The complexity of global cardinality constraints

Andrei A. Bulatov[*] Dániel Marx[†]

Abstract

In a constraint satisfaction problem (CSP) the goal is to find an assignment of a given set of variables subject to specified constraints. A global cardinality constraint is an additional requirement that prescribes how many variables must be assigned a certain value. We study the complexity of the problem $\mathrm{CCSP}(\Gamma)$*, the constraint satisfaction problem with global cardinality constraints that allows only relations from the set* Γ*. The main result of this paper characterizes sets* Γ *that give rise to problems solvable in polynomial time, and states that the remaining such problems are NP-complete.*

1 Introduction

In a constraint satisfaction problem (CSP) we are given a set of variables, and the goal is to find an assignment of the variables subject to specified constraints, and a constraint is usually expressed as a requirement that combinations of values of a certain (usually small) set of variables belong to a certain relation. CSPs have been intensively studied in both theoretical and practical perspectives. On the theoretical side the key research direction has been the complexity of the CSP when either the way the constraints interact (more precisely, the hypergraph formed by the variable sets of the constraints) is restricted [12, 13, 14], or restrictions are on the type of allowed relations [16, 8, 6, 7, 2]. In the latter direction the main focus has been on the so called *Dichotomy conjecture* [10] suggesting that every CSP restricted in this way is either solvable in polynomial time or is NP-complete.

This 'pure' constraint satisfaction problem is sometimes not enough to model practical problems, as some constraint that have to be satisfied are not 'local' in the sense that they cannot be viewed as applied to only a limited number of variables. Constraints of this type are called *global*. Global constraints are very diverse, the current Clobal Constraint Catalog (see `http://www.emn.fr/x-info/sdemasse/gccat/`) lists 313 types of such constraints. In this paper we focus on *global cardinality constraints* [3, 5, 19]. A global cardinality constraint π is specified for a set of values D and a set of variables V, and is given by a mapping $\pi : D \to \mathbb{N}$ that assigns a natural number to each element of D such that $\sum_{a\in D}\pi(a) = |V|$. An assignment of variables V satisfies π if for each $a \in D$ the number of variables that take value a equals $\pi(a)$. In a CSP with global cardinality constraints, given a CSP instance and a global cardinality constraint π, the goal is to decide if there is a solution of the CSP instance satisfying π. We consider the following problem: Characterize sets of relations Γ such that CSP with global cardinality constraint that uses relations from Γ, denoted by $\mathrm{CCSP}(\Gamma)$, is solvable in polynomial time.

The complexity of $\mathrm{CCSP}(\Gamma)$ has been studied in [9] for sets Γ of relations on a 2-element set. It was shown that $\mathrm{CCSP}(\Gamma)$ is solvable in polynomial time if and only if every relation in Γ is width-2-affine, i.e. it can be expressed as the set of solutions of system of linear equations over a 2-element field containing at most 2 variables. Otherwise it is NP-complete. In this case $\mathrm{CCSP}(\Gamma)$ is also known as the k-$\textsc{Ones}(\Gamma)$ problem, since a global cardinality constraint can be expressed by specifying how many ones (the set of values is thought to be $\{0, 1\}$) one wants to have among the values of variables. The parametrized complexity of k-$\textsc{Ones}(\Gamma)$ has also been studied [18], where k is used as a parameter.

In this paper we characterize sets of relations Γ on an arbitrary finite set D that give rise to a $\mathrm{CCSP}(\Gamma)$ problem solvable in polynomial time, and prove that in all other cases the problem is NP-complete. For 2-element domains [9], the polynomial-time solvable cases rely on the fact that if the value of a variable is set, then this forces a unique assignment on the component of the variable. Generalizing this property, we can obtain tractable cases for larger domains: for example, if Γ contains only binary one-to-one mappings, then the value of a variable clearly defines the assignment of its component. However, there are further polynomial-time cases. The problem does not become more

[*]School of Computing Science, Simon Fraser University, Burnaby, Canada, abulatov@cs.sfu.ca

[†]Budapest University of Technology and Economics, Budapest, Hungary, dmarx@cs.bme.hu. Research supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (OTKA grant 67651)

difficult if a value is replaced by a set of equivalent values, thus in particular the problem is tractable if Γ consists of a binary relation that is a one-to-one mapping between equivalence classes of the domain. The situation becomes significantly more complicated if there are several such relations: to ensure tractability, the equivalence classes have to be coordinated in a certain way. We do not see an easy way of giving a combinatorial characterization of the tractable cases. However, we can obtain a compact characterization using logical definability.

Sets of relations Γ that give rise to polynomial time solvable problem are given by the following 3 conditions: (1) every relation R that can be derived from Γ can be expressed as a conjunction of binary relations; (2) every such binary relation Q involved in the definition of R is a *thick mapping*, i.e. $Q \subseteq A \times B$ for some sets A, B and there are equivalence relations α, β on A, B, respectively, and a mapping $\varphi : A/_{\alpha} \to B/_{\beta}$ such that $(a, b) \in Q$ if and only if $b^{\beta} = \varphi(a^{\alpha})$; (3) any pair of equivalence relations α, β that appear in the definition of binary projections of any such derivable relation R is *non-crossing*, that is, for any α-class C and any β-class D either $C \cap D = \emptyset$, or $C \subseteq D$, or $D \subseteq C$.

The paper is structured as follows. After introducing in Section 2 necessary definition and notation, in Section 3 we study properties of thick mappings, state the main result, and prove that recognizing if a set Γ gives rise to a polynomial time problem can also be done in polynomial time. In Section 4 we present an algorithm solving $\mathrm{CCSP}(\Gamma)$. Then a result similar to the key result of the algebraic approach to the CSP is proved in Section 5.1: Adding to Γ a relation definable in Γ by a primitive positive formula does not increase the complexity of the problem. We also prove in Section 5.2 that adding the *constant* relations does not increase the complexity of $\mathrm{CCSP}(\Gamma)$. Section 6 proves the hardness part of the theorem.

2 Preliminaries

Relations and constraint languages. The set of all tuples of elements from a set D is denoted by D^n. We denote tuples in boldface, e.g., $\mathbf{a}$, and their components by $\mathbf{a}[1], \mathbf{a}[2], \ldots$. For a subset $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ with $i_1 < \ldots < i_k$ and an n-tuple $\mathbf{a}$, by $\mathrm{pr}_I \mathbf{a}$ we denote the *projection of* $\mathbf{a}$ *onto* I, the k-tuple $(\mathbf{a}[i_1], \ldots, \mathbf{a}[i_k])$. An n-ary *relation* on set D is any subset of D^n. A set of relations over D is called a *constraint language* over D. Sometimes we use instead of relation R the corresponding predicate $R(x_1, \ldots, x_n)$. Using predicates we can *express* or *define* relations through other relations by means of logical formulas. The *projection* $\mathrm{pr}_I R$ of R is the k-ary relation $\{\mathrm{pr}_I \mathbf{a} \mid \mathbf{a} \in R\}$.

Pairs from equivalence relations play a special role, so such pairs will be denoted by, e.g., $\langle a, b \rangle$. If α is an equivalence relation on a set D then $D/_{\alpha}$ denotes the set of α-classes, and a^{α} for $a \in D$ denotes the α-class containing α. Sometimes we need to emphasize that the unary projections $\mathrm{pr}_1 R$, $\mathrm{pr}_2 R$ of a binary relation R are sets A and B. We denote this by $R \subseteq A \times B$.

Constraint Satisfaction Problem with cardinality constraints. Let D be a finite set and Γ a constraint language over D. An instance of the *Constraint Satisfaction Problem* (CSP for short) $\mathrm{CSP}(\Gamma)$ is a pair $\mathcal{P} = (V, \mathcal{C})$, where V is a finite set of *variables* and $\mathcal{C}$ is a set of *constraints*. Every constraint is a pair $C = \langle \mathbf{s}, R \rangle$ consisting of an n_C-tuple $\mathbf{s}$ of variables, called the *constraint scope* and an n_C-ary relation $R \in \Gamma$, called the *constraint relation*. A solution of $\mathcal{P}$ is a mapping $\varphi \colon V \to D$ such that for every constraint $C = \langle \mathbf{s}, R \rangle$ the tuple $\varphi(\mathbf{s})$ belongs to R.

