^i_{j,S}$ whose entries are elements of A such that:

(1) each row of $N^i_{j,S}$ is a tuple of R, and
(2) for any $1 \leq s \leq r$ the set of entries in the s-th column is exactly

$$\mathrm{pr}_s(R \cap (G^{i-1}_{v_1} \times \cdots \times G^{i-1}_{v_{j-1}} \times (G^{i-1}_{v_j} \cap S) \times G^{i-1}_{v_{j+1}} \times \cdots \times G^{i-1}_{v_r}))$$

That is, the matrix can be seen as a sequence of m tuples $\mathbf{t_1}, \ldots, \mathbf{t_m}$ (the rows) of R such that $\{\mathbf{t_1}, \ldots, \mathbf{t_m}\} = R \cap (G^{i-1}_{v_1} \times \cdots \times G^{i-1}_{v_{j-1}} \times (G^{i-1}_{v_j} \cap S) \times G^{i-1}_{v_{j+1}} \times \cdots \times G^{i-1}_{v_r})$. This is easily achieved by placing in the matrix all tuples in $R \cap (G^{i-1}_{v_1} \times \cdots \times G^{i-1}_{v_{j-1}} \times (G^{i-1}_{v_j} \cap S) \times G^{i-1}_{v_{j+1}} \times \cdots \times G^{i-1}_{v_r})$ and repeating some of them if necessary. By condition (7) the set of all entries in the l-th column of $N^i_{j,S}$ belongs to $\mathcal{S}^i_{v_l}$ and, by construction, it must be a subset of $G^{i-1}_{v_l}$. It follows that if S is minimal in $\mathcal{S}^i_{v_j}$ then the set of entries in the j-th column is precisely S.

Recall that $n = L(2^{|A|} - 1)$. For every $1 \leq i \leq k$ construct a $(mn \times r)$-matrix N^i as follows. It is divided into n layers of consecutive m rows, each layer is a matrix $N^i_{j,S}$ for some $1 \leq j \leq r$ and some $S \in \mathcal{S}^i_{v_j}$, and each matrix of this form appears as a layer. By the choice of n, this is possible. For every $1 \leq i \leq k$ and every $1 \leq j \leq r$ we shall denote by $\mathcal{T}^i_{v_j}$ the set containing all those $T \subseteq A$ such that T is the set of all entries of the jth column for some matrix $N^i_{j,S}$ included in N^i. By the remarks made after the construction of $N^i_{j,S}$ we have that $\min \mathcal{T}^i_{v_j} = \min \mathcal{S}^i_{v_j}$ for every $1 \leq j \leq r$ and $1 \leq i \leq k$.

Finally form the $(mkn \times r)$-matrix N whose first mn rows are occupied by matrix N^1, next mn rows are occupied by matrix N^2 and so on. Let $(a_1, \ldots, a_r)$ be the result of applying f column-wise to N, which must be tuple of R since f preserves R. To complete our proof we shall see that $(a_1, \ldots, a_r)$ is precisely $(t(v_1), \ldots, t(v_r))$.

Let $j \in \{1, \ldots, r\}$. By the construction of N it follows that $a_j = f(\mathcal{T}^1_{v_j}, \ldots, \mathcal{T}^k_{v_j})$. By the construction of the matrices, for every $1 \leq i \leq k$, every element in $\mathcal{T}^i_{v_j}$ is a subset of $G^{i-1}_{v_j}$ implying that $(\mathcal{T}^1_{v_j},\dots,\mathcal{T}^k_{v_j})$ is a nested sequence. It follows by the absorption property that $f(\mathcal{T}^1_{v_j},\dots,\mathcal{T}^k_{v_j}) = f(\min \mathcal{T}^1_{v_j},\dots,\min \mathcal{T}^k_{v_j})$ and since $\min(\mathcal{T}^i_{v_j}) = \min(\mathcal{S}^i_{v_j})$ for every $1 \le i \le k$ it follows that $f(\min \mathcal{T}^1_{v_j},\dots,\min \mathcal{T}^k_{v_j}) = f(\min \mathcal{S}^1_{v_j},\dots,\min \mathcal{S}^k_{v_j}) = t(v_j)$. ■

This completes the proof of Theorem 17. ■

The case $k = 1$ of Theorem 17 (i.e. for structures with caterpillar duality) has been previously shown in [41]. With some local modifications in the proof of Theorem 17 we can extend this result to structures with jellyfish duality. Our result gives the currently most general sufficient condition for robust solvability with linear loss. It will also be useful in Section 6.

