On Planar Valued CSPs☆

Peter Fulla[a], Stanislav Živný[a,*]

[a] *University of Oxford, Oxford, UK*

Abstract

We study the computational complexity of planar valued constraint satisfaction problems (VCSPs), which require the *incidence* graph of the instance be planar. 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. 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].

Keywords: constraint satisfaction, planarity, multimorphisms, valued constraint satisfaction

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 [2, 3]. 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. (We note that we allow zero weights and for $w_i = 0$ we define $w_i \cdot \infty = \infty$.) 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. To see this, observe that any instance of VCSP(Γ_{nae}) is equivalent to an instance of NAE-3-Sat over the same variables, each constraint giving a ternary clause (weights are without effect in this case). 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 weighted 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 weighted 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 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.

☆An extended abstract of part of this work appeared in the *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 2016 [1].

*Corresponding Author

Email addresses: peter.fulla@cs.ox.ac.uk (Peter Fulla), standa.zivny@cs.ox.ac.uk (Stanislav Živný)

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 [4] whereas Γ_{nae}, Γ_{cut}, Γ_{is} are intractable [5].

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 [4]. 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 [6, 7] whereas Γ_{is} is planarly-intractable [8]. 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 [4]. Our result subsumes the analogous result obtained for $\{0, \infty\}$-valued languages [9]. 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) [5] 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 [10], with a recent simplification of both the algorithmic and the hardness part [11, 12]. 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. We remark that conservative (V)CSPs constitute a large and important fragment of CSPs [13] and VCSPs [10]. In fact, in practice most (V)CSPs are conservative [14].

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 [15] 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 [16, 17] 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 [11] 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, [11, 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 [10]. 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 [10], 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 [18]. There has been a lot of work on decision CSPs, see [19] for a recent survey. Most results have been obtained for CSPs parametrised by a constraint language, see [20] for a recent survey. Some of the algebraic methods developed for CSPs [21] have been extended to VCSPs [16, 22, 17, 23] and successfully used in classifying various fragments of VCSPs [24, 25, 26, 27, 11]. However, it is unclear how to use algebraic methods for instance-restricted classes of VCSPs (sometimes called *hybrid* [19]), even though there are some recent investigations in this direction [28, 29].

Following [9], we define planar VCSPs by requiring the *incidence* graph be planar. We note that an alternative option when structurally restricting classes of (V)CSPs is to consider the *Gaifman* graph, as was done for CSPs [30], counting CSPs [31], special cases of VCSPs [32] and also in the setting of parametrised counting [33]. However, we believe that the incidence graph is the more natural option for the planarity requirement since restricting the Gaifman graph would exclude (V)CSPs with, for instance, any constraint of arity at least 5.

Planar restrictions have been studied for Boolean (decision) CSPs [9, 34], for Boolean symmetric counting CSPs with real [35] and complex [36] weights, and also for Boolean CSPs with respect to polynomial-time approximation schemes [37, 38].

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 $\gamma(x_1, \dots, x_{\mathrm{ar}(\gamma)}) \in S$ and $1 \leq i \leq \mathrm{ar}(\gamma)$.

We are interested in VCSP instances with *planar* incidence graphs. Following [9], 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 [9], 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 $b(\phi(i)) = x_1 x_2 \dots x_{\mathrm{ar}(\gamma_i)} x_1$.

Example 2. Let $V = \{x_1, x_2, x_3, x_4\}$ and $C(x_1, x_2, x_3, x_4) = 2 \cdot \gamma_1(x_1) + 0 \cdot \gamma_2(x_2, x_3, x_1) + \gamma_3(x_3, x_2) + \frac{5}{3} \cdot \gamma_4(x_3, x_4)$. The (non-planar drawing of the planar) incidence graph of this instance is depicted in Figure 1(a). The plane graph of the instance from Definition 1 is depicted in Figure 1(b).

