On Planar Valued CSPs*

Peter Fulla[1] and Stanislav Živný[1]

1 **Department of Computer Science, University of Oxford, UK**
{peter.fulla,standa.zivny}@cs.ox.ac.uk

Abstract

We study the computational complexity of planar valued constraint satisfaction problems (VCSPs). First, we show that intractable *Boolean* VCSPs have to be *self-complementary* to be tractable in the planar setting, thus extending a corresponding result of Dvořák and Kupec [ICALP'15] from CSPs to VCSPs. Second, we give a complete complexity classification of *conservative* planar VCSPs on arbitrary finite domains. As it turns out, in this case planarity does not lead to any new tractable cases, and thus our classification is a sharpening of the classification of conservative VCSPs by Kolmogorov and Živný [JACM'13].

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases constraint satisfaction, valued constraint satisfaction, planarity, polymorphisms, multimorphisms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.39

1 Introduction

The valued constraint satisfaction problem (VCSP) is a far-reaching generalisation of many natural satisfiability, colouring, minimum-cost homomorphism, and min-cut problems [18, 21]. It is naturally parametrised by its domain and a valued constraint language. A *domain* D is an arbitrary finite set. A *valued constraint language*, or just a language, Γ is a (usually finite) set of weighted relations; each weighted relation $\gamma \in \Gamma$ is a function $\gamma : D^{\mathrm{ar}(\gamma)} \to \overline{\mathbb{Q}}$, where $\mathrm{ar}(\gamma) \in \mathbb{N}^+$ is the *arity* of γ and $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ is the set of extended rationals.

An *instance* $I = (V, D, C)$ of the VCSP on domain D is given by a finite set of n variables $V = \{x_1, \ldots, x_n\}$ and an objective function $C : D^n \to \overline{\mathbb{Q}}$ expressed as a weighted sum of *valued constraints* over V, i.e. $C(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}_i)$, where γ_i is a weighted relation, $w_i \in \mathbb{Q}_{\geq 0}$ is the *weight* and $\mathbf{x}_i \in V^{\mathrm{ar}(\gamma_i)}$ the *scope* of the ith valued constraint. Given an instance I, the goal is to find an assignment $s : V \to D$ of domain labels to the variables that *minimises* C. Given a language Γ, we denote by VCSP(Γ) the class of all instances I that use only weighted relations from Γ in their objective function.

We now provide a few examples of languages on $D = \{0, 1\}$. If $\Gamma_{\mathsf{nae}} = \{\rho\}$ with $\rho(x, y, z) = \infty$ if $x = y = z$ and $\rho(x, y, z) = 0$ otherwise, then VCSP(Γ_{nae}) captures precisely the NAE-3-Sat (Not-All-Equal 3-Satisfiability) problem. If $\Gamma_{\mathsf{cut}} = \{\gamma\}$ with $\gamma(x, y) = 1$ if $x = y$ and $\gamma(x, y) = 0$ otherwise, then VCSP(Γ_{cut}) captures precisely the Min-UnCut problem. If $\Gamma_{\mathsf{is}} = \{\rho, \gamma\}$ with $\rho(x, y) = \infty$ if $x = y = 1$ and $\rho(x, y) = 0$ otherwise, and $\gamma(x) = 1 - x$, then VCSP(Γ_{is}) captures precisely the Maximum Independent Set problem. Minimisation of bounded-arity submodular functions (or equivalently, submodular pseudo-Boolean polynomials of bounded degree) corresponds to VCSP(Γ_{sub}) for Γ_{sub} consisting of

* The authors were supported by a Royal Society Research Grant. The work was done while the authors were visiting the Simons Institute for the Theory of Computing at UC Berkeley. Stanislav Živný was supported by a Royal Society University Research Fellowship.

all weighted relations γ that satisfy $\gamma(\min(\mathbf{x},\mathbf{y}))+\gamma(\max(\mathbf{x},\mathbf{y}))\leq\gamma(\mathbf{x})+\gamma(\mathbf{y})$, where min and max are applied componentwise.

We will be concerned with *exact* solvability of VCSPs. A language Γ is called *tractable* if VCSP(Γ') can be solved (to optimality) in polynomial time for every finite subset $\Gamma'\subseteq\Gamma$, and Γ is called *intractable* if VCSP(Γ') is NP-hard for some finite $\Gamma'\subseteq\Gamma$. For instance, Γ_{sub} is tractable [8] whereas Γ_{nae}, Γ_{cut}, Γ_{is} are intractable [15].

1.1 Contribution

Languages on a two-element domain are called *Boolean.* The complexity of Boolean valued constraint languages is well understood and eight tractable cases have been identified [8]. Suppose that a Boolean language Γ is intractable. We are interested in restrictions that can be imposed on input instances of VCSP(Γ) that make the problem tractable. A natural way is to restrict the *incidence graph* of the instance (the precise definition is given in Section 2). In this paper we initiate the study of the *planar* variant of the VCSP.

We denote by $\mathrm{VCSP_p}(\Gamma)$ the class of instances I of VCSP(Γ) with planar incidence graph (with an additional requirement that leads to a finer classification, as discussed in detail in Section 2). Language Γ is called *planarly-tractable* if $\mathrm{VCSP_p}(\Gamma')$ can be solved (to optimality) in polynomial time for every finite subset $\Gamma'\subseteq\Gamma$, and it is called *planarly-intractable* if $\mathrm{VCSP_p}(\Gamma')$ is NP-hard for some finite $\Gamma'\subseteq\Gamma$. For instance, while Γ_{nae}, Γ_{cut}, and Γ_{is} are intractable, it is known that Γ_{nae} and Γ_{cut} are planarly-tractable [28, 17] whereas Γ_{is} is planarly-intractable [14]. The problem of classifying all intractable languages as planarly-tractable and planarly-intractable is challenging and open even for Boolean valued constraint languages.

A Boolean valued constraint language Γ is called *self-complementary* if every $\gamma\in\Gamma$ satisfies $\gamma(\mathbf{x})=\gamma(\overline{\mathbf{x}})$ for every $\mathbf{x}\in D^{\mathrm{ar}(\gamma)}$, where $\overline{\mathbf{x}}=(1-x_1,\ldots,1-x_{\mathrm{ar}(\gamma)})$ for $\mathbf{x}=(x_1,\ldots,x_{\mathrm{ar}(\gamma)})$. As our first contribution, we show in Section 3 that intractable Boolean valued constraint languages that are *not* self-complementary are planarly-intractable. We prove this by carefully constructing planar NP-hardness gadgets for any intractable Boolean valued constraint language that is not self-complementary, relying on the fact that all tractable Boolean valued constraint languages are known [8]. Our result subsumes the analogous result obtained for $\{0,\infty\}$-valued languages [10]. We remark that focusing on Boolean languages is natural as it avoids a number of difficulties intrinsic to the planar setting. Let $\Gamma_{\mathsf{col}}=\{\gamma\}$ with $\gamma(x,y)=0$ if $x\neq y$ and $\gamma(x,y)=\infty$ otherwise. Then Γ_{col} on domain D with $|D|=3$ is planarly intractable (since $\mathrm{VCSP_p}(\Gamma_{\mathsf{col}})$ captures precisely the 3-Colouring problem on planar graphs) [15] but is tractable on D with $|D|=4$ for highly nontrivial reasons, namely the Four Colour Theorem.