A *global cardinality constraint* for a CSP instance $\mathcal{P}$ is a mapping $\pi : D \to \mathbb{N}$ with $\sum_{a \in D} \pi(a) = |V|$. A solution φ of $\mathcal{P}$ satisfies the cardinality constraint π if the number of variables mapped to each $a \in D$ equals $\pi(a)$. The variant of $\mathrm{CSP}(\Gamma)$ allowing global cardinality constraints will be denoted by $\mathrm{CCSP}(\Gamma)$; the question is, given an instance $\mathcal{P}$ and a cardinality constraint π, whether there is a solution of $\mathcal{P}$ satisfying π.

Example 1 *If Γ is a constraint language on the 2-element set $\{0, 1\}$ then to specify a global cardinality constraint it suffices to specify the exact number of ones we want to have in a solution. This problem is also known as the k-$\mathrm{ONES}(\Gamma)$ problem, [9].*

Sometimes it is convenient to use arithmetic operations on cardinality constraints. Let $\pi, \pi' : D \to \mathbb{N}$ be cardinality constraints on a set D, and $c \in \mathbb{N}$. Then $\pi + \pi'$ and $c\pi$ denote cardinality constraints given by $(\pi + \pi')(a) = \pi(a) + \pi'(a)$ and $(c\pi)(a) = c \cdot \pi(a)$, respectively, for any $a \in D$. Furthermore, we extend addition to sets Π, Π' of cardinality vectors in a convolution sense: $\Pi + \Pi'$ is defined as $\{\pi + \pi' \mid \pi \in \Pi, \pi' \in \Pi'\}$.

Primitive positive definitions and polymorphisms. We now introduce the algebraic tools that will assist us throughout the paper. Let Γ be a constraint language on a set D. A relation R is *primitive positive (pp-) definable* in Γ if it can be expressed using (a) relations from Γ, (b) conjunction, (c) existential quantifiers, and (d) the binary equality relations. The set of all relations pp-definable in Γ will be denoted by $\langle\langle \Gamma \rangle\rangle$.

Example 2 *An important example of pp-definitions that will be used throughout the paper is the* product *of binary relations. Let R, Q be binary relations. Then $R \circ Q$ is the binary relation given by*

$$(R \circ Q)(x, y) = \exists z R(x, z) \wedge Q(z, y).$$

In this paper we will need a slightly weaker notion of definability. We say that R is *pp-definable* in Γ *without equalities* if it can be expressed using only items (a)–(c) from above. The set of all relations pp-definable in Γ without equalities will be denoted by $\langle\langle\Gamma\rangle\rangle'$. Clearly, $\langle\langle\Gamma\rangle\rangle' \subseteq \langle\langle\Gamma\rangle\rangle$. The two sets are different only on relations with redundancies. Let R be a (say, n-ary) relation. A *redundancy* of R is a pair i, j of its coordinate positions such that, for any $\mathbf{a} \in R$, $\mathbf{a}[i] = \mathbf{a}[j]$.

Lemma 3 *For every constraint language Γ, every $R \in \langle\langle\Gamma\rangle\rangle$ without redundancies belongs to $\langle\langle\Gamma\rangle\rangle'$.*

A *polymorphism* of a (say, n-ary) relation R on D is a mapping $f : D^k \to D$ for some k such that for any tuples $\mathbf{a}_1, \ldots, \mathbf{a}_k \in R$ the tuple

$$\begin{aligned} &f(\mathbf{a}_1, \ldots, \mathbf{a}_k) \\ &\quad = \quad (f(\mathbf{a}_1[1], \ldots, \mathbf{a}_k[1]), \ldots, f(\mathbf{a}_1[n], \ldots, \mathbf{a}_k[n])) \end{aligned}$$

belongs to R. Operation f is a polymorphism of a constraint language Γ if it is a polymorphism of every relation from Γ. There is a tight connection, a Galois correspondence, between polymorphisms of a constraint language and relations pp-definable in the language, see [11, 4]. This connection has been extensively exploited to study the ordinary constraint satisfaction problems [16, 8]. Here we do not need the full power of this Galois correspondence, we only need the following result:

Lemma 4 *If operation f is a polymorphism of a constraint language Γ, then it is also a polymorphism of any relation from $\langle\langle\Gamma\rangle\rangle$, and therefore of any relation from $\langle\langle\Gamma\rangle\rangle'$.*

Consistency. Let us fix a constraint language Γ on a set D and let $\mathcal{P} = (V, \mathcal{C})$ be an instance of $\mathrm{CSP}(\Gamma)$. A *partial solution* of $\mathcal{P}$ on a set of variables $W \subseteq V$ is a mapping $\psi : W \to D$ that satisfies the constraint $\langle W \cap \mathbf{s}, \mathrm{pr}_{W\cap\mathbf{s}} R\rangle$ for every $\langle \mathbf{s}, R\rangle \in \mathcal{C}$. Here $W \cap \mathbf{s}$ denotes the subtuple of $\mathbf{s}$ consisting of those entries of $\mathbf{s}$ that belong to W. Instance $\mathcal{P}$ is said to be *k-consistent* if for any k-element set $W \subseteq V$ and any $v \in V \setminus W$ any partial solution on W can be extended to a partial solution on $W \cup \{v\}$. As we only need $k = 2$, all further definitions are given under this assumption.

Any instance $\mathcal{P} = (V, \mathcal{C})$ can be transformed to a 2-consistent instance by means of a standard 2-CONSISTENCY algorithm. This polynomial-time algorithm works as follows. First, for each pair $v, w \in V$ it creates a constraint $\langle (v, w), R_{v,w}\rangle$ where $R_{v,w}$ is the binary relation consisting of all partial solutions ψ on $\{v, w\}$, i.e. $R_{v,w}$ includes pairs $(\psi(v), \psi(w))$. These new constraints are added to $\mathcal{C}$, let the resulting instance be denoted by $\mathcal{P}' = (V, \mathcal{C}')$. Second, for each pair $v, w \in V$, every partial solution $\psi \in R_{v,w}$, and every $u \in V \setminus \{v, w\}$, the algorithm checks if ψ can be extended to a partial solution of $\mathcal{P}'$ on $\{v, w, u\}$. If not it updates $\mathcal{P}'$ by removing ψ from $R_{v,w}$. This step is repeated until no more changes happen.

Lemma 5 *Let $\mathcal{P} = (V, \mathcal{C})$ be an instance of* $\mathrm{CSP}(\Gamma)$.
(a) The problem obtained from $\mathcal{P}$ by applying 2-CONSISTENCY *is 2-consistent;*
(b) On every step of 2-CONSISTENCY *for any pair $v, w \in V$ the relation $R_{v,w}$ belongs to $\langle\langle\Gamma\rangle\rangle'$.*

3 The results

3.1 Decomposability, thick mapping, and cardinality constraints

We introduce several properties of relations that are necessary to describe the relations for which, as we will prove, $\mathrm{CCSP}(\Gamma)$ is solvable in polynomial time.

An n-ary relation R is said to be *2-decomposable* if $\mathbf{a} \in R$ if and only if, for any $i, j \in \{1, \ldots, n\}$, $\mathrm{pr}_{i,j}\mathbf{a} \in \mathrm{pr}_{i,j} R$.

A binary relation $R \subseteq A \times B$ is called a *thick mapping* if there are equivalence relations α and β on A and B, respectively, and a one-to-one mapping $\varphi\colon A/_\alpha \to B/_\beta$ (thus, in particular, $|A/_\alpha| = |B/_\beta|$) such that $(a, b) \in R$ if and only if $b^\beta = \varphi(a^\alpha)$. In this case we shall also say that R is a thick mapping with respect to α, β, and φ. Given a thick mapping R the corresponding equivalence relations will be denoted by α^1_R and α^2_R. Thick mapping R is said to be *trivial* if α^1_R and α^2_R are the total equivalence relations $(\mathrm{pr}_1 R)^2$ and $(\mathrm{pr}_2 R)^2$, respectively.