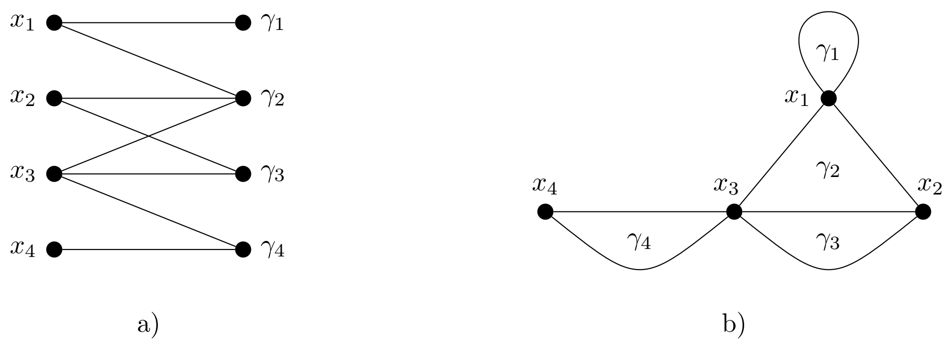

Figure 1: Graphs from Example 2.

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 [9], checking whether a VCSP instance I has a planar representation, and if so then finding (I, G, ϕ), can be done in polynomial time [39]. 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.

We define *relations* as a special case of weighted relations (also called *crisp*) with range $\{0, \infty\}$, where value 0 is assigned to tuples that are elements of the relation in the conventional sense. 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 $\mathrm{Opt}(\gamma) = \{\mathbf{x} \in \mathrm{Feas}(\gamma) \mid \gamma(\mathbf{x}) \leq \gamma(\mathbf{y}) \text{ for every } \mathbf{y} \in D^r\}$ 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 $V = \{x_1, \ldots, x_n\}$ is called *feasible* if $C(s(x_1), \ldots, s(x_n)) < \infty$.

Definition 3. 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 $\mathbf{v} = (v_1, \ldots, v_r)$ be an r-tuple of variables from V such that $b(f_o) = v_r v_{r-1} \ldots v_1 v_r$. 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 $\pi_{\mathbf{v}}(I) = \gamma$.

Example 4. Let $V = \{x_1, x_3, x_3, z\}$, $D = \{0, 1\}$, and $C(x_1, x_3, x_3, z) = \gamma(x_1, z) + \gamma(x_2, z) + \gamma(x_3, z)$ be a plane VCSP instance (I, G, ϕ) depicted in Figure 2, where γ is the binary "cut" weighted relation from Section 1; i.e., $\gamma(x, y) = 1$ if $x = y$ and $\gamma(x, y) = 0$ otherwise. Then $\rho = \pi_{(x_1, x_2, x_3)}(I)$ is a ternary weighted relation planarly expressible from $\{\gamma\}$, where $\rho(x, y, z) = 0$ if $x = y = z$ and $\rho(x, y, z) = 1$ otherwise.

To see that planar expressibility is a proper restriction of (unrestricted) expressibility [4], consider relations $\rho_= = \{(0, 0), (1, 1)\}$ and $\rho_{\mathrm{cross}} = \{(0, 0, 0, 0), (0, 1, 0, 1), (1, 0, 1, 0), (1, 1, 1, 1)\}$ on domain $D = \{0, 1\}$. Relation ρ_{cross} is expressible from the binary equality relation $\rho_=$, because $\rho_{\mathrm{cross}}(x_1, x_2, x_3, x_4) = \rho_=(x_1, x_3) + \rho_=(x_2, x_4)$. However, it is not planarly expressible. This can be proved unconditionally but here we give a simpler argument assuming P $\neq$ NP:

Relation $\rho_=$ can be included in any valued constraint language without affecting its complexity (see Lemma 6 and Theorem 7 below). On the other hand, relation ρ_{cross} enables bypassing the planarity restriction; languages from which ρ_{cross} is planarly expressible have the same complexity in the planar setting as

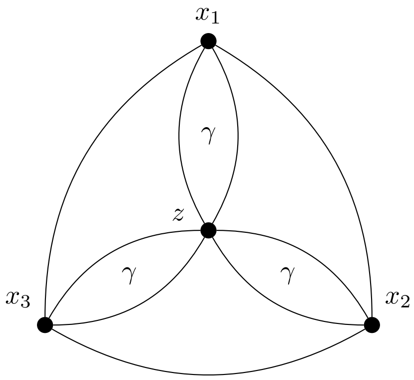

Figure 2: Instance from Example 4.

in general [9]. Consequently, if ρ_{cross} is planarly expressible from $\rho_=$ then (say) the NAE-3-Sat problem on general instances can be solved in polynomial time.