A valued constraint language Γ on D is called *conservative* if Γ contains all $\{0,1\}$-valued unary weighted relations. The complexity of conservative valued constraint languages is well understood: a complete complexity classification has been obtained in [26], with a recent simplification of both the algorithmic and the hardness part [35]. As our second contribution, we give a complete complexity classification of conservative valued constraint languages on arbitrary finite domains with respect to planar-tractability. In particular, we show that every intractable conservative valued constraint language is also planarly-intractable. Hence there are no new tractable cases in the conservative planar setting. This may seem unsurprising but the proof is not trivial.

Note that for Boolean valued constraint languages that are conservative the claim follows immediately from our first result: any intractable Boolean language containing both $\gamma_0(x)=x$ and $\gamma_1(x)=1-x$ (guaranteed by the conservativity assumption) is not self-complementary, and thus is planarly-intractable. This shows that $\Gamma = \Gamma_{\mathsf{cut}} \cup \{\gamma_0, \gamma_1\}$ is intractable, a result originally obtained in [1] since $\mathrm{VCSP_p}(\Gamma)$ captures precisely the planar MIN-UNCUT problem with unary weights. (In fact, the same argument shows that both $\Gamma_{\mathsf{cut}} \cup \{\gamma_0\}$ and $\Gamma_{\mathsf{cut}} \cup \{\gamma_1\}$ are planarly-intractable.)

As it is common in the world of CSPs, dealing with non-Boolean domains is considerably more difficult than the case of Boolean domains. For valued constraint languages we have a Galois connection with certain algebraic objects [6, 13] but no Galois connection is known for valued constraint languages in the planar setting. Moreover, it is unclear how to use the recent relatively simple proof of the complexity classification of conservative valued constraint languages [35] and make it work in the planar setting since the proof depends on linear programming duality. (This is related to the lack of a Galois connection in the planar setting. In particular, [35, Lemma 2], which relates (non-planar) expressibility and operator Opt, is proved via LP duality, and it is unclear how to prove it in the planar setting.)

Our approach is to follow the original proof of the classification of conservative valued constraint languages [26]. In order to adapt the proof for the planar setting, we significantly simplify it and generalise necessary parts. Details on proof differences as well as challenges that we needed to overcome to make the proof work are outlined in Section 4. We believe that our proof techniques, and in particular the now simplified and generalised technique from [26], will be useful in future work on planar (V)CSPs.

1.2 Related work

VCSPs with $\{0, \infty\}$-valued weighted relations are just (ordinary) decision CSPs [11]. There has been a lot of work on decision CSPs, see [5] for a recent survey. Most results have been obtained for CSPs parametrised by a constraint language, see [2] for a recent survey. Some of the algebraic methods developed for CSPs [3] have been extended to VCSPs [6, 34, 13, 27] and successfully used in classifying various fragments of VCSPs [20, 25, 33, 23, 35]. However, it is unclear how to use algebraic methods for instance-restricted classes of VCSPs (sometimes called *hybrid* [5]), even though there are some recent investigations in this direction [24, 32].

Planar restrictions have been studied for Boolean (decision) CSPs [10], for Boolean symmetric counting CSPs with real [4] and complex [16] weights, and also for Boolean CSPs with respect to polynomial-time approximation schemes [22, 9].

2 Preliminaries

2.1 Planar VCSPs

Let I be a VCSP instance with variables V and valued constraints S. The *incidence graph* of I is the bipartite multigraph with vertex set $S \cup V$ and edges (γ, x_i) for every $\gamma(x_1, \ldots, x_{\mathrm{ar}(\gamma)}) \in S$ and $1 \leq i \leq \mathrm{ar}(\gamma)$.

We are interested in VCSP instances with *planar* incidence graphs. Following [10], we additionally require the order of edges around constraint vertices in the plane drawing of the incidence graph respect the order of arguments of the corresponding constraint. Note that the variant without this additional restriction can be easily modelled by replacing each weighted relation γ in a language by all weighted relations obtained from γ by permuting the order of its inputs. Hence, this choice leads to a finer classification.

Following [10], rather than working with the incidence graph, we equivalently define the problem in terms of a related plane graph where variables correspond to vertices and valued constraints to faces. We note that our graphs are allowed to have loops, possibly several at a single vertex, and parallel edges.

For a connected plane graph G, we denote by $F(G)$ the set of its faces. For any face $f \in F(G)$, let $b(f)$ denote a closed walk bounding f, enumerated in the clockwise order around f.

▶ **Definition 1.** A *plane* VCSP *instance* (I, G, ϕ) is given by a VCSP instance I with variables V and objective function C with q valued constraints, a connected plane graph G over vertices V, and an injective mapping $\phi : \{1, \dots, q\} \to F(G)$ such that for every valued constraint $\gamma_i(x_1, x_2, \dots, x_{\mathrm{ar}(\gamma_i)})$ it holds $b(\phi(i)) = x_1 x_2 \dots x_{\mathrm{ar}(\gamma_i)} x_1$.

We note that the definition of a *planar* VCSP *instance*, in which case the graph G and mapping ϕ are not given, is equivalent to Definition 1. This is because, as mentioned in [10], checking whether a VCSP instance I has a planar representation, and if so then finding (I, G, ϕ), can be done in polynomial time [19]. For simplicity of presentation, we will assume that graph G and mapping ϕ are given.

We denote by $\mathrm{VCSP_p}(\Gamma)$ the class of plane VCSP instances over the language Γ.

2.2 Planar Weighted Relational Clones

In this section, we define planar weighted relational clones, which are closures of valued constraint languages that do not change the tractability of corresponding planar VCSPs.

Relations can be seen as a special case of weighted relations with range $\{0, \infty\}$ (also called *crisp*). For a weighted relation $\gamma : D^r \to \overline{\mathbb{Q}}$, we denote by $\mathrm{Feas}(\gamma) = \{\mathbf{x} \in D^r \mid \gamma(\mathbf{x}) < \infty\}$ the underlying *feasibility relation*, and by $\mathrm{Opt}(\gamma) = \{\mathbf{x} \in \mathrm{Feas}(\gamma) \mid \gamma(\mathbf{x}) \leq \gamma(\mathbf{y})$ for every $\mathbf{y} \in D^r\}$ the relation of minimal-value (or *optimal*) tuples. We also write $\mathrm{Feas}(\gamma) = 0 \cdot \gamma$ and see the Feas operator as scaling a weighted relation by zero, where we define $0 \cdot \infty = \infty$.