Observation 6 *Binary relation $R \subseteq A \times B$ is a thick mapping if and only if whenever $(a, c), (a, d), (b, d) \in R$, the pair (b, c) also belongs to R.*

We say that two sets C and D are *non-crossing* if $C \cap D = \emptyset$, or $C \subseteq D$, or $D \subseteq C$. A pair α, β of equivalence relations is *non-crossing* if every α-class C forms a non-crossing pair with every β-class D. Note that this is equivalent to saying that $\alpha \vee \beta = \alpha \cup \beta$ holds, where $\alpha \vee \beta$ denotes the smallest equivalence relation containing both α and β. A pair of thick mappings $R \subseteq A_1 \times A_2$ and $R' \subseteq B_1 \times B_2$ is called *non-crossing* if α^i_R and $\alpha^j_{R'}$ are non-crossing for any $i, j \in \{1, 2\}$.

Observation 7 *If α, β are non-trivial non-crossing equivalence relations, then $\alpha \vee \beta = \alpha \cup \beta$ is non-trivial.*

Lemma 8 *Let R_1, R_2 be a pair of thick mappings.*

(1) $R = R_1 \cap R_2$ is a thick mapping. If R_1, R_2 are non-crossing, then R, R_1 and R, R_2 are also non-crossing.

(2) If R_1, R_2 is a non-crossing pair then $R' = R_1 \circ R_2$ is a thick mapping.

For a set Γ of thick mappings on a set D let $[\Gamma]$ denote the set of binary relations that can be obtained from Γ by means of intersections and products. A set Γ of thick mappings is said to be *non-crossing* if $\Gamma = [\Gamma]$, and the members of Γ are pairwise non-crossing.

A (say, n-ary) relation R is said to be *non-crossing decomposable* if it is 2-decomposable and all the binary projections $\mathrm{pr}_{ij} R$ belong to a certain non-crossing set of thick mappings. Sometimes we need to stress that the binary projections belong to a non-crossing set Δ. Then R is called Δ-non-crossing decomposable. A key point in our algorithmic results is that the property Δ-non-crossing decomposable is closed under pp-definitions (Lemma 14). Note that this is not true for the property 2-decomposable.

Now we are able to state the main result of the paper:

Theorem 9 *Let Γ be a constraint language. The problem $\mathrm{CCSP}(\Gamma)$ is polynomial time if there is a non-crossing set Δ of thick mappings such that every relation from Γ is Δ-non-crossing decomposable and NP-complete otherwise.*

3.2 Meta-Problem

We also consider the *meta-problem* for $\mathrm{CCSP}(\Gamma)$. Suppose D is fixed. Given a finite constraint language Γ on D, decide if $\mathrm{CCSP}(\Gamma)$ is solvable in polynomial time.

Theorem 10 *Let D be a finite set. The meta-problem for $\mathrm{CCSP}(\Gamma)$ is polynomial time solvable.*

To prove Theorem 10 we need several auxiliary statements. For a non-crossing set Δ of thick mappings $\mathsf{Un}(\Delta)$ denotes the set $\{\mathrm{pr}_i R \mid R \in \Delta,\ i \in \{1,2\}\}$; and $\mathsf{Eqv}(\Delta) = \{\alpha_R^1, \alpha_R^2 \mid R \in \Delta\}$. As is easily seen, $\mathsf{Eqv}(\Delta) \subseteq \Delta$, since for any $R \in \Delta$ we have $\alpha_R^1 = R \circ R^{-1}$ and $\alpha_R^2 = R^{-1} \circ R$.

For a subset $A \subseteq D$ by $\mathsf{Sg}_\Delta(A)$ we denote the smallest set from $\mathsf{Un}(\Delta)$ that contains A if $A \subseteq B$ for some $B \in \mathsf{Un}(\Delta)$; otherwise $\mathsf{Sg}_\Delta(A) = D$. Observe that if $B, C \in \mathsf{Un}(\Delta)$ then $B \cap C \in \mathsf{Un}(\Delta)$. Indeed, let $B = \mathrm{pr}_1 R$, $C = \mathrm{pr}_1 R'$ where $R, R' \in \Delta$. Then $\alpha_R^1, \alpha_{R'}^1 \in \Delta$ and $B \cap C = \mathrm{pr}_1(\alpha_R^1 \cap \alpha_{R'}^1)$. Thus there is a unique minimal set in $\mathsf{Un}(\Delta)$ containing A.

Let $A \in \mathsf{Un}(\Delta)$. The set of all equivalence relations from $\mathsf{Eqv}(\Delta)$ that are relations on A is denoted by $\mathsf{Eqv}_\Delta(A)$. For a subset $A \subseteq D$ and a set $B \subseteq A^2$ by $\mathsf{Eg}_{\Delta,A}(B)$ we denote the smallest relation from $\mathsf{Eqv}_\Delta(\mathsf{Sg}_\Delta(A))$ such that $B \subseteq \mathsf{Eg}_{\Delta,A}(B)$. For any $\alpha, \beta \in \mathsf{Eqv}_\Delta(A)$ the relations $\alpha \wedge \beta$ and $\alpha \vee \beta$ belong to $\mathsf{Eqv}_\Delta(A)$. To show that $\alpha \vee \beta \in \mathsf{Eqv}_\Delta(A)$ we need $\alpha \vee \beta = \alpha \cup \beta = \alpha \circ \beta$ that follow from the fact that Δ is non-crossing. Thus $\mathsf{Eg}_{\Delta,A}(B)$ is properly defined.

Lemma 11 *Let $A = \{a,b,c\}$ and $\eta_1 = \mathsf{Eg}_{\Gamma,A}(\{\langle a,b\rangle\})$, $\eta_2 = \mathsf{Eg}_{\Gamma,A}(\{\langle b,c\rangle\})$, $\eta_3 = \mathsf{Eg}_{\Gamma,A}(\{\langle c,a\rangle\})$. Then η_1, η_2, η_3 are all comparable.*

Now let Δ be a non-crossing set on D. We define a ternary operation m that is a polymorphism of Δ and a *majority* operation, that is, m satisfies equations $m(x,x,y) = m(x,y,x) = m(y,x,x) = x$. Let $A = \{a,b,c\} \subseteq D$, and let η_1, η_2, η_3 are given by $\eta_1 = \mathsf{Eg}_{\Delta,A}(\{\langle a,b\rangle\})$, $\eta_2 = \mathsf{Eg}_{\Delta,A}(\{\langle b,c\rangle\})$, $\eta_3 = \mathsf{Eg}_{\Delta,A}(\{\langle c,a\rangle\})$. Then

$$m(a,b,c) = \begin{cases} a, & \text{if } \eta_1 \subseteq \eta_2, \eta_3, \\ b, & \text{if } \eta_2 \subset \eta_1 \text{ and } \eta_2 \subseteq \eta_3, \\ a, & \text{if } \eta_3 \subset \eta_1, \eta_2. \end{cases}$$

Lemma 12 *Operation m is a majority operation and is a polymorphism of Δ.*

Corollary 13 *Let Δ be a non-crossing set of thick mappings and Γ is a set of Δ-non-crossing decomposable relations. Then Γ has a majority polymorphism.*

Proof: (of Theorem 10) By Theorem 9, given a constraint language Γ, it suffices to check whether or not Γ is Δ-non-crossing decomposable for a certain non-crossing set of thick mappings Δ.

Set Δ_0 to be the set of all binary projections of relations from Γ. It follows from the definition of non-crossing decomposable constraint languages, that if Γ is Δ'-non-crossing decomposable for some Δ' then it is Δ-non-crossing decomposable for $\Delta = [\Delta_0]$. First, compute Δ by setting initially $\Delta = \Delta_0$, and then iteratively finding intersections and products of relations from Δ and adding the result to Δ if it is not already there. Since D is fixed, the maximal number of members in Δ, and therefore the number of iterations of the process above is bounded by the constant $2^{|D|^2}$. Second, check if Δ contains a relation that is not a thick mapping, and that all pairs of thick mappings are non-crossing. Again, as the number of relations in Δ is bounded by a constant, this can be done in constant time. Third, construct the majority operation m as described above. Finally, check if m is a polymorphism of Γ. This last step can be done in a time cubic in the total size of relations in Γ, since it suffices for each relation $R \in \Gamma$ to apply m to every triple of tuples in R. By Corollary 13, if Γ is Δ-non-crossing decomposable then m is a polymorphism of Γ. On the other hand, if m is a polymorphism of Γ then by [1] Γ is 2-decomposable. Furthermore, as is checked before, all binary projections of relations from Γ belong to the non-crossing set Δ, implying Γ is non-crossing decomposable. $\square$

4 Algorithm

In this section we fix a non-crossing set Δ of thick mappings, and a Δ-non-crossing decomposable set Γ. We present a polynomial-time algorithm for solving $\mathrm{CCSP}(\Gamma)$ in this case.