Definition 5. 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 $\mathrm{wClone}_\mathrm{p}(\Gamma)$.

An analogous notion of weighted relational clones closed under *general* (i.e. not necessarily planar) expressibility [16, 17] has been used to study the complexity of VCSPs.

Lemma 6. *For any domain D and language Γ on D, the binary equality relation $\rho_=$ on D belongs to* $\mathrm{wClone}_\mathrm{p}(\Gamma)$.

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 7. *For any valued constraint language Γ, Γ is planarly-tractable if, and only if,* $\mathrm{wClone}_\mathrm{p}(\Gamma)$ *is planarly-tractable, and Γ is planarly-intractable if, and only if,* $\mathrm{wClone}_\mathrm{p}(\Gamma)$ *is planarly-intractable.*

Proof. We show that $\mathrm{VCSP}_\mathrm{p}(\mathrm{wClone}_\mathrm{p}(\Gamma))$ is polynomial-time reducible to $\mathrm{VCSP}_\mathrm{p}(\Gamma)$. Given an instance I over $\mathrm{wClone}_\mathrm{p}(\Gamma)$, 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. □

We now define a few operations on weighted relations that will occur frequently throughout the paper. As shown in the lemma below, these operations are planarly expressible.

Definition 8. 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 $\gamma'(x_1, \ldots, x_r) = \gamma(x_1, \ldots, x_r)$ if $x_i \in D'$ and $\gamma'(x_1, \ldots, x_r) = \infty$ otherwise. A *pinning* of γ to $a \in D$ at coordinate i is the $(r-1)$-ary weighted relation defined as $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \gamma(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_r)$. Finally, 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 binary weighted relation γ is a *join* of two binary weighted relations γ_1 and γ_2 if it can be written as $\gamma(x, y) = \min_{z \in D}(\gamma_1(u_1, v_1) + \gamma_2(u_2, v_2))$ where $\{u_1, v_1\} = \{x, z\}$, $\{u_2, v_2\} = \{y, z\}$.

Lemma 9. *Let us denote by $\rho_{D'}$ the unary relation corresponding to a subdomain $D' \subseteq D$ (i.e. $\rho_{D'}(x) = 0$ if $x \in D'$ and $\rho_{D'}(x) = \infty$ otherwise).*

For any language Γ, $\mathrm{wClone_p}(\Gamma)$ is closed under addition of unary weighted relations to weighted relations of arbitrary arity, minimisation, and join. If $\rho_{D'} \in \mathrm{wClone_p}(\Gamma)$, it is closed under domain restriction to $D' \subseteq D$. If $\rho_{\{a\}} \in \mathrm{wClone_p}(\Gamma)$, it is closed under pinning to $a \in D$.

Proof. A unary weighted relation γ imposed on variable x_i can be planarly expressed by adding a parallel edge $x_i - x_{i+1}$ and a self-loop at x_i hidden in the just formed face. Minimisation over x_i can be achieved by adding an edge in the outer face between vertices x_{i-1} and x_{i+1}, thus hiding vertex x_i. A join $\gamma(x, y)$ can be achieved by adding two edges between x and y to hide z from the outer face (similarly as in Figure 2). Domain restriction is planarly expressible by imposing unary relation $\rho_{D'}$ on variable x_i; pinning can be expressed by domain restriction to $\{a\}$ and subsequent minimisation at coordinate i. $\square$

Proving results for conservative languages in Section 4, we will need only a limited subset of $\mathrm{wClone_p}(\Gamma)$ which is defined as follows.

Definition 10. For any 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.

Set Γ^* is also closed under domain restriction and pinning, as these operations can be achieved by adding unary weighted relations and minimisation.

Note that for conservative languages we have $\Gamma^* \subseteq \mathrm{wClone_p}(\Gamma)$, 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. By Theorem 7, Γ^* has the same complexity as Γ.

Lemma 12 will be useful for proving results about both Boolean and conservative valued constraint languages. Before its statement, we need to define 2-decomposable relations and introduce some notation.