An assignment $s : V \to D$ for a VCSP instance (V, D, C) with $V = \{x_1, \dots, x_n\}$ is called *feasible* if $C(s(x_1), \dots, s(x_n)) < \infty$.

▶ **Definition 2.** Let (I, G, ϕ) be a plane VCSP instance such that ϕ does not map any i to the outer face f_o of G, and let $\mathbf{v} = (v_1, \dots, v_r)$ be an r-tuple of variables from V such that $b(f_o) = v_r v_{r-1} \dots v_1 v_r$. We denote by $\pi_{\mathbf{v}}(I)$ the r-ary weighted relation mapping any $\mathbf{x} \in D^r$ to the minimum objective value obtained by feasible assignments s of I with $s(\mathbf{v}) = \mathbf{x}$, or ∞ if no such feasible assignment exists.

An r-ary weighted relation γ is *planarly expressible* from a valued constraint language Γ if there exists a plane instance I over Γ and an r-tuple $\mathbf{v}$ of its variables such that $\pi_{\mathbf{v}}(I) = \gamma$.

▶ **Definition 3.** A *planar weighted relational clone* is a non-empty set of weighted relations over the same domain that is closed under planar expressibility, scaling by non-negative rational constants, addition of rational constants, and operator Opt. We will denote the smallest planar weighted relational clone containing a valued constraint language Γ by $\mathrm{wClone_p}(\Gamma)$.

An analogous notion of weighted relational clones closed under *general* (i.e. not necessarily planar) expressibility [6, 13] has been used to study the complexity of VCSPs.

▶ **Lemma 4.** *For any domain D and language Γ on D, the binary equality relation $\rho_=$ on D belongs to* $\mathrm{wClone_p}(\Gamma)$.

Proof. Relation $\rho_=$ is planarly expressible by a plane instance consisting of a single variable x with two self-loops, and $\mathbf{v} = (x, x)$. ◀

▶ **Theorem 5.** *For any valued constraint language* Γ, Γ *is planarly-tractable if, and only if,* $\mathrm{wClone_p}(\Gamma)$ *is planarly-tractable, and* Γ *is planarly-intractable if, and only if,* $\mathrm{wClone_p}(\Gamma)$ *is planarly-intractable.*

Proof. We show that $\mathrm{VCSP_p}(\mathrm{wClone_p}(\Gamma))$ is polynomial-time reducible to $\mathrm{VCSP_p}(\Gamma)$. Given an instance I over $\mathrm{wClone_p}(\Gamma)$, we replace in it all weighted relations planarly expressible from Γ by their plane instances. Scaling, which includes Feas, can be achieved by adjusting the weights of the valued constraints. Adding a constant to a weighted relation affects the value of every feasible assignment by the same amount, and therefore can be ignored.

Relation $\mathrm{Opt}(\gamma)$ can be simulated by scaling γ by a sufficiently large constant. Let W equal an upper bound on the maximum objective value of a feasible assignment of I. Without loss of generality, we may assume that no weighted relation of I assigns a negative value and that the smallest value assigned by γ is 0. Let d equal the second smallest value assigned by γ. We replace $\mathrm{Opt}(\gamma)$ with $(W/d+1)\cdot\gamma$, so that any assignment of I that would incur an infinite value from $\mathrm{Opt}(\gamma)$ has now objective value exceeding W. ◀

2.3 Algebraic Properties

For any r-tuple $\mathbf{x} \in D^r$, we write x_i for its ith component. We apply a k-ary operation $f : D^k \to D$ to k r-tuples componentwise; that is, if $\mathbf{x}^1 = (x_1^1, \dots, x_r^1), \mathbf{x}^2 = (x_1^2, \dots, x_r^2), \dots, \mathbf{x}^k = (x_1^k, \dots, x_r^k)$, then

$$f(\mathbf{x}^1, \dots, \mathbf{x}^k) \;=\; (f(x_1^1, x_1^2, \dots, x_1^k), f(x_2^1, x_2^2, \dots, x_2^k), \dots, f(x_r^1, x_r^2, \dots, x_r^k))\,.$$

The following notion is at the heart of the algebraic approach to decision CSPs [3].

▶ **Definition 6.** Let γ be a weighted relation on D. A k-ary operation $f : D^k \to D$ is a *polymorphism* of γ (and γ is *invariant under* or *admits* f) if, for every $\mathbf{x}^1, \dots, \mathbf{x}^k \in \mathrm{Feas}(\gamma)$, we have $f(\mathbf{x}^1, \dots, \mathbf{x}^k) \in \mathrm{Feas}(\gamma)$. We say that f is a polymorphism of a language Γ if it is a polymorphism of every $\gamma \in \Gamma$. We denote by $\mathrm{Pol}(\Gamma)$ the set of all polymorphisms of Γ.

A k-ary *projection* is an operation of the form $\pi_i^{(k)}(x_1, \dots, x_k) = x_i$ for some $1 \le i \le k$. Projections are (trivial) polymorphisms of all valued constraint languages.

The following notion, which involves a collection of k k-ary polymorphisms, played an important role in the complexity classification of Boolean valued constraint languages [8].

▶ **Definition 7.** Let γ be a weighted relation on D. A list $\langle f_1, \dots, f_k \rangle$ of k-ary polymorphisms of γ is a k-ary *multimorphism* of γ (and γ *admits* $\langle f_1, \dots, f_k \rangle$) if, for every $\mathbf{x}^1, \dots, \mathbf{x}^k \in \mathrm{Feas}(\gamma)$, we have

$$\sum_{i=1}^{k} \gamma(f_i(\mathbf{x}^1, \dots, \mathbf{x}^k)) \;\le\; \sum_{i=1}^{k} \gamma(\mathbf{x}^i)\,. \tag{1}$$

We say that $\langle f_1, \dots, f_k \rangle$ is a multimorphism of a language Γ if it is a multimorphism of every $\gamma \in \Gamma$.

It is known that weighted relational clones preserve polymorphisms and multimorphisms [6] and thus planar weighted relational clones do as well.

▶ **Example 8.** The class of submodular functions on $D = \{0, 1\}$ [30] can be defined as the valued constraint language Γ_{sub} that admits $\langle \min, \max \rangle$ as a multimorphism; that is, for every $\gamma \in \Gamma_{\mathsf{sub}}$, we have $\gamma(\min(\mathbf{x}, \mathbf{y})) + \gamma(\max(\mathbf{x}, \mathbf{y})) \le \gamma(\mathbf{x}) + \gamma(\mathbf{y})$.

3 Boolean Valued CSPs

In this section, we will consider only languages on a Boolean domain $D = \{0,1\}$. Our first result is that self-complementarity is necessary for planar-tractability of intractable Boolean languages.