Theorem 18. *If (A,Γ) has jellyfish duality then there is a polynomial-time algorithm that $(1-\epsilon, 1-O(\epsilon))$-aproximates* CSP($\Gamma$) *for every $\epsilon \ge 0$.*

Proof. By Lemma 6 and Theorem 15, we can assume that Γ is preserved by $x \sqcup (y \sqcap z)$ for some distributive lattice $(A,\sqcup,\sqcap)$. In the proof of Theorem 17, set $H > 2|A|^L$, $k = 1$, and modify step (3) of the rounding algorithm by setting $t(v)$ to be $\sqcap_{S\in\mathcal{S}^1_v} \sqcup S$ where $\sqcup S = \sqcup_{a\in S}\, a$. It is only required to adapt the proof of Lemma 9 to show that if θ is good for C then C is satisfied by t. Construct matrix $N = N^1$ as in the proof of Theorem 17.

In our proof we shall use the following two properties of lattices.

(i) $x_0 \sqcup ((x_{11} \sqcup \cdots \sqcup x_{1m}) \sqcap \cdots \sqcap (x_{n1} \sqcup \cdots \sqcup x_{nm})) = (x_{11} \sqcup \cdots \sqcup x_{1m}) \sqcap \cdots \sqcap (x_{n1} \sqcup \cdots \sqcup x_{nm})$ whenever $x_0 \in \{x_{u1},\dots x_{um}\}$ for every $u \in \{1,\dots,n\}$

(ii) $(x_{11} \sqcup \cdots \sqcup x_{1m}) \sqcap \cdots \sqcap (x_{n1} \sqcup \cdots \sqcup x_{nm})$ is a 1-layered m-ABS operation.

Both properties follow directly from the definitions.

Now, let $\mathbf{b} = (b_1,\dots,b_r)$ the any tuple in A^r with $p_C(\mathbf{b}) \ge 1/|A|^r$ which must necessarily exist because $p_C(A^r) = 1$. First, we prove that $\mathbf{b}$ appears in every of the matrices $N^1_{j,S}$ used to construct N^1.

By construction, for every $1 \le j \le r$ and every $S \in \mathcal{S}^1_{v_j}$, $N^1_{j,S}$ contains all the tuples of

$$R \cap (G^0_{v_1} \times \cdots \times G^0_{v_{j-1}} \times (G^0_{v_j} \cap S) \times G^0_{v_{j+1}} \times \cdots \times G^0_{v_r}) = R \cap (A^{j-1} \times S \times A^{r-j})$$

By consistency of marginals $p_C(A^{j-1} \times S \times A^{r-j}) = p_{v_j}(S) \ge 1 - M^1_\theta$. Then, by the union bound $p_C(R \cap (A^{j-1} \times S \times A^{r-j})) \ge 1 - \epsilon_C - M^1_\theta$ which is not smaller than $1 - 2M^1_\theta \ge 1 - 2H^{-1} > 1 - 1/|A|^L$ by (4). Since $L \ge r$ it follows that $\mathbf{b}$ must necessarily belong to $R \cap (A^{j-1} \times S \times A^{r-j})$ and hence to $N^1_{j,S}$.

Now, if $N^1_{11},\dots,N^1_{1m},\dots,N^1_{n1},\dots N^1_{nm}$ are the rows in N^1 then let $(a_1,\dots,a_r)$ be the tuple $\mathbf{b} \sqcup ((N^1_{11} \sqcup \cdots \sqcup N^1_{1m}) \sqcap \cdots \sqcap (N^1_{n1} \sqcup \cdots \sqcup N^1_{nm}))$ where $\sqcup$ and $\sqcap$ are applied component-wise.

We want to show that $(a_1,\dots,a_r) \in R$. Observe that for every pair of tuples $\mathbf{t},\mathbf{t}' \in R$ we have that $\mathbf{t} \sqcup \mathbf{t}'$ belongs to R as $\mathbf{t} \sqcup \mathbf{t}' = \mathbf{t} \sqcup (\mathbf{t}' \sqcap \mathbf{t}')$ and the latter must be in R. Alternatively we can say that the binary operation $x \sqcup y$ preserves R because it can be obtained from $x \sqcup (y \sqcap z)$ by composition.