Definition 11. 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

$$(a_i, a_j) \in \mathrm{Pr}_{i,j}(\rho) \iff (\exists \mathbf{x} \in \rho)\ x_i = a_i \wedge x_j = a_j \,. \tag{1}$$

Relation ρ is *2-decomposable* if

$$\mathbf{x} \in \rho \iff \bigwedge_{1 \le i,j \le r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho) \,. \tag{2}$$

Note that all unary and binary relations are 2-decomposable.

For any r-tuple $\mathbf{z}$, we denote its ith component by z_i. Let $I \subseteq \{1, \dots, r\}$ be a subset of coordinates, 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 12. *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*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J) \,, \tag{3}$$

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

$$\gamma_{i,j}(x_i, x_j) + \gamma_{i,j}(y_i, y_j) < \gamma_{i,j}(x_i, y_j) + \gamma_{i,j}(y_i, x_j) \,. \tag{4}$$

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 $(y_i, x_j) \in \mathrm{Feas}(\gamma_{i,j})$.

Proof. We prove the lemma by induction on the arity of γ. If $|I| = 0$, $|J| = 0$, or $|I| = |J| = 1$, the claim holds trivially. Otherwise we may without loss of generality assume that $|J| \geq 2$. Let $k \in J$ be an arbitrary coordinate and define $J' = J \setminus \{k\}$. We extend our notation $\cdot$ to $I, J', \{k\}$ as a finer partition of $\{1, \ldots, r\}$, and write for instance $\mathbf{x}$ as $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot x_k$.

We first consider the case when $\mathbf{x}_I \cdot \mathbf{y}_{J'} \cdot x_k, \mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k \notin \mathrm{Feas}(\gamma)$. We restrict the domain at coordinate k to $\{x_k, y_k\}$ and minimise over it to obtain an $(r-1)$-ary weighted relation γ' with coordinates partition I, J'. It holds $\gamma'(\mathbf{x}_I \cdot \mathbf{x}_{J'}) \leq \gamma(\mathbf{x}), \gamma'(\mathbf{y}_I \cdot \mathbf{y}_{J'}) \leq \gamma(\mathbf{y}), \gamma'(\mathbf{x}_I \cdot \mathbf{y}_{J'}) = \gamma(\mathbf{x}_I \cdot \mathbf{y}_J), \gamma'(\mathbf{y}_I \cdot \mathbf{x}_{J'}) = \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)$, and the claim follows directly from the induction hypothesis for γ'.

We may now assume without loss of generality that $\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$. If

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot x_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k) < \gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot x_k)\,, \tag{5}$$

we pin γ at every coordinate $j' \in J'$ to its respective label $x_{j'}$ to obtain a weighted relation γ' with coordinates partition $I, \{k\}$. The claim then follows from the induction hypothesis for γ'. Note that $\mathbf{x}_I \cdot \mathbf{y}_J \in \mathrm{Feas}(\gamma)$ implies $(x_i, y_k) \in \mathrm{Pr}_{i,k}(\mathrm{Feas}(\gamma))$ for all $i \in I$; together with $(x_{j'}, y_k) \in \mathrm{Pr}_{j',k}(\mathrm{Feas}(\gamma)), (x_i, x_{j'}) \in \mathrm{Pr}_{i,j'}(\mathrm{Feas}(\gamma))$ for all $i \in I, j' \in J'$ (as $\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k, \mathbf{x} \in \mathrm{Feas}(\gamma)$) this implies $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$ if $\mathrm{Feas}(\gamma)$ is 2-decomposable.

If (5) does not hold, we have $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$, and therefore

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_{J'} \cdot y_k) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k)\,, \tag{6}$$

otherwise the sum of negated (5) and (6) would contradict (3). We resolve this case analogously to the previous one, this time pinning γ at coordinate k to y_k. $\square$

2.3. Algebraic Properties

We apply a k-ary operation $f : D^k \to D$ to k r-tuples componentwise; that is, if $\mathbf{x}^1 = (x_1^1, \ldots, x_r^1), \mathbf{x}^2 = (x_1^2, \ldots, x_r^2), \ldots, \mathbf{x}^k = (x_1^k, \ldots, x_r^k)$, then

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

The following notion is at the heart of the algebraic approach to decision CSPs [21].