▶ **Theorem 9.** *Let Γ be a Boolean valued constraint language that is intractable. If Γ is not self-complementary then it is planarly-intractable.*

We start with some notation for important operations on D. For any $a \in D$, c_a is the constant unary operation such that $c_a(x) = a$ for all $x \in D$. Operation $\neg$ is the unary negation, i.e. $\neg(0) = 1$ and $\neg(1) = 0$. Binary operation min (max) is the minimum (maximum) operation with respect to the order $0 < 1$. Ternary operation Mn (Mj) is the unique minority (majority) operation.

Next we define some useful relations. For any $a \in D$, we denote by ρ_a the unary constant relation $\{(a)\}$. Relation $\rho_{\neq}$ is the binary disequality relation, i.e. $\rho_{\neq} = \{(0,1),(1,0)\}$. Ternary relation $\rho_{\text{1-in-3}}$ corresponds to the 1-in-3 Positive 3-Sat problem, i.e. $\rho_{\text{1-in-3}} = \{(0,0,1),(0,1,0),(1,0,0)\}$. Weighted relations $\gamma_0, \gamma_1, \gamma_{\neq}$ are defined as soft-constraint variants of $\rho_0, \rho_1, \rho_{\neq}$ assigning value 0 to allowed tuples and 1 to disallowed tuples.

Note that Γ is self-complementary if, and only if, Γ admits multimorphism $\langle \neg \rangle$. The proof of Theorem 9 is based on the following four lemmas.

▶ **Lemma 10.** *Let Γ be a valued constraint language that admits neither of the multimorphisms $\langle c_0 \rangle$, $\langle c_1 \rangle$. Then $\rho_0, \rho_1 \in \mathrm{wClone_p}(\Gamma)$ or $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$.*

▶ **Lemma 11.** *Let Γ be a valued constraint language that admits neither of the multimorphisms $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$. If $\rho_0, \rho_1 \in \mathrm{wClone_p}(\Gamma)$, then $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$.*

▶ **Lemma 12.** *Let Γ be a valued constraint language that does not admit multimorphism $\langle \neg \rangle$. If $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$, then $\rho_0, \rho_1 \in \mathrm{wClone_p}(\Gamma)$.*

▶ **Lemma 13.** *Let Γ be a valued constraint language that admits neither of the multimorphisms $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. If $\rho_0, \rho_1, \rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$, then $\rho_{\mathit{1\text{-}in\text{-}3}} \in \mathrm{wClone_p}(\Gamma)$.*

Proof (of Theorem 9). Since Γ is intractable we know, by [8, Theorem 7.1], that Γ admits neither of the multimorphisms $\langle c_0 \rangle$, $\langle c_1 \rangle$, $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. By assumption, Γ is not self-complementary and hence does not admit the $\langle \neg \rangle$ multimorphism.

By Lemmas 10, 11, and 12, we have $\rho_0, \rho_1, \rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$ and hence by Lemma 13 $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$. Planar 1-in-3 Positive 3-Sat problem is NP-complete [29], and therefore Γ is planarly-intractable by Theorem 5. ◀

4 Conservative Valued CSPs

A valued constraint language Γ is called *conservative* if Γ includes all $\{0,1\}$-valued unary weighted relations. As our second result, we prove that planarity does not add any tractable cases for conservative valued constraint languages.

▶ **Theorem 14.** *Let Γ be an intractable conservative valued constraint language. Then Γ is planarly-intractable.*

Consequently, we obtain a complexity classification of all conservative valued constraint languages in the planar setting, thus sharpening the classification of conservative valued constraint languages [26, 35]. As mentioned in Section 1, for Boolean domains Theorem 14 follows from Theorem 9. Thus, the only tractable Boolean conservative languages in the planar setting are given by the multimorphisms $\langle \min, \max \rangle$ and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ [8].

We now define certain special types of multimorphisms.

A k-ary operation $f : D^k \to D$ if called *conservative* if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for every $x_1, \ldots, x_k \in D$. A multimorphism $\langle f_1, \ldots, f_k \rangle$ is called *conservative* if applying $\langle f_1, \ldots, f_k \rangle$ to $(x_1, \ldots, x_k)$ returns a permutation of $(x_1, \ldots, x_k)$.

▶ **Definition 15.** A binary multimorphism $\langle f, g \rangle$ of Γ is called a *symmetric tournament pair* (STP) if it is conservative and both f and g are commutative operations.

It was shown in [7] that languages admitting an STP multimorphism are tractable.

A ternary operation $f : D^3 \to D$ is called a *majority* operation if $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$ for all $x, y \in D$, and a *minority* operation if $f(x, x, y) = f(x, y, x) = f(y, x, x) = y$ for all $x, y \in D$.

▶ **Definition 16.** A ternary multimorphism $\langle f, g, h \rangle$ is called an MJN if f and g are (possibly equal) majority operations and g is a minority operation.

It was shown in [26] that languages admitting an MJN multimorphism are tractable.

▶ **Theorem 17** ([26]). *Let Γ be a conservative valued constraint language on D. Then either Γ admits a conservative binary multimorphism $\langle f, g \rangle$ and a conservative ternary multimorphism $\langle f', g', h' \rangle$ and there is a family M of 2-element subsets of D, such that*

- *for every $\{a, b\} \in M$, $\langle f, g \rangle$ restricted to $\{a, b\}$ is a symmetric tournament pair, and*
- *for every $\{a, b\} \notin M$, $\langle f', g', h' \rangle$ restricted to $\{a, b\}$ is an MJN multimorphism,*

in which case Γ is tractable, or else Γ is intractable.

The idea of the proof of Theorem 17 is as follows: given a conservative valued constraint language Γ, we define a certain graph G_Γ whose vertices are pairs of different labels from D and an edge $(a, b) - (c, d)$ is present if there is a binary weighted relation $\gamma \in \mathrm{wClone}(\Gamma)$ that is "non-submodular with respect to the order $a < b$ and $c < d$". The edges of G_Γ are then classified as soft and hard. It is shown that a soft self-loop implies intractability of Γ. Otherwise, the vertices of G_Γ are partitioned into $M \cup \overline{M}$, where M denotes the set of loopless vertices and $\overline{M}$ denotes the rest (i.e. vertices with hard loops). It is then shown that G_Γ restricted to M is bipartite, which is in turn used to construct a binary multimorphism and a ternary multimorphism of Γ such that the binary multimorphism is an STP on M and the ternary multimorphism is an MJN on $\overline{M}$. (Proving that the constructed objects are multimorphisms of Γ is the most technical part of the proof.) Any such language is then tractable via an involved algorithm from [26] that relies on [7], or by an LP relaxation [35].