Proceeding in this way we shall prove that the $(nm+1)$-ary operation $x_0 \sqcup ((x_{11} \sqcup \cdots \sqcup x_{1m}) \sqcap \cdots \sqcap (x_{n1} \sqcup \cdots \sqcup x_{nm}))$ preserves R implying that $(a_1,\dots,a_r) \in R$. First, we observe that the m-ary operation $x_1 \sqcup \cdots \sqcup x_m$ preserves R as it can be obtained from composition from $x \sqcup y$ by $x_1 \sqcup (x_2 \sqcup (x_3 \sqcup \cdots \sqcup (x_{m-1} \sqcup x_n) \cdots))$. In a bit more complicated fashion we can show that $x_0 \sqcup (x_1 \sqcap \cdots \sqcap x_n)$ preserves R. If $n = 3$ it follows from the properties of distributive lattices that $x_0 \sqcup ((x_0 \sqcup (x_1 \sqcap x_2)) \sqcap x_3)$ is equal to $x_0 \sqcup (x_1 \sqcap x_2 \sqcap x_3)$. The pattern generalizes easily to arbitrary values for n. Finally, one obtains $x_0 \sqcup ((x_{11} \sqcup \cdots \sqcup x_{1m}) \sqcap \cdots \sqcap (x_{n1} \sqcup \cdots \sqcup x_{nm})$ by suitably composing $x_0 \sqcup (x_1 \sqcap \cdots \sqcap x_n)$ and $x_1 \sqcup \cdots \sqcup x_m$. This finishes the proof that $(a_1, \dots, a_r) \in R$.

Finally, we have:

$$\begin{aligned} b_j \sqcup ((N^1_{11,j} \sqcup \cdots \sqcup N^1_{1m,j}) \sqcap \cdots \sqcap (N^1_{11,j} \sqcup \cdots \sqcup N^1_{nm,j})) = \\ ((N^1_{11,j} \sqcup \cdots \sqcup N^1_{1m,j}) \sqcap \cdots \sqcap (N^1_{11,j} \sqcup \cdots \sqcup N^1_{nm,j})) = \\ \sqcap_{S \in \mathcal{T}^1_{v_j}} \sqcup S = \\ \sqcap_{S \in \mathcal{S}^1_{v_j}} \sqcup S = \\ t(v_j) \end{aligned}$$

which implies that $(t(v_1), \dots, t(v_r)) = (a_1, \dots, a_r)$ and, hence, that t satisfies C.

The first equality follows from property (i) and the fact that $\mathbf{b}$ appears in all the matrices used to construct N^1. The third equality follows from property (2) and the fact that $\min(\mathcal{T}^1_{v_j}) = \min(\mathcal{S}^1_{v_j})$. ■

6. The Boolean Classification

Theorem 18 is the only missing piece to complete the classification of the Boolean case.

Theorem 19. *Let Γ be a finite set of Boolean relations which is a core. The following conditions hold:*

(1) *If* $\mathrm{Pol}(\Gamma)$ *contains the operation $x \vee (y \wedge z)$ or $x \wedge (y \vee z)$ then*
- *there is a polynomial-time algorithm that $(1-\epsilon, 1-O(\epsilon))$-approximates* $\mathrm{CSP}(\Gamma)$ *for every $\epsilon \geq 0$.*

(2) *otherwise, if* $\mathrm{Pol}(\Gamma)$ *contains the majority operation $(x \vee y) \wedge (y \vee z) \wedge (x \vee z)$ then*
- *there is a polynomial-time algorithm that $(1-\epsilon, 1-O(\sqrt{\epsilon}))$-approximates* $\mathrm{CSP}(\Gamma)$ *for every $\epsilon \geq 0$, but*
- *there is no polynomial-time algorithm that $(1-\epsilon, 1-o(\sqrt{\epsilon}))$-approximates* $\mathrm{CSP}(\Gamma)$ *for all $\epsilon \geq 0$ unless the UG conjecture is false.*

(3) *otherwise, if* $\mathrm{Pol}(\Gamma)$ *contains $x \vee y$ or $x \wedge y$ then*
- *there is a polynomial-time algorithm that $(1-\epsilon, 1-O(1/\log(1/\epsilon)))$-approximates* $\mathrm{CSP}(\Gamma)$ *for every $\epsilon \geq 0$ but*
- *there is no polynomial-time algorithm that $(1-\epsilon, 1-o(1/\log(1/\epsilon)))$-approximates* $\mathrm{CSP}(\Gamma)$ *for all $\epsilon \geq 0$ unless the UG conjecture is false.*

(4) *otherwise* $\mathrm{CSP}(\Gamma)$ *is not robustly tractable unless* $\mathrm{P} = \mathrm{NP}$.

Proof. The family of all sets of the form $\mathrm{Pol}(\Gamma)$ where Γ is a set of boolean relations was completely described by Post (see [48] for example). The following result follows directly from Post's description.