4.1 Prerequisites

Let Γ be a constraint language and let $\mathcal{P} = (V, \mathcal{C})$ be a 2-consistent instance of $\mathrm{CCSP}(\Gamma)$. By $\mathsf{bin}(\mathcal{P})$ we denote the instance $(V, \mathcal{C}')$ such that $\mathcal{C}'$ is the set of all constraints of the form $\langle (v, w), R_{v,w} \rangle$ where $v, w \in V$ and $R_{v,w}$ is the set of all partial solutions on $\{v, w\}$.

Lemma 14 *Let Δ be a non-crossing set of thick mappings, and let Γ be a set of Δ-non-crossing decomposable relations.*

(1) Any R pp-definable in Γ is Δ-non-crossing decomposable.

(2) If $\mathcal{P}$ is a 2-consistent instance of $\mathrm{CCSP}(\Gamma)$ then $\mathsf{bin}(\mathcal{P})$ has the same solutions as $\mathcal{P}$.

Let $\mathcal{P} = (V, \mathcal{C})$ be an instance of $\mathrm{CCSP}(\Gamma)$. Applying algorithm 2-CONSISTENCY we may assume that $\mathcal{P}$ is 2-consistent, and, by Lemma 14, as all relations of Γ are 2-decomposable, that every constraint relation of $\mathcal{P}$ is 2-decomposable, and therefore every constraint of $\mathcal{P}$ can be assumed to be binary, and every constraint relation belongs to $[\Delta] = \Delta$. Let constraints of $\mathcal{P}$ be $\langle (v, w), R_{vw} \rangle$ for each pair of different $v, w \in V$. Let $\mathcal{S}_v$, $v \in V$, denote the set of $a \in D$ such that there is a solution φ of $\mathcal{P}$ with $\varphi(v) = a$. By [15] if a constraint language has a majority polymorphism, then every 2-consistent problem is *globally consistent*, that is every partial solution can be extended to a global solution of the problem. In particular, $\mathcal{P}$ is globally consistent, therefore, $\mathcal{S}_v = \mathrm{pr}_1 R_{vw}$ for any $w \in V$, $w \neq v$. Constraint $\langle (v, w), R_{vw} \rangle$ is said to be *trivial* if $R_{vw} = \mathcal{S}_v \times \mathcal{S}_w$, otherwise it is said to be *non-trivial*.

The *graph of $\mathcal{P}$*, denoted $G(\mathcal{P})$, is a graph with vertex set V and edge set $E = \{vw \mid v, w \in V \text{ and } \langle (v, w), R_{vw} \rangle \text{ is non-trivial}\}$.

Observation 15 *By the 2-consistency of $\mathcal{P}$, for any $u, v, w \in V$, $R_{uv} \subseteq R_{uw} \circ R_{wv}$.*

Lemma 16 *Let R, R' be a non-crossing pair of non-trivial thick mappings such that $\mathrm{pr}_2 R = \mathrm{pr}_1 R'$. Then $R \circ R'$ is also non-trivial.*

Suppose that $G(\mathcal{P})$ is connected and fix $v \in V$. By Observation 15 and Lemma 16, for any $w \in V$ the constraint $\langle (v, w), R_{vw} \rangle$ is non-trivial. Note that due to 2-consistency, all the $\alpha^1_{R_{vw}}$ are over the same set. Set $\eta_v = \bigvee_{w \in V - \{v\}} \alpha^1_{R_{vw}}$. If $|V| = 1$ we set η_v to be the equality relation.

Lemma 17 *If $G(\mathcal{P})$ is connected then the equivalence relations η_v and $\alpha^1_{R_{vw}}$ (for any $w \in V - \{v\}$) are non-trivial.*

Lemma 18 *Suppose $G(\mathcal{P})$ is connected.*

(1) For any $v, w \in V$ there is a one-to-one correspondence ψ_{vw} between $\mathcal{S}_v/_{\eta_v}$ and $\mathcal{S}_w/_{\eta_w}$ such that for any solution φ of $\mathcal{P}$ if $\varphi(v) \in A \in \mathcal{S}_v/_{\eta_v}$, then $\varphi(w) \in \psi_{vw}(A) \in \mathcal{S}_w/_{\eta_w}$.

(2) The mappings ψ_{vw} are consistent, i.e. for any $u, v, w \in V$ we have $\psi_{uw}(x) = \psi_{vw}(\psi_{uv}(x))$ for every x.

4.2 Algorithm

We split the algorithm into two parts. Algorithm CARDINALITY (Figure 1) just ensures 2-consistency and initializes a recursive process. The main part of the work is done by EXT-CARDINALITY (Figure 2).

Algorithm EXT-CARDINALITY solves the more general problem of computing the set of all cardinality constraints π that can be satisfied by a solution of $\mathcal{P}$. Thus it can be used to solve directly CSP with *extended global cardinality constraints*, where the input contains a set Π of allowed cardinality constraints and the solution can satisfy any one of them.

The algorithm considers three cases. Step 2 handles the trivial case when the instance consists of a single variable and there is only one possible value it can be assigned. Otherwise, we decompose the instance either by partitioning the variables or by partitioning the domain of the variables. If $G(\mathcal{P})$ is not connected, then the satisfying assignments of $\mathcal{P}$ can be obtained from the satisfying assignments of the connected components. Thus a cardinality constraint π can be satisfied if it arises as the sum $\pi_1 + \cdots + \pi_k$ of cardinality constraints such that the i-th component has a solution satisfying π_i. Instead of considering all such sums (which would not be possible in polynomial time), we follow the standard dynamic programming approach of going through the components one by one, and determining all possible cardinality constraints that can be satisfied by a solution for the first i components (Step 3).

If the graph $G(\mathcal{P})$ is connected, then we fix a variable v_0 and go through each class A of the partition η_{v_0} (Step 4). If v_0 is restricted to A, then this implies a restriction for every other variable w. We recursively solve the problem for the restricted instance arising for each class A; if constraint π can be satisfied, then it can be satisfied for one of the restricted instances.

The correctness of the algorithm follows from the discussion above. The only point that has to be verified is that the instance remains 2-consistent after the recursion. This is obvious if we recurse on the connected components (Step 3). In Step 4, 2-consistency follows from the fact that if $(a, b) \in R_{vw}$ can be extended by $c \in S_u$, then in every subproblem either these three values satisfy the instance restricted to $\{v, w, u\}$ or a, b, c do not appear in the domain of v, w, u, respectively.

To show that the algorithm runs in polynomial time, observe first that every step of the algorithm (except the recursive calls) can be done in polynomial time. Here we use that D is fixed, thus the size of the set Π is polynomially bounded. Thus we only need to bound the size of the recursion tree. If we recurse in Step 3, then we produce instances whose graphs are connected, thus it cannot be followed by recursing again in Step 3. In Step 4, the domain of every variable is decreased: by Lemma 17, η_w is nontrivial for any variable w. Thus in any branch of the recursion tree, recursion in Step 4 can occur at most $|D|$ times, hence the depth of the recursion tree is $O(|D|)$. As the number of branches is polynomial in each step, the size of the recursion tree is polynomial.

INPUT: An instance $\mathcal{P} = (V, \mathcal{C})$ of $\mathrm{CCSP}(\Gamma)$, and a cardinality constraint π
OUTPUT: YES if $\mathcal{P}$ has a solution satisfying π, NO otherwise

Step 1. **apply** 2-CONSISTENCY to $\mathcal{P}$
Step 2. **set** Π :=EXT-CARDINALITY($\mathcal{P}$)
Step 3. **if** $\pi \in \Pi$ **output** YES
else output NO

Figure 1. Algorithm CARDINALITY**.**

5 Definable relations, constant relations, and the complexity of CCSP

We present two reductions that will be crucial for the proofs in Section 6. In Section 5.1, we show that adding relations that are pp-definable (without equalities) does not make the problem harder, while in Section 5.2, we show the same for unary constant relations.