Our approach is to follow the above-described proof and adapt it to the planar setting. It is natural to replace $\mathrm{wClone}(\Gamma)$ by $\mathrm{wClone_p}(\Gamma)$ in the definition of G_Γ. But this simple change does not immediately yield the desired result. There are two main obstacles. First, the proof of Theorem 17 from [26] heavily relies on [31], which guarantees the existence of a majority polymorphism. However, this is proved in [31] using (functional) clones and depends on the Galois connection between clones and relational co-clones; such a connection is not known for planar expressibility! Second, some of the gadgets (and in particular the "i-expansion" from [26, Section 6.4]) are not necessarily planar.

To avoid these obstacles, we modify, significantly simplify, and generalise the proof so that it works in the planar setting. The key changes are the following. (i) We use a closure of a language, denoted Γ^* below, that is a subset of the planar weighted relational clone of a conservative language. (ii) We do *not* rely on Takhanov's result on the existence of a majority polymorphism [31] but instead prove directly without using [31] that (the closure of) Γ is 2-decomposable. (iii) We define different STP and MJN multimorphisms that allow us to simplify the proof that these are indeed multimorphisms of Γ. In particular, we will prove modularity of weighted relations on $\overline{M}$ and show that the ternary multimorphism satisfies Inequality (1) with equality, thus obtaining a better structural understanding of tractable languages. The main simplification is that we define MJN as close to projection operations as possible, and in particular not depending on the STP multimorphism as in [26].

We now define a few operations on weighted relations.

▶ **Definition 18.** Let γ be an r-ary weighted relation on D. A *domain restriction* of γ to $D' \subseteq D$ at coordinate i is the r-ary weighted relation defined as $\gamma'(x_1, \ldots, x_r) = \gamma(x_1, \ldots, x_r) + \rho_{D'}(x_i)$, where $\rho_{D'}(x) = 0$ if $x \in D'$ and $\rho_{D'}(x) = \infty$ otherwise. A *pinning* of γ to $a \in D$ at coordinate i is the $(r-1)$-ary weighted relation defined as $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \min_{x_i \in D} \gamma(x_1, \ldots, x_r) + \rho_{\{a\}}(x_i)$. A *minimisation* of γ at coordinate i is the $(r-1)$-ary weighted relation defined as $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \min_{x_i \in D} \gamma(x_1, \ldots, x_r)$.

A *join* of two binary weighted relations γ_1 and γ_2 is the weighted relation $\gamma(x, y) = \min_{z \in D} \gamma_1(z, x) + \gamma_2(z, y)$.

We will make use only of a limited subset of $\mathrm{wClone_p}(\Gamma)$, which is defined below.

▶ **Definition 19.** For a conservative valued constraint language Γ on D, we define Γ^* to be the smallest set containing Γ, all unary weighted relations and the binary equality relation on D, and closed under operators Feas and Opt, addition of unary weighted relations to weighted relations of arbitrary arity, minimisation, and join.

Note that $\Gamma^* \subseteq \mathrm{wClone_p}(\Gamma)$, as any unary weighted relation can be obtained from the set of all $\{0, 1\}$-valued unary weighted relations by addition of unary weighted relations, scaling, addition of constants, and operator Opt. It is easy to show that addition of unary weighted relations, minimisation, and join are planarly-expressible. Set Γ^* is also closed under domain restriction and pinning, as these operations can be achieved by adding unary weighted relations and minimisation. By Theorem 5, Γ^* has the same complexity as Γ.

▶ **Definition 20.** Let Γ be a conservative language. We define an undirected graph G_Γ on vertices (a, b) for all $a, b \in D, a \neq b$. For any vertex $v = (a, b)$, we will denote by $\overline{v}$ vertex (b, a). Graph G_Γ is allowed to have self-loops. It contains edge $(a_1, b_1) - (a_2, b_2)$ if there is a binary weighted relation $\gamma \in \Gamma^*$ such that $(a_1, b_2), (b_1, a_2) \in \mathrm{Feas}(\gamma)$ and

$$\gamma(a_1, b_2) + \gamma(b_1, a_2) < \gamma(a_1, a_2) + \gamma(b_1, b_2)\,. \tag{2}$$

If there exists such a weighted relation γ with at least one of $(a_1, a_2), (b_1, b_2)$ belonging to $\mathrm{Feas}(\gamma)$, we will call the edge *soft*, otherwise the edge is *hard*. We denote by $\overline{M}$ and M the set of vertices with and without self-loops respectively.

We will show in Theorem 25, proved in Section 5, that if G_Γ has a soft self-loop then Γ is planarly-intractable. Our goal, assuming G_Γ has no soft self-loops, is to prove the following.

▶ **Theorem 21.** *If G_Γ has no soft self-loop, then Γ admits a binary multimorphism $\langle \sqcap, \sqcup \rangle$ that is an STP on M, and a ternary multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ that is an MJN on $\overline{M}$.*

5 Proof of Theorem 21

We will need the following definition.

▶ **Definition 22.** Let ρ be an r-ary relation. For any $i, j \in \{1, \dots, r\}$, we will denote by $\mathrm{Pr}_{i,j}(\rho)$ the projection of ρ on coordinates i and j, i.e. the binary relation defined as

$$(a_i, a_j) \in \mathrm{Pr}_{i,j}(\rho) \iff (\exists \mathbf{x} \in \rho)\ x_i = a_i \wedge x_j = a_j\,. \tag{3}$$

Relation ρ is *2-decomposable* if

$$\mathbf{x} \in \rho \iff \bigwedge_{1 \le i,j \le r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho)\,. \tag{4}$$

The following lemma will be useful in proving results about both Boolean and conservative valued constraint languages. For any r-tuple $\mathbf{z}$ and a subset of coordinates $I \subseteq \{1, \dots, r\}$, we denote by $\mathbf{z}_I$ the projection of $\mathbf{z}$ onto I. For any partition of coordinates $I, J \subseteq \{1, \dots, r\}$, we then write $\cdot$ for the inverse operation, i.e. $\mathbf{z}_I \cdot \mathbf{z}_J = \mathbf{z}$.

▶ **Lemma 23.** *Let γ be an r-ary weighted relation and $I, J \subseteq \{1, \dots, r\}$ a partition of its coordinates. If $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ and*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{5}$$

then there exist coordinates $i \in I, j \in J$ and a binary weighted relation $\gamma_{i,j} \in \{\gamma\}^*$ such that $(x_i, x_j), (y_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and

$$\gamma_{i,j}(x_i, x_j) + \gamma_{i,j}(y_i, y_j) < \gamma_{i,j}(x_i, y_j) + \gamma_{i,j}(y_i, x_j)\,. \tag{6}$$

Moreover, if every relation in $\{\gamma\}^*$ is 2-decomposable, then $\mathbf{x}_I \cdot \mathbf{y}_J \in \mathrm{Feas}(\gamma)$ implies $(x_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and $\mathbf{y}_I \cdot \mathbf{x}_J \in \mathrm{Feas}(\gamma)$ implies $(y_i, x_j) \in \mathrm{Feas}(\gamma_{i,j})$.