Lemma 10. *If $\{x \vee (y \wedge z), x \wedge (y \vee z)\} \cap \mathrm{Pol}(\Gamma) = \emptyset$ then* $\mathrm{Pol}(\Gamma)$ *is included in* $\mathrm{Pol}(\{\neq_2\})$, $\mathrm{Pol}(3\text{-HORN})$, $\mathrm{Pol}(3\text{-DualHORN})$, *or* $\mathrm{Pol}(3\mathrm{EQ\text{-}LIN}(\mathbb{Z}_2))$ *where $\mathbb{Z}_2$ is the 2-element cyclic group.*

Let us consider each of the items of the thorem separately.

(1) Follows from Theorems 18 and 15.

(2) The existence of the approximation algorithm follows from the well-known fact that every relation preserved by $(x \vee y) \wedge (y \vee z) \wedge (x \vee z)$ can be written as a conjunction of clauses in 2-SAT, Theorem 3, and Lemma 1. The hardness follows from Lemma 10, Lemma 1, and Theorems 8 and 4.

(3) If $(x \wedge y) \in \mathrm{Pol}(\Gamma)$ then the existence of the approximation algorithm follows from the well-knwon fact that every relation preserved by $x \wedge y$ is pp-definable from 3-HORN, Theorem 3, and Lemma 1. The case $x \vee y$ follows similarly replacing horn by dual horn. For the hardness, since $(x \vee y) \wedge (y \vee z) \wedge (x \vee z)$ preserves $\mathrm{CSP}(\neq_2)$ and $(x \vee y) \wedge (y \vee z) \wedge (x \vee z) \notin \mathrm{Pol}(\Gamma)$ it follows from Lemma 10 that $\mathrm{Pol}(\Gamma)$ is included in Pol(3-HORN), Pol(3-DualHORN), or Pol(3EQ-LIN($\mathbb{Z}_2$)). Now, we only need to apply Lemma 1, and Theorems 8 and 4.

(4) Since $x \wedge y$ preserves 3-HORN, $x \vee y$ preserves 3-DualHORN, $(x \vee y) \wedge (y \vee z) \wedge (x \vee z)$ preserves $\neq$, and none of these three operations is in $\mathrm{Pol}(\Gamma)$ we can infer by Lemma 10 that $\mathrm{Pol}(\Gamma) \subseteq$ Pol(3EQ-LIN($\mathbb{Z}_2$)). Hardness follows again from Lemma 1, and Theorems 8 and 4. ∎

7. Conclusion

We have adapted the universal-algebraic framework to study robustly satisfiable problems $\mathrm{CSP}(\Gamma)$ with a given error function, and we used it to derive some hardness conditions. We described three classes of CSPs that can be robustly solved with exponential, polynomial, and linear loss. We would like to mention some open problems arising from our research.

Problem 1. *Which problems* $\mathrm{CSP}(\Gamma)$ *can be robustly solved with polynomial or linear loss?*

Problem 2. *Consider the set of numbers* $k \geq 1$ *such that there is a problem* $\mathrm{CSP}(\Gamma)$ *that can be* $(1-\epsilon, 1-O(\epsilon^{1/k}))$*-approximated, but not* $(1-\epsilon, 1-o(\epsilon^{1/k}))$*-approximated, modulo some complexity-theoretic assumptions. Is this set infinite? Does it contain all positive integers?*

References

[1] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[2] S. Arora, B. Barak, and D. Steurer. Subexponential algorithms for unique games and related problems. In *FOCS'10*, pages 563–572, 2010.

[3] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS'09)*, pages 595–603, 2009.

[4] L. Barto and M. Kozik. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 19–22, 2012.

[5] L. Barto, M. Kozik, and R. Willard. Near unanimity constraints have bounded pathwidth duality. Accepted to LICS'12.

[6] V.G. Bodnarchuk, L.A. Kaluzhnin, V.N. Kotov, and B.A. Romov. Galois theory for post algebras, i. *Kibernetika*, (3):1–10, 1969.

[7] A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[8] A. Bulatov. Bounded relational width. Manuscript, 2009.

[9] A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic*, 12(4):Article 24, 2011.

[10] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[11] A. Bulatov, A. Krokhin, and B. Larose. Dualities for constraint satisfaction problems. In *Complexity of Constraints*, volume 5250 of *LNCS*, pages 93–124, 2008.

[12] S. Burris and H.P Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Text in Mathematics*. Springer-Verlag, 1981.

[13] C. Carvalho, V. Dalmau, and A. Krokhin. CSP duality and trees of bounded pathwidth. *Theoretical Computer Science*, 411(34-36):3188–3208, 2010.

[14] C. Carvalho, V. Dalmau, and A. Krokhin. Two new homomorphism dualities and lattice operations. *Journal of Logic and Computation*, 21(6):1065–1092, 2011.