5.1 Definable relations and the complexity of cardinality constraints

Theorem 19 *Let Γ be a constraint language and R a relation pp-definable in Γ without equalities. Then $\mathrm{CCSP}(\Gamma \cup \{R\})$ is reducible to $\mathrm{CCSP}(\Gamma)$.*

Proof (sketch): We proceed by induction on the structure of pp-formulas. The base case of induction is given by $R \in \Gamma$. There are two cases: when R is defined by conjunction of two relations, and when $R(x_1, \ldots, x_n) = \exists x R'(x_1, \ldots, x_n, x)$. In the first case it suffices to replace in an instance of $\mathrm{CCSP}(\Gamma)$ every constraint using R with two constraints using the conjuncts. So, we consider the second case.

INPUT: A 2-consistent instance $\mathcal{P} = (V, \mathcal{C})$ of $\mathrm{CCSP}(\Gamma)$
OUTPUT: The set of cardinality constraints π such that $\mathcal{P}$ has a solution that satisfies π

Step 1. **construct** the graph $G(\mathcal{P}) = (V, E)$
Step 2. **if** $|V| = 1$ and the domain of this variable is a singleton $\{a\}$ **then do**
Step 2.1 **set** $\Pi := \{\pi\}$ where $\pi(x) = 0$ except $\pi(a) = 1$
Step 3. **else if** $G(\mathcal{P})$ is disconnected and $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ are its connected components **do**
Step 3.1 **set** $\Pi := \{\pi\}$ where $\pi(x) = 0$
Step 3.2 **for** $i = 1$ **to** k **do**
Step 3.2.1 **set** $\Pi := \Pi +$ EXT-CARDINALITY($\mathcal{P}_{|V_i}$)
endfor
endif
Step 4. **else do**
Step 4.1 **for each** $v \in V$ **find** η_v
Step 4.2 **fix** $v_0 \in V$ **and set** $\Pi := \emptyset$
Step 4.3 **for each** η_{v_0}-class A **do**
Step 4.3.1 **set** $\mathcal{P}_A := (V, \mathcal{C}_A)$ where for every $v, w \in V$ the set $\mathcal{C}_A$ includes the constraint $\langle (v, w), R_{vw} \cap (\psi_{v_0 v}(A) \times \psi_{v_0 w}(A)) \rangle$
Step 4.3.2 **set** $\Pi := \Pi \cup$ EXT-CARDINALITY($\mathcal{P}_A$)
endfor
enddo
Step 4. **output** Π

Figure 2. Algorithm EXT-CARDINALITY**.**

Let $\mathcal{P} = (V, \mathcal{C})$ be a $\mathrm{CCSP}(\Gamma \cup \{R\})$ instance. W.l.o.g. let $C_1, \ldots, C_q$ be the constraints involving R. Instance $\mathcal{P}'$ of $\mathrm{CCSP}(\Gamma)$ is constructed as follows.
Variables: Replace every variable v from V with a set W_v of variables of size $q|D|$ and introduce a set of $|D|$ variables for each constraint involving R. Formally,

$$W = \bigcup_{v \in V} W_v \cup \{w_1, \ldots, w_q\} \cup \bigcup_{i=1}^{q} \{w_i^1, \ldots, w_i^{|D|-1}\}.$$

Non-R constraints: For every $C_i = \langle (v_1, \ldots, v_\ell), Q \rangle$ with $i > q$, introduce all possible constraints of the form $\langle (u_1, \ldots, u_\ell), Q \rangle$, where $u_j \in W_{v_j}$ for $j \in \{1, \ldots, \ell\}$.
R constraints: For every $C_i = \langle (v_1, \ldots, v_\ell), R \rangle$, $i \leq q$, introduce all possible constraints of the form $\langle (u_1, \ldots, u_\ell, w_i), R' \rangle$, where $u_j \in W_{v_j}$, $j \in \{1, \ldots, \ell\}$.

It is not hard to see that if $\mathcal{P}$ has a solution satisfying cardinality constraint π then $\mathcal{P}'$ has a solution satisfying the cardinality constraint $\pi' = |W_v| \cdot \pi + q$. Thus it suffices to show that if $\mathcal{P}'$ has a solution ψ satisfying π', then $\mathcal{P}$ has a solution satisfying π.

Let $a \in D$ and $U_a(\psi) = \psi^{-1}(a) = \{u \in W \mid \psi(u) = a\}$. Observe first that if $\varphi : V \to D$ is a mapping such that $U_{\varphi(v)}(\psi) \cap W_v \neq \emptyset$ for every $v \in V$ (i.e., $\varphi(v)$ appears on at least one variable $v' \in W_v$ in ψ), then φ satisfies all the constraints of $\mathcal{P}$. Then we show that it is possible to construct such a φ that also satisfies the cardinality constraint π. Since $|W_v| = q|D|$, even if set $U_a(\psi)$ contains all $q|D|$ variables of the form w_i and w_i^j, it has to intersect at least $\pi(a)$ sets W_v. Using this observation we construct a bipartite graph indicating which intersections $U_a(\psi) \cap W_v$ are nonempty, show that required solutions correspond to perfect matchings in this graph, and prove that such a perfect matching exists using Hall's Theorem.

5.2 Constant relations and the complexity of cardinality constraints

Let D be a set, and let $a \in D$. The *constant relation* C_a is the unary relation that contains only one tuple, (a). If a constraint language Γ over D contains all the constant relations, then they can be used in the corresponding constraint satisfaction problem to force certain variables to take some fixed values. The goal of this section is to show that for any constraint language Γ the problem $\mathrm{CCSP}(\Gamma \cup \{C_a \mid a \in D\})$ is polynomial time reducible to $\mathrm{CCSP}(\Gamma)$. For the ordinary decision CSP such a reduction exists when Γ does not have unary polymorphisms that are not permutations, see [8].

Let R be a (say, n-ary) relation on a set D, and let f be a mapping from D to 2^D, the powerset of D. Mapping f is said to be a *multi-valued morphism* of R if for any tuple $(a_1, \dots, a_n) \in R$ the set $f(a_1) \times \dots \times f(a_n)$ is a subset of R. Mapping f is a multi-valued morphism of a constraint language Γ if it is a multi-valued morphism of every relation in Γ. Mappings of this kind are also known as *hyperoperations*, see e.g. [20].

Theorem 20 *Let Γ be a finite constraint language over a set D. Then* $\mathrm{CCSP}(\Gamma \cup \{C_a \mid a \in D\}) \leq \mathrm{CCSP}(\Gamma)$.

Proof: Let $D = \{d_1, \dots, d_k\}$ and $a = d_1$. We show that $\mathrm{CCSP}(\Gamma \cup \{C_a\}) \leq \mathrm{CCSP}(\Gamma)$. This clearly implies the result. We make use of the following multi-valued morphism gadget $\mathsf{MVM}(\Gamma, n)$ (i.e. a CSP instance). Observe that it is somewhat similar to the *indicator problem* [17].

- The set of variables is $V(n) = \bigcup_{i=1}^k V_{d_i}$, where V_{d_i} contains n^{k+1-i} elements. All sets V_{d_i} are assumed to be disjoint.
- The constraints are as follows: For every $R \in \Gamma$ and every $(a_1, \dots, a_r) \in R$ we include all possible constraints of the form $\langle (v_1, \dots, v_r), R \rangle$ where $v_i \in V_{a_i}$ for $i \in \{1, \dots, k\}$.

Given an instance $\mathcal{P} = (V, \mathcal{C})$ of $\mathrm{CCSP}(\Gamma \cup \{C_a\})$, we construct instance $\mathcal{P}' = (V', \mathcal{C}')$ of $\mathrm{CCSP}(\Gamma)$.