The following lemma gives a useful alternative characterisation of an edge in G_Γ.

▶ **Lemma 24.** *Graph G_Γ contains edge $(a_1, b_1) - (a_2, b_2)$ if, and only if, binary relation $\{(a_1, b_2), (b_1, a_2)\}$ belongs to Γ^*. The edge is soft if, and only if, at least one of binary relations $\{(a_1, a_2), (a_1, b_2), (b_1, a_2)\}$, $\{(b_1, b_2), (a_1, b_2), (b_1, a_2)\}$ belongs to Γ^*.*

▶ **Theorem 25.** *If G_Γ has a soft self-loop, language Γ is planarly-intractable.*

Proof. Let (a, b) be a vertex of G_Γ with a soft self-loop. Without loss of generality, we have $\rho = \{(a, a), (a, b), (b, a)\} \in \Gamma^*$ by Lemma 24. We denote by γ_a, γ_b the unary weighted relations defined as $\gamma_a(a) = \gamma_b(b) = 0$, $\gamma_a(b) = \gamma_b(a) = 1$, and $\gamma_a(x) = \gamma_b(x) = \infty$ for $x \notin \{a, b\}$. Set $\Gamma' = \{\rho, \gamma_a, \gamma_b\} \subseteq \Gamma^*$ can be viewed as a conservative language over a Boolean domain $\{a, b\}$. Observe that Γ' is intractable (via checking that it does not fall into either of the two tractable cases for Boolean conservative valued constraint languages [8] corresponding to the $\langle \min, \max \rangle$ and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ multimorphisms) and not self-complementary (neither of its weighted relations is self-complementary), and hence planarly-intractable by Theorem 9. Alternatively, just take the obvious encoding of the planar MAXIMUM INDEPENDENT SET problem as discussed in Section 1. ◀

In order to prove Theorem 21, we now introduce several lemmas. From now on we will assume that G_Γ has no soft self-loop.

▶ **Lemma 26.** *For any vertex v, graph G_Γ contains edge $v - \overline{v}$. There is no edge between M and $\overline{M}$, no odd cycle in M, and no soft edge in $\overline{M}$.*

Proof. As the binary equality relation belongs to Γ^*, we have edge $v - \overline{v}$ for all vertices v.

Consider any sequence of vertices v_1, v_2, v_3, v_4 such that there is an edge between every two consecutive ones, and denote $v_i = (a_i, b_i)$. By Lemma 24, there exist binary relations $\rho_i = \{(a_i, b_{i+1}), (b_i, a_{i+1})\} \in \Gamma^*$ for $i \in \{1, 2, 3\}$. Their join equals $\{(a_1, b_4), (b_1, a_4)\} \in \Gamma^*$, and hence G_Γ contains edge $v_1 - v_4$. If any of edges $v_1 - v_2, v_2 - v_3, v_3 - v_4$ is soft, we can replace the corresponding relation ρ_i with $\{(a_i, a_{i+1}), (a_i, b_{i+1}), (b_i, a_{i+1})\}$ or $\{(b_i, b_{i+1}), (a_i, b_{i+1}), (b_i, a_{i+1})\}$ to show that $v_1 - v_4$ is also soft.

Suppose that there is an edge between $s \in M$ and $t \in \overline{M}$. Then we have edges $s - t, t - t, t - s$, and hence also self-loop $s - s$, which is a contradiction.

If there is an odd cycle in M, let us choose a shortest one and denote its vertices $v_1, \ldots, v_k$ ($k \geq 3$). We have a sequence of adjacent vertices v_k, v_1, v_2, v_3, and hence v_3 and v_k are also adjacent. But that means there is a shorter odd cycle (or a self-loop) $v_3, \ldots, v_k$; a contradiction.

Finally, suppose that $s, t \in \overline{M}$ and edge $s - t$ is soft. Then we have edges $s - t, t - t, t - s$, and hence a soft self-loop at s, which is a contradiction. ◀

▶ **Lemma 27.** *Every relation in Γ^* is 2-decomposable.*

Proof. Let $\rho \in \Gamma^*$ be an r-ary relation. By definition, $\mathbf{x} \in \rho$ implies $\bigwedge_{1 \leq i,j \leq r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho)$ for every relation. We prove the converse implication by induction on r. If $r \leq 2$, relation ρ is trivially 2-decomposable. Let $r = 3$. Suppose for the sake of contradiction that $(x_1, x_2, x_3) \notin \rho$ even though $(y_1, x_2, x_3), (x_1, y_2, x_3), (x_1, x_2, y_3) \in \rho$ for some $y_1, y_2, y_3 \in D$. Let $\rho_1 \in \Gamma^*$ be the binary relation obtained from ρ by pinning it at the first coordinate to label x_1; we have $(x_2, y_3), (y_2, x_3) \in \rho_1$, $(x_2, x_3) \notin \rho_1$, and thus graph G_Γ contains edge $(x_2, y_2) - (x_3, y_3)$. Analogously, the graph contains edges $(x_3, y_3) - (x_1, y_1)$ and $(x_1, y_1) - (x_2, y_2)$. This is an odd cycle, so it must hold $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in \overline{M}$. Let γ be a unary weighted relation with $\gamma(x_1) = 0, \gamma(y_1) = 1$ and $\gamma(z) = \infty$ for all $z \in D \setminus \{x_1, y_1\}$. By adding γ to ρ at the first coordinate and then minimising over it we show that edge $(x_2, y_2) - (x_3, y_3)$ is soft, which is a contradiction.

It remains to prove the lemma for $r \geq 4$. Let $\rho_1 \in \Gamma^*$ be the relation obtained from ρ by minimisation over the first coordinate. Relation ρ_1 is 2-decomposable by the induction hypothesis, so $(x_2, \ldots, x_r) \in \rho_1$, and hence $(y_1, x_2, \ldots, x_r) \in \rho$ for some $y_1 \in D$. Analogously, we have $(x_1, y_2, x_3, \ldots, x_r), (x_1, x_2, y_3, x_4, \ldots, x_r) \in \rho$ for some $y_2, y_3 \in D$. Pinning ρ at every coordinate $k \geq 4$ to its respective label x_k gives a ternary 2-decomposable relation ρ' such that $(x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho')$ for all $i, j \in \{1, 2, 3\}$. Therefore, $(x_1, x_2, x_3) \in \rho'$ and $\mathbf{x} \in \rho$. ◀

The following lemma involves a generalisation of the definition of an edge in G_Γ to non-binary weighted relations.

▶ **Lemma 28.** *Let $\gamma \in \Gamma^*$ be an r-ary weighted relation and $I, J \subseteq \{1, \ldots, r\}$ a partition of its coordinates. If $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ and*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{7}$$

then graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$ for some $i \in I, j \in J$. If at least one of $\mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J$ belongs to $\mathrm{Feas}(\gamma)$, the edge is soft.