[15] M. Charikar, K. Makarychev, and Y. Markarychev. Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Transactions on Algorithms*, 5(3), 2009.

[16] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. Supermodular functions and the complexity of Max CSP. *Discrete Applied Mathematics*, 149(1-3):53–72, 2005.

[17] D. Cohen and P. Jeavons. The complexity of constraint languages. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, chapter 8. Elsevier, 2006.

[18] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. 2001.

[19] N. Creignou, Ph.G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints*, volume 5250 of *LNCS*. Springer, 2008.

[20] V. Dalmau. Linear datalog and bounded path duality of relational structures. *Logical Methods in Computer Science*, 1(1), 2005.

[21] V. Dalmau and A Krokhin. Majority constraints have bounded pathwidth duality. *Eur. J. Comb.*, 29(4):821–837, 2008.

[22] V. Dalmau and J. Pearson. Set functions and width 1 problems. In *CP'99*, volume 1713 of *LNCS*, pages 159–173, 1999.

[23] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin. The approximability of Max CSP with fixed-value constraints. *Journal of the ACM*, 55(4):Article 16, 2008.

[24] I. Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3), 2007.

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

[26] R. Freese, M. Kozik, A Krokhin, B. Larose, M. Maroti, M. Valeriote, and R. Willard. On Maltsev conditions and omitting types. In preparation.

[27] D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27(1):95–100, 1968.

[28] G. Gottlob, L. Leone, and F. Scarcello. Tractable optimization problems through hypergraph-based structural restrictions. In *Proceedings of ICALP'09*, pages 16–30, 2009.

[29] G. Grätzer. *General lattice theory*. Birkhäuser, 2002.

[30] V. Guruswami, Y. Makarychev, P. Raghavendra, D. Steurer, and Y. Zhou. Finding almost perfect graph bisections. In *Innovations in Computer Science (ICS'11)*, 2011. to appear.

[31] V. Guruswami and Y. Zhou. Tight bounds on the approximability of Almost-satisfiable Horn SAT and Exact Hitting Set. *Theory of Computing*, 8:Article 11, 2012.

[32] J. Håstad. Some optimal inapproximability results. *Journal of the ACM.*, 48:798–859, 2001.

[33] J. Håstad. Every 2-CSP allows nontrivial approximation. *Computational Complexity*, 17(4):549–566, 2008.

[34] D. Hobby and R. McKenzie. *The Structure of Finite Algebras*. Contemporary Mathematics. American Mathematical Society, 1988.

[35] P. Jonsson, M. Klasson, and A. Krokhin. The approximability of three-valued Max CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[36] P. Jonsson, F. Kuivinen, and J. Thapper. Min CSP on four elements: Moving beyond submodularity. In *Constraint Programming, CP'11*, pages 438–453, 2011.

[37] S. Khot. On the power of unique 2-prover 1-round games. In *34th ACM Symposium on Theory of Computing*, pages 767–775, 2002.

[38] S. Khot. On the unique games conjecture. In *Colloquium on Computational Complexity, CCC'10*, pages 99–121, 2010.

[39] S. Khot, G. Kindley, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps. *SIAM Journal on Computing*, 37(1):319–357, 2007.

[40] P. Kolaitis and M. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61:302–332, 2000.

[41] G. Kun, R. O'Donnell, S. Tamaki, Y. Yoshida, and Y. Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Innovations in Computer Science, ITCS'12*, pages 484–495, 2012.

[42] B. Larose and P. Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theoretical Computer Science*, 410(18):1629–1647, 2009.

[43] C. Loten and C. Tardif. Majority functions on structures with finite duality. *European Journal of Combinatorics*, 29(4):979–986, 2008.

[44] M. Maróti and R. McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3-4):463–489, 2008.

[45] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 735–744, 2010.

[46] J. Nešetřil and C. Tardif. Duality theorems for finite structures (characterising gaps and good characterisations). *Journal of Combinatorial Theory, Ser.B*, 80:80–97, 2000.

[47] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC'08*, pages 245–254, 2008.

[48] A. Szendrei. *Clones in universal algebra.*, volume 99 of *Séminaire de Mathématiques Supérieures*. Les Presses de l'Université de Montréal, 1986.

[49] A. Szendrei. A survey on strictly simple algebras and minimal varieties. In *Research and Exposition in Mathematics*, pages 209–239. Heldermann Verlag, 1992.

[50] M. Valeriote. A subalgebra intersection property for congruence distributive varieties. *Canadian Journal of Mathematics*, 61(2):451–464, 2009.

[51] U. Zwick. Finding almost-satisfying assignments. In *30th Annual ACM Symposium on Theory of Computing*, pages 551–560, 1998.