- Let $W \subseteq V$ be the set of variables, on which the constant relation C_a is imposed, that is, $\mathcal{C}$ contains the constraint $\langle (v), C_a \rangle$. Set $n = |V|$. The set V' of variables of $\mathcal{P}'$ is the disjoint union of the set $V(n)$ of variables of $\mathsf{MVM}(\Gamma, n)$ and $V \setminus W$.
- The set $\mathcal{C}'$ of constraints of $\mathcal{P}'$ consists of three parts:
  - (a) $\mathcal{C}'_1$, the constraints of $\mathsf{MVM}(\Gamma, n)$;
  - (b) $\mathcal{C}'_2$, the constraints of $\mathcal{P}$ that do not include variables from W;
  - (c) $\mathcal{C}'_3$, for any constraint $\langle (v_1, \dots, v_m), R \rangle \in \mathcal{C}$ whose scope contains variables constrained by C_a (without loss of generality let $v_1, \dots, v_\ell$ be such variables), $\mathcal{C}'_3$ contains all constraints of the form $\langle (w_1, \dots, w_k, v_{\ell+1}, \dots, v_m), R \rangle$, where $w_1, \dots, w_\ell \in V_a$.

We show that $\mathcal{P}$ has a solution satisfying a cardinality constraint π if and only if $\mathcal{P}'$ has a solution satisfying cardinality constraint π' given by

$$\pi'(d_i) = \begin{cases} \pi(a) + (|V_a| - |W|), & \text{if } i = 1, \\ \pi(d_i) + |V_{d_i}|, & \text{otherwise.} \end{cases}$$

Suppose that $\mathcal{P}$ has a right solution φ. Then a required solution for $\mathcal{P}'$ is given by

$$\psi(v) = \begin{cases} \varphi(v), & \text{if } v \in V \setminus W, \\ d_i, & \text{if } v \in V_{d_i}. \end{cases}$$

It is clear that ψ is a solution to $\mathcal{P}'$ and it satisfies π'.

Suppose that $\mathcal{P}'$ has a solution ψ that satisfies π'. Since $\pi'(a) > |V' \setminus V_a|$ (using $d_1 = a$), there is a $v \in V_a$ such that $\psi(v) = a$. Thus the assignment

$$\varphi(v) = \begin{cases} \psi(v), & \text{if } v \in V \setminus W, \\ a & \text{if } v \in W \end{cases}$$

is a satisfying assignment $\mathcal{P}$, but it might not satisfy π. Using the following claims one can show that $\mathcal{P}'$ has a solution ψ, where φ obtained this way satisfies π. Observe that what we need is that in ψ value d_i appears on exactly $\pi'(d_i) - |V_{d_i}|$ variables of $V \setminus W$.

CLAIM 1. Mapping f taking every $d_i \in D$ to the set $\{\psi(v) \mid v \in V_{d_i}\}$ is a multi-valued morphism of Γ.

Proof of this claim is straightforward.

CLAIM 2. Let f be the mapping defined in Claim 1. Then f^* defined by $f^*(b) := f(b) \cup \{b\}$ for every $b \in D$ is also a multi-valued morphism of Γ.

We show that for every $d_i \in D$, there is an $m_i \geq 1$ such that $d_i \in f^j(d_i)$ for every $j \geq m_i$. Taking the maximum

m of these values, we get $d_i \in f^{m+1}(d_i)$ and $f(d_i) \subseteq f^{m+1}(d_i)$ (as $d_i \in f^m(d_i)$) for every i, proving the claim.

The proof is by induction on i. If $d_i \in f(d_i)$, then we are done as we can set $m_i = 1$ (note that this is always the case for $i = 1$, since we observed above that value d_1 has to appear on a variable of V_{d_1})). So let us suppose that $d_i \notin f(d_i)$. Let $D_i = \{d_1, \dots, d_i\}$ and let $g_i : D_i \to 2^{D_i}$ defined by $g_i(d_j) := f(d_j) \cap D_i$. Observe that $g_i(d_j)$ is well-defined, i.e., $g_i(d_j) \neq \emptyset$: the set V_{d_j} contains $n^{k+1-j} \geq n^{k+1-i}$ variables, while the number of variables where values not from D_i appear is $\sum_{\ell=i+1}^{k} \pi'(d_\ell) \leq n + \sum_{\ell=i+1}^{k} n^{k+1-\ell} < n^{k+1-i}$.

Let $T := \bigcup_{\ell \geq 1} g_i^\ell(d_i)$. We claim that $d_i \in T$. Suppose that $d_i \notin T$. By the definition of T and the assumption $d_i \notin f(d_i)$, for every $b \in T \cup \{d_i\}$, the variables in V_b can have values only from T and from $D \setminus D_i$. The total number of variables in V_b, $b \in T \cup \{d_i\}$ is $\sum_{b \in T \cup \{d_i\}} n^{k+1-b}$, while the total cardinality constraint of the values from $T \cup (D \setminus D_i)$ is

$$\sum_{b \in T \cup (D \setminus D_i)} \pi'(b) < n + \sum_{b \in T} n^{k+1-b} + \sum_{\ell=i+1}^{k} n^{k+1-\ell}$$
$$< \sum_{b \in T} n^{k+1-b} + n^{k+1-i} = \sum_{b \in T \cup \{d_i\}} n^{k+1-b},$$

a contradiction. Thus $d_i \in T$, that is, there is a value $j < i$ such that $d_j \in f(d_i)$ and $d_i \in f^s(d_j)$ for some $s \geq 1$. By the induction hypothesis, $d_j \in f^m(d_j)$ for every $m \geq n_j$, thus we have that $d_i \in f^m(d_i)$ if m is at least $m_i := 1 + m_j + s$. This concludes the proof of Claim 2.

Let D^+ (resp., D^-) be the set of those values $d_i \in D$ that appear on more than (resp., less than) $\pi'(i) - |V_{d_i}|$ variables of $V \setminus W$. It is clear that if $|D^+| = |D^-| = 0$, then φ obtained from ψ satisfies π. Futhermore, if $|D^+| = 0$, then $|D^-| = 0$ as well. Thus suppose that $D^+ \neq \emptyset$ and let $S := \bigcup_{b \in D^+, s \geq 1} f^s(b)$.

CLAIM 3. $S \cap D^- \neq \emptyset$.

We skip the proof this claim.

By Claim 3, there is a value $d^- \in S \cap D^-$, which means that there is a $d^+ \in D^+$ and a sequence of distinct values $b_0 = d^+, b_1, \dots, b_\ell = d^-$ such that $b_{i+1} \in f(b_i)$ for every $0 \leq i < \ell$. Let $v \in V \setminus W$ be an arbitrary variable with value d^+. We modify ψ the following way:

1. The value of v is changed from d^+ to d^-.
2. For every $0 \leq i < \ell$, one variable in V_{b_i} with value b_{i+1} is changed to b_i.

Note that these changes do not modify the cardinalities of the values: the second step increases only the cardinality of $b_0 = d^+$ (by one) and decreases only the cardinality of $b_\ell = d^-$ (by one). We have to argue that the transformed assignment still satisfies the constraints of $\mathcal{P}'$. Since $d^- \in f^\ell(d^+)$, the multi-valued morphism f^* of Claim 2 implies that changing d^+ to d^- on a single variable and not changing anything else also gives a satisfying assignment. The rest of the proof is fairly straightforward. □

We will use the following simple lemma:

Lemma 21 *Let α be an equivalence relation on a set D and $a \in D$. Then $a^\alpha \in \langle\langle \alpha, C_a \rangle\rangle'$.*

6 Hardness

We prove that if Γ does not satisfy the conditions of Theorem 9 then $\mathrm{CCSP}(\Gamma)$ is NP-complete.

For a (say, n-ary) relation R over a set D and a subset $D' \subseteq D$, by $R_{|D'}$ we denote the relation $\{(a_1, \dots, a_n) \mid (a_1, \dots, a_n) \in R \text{ and } a_1, \dots, a_n \in D'\}$. For a constraint language Γ over D we use $\Gamma_{|D'}$ to denote the constraint language $\{R_{|D'} \mid R \in \Gamma\}$. We can easily simulate the restriction to a subset of the domain by setting to 0 the cardinality constraint on the unwanted values:

Lemma 22 *For any constraint language Γ over a set D and any $D' \subseteq D$, the problem $\mathrm{CCSP}(\Gamma_{|D'})$ is polynomial time reducible to $\mathrm{CCSP}(\Gamma)$.*

Suppose now that a constraint language Γ does not satisfy the conditions of Theorem 9. Then one of the following cases takes place: (a) $\langle\langle\Gamma\rangle\rangle'$ contains a binary relation wich is not a thick mapping; or (b) $\langle\langle\Gamma\rangle\rangle'$ contains two equivalence relations that are not a non-crossing pair; or (c) Γ contains a relation which is not 2-decomposable. We consider these three cases in turn.