Proof. By Lemma 23, there are coordinates $i \in I, j \in J$ and a binary weighted relation $\gamma_{i,j} \in \Gamma^*$ such that $(x_i, x_j), (y_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and $\gamma_{i,j}(x_i, x_j) + \gamma_{i,j}(y_i, y_j) < \gamma_{i,j}(x_i, y_j) + \gamma_{i,j}(y_i, x_j)$, so graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$. If $\mathbf{x}_I \cdot \mathbf{y}_J$ or $\mathbf{y}_I \cdot \mathbf{x}_J$ belongs to $\mathrm{Feas}(\gamma)$, then (x_i, y_j) or (y_i, x_j) belongs to $\mathrm{Feas}(\gamma_{i,j})$ (as $\mathrm{Feas}(\gamma)$ is 2-decomposable by Lemma 27), and hence the edge is soft. ◀

▶ **Lemma 29.** *Let $\gamma \in \Gamma^*$ be an r-ary weighted relation and $I, J \subseteq \{1, \dots, r\}$ a partition of its coordinates. If $\mathbf{x}, \mathbf{y}, \mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J \in \mathrm{Feas}(\gamma)$ and, for all $i \in I$, $(x_i, y_i) \in \overline{M}$, then*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) = \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,. \tag{8}$$

Proof. Suppose for the sake of contradiction that the equality does not hold. Without loss of generality, we may assume that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)$. By Lemma 28, graph G_Γ contains a soft edge incident to (x_i, y_i) for some $i \in I$, which contradicts Lemma 26. ◀

Graph G_Γ does not have any odd cycle on vertices M. Therefore, there is a partition of M into two independent sets M_1, M_2. (In fact, it can be shown that every connected component of G_Γ restricted to M is a complete bipartite graph but we do not need this fact here.) Note that $(a, b) \in M_1$ if, and only if, $(b, a) \in M_2$, as every vertex $v \in M$ is adjacent to $\overline{v}$. We define multimorphism $\langle \sqcap, \sqcup \rangle$ as follows:

$$\langle \sqcap, \sqcup \rangle(x, y) = \begin{cases} (x, y) & \text{if } (x, y) \in M_1, & \text{(9a)} \\ (y, x) & \text{if } (x, y) \in M_2, & \text{(9b)} \\ (x, y) & \text{otherwise.} & \text{(9c)} \end{cases}$$

By definition, $\langle \sqcap, \sqcup \rangle$ is commutative on M.

▶ **Theorem 30.** *$\langle \sqcap, \sqcup \rangle$ is a multimorphism of Γ.*

Proof. Let $\gamma \in \Gamma$ be an r-ary weighted relation and $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$. Suppose for the sake of contradiction that (1) does not hold. We partition set $\{1, \dots, r\}$ into I and J: Set J consists of all coordinates j such that case (9b) applies to (x_j, y_j); set I covers the other two cases. For any $i \in I$, either $x_i = y_i$ or $(x_i, y_i) \in M_1 \cup \overline{M}$. For any $j \in J$, $(x_j, y_j) \in M_2$ and hence $(y_j, x_j) \in M_1$. $\langle \sqcap, \sqcup \rangle$ maps $\mathbf{x}, \mathbf{y}$ to $\mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J$, so we have $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)$. By Lemma 28, graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$ for some $i \in I, j \in J$, which contradicts Lemma 26. ◀

The following definition corresponds to the "μ function" from [26, Section 6].

▶ **Definition 31.** For any $a, b, c \in D$, we say that $ab|c$ holds if a, b, c are all different labels and there exist $(s, t) \in \overline{M}$ such that binary relation $\{(a, s), (b, s), (c, t)\}$ belongs to Γ^*.

The intuition is that if $ab|c$ holds, then any minority operation on $\overline{M}$ must map any permutation of $\{a, b, c\}$ to c.

▶ **Lemma 32.** *For any $a, b, c \in D$, at most one of $ab|c$, $ca|b$, $bc|a$ holds. If $ab|c$, then $(a, c), (b, c) \in \overline{M}$.*

Proof. Suppose that both $ca|b$ and $bc|a$ hold. Then there are $(s_1, t_1), (s_2, t_2) \in \overline{M}$ and binary relations $\rho_1, \rho_2 \in \Gamma^*$ such that $\rho_1 = \{(c, s_1), (a, s_1), (b, t_1)\}$, $\rho_2 = \{(b, s_2), (c, s_2), (a, t_2)\}$. We construct binary relation ρ as $\rho(x, y) = \min_{z \in D} \rho_1(z, x) + \rho_2(z, y)$. We have $\rho \in \Gamma^*$ and $\rho = \{(s_1, s_2), (s_1, t_2), (t_1, s_2)\}$, which implies a soft edge in $\overline{M}$ and hence a contradiction.

If $ab|c$, then there are $(s, t) \in \overline{M}$ such that $\{(a, s), (b, s), (c, t)\} \in \Gamma^*$. By restricting this relation at the first coordinate to labels $\{a, c\}$ we get edge $(a, c) - (t, s)$ and thus $(a, c) \in \overline{M}$; analogously by restricting to $\{b, c\}$ we get $(b, c) \in \overline{M}$. ◀

We define multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ as follows:

$$\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle(x,y,z) = \begin{cases} (x,y,z) & \text{if } x = y \wedge (y,z) \in \overline{M} \text{ or } xy|z, \quad (10a) \\ (z,x,y) & \text{if } z = x \wedge (x,y) \in \overline{M} \text{ or } zx|y, \quad (10b) \\ (y,z,x) & \text{if } y = z \wedge (z,x) \in \overline{M} \text{ or } yz|x, \quad (10c) \\ (x,y,z) & \text{otherwise.} \quad (10d) \end{cases}$$

Note that the operations of $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ are majorities and a minority on $\overline{M}$. Also note that in the subcase $x = y \wedge (y,z) \in \overline{M}$ of case (10a), the output has to be (x,y,z) for $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ to be an MJN multimorphism of Γ on $\overline{M}$ (and similarly for the first subcase of case (10b) and case (10c)). It is the other cases where there is some freedom and where we differ from [26].

▶ **Theorem 33.** $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ *is a multimorphism of* Γ.

We will actually prove that (1) holds with *equality.*

Proof. Suppose for the sake of contradiction this is not true for some r-ary weighted relation $\gamma \in \Gamma^*$ and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathrm{Feas}(\gamma)$; we choose γ so that it has the minimum arity among such counterexamples. We denote the r-tuples to which $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ by $(\mathbf{f}, \mathbf{g}, \mathbf{h})$.

First we show that case (10b) does not occur. Let I be the set of coordinates i such that case (10b) applies to (x_i, y_i, z_i) and let J cover the remaining cases. Suppose that I is non-empty, and note that $\mathbf{f}_I = \mathbf{z}_I, \mathbf{g}_I = \mathbf{x}_I, \mathbf{h}_I = \mathbf{y}_I$. For every $i \in I$, it holds $(x_i, y_i), (z_i, y_i) \in \overline{M}$ (directly or by Lemma 32), and either $z_i = x_i$ or $z_i x_i | y_i$.

We claim that $\{x_i, y_i, z_i\} \times \{x_j, y_j, z_j\} \subseteq \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ for all $i \in I, j \in J$. (A detailed proof of the claim is given in the full version of this paper [12].)

Because $\mathrm{Feas}(\gamma)$ is 2-decomposable by Lemma 27, we have $\mathbf{u}_I \cdot \mathbf{v}_J \in \mathrm{Feas}(\gamma)$ for any $\mathbf{u}, \mathbf{v} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$. It must hold

$$\gamma(\mathbf{y}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J) = \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{h}_J)\,, \tag{11}$$

otherwise we would obtain a smaller counterexample by pinning γ at every coordinate $i \in I$ to its respective label y_i. This gives $\mathbf{y}_I \cdot \mathbf{f}_J, \mathbf{y}_I \cdot \mathbf{g}_J, \mathbf{y}_I \cdot \mathbf{h}_J \in \mathrm{Feas}(\gamma)$ and hence $\mathbf{u}_I \cdot \mathbf{v}_J \in \mathrm{Feas}(\gamma)$ for any $\mathbf{u}, \mathbf{v} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{f}, \mathbf{g}, \mathbf{h}\}$. By Lemma 29, it holds

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) = \gamma(\mathbf{x}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{12}$$

$$\gamma(\mathbf{z}_I \cdot \mathbf{z}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) = \gamma(\mathbf{z}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J)\,. \tag{13}$$

Adding (11), (12), and (13) shows that (1) holds as equality, which is a contradiction. Therefore, case (10b) does not apply at any coordinate.

Suppose that case (10c) applies at some coordinate i. $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{y}, \mathbf{x}, \mathbf{z})$ to $(\mathbf{g}, \mathbf{f}, \mathbf{h})$, which gives us another smallest counterexample to the theorem. However, at coordinate i is now applied case (10b), which was proved impossible.

Finally, we have that only cases (10a) and (10d) may occur in a smallest counterexample. But then $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ to $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and hence the stated equality holds. ◀

References

1 Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, 1982. doi:10.1088/0305-4470/15/10/028.

2 Libor Barto. Constraint satisfaction problem and universal algebra. *ACM SIGLOG News*, 1(2):14–24, 2014. doi:10.1145/2677161.2677165.

3 Andrei Bulatov, Andrei Krokhin, and Peter Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005. doi:10.1137/S0097539700376676.

4 Jin-yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar #CSP. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 427–436. IEEE Computer Society, 2010.

5 Clément Carbonnel and Martin C. Cooper. Tractability in constraint satisfaction problems: a survey. *Constraints*, 21(2):115–144, 2016. doi:10.1007/s10601-015-9198-6.

6 David A. Cohen, Martin C. Cooper, Páidí Creed, Peter Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing*, 42(5):915–1939, 2013. URL: http://zivny.cz/publications/cccjz13sicomp-preprint.pdf, doi:10.1137/130906398.

7 David A. Cohen, Martin C. Cooper, and Peter G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. *Theoretical Computer Science*, 401(1-3):36–51, 2008. doi:10.1016/j.tcs.2008.03.015.

8 David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006. doi:10.1016/j.artint.2006.04.002.

9 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

10 Zdeněk Dvořák and Martin Kupec. On Planar Boolean CSP. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 432–443. Springer, 2015.

11 Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

12 Peter Fulla and Stanislav Živný. On Planar Valued CSPs. *CoRR*, abs/1602.06323, 2016. URL: http://arxiv.org/abs/1602.06323.

13 Peter Fulla and Stanislav Živný. A Galois Connection for Valued Constraint Languages of Infinite Size. *ACM Transactions on Computation Theory*, 8(3), 2016. Article No. 9. URL: http://www.cs.ox.ac.uk/Stanislav.Zivny/homepage/publications/fz16toct-preprint.pdf, doi:10.1145/2898438.

14 M. R. Garey and David S. Johnson. The rectilinear steiner tree problem in NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

15 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

16 Heng Guo and Tyson Williams. The complexity of planar Boolean #CSP with complex weights. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 516–527. Springer, 2013.

17 F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975. doi:10.1137/0204019.

18 Pavol Hell and Jaroslav Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008. doi:10.1016/j.cosrev.2008.10.003.

19 John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

20 Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 43(3):1064–1084, 2014. doi:10.1137/120893549.

21 Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 113:21–55, 2014. URL: http://zivny.cz/publications/jkz14.pdf.

22 Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of PTAS. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96)*, pages 329–337, 1996. doi:10.1145/237814.237979.

23 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS'15)*. IEEE Computer Society, 2015.

24 Vladimir Kolmogorov, Michal Rolínek, and Rustem Takhanov. Effectiveness of Structural Restrictions for Hybrid CSPs. *CoRR*, abs/1504.07067, 2015. URL: http://arxiv.org/abs/1504.07067.

25 Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, 44(1):1–36, 2015. doi:10.1137/130945648.

26 Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *Journal of the ACM*, 60(2), 2013. Article No. 10. URL: http://zivny.cz/publications/kz13jacm-preprint.pdf, doi:10.1145/2450142.2450146.

27 Marcin Kozik and Joanna Ochremiak. Algebraic properties of valued constraint satisfaction problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2015. doi:10.1007/978-3-662-47672-7_69.

28 Bernard M. E. Moret. Planar NAE3SAT is in P. *SIGACT News*, 19(2):51–54, 1988.

29 Wolfgang Mulzer and Günter Rote. Minimum-weight Triangulation is NP-hard. *Journal of the ACM*, 55(2):11:1–11:29, 1998. doi:10.1145/1346330.1346336.

30 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

31 Rustem Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, pages 657–668, 2010. doi:10.4230/LIPIcs.STACS.2010.2493.

32 Rustem Takhanov. Hybrid (V)CSPs and algebraic reductions. *CoRR*, abs/1506.06540, 2015. URL: http://arxiv.org/abs/1506.06540.

33 Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. *Journal of the ACM*. To appear. URL: http://arxiv.org/abs/1210.2977v3.

34 Johan Thapper and Stanislav Živný. Necessary Conditions on Tractability of Valued Constraint Languages. *SIAM Journal on Discrete Mathematics*, 29(4):2361–2384, 2015. URL: http://zivny.cz/publications/tz15sidma-preprint.pdf, doi:10.1137/140990346.

35 Johan Thapper and Stanislav Živný. Sherali-Adams relaxations for valued CSPs. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 1058–1069. Springer, 2015. URL: http://zivny.cz/publications/tz15icalp-preprint.pdf, doi:10.1007/978-3-662-47672-7_86.