One of the NP-complete problems we will reduce to $\mathrm{CCSP}(R)$ is the BIPARTITE INDEPENDENT SET problem (or BIS for short). In this problem, given a connected bipartite graph G with bipartition V_1, V_2 and numbers k_1, k_2, the goal is to verify if there exists an independent set S of G such that $|S \cap V_1| \geq k_1$ and $|S \cap V_2| \geq k_2$. It is easy to see that the problem is hard even for graphs containing no isolated vertices. By representing the edges of a bipartite graph with the relation $R = \{(a, c), (a, d), (b, d)\}$, we can express the problem of finding an bipartite independent set. Value b (resp., a) represents selected (resp., unselected) vertices in V_1, while value c (resp., d) represents selected (resp., unselected) vertices in V_2. With this interpretation, the only combination that relation R excludes is that two selected vertices are adjacent. By Observation 6, if a binary relation is not a thick mapping, then it contains something very similar to R. However, some of the values a, b, c, and d might coincide and the relation might contain further tuples such as (c, d). Thus we need a delicate case analysis to show that the problem is NP-hard for binary relations that are not thick mappings.

Lemma 23 *Let R be a binary relation which is not a thick mapping. Then $\mathrm{CCSP}(\{R\})$ is NP-complete.*

Next we show hardness in the case when there are two equivalence relations that are crossing.

Lemma 24 *Let R, Q be a crossing pair of equivalence relations. Then $\mathrm{CCSP}(\{R, Q\})$ is NP-complete.*

Proof: Let R, Q be equivalence relations on D and D', respectively. As these relations are not a non-crossing pair there are $a, b, c \in D \cap D'$ such that $\langle a, c\rangle \in R \setminus Q$ and $\langle c, b\rangle \in Q \setminus R$. Let $R' = R_{|\{a,b,c\}}$ and $Q' = Q_{|\{a,b,c\}}$. Clearly,

$$R' = \{(a, a), (b, b), (c, c), (a, c), (c, a)\},$$
$$Q' = \{(a, a), (b, b), (c, c), (b, c), (c, b)\}.$$

By Lemma 22, $\mathrm{CCSP}(\{R', Q'\})$ is polynomial time reducible to $\mathrm{CCSP}(\{R, Q\})$. Consider $R''(x, y) = \exists z(R'(x, z) \wedge Q'(z, y))$. We have that $\mathrm{CCSP}(R'')$ is reducible to $\mathrm{CCSP}(\{R', Q'\})$ and

$$R'' = \{(a, a), (b, b), (c, c), (a, c), (c, a), (b, c), (c, b), (a, b)\}.$$

Observe that R'' is not a thick mapping and by Lemma 23, $\mathrm{CCSP}(R'')$ is NP-complete. $\square$

Finally, we prove hardness in the case when there is a relation that is not 2-decomposable. An example of such a relation is a ternary Boolean affine relation $x + y + z = c$ for $c = 0$ or $c = 1$. The CSP with global cardinality constraints for this relation is NP-complete by [9]. Our strategy is to obtain such a relation from a relation that is not 2-decomposable. However, as in Lemma 23, we have to consider several cases.

Lemma 25 *Let R be a relation whose binary projections is contained in a non-crossing set of thick mappings, but R is not 2-decomposable. Then $\mathrm{CCSP}(\{R\})$ is NP-complete.*

Proof: We choose R to be the 'smallest' non-2-decomposable relation in the sense that every relation $R' \in \langle\langle\{R\} \cup \{C_a \mid a \in D\}\rangle\rangle'$ that either have smaller arity, or $R' \subset R$, is non-crossing decomposable, and every relation obtained from R by restricting on a proper subset of D is also non-crossing decomposable. By Theorems 19, 20, and Lemmas 22, 23, 24, it suffices to consider relations satisfying these conditions.

Relation R is ternary. Clearly, it is not binary; suppose that its arity is more than 3. Let $\mathbf{a} \notin R$ be a tuple such that $\mathrm{pr}_{ij}\mathbf{a} \in \mathrm{pr}_{ij} R$ for any i, j. Let

$$R'(x, y, z) = \exists x_4, \ldots, x_n(R(x, y, z, x_4, \ldots, x_n) \wedge C_{\mathbf{a}[4]}(x_4) \wedge \ldots \wedge C_{\mathbf{a}[n]}(x_n)).$$

By the minimality of R all binary projections of R' are pairwise non-crossing thick mappings. It is straightforward that $(\mathbf{a}[1], \mathbf{a}[2], \mathbf{a}[3]) \notin R'$, while, since any proper projection of R is 2-decomposable, $\mathrm{pr}_{\{2,\ldots,n\}}\mathbf{a} \in \mathrm{pr}_{\{2,\ldots,n\}} R$, $\mathrm{pr}_{\{1,3,\ldots,n\}}\mathbf{a} \in \mathrm{pr}_{\{1,3,\ldots,n\}} R$, $\mathrm{pr}_{\{1,2,4,\ldots,n\}}\mathbf{a} \in \mathrm{pr}_{\{1,2,4,\ldots,n\}} R$, implying $(\mathbf{a}[1], \mathbf{a}[2]) \in \mathrm{pr}_{12} R'$, $(\mathbf{a}[2], \mathbf{a}[3]) \in \mathrm{pr}_{23} R'$, $(\mathbf{a}[1], \mathbf{a}[3]) \in \mathrm{pr}_{13} R'$. Thus R' is not 2-decomposable, a contradiction.

Let $(a, b, c) \notin R$ and $(a, b, d), (a, e, c), (f, b, c) \in R$, and let $B = \{a, b, c, d, e, f\}$. As $R_{|B}$ is not 2-decomposable, we should have $R = R_{|B}$.

If $R_{12} = \mathrm{pr}_{12} R$ is a thick mapping with respect to η_{12}, η_{21}, $R_{13} = \mathrm{pr}_{13} R$ is a thick mapping with respect to η_{13}, η_{31}, and $R_{23} = \mathrm{pr}_{23} R$ is a thick mapping with respect to η_{23}, η_{32}, then $\langle a, f\rangle \in \eta_{12} \cap \eta_{13}$, $\langle b, e\rangle \in \eta_{21} \cap \eta_{23}$, and $\langle c, d\rangle \in \eta_{31} \cap \eta_{32}$. Let the corresponding classes of $\eta_{12} \cap \eta_{13}$, $\eta_{21} \cap \eta_{23}$, and $\eta_{31} \cap \eta_{32}$ be B_1, B_2, and B_3, respectively. Then $B_1 = \mathrm{pr}_1 R$, $B_2 = \mathrm{pr}_2 R$, $B_3 = \mathrm{pr}_3 R$. Indeed, if one of these equalities is not true, since by Lemma 21 B_1, B_2, B_3 are pp-definable in R without equalities, the relation $R'(x, y, z) = R(x, y, z) \wedge B_1(x) \wedge B_2(y) \wedge B_3(z)$ is pp-definable in R and the constant relations, is smaller than R, and is not 2-decomposable.

Next we show that $(a, g) \in \mathrm{pr}_{12} R$ for all $g \in \mathrm{pr}_2 R$. If there is g with $(a, g) \notin \mathrm{pr}_{12} R$ then setting $C(y) = \exists z(\mathrm{pr}_{12} R(z, y) \wedge C_a(z))$ we have $b, e \in C$ and $C \neq \mathrm{pr}_2 R$. Thus $R'(x, y, z) = R(x, y, z) \wedge C(y)$ is smaller than R and is not 2-decomposable. The same is true for a and $\mathrm{pr}_3 R$, and for b and $\mathrm{pr}_3 R$. Since every binary projection of R is a thick mapping this implies that $\mathrm{pr}_{12} R = \mathrm{pr}_1 R \times \mathrm{pr}_2 R$, $\mathrm{pr}_{23} R = \mathrm{pr}_2 R \times \mathrm{pr}_3 R$, and $\mathrm{pr}_{13} R = \mathrm{pr}_1 R \times \mathrm{pr}_3 R$.

For each $i \in \{1, 2, 3\}$ and every $p \in \mathrm{pr}_i R$, the relation $R_i^p(x_j, x_k) = \exists x_i(R(x_1, x_2, x_3) \wedge C_p(x_i))$, where $\{j, k\} = \{1, 2, 3\} \setminus \{i\}$, is definable in R and therefore is a thick mapping with respect to, say, η_{ij}^p, η_{ik}^p. Our next step is to show that R can be chosen such that η_{ij}^p does not depend on the choice of p and i.

If one of these relations, say, R_1^p, equals $\mathrm{pr}_2 R \times \mathrm{pr}_3 R$, while another one, say R_1^q does not, then consider R_3^c. We have $\{p\} \times \mathrm{pr}_2 R \subseteq R_3^c$. Moreover, since by the choice of R relation R_1^q is a non-trivial thick mapping there is $r \in \mathrm{pr}_2 R$ such that $(r, c) \notin R_1^q$, hence $(q, r) \notin R_3^c$. Therefore R_3^c is not a thick mapping, a contradiction. Since R_1^a does not equal $\mathrm{pr}_2 R \times \mathrm{pr}_3 R$, every η_{ij}^p is non-trivial. Let

$$\eta_i = \bigvee_{\substack{j \in \{1,2,3\} \setminus \{i\} \\ p \in \mathrm{pr}_j R}} \eta_{ji}^p = \bigcup_{\substack{j \in \{1,2,3\} \setminus \{i\} \\ p \in \mathrm{pr}_j R}} \eta_{ji}^p.$$

As we observed before Lemma 11, η_i is pp-definable in R and constant relations without equalities. Since all the η_{ji}^p are non-trivial, η_i is also non-trivial. We set

$$R'(x, y, z) = \exists x', y', z'(R(x', y', z') \wedge \eta_1(x, x') \wedge \eta_2(y, y') \wedge \eta_3(z, z')).$$

Let Q_i^p be defined for R' in the same way as R_i^p for R. Observe that since $(p,q,r) \in R'$ if and only if there is $(a',b',c') \in R$ such that $\langle a,a'\rangle \in \eta_1$, $\langle b,b'\rangle \in \eta_2$, $\langle c,c'\rangle \in \eta_3$, the relations Q_1^p, Q_2^q, Q_3^r for $p \in \mathrm{pr}_1 R', q \in \mathrm{pr}_2 R', r \in \mathrm{pr}_3 R'$ are thick mappings with respect to the equivalence relations η_1,η_2, relations η_2,η_3, and relations η_1,η_3, respectively. Since all the binary projections of R' equal to the direct product of the corresponding unary projections and η_1,η_2,η_3 are non-trivial, R' is not 2-decomposable, and we can replace R with R'. Thus we have achieved that η_{ij}^p does not depend on the choice of p and i.

Next we show that R can be chosen such that $\mathrm{pr}_1 R = \mathrm{pr}_2 R = \mathrm{pr}_3 R$, $\eta_1 = \eta_2 = \eta_3$, and for each $i \in \{1,2,3\}$ there is $r \in \mathrm{pr}_i R$ such that R_i^r is a reflexive relation. If, say, $\mathrm{pr}_1 R \neq \mathrm{pr}_2 R$, or $\eta_1 \neq \eta_2$, or R_3^r is not reflexive for any $r \in \mathrm{pr}_3 R$, consider the following relation

$$R'(x,y,z) = \exists y', z'(R(x,y',z) \wedge R(y,y',z') \wedge C_d(z')).$$

First, observe that $\mathrm{pr}_{ij} R' = \mathrm{pr}_i R' \times \mathrm{pr}_j R'$ for any $i,j \in \{1,2,3\}$. Then, for any fixed $r \in \mathrm{pr}_3 R' = \mathrm{pr}_3 R$ the relation $Q_3^r = \{(p,q) \mid (p,q,r) \in R'\}$ is the product $R_3^r \circ (R_3^d)^{-1}$, that is, a non-trivial thick mapping. Thus R' is not 2-decomposable. Furthermore, $\mathrm{pr}_1 R' = \mathrm{pr}_2 R' = \mathrm{pr}_1 R$, for any $r \in \mathrm{pr}_3 R'$ the relation Q_3^r is a thick mapping with respect to η_1,η_1, and Q_3^d is reflexive. Repeating this procedure for the first and third coordinate positions, we obtain a relation R'' with the required properties. Replacing R with R'' if necessary, we may assume that R also has all these properties.

Set $B = \mathrm{pr}_1 R = \mathrm{pr}_2 R = \mathrm{pr}_3 R$ and $\eta = \eta_1 = \eta_2 = \eta_3$. Let $p \in B$ be such that R_1^p is reflexive. Let also $q \in B$ be such that $\langle p,q\rangle \notin \eta$. Then $(p,p,p),(p,q,q) \in R$ while $(p,p,q) \notin R$. Choose r such that $(r,p,q) \in R$. Then the restriction of R onto 3-element set $\{p,q,r\}$ is not 2-decomposable. Thus R can be assumed to be a relation on a 3-element set.

If η is not the equality relation, say, $\langle p,r\rangle \in \eta$, then as the restriction of R onto $\{p,q\}$ is still a not 2-decomposable relation, R itself is a relation on the set $\{p,q\}$. It is not hard to see that it is the affine relation $x+y+z=0$ on $\{p,q\}$. The CSP with global cardinality constraints for this relation is NP-complete by [9].

Suppose that η is the equality relation. Since each of R_1^p, R_1^q, R_1^q is a mapping and $R_1^p \cup R_1^q \cup R_1^r = B^2$, it follows that the three relations are disjoint. As R_1^r is the identity mapping, R_1^q and R_1^r are two cyclic permutations of (the 3-element set) B. Hence either (p,q) or (q,p) belongs to R_1^q. Let it be (p,q). Restricting R onto $\{p,q\}$ we obtain a relation R' whose projection $\mathrm{pr}_{23} R'$ equals $\{(p,p),(q,q),(p,q)\}$, which is not a thick mapping. A contradiction with the choice of R. $\square$

References

[1] K. Baker and A. Pixley. Polynomial interpolation and the chinese remainder theorem. *Mathematische Zeitschrift*, 143:165–174, 1975.

[2] L. Barto, M. Kozik, and T. Niven. Graphs, polymorphisms and the complexity of homomorphism problems. In *STOC*, pages 789–796, 2008.

[3] C. Bessière, E. Hebrard, B. Hnich, and T. Walsh. The complexity of global constraints. In *AAAI*, pages 112–117, 2004.

[4] V. Bodnarchuk, L. Kaluzhnin, V. Kotov, and B. Romov. Galois theory for post algebras. I. *Kibernetika*, 3:1–10, 1969.

[5] S. Bourdais, P. Galinier, and G. Pesant. Hibiscus: A constraint programming application to staff scheduling in health care. In *CP*, pages 153–167, 2003.

[6] A. Bulatov. Tractable conservative constraint satisfaction problems. In *LICS*, pages 321–330, 2003.

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

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

[9] N. Creignou, H. Schnoor, and I. Schnoor. Non-uniform boolean constraint satisfaction problems with cardinality constraint. In *CSL*, pages 109–123, 2008.

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

[11] D. Geiger. Closed systems of function and predicates. *Pacific Journal of Mathematics*, pages 95–100, 1968.

[12] G. Gottlob, L. Leone, and F. Scarcello. Hypertree decompositions: A survey. In *MFCS*, volume 2136 of *LNCS*, pages 37–57. Springer-Verlag, 2001.

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

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

[15] P. Jeavons, D. Cohen, and M. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

[16] P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *J. ACM*, 44:527–548, 1997.

[17] P. Jeavons, D. Cohen, and M. Gyssens. How to determine the expressive power of constraints. *Constraints*, 4:113–131, 1999.

[18] D. Marx. Parameterized complexity of constraint satisfaction problems. *Comput. Complexity*, 14(2):153–183, 2005.

[19] J.-C. Régin and C. P. Gomes. The cardinality matrix constraint. In *CP*, pages 572–587, 2004.

[20] I. G. Rosenberg. An algebraic approach to hyperalgebras. In *ISMVL*, pages 203–208, 1996.