Definition 13. 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 $\mathbf{x}^1, \ldots, \mathbf{x}^k \in \mathrm{Feas}(\gamma)$, we have $f(\mathbf{x}^1, \ldots, \mathbf{x}^k) \in \mathrm{Feas}(\gamma)$. 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 $\pi_i^{(k)}(x_1, \ldots, x_k) = x_i$ for some $1 \leq i \leq k$. 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 [4].

Definition 14. Let γ be a weighted relation on D. A list $\langle f_1, \ldots, f_k \rangle$ of k-ary polymorphisms of γ is a k-ary *multimorphism* of γ (and γ *admits* $\langle f_1, \ldots, f_k \rangle$) if, for every $\mathbf{x}^1, \ldots, \mathbf{x}^k \in \mathrm{Feas}(\gamma)$, we have

$$\sum_{i=1}^{k} \gamma(f_i(\mathbf{x}^1, \ldots, \mathbf{x}^k)) \;\leq\; \sum_{i=1}^{k} \gamma(\mathbf{x}^i)\,. \tag{7}$$

We say that $\langle f_1, \ldots, f_k \rangle$ 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 [16] and thus planar weighted relational clones do as well.

Example 15. The class of submodular functions on $D = \{0, 1\}$ [40] 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 $\gamma(\min(\mathbf{x}, \mathbf{y})) + \gamma(\max(\mathbf{x}, \mathbf{y})) \leq \gamma(\mathbf{x}) + \gamma(\mathbf{y})$.

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 $x, y \in D$.

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 16. *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 on D.

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 16 is based on Lemmas 20–23 proved below.

We will need the following definition and an easy lemma.

Definition 17. Let γ be an r-ary weighted relation and $i \in \{1, \ldots, r\}$. The *$=$-restriction* of γ at i is the r-ary weighted relation γ' such that $\gamma'(\mathbf{x}) = \gamma(\mathbf{x})$ if $x_i = x_{i+1}$ (where $x_{r+1} = x_1$) and $\gamma'(\mathbf{x}) = \infty$ otherwise. The *$\neq$-restriction* of γ at i is the r-ary weighted relation γ' such that $\gamma'(\mathbf{x}) = \gamma(\mathbf{x})$ if $x_i \neq x_{i+1}$ and $\gamma'(\mathbf{x}) = \infty$ otherwise.

We will denote by $\oplus$ the addition modulo 2 operation on $\{0, 1\}$ and its extension to tuples. Let $\mathbf{0}^r$ ($\mathbf{1}^r$) be the zero (one) r-tuple. The *negation* of an r-tuple $\mathbf{x}$ is $\overline{\mathbf{x}} = \mathbf{x} \oplus \mathbf{1}^r$. Let $\mathbf{e}_i^r$ be the r-tuple with a one at coordinate i and zeros elsewhere. The *twist* of γ at i is the r-ary weighted relation γ' defined as $\gamma'(\mathbf{x}) = \gamma(\mathbf{x} \oplus \mathbf{e}_i^r)$.

In other words, a twist switches roles of labels 0 and 1 at a single coordinate.

Example 18. Let ρ be the ternary "not-all-equal" relation from Section 1; i.e., $\rho(x, y, z) = \infty$ if $x = y = z$ and $\rho(x, y, z) = 0$ otherwise. The twist of ρ at the first coordinate is the ternary relation ρ' defined by $\rho'(x, y, z) = \infty$ if $x = 0$ and $y = z = 1$, or $x = 1$ and $y = z = 0$; in all other cases $\rho'(x, y, z) = 0$.

Lemma 19. *Let Γ be a valued constraint language and $\gamma \in \mathrm{wClone_p}(\Gamma)$ a weighted relation. Then*

- *all $=$-restrictions of γ belong to $\mathrm{wClone_p}(\Gamma)$,*
- *if $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$, all $\neq$-restrictions and twists of γ belong to $\mathrm{wClone_p}(\Gamma)$,*
- *if $\rho_0, \rho_1 \in \mathrm{wClone_p}(\Gamma)$, all pinnings of γ belong to $\mathrm{wClone_p}(\Gamma)$.*

Proof. Both $=$-restriction and $\neq$-restriction are planarly expressible by adding a parallel edge between vertices x_i, x_{i+1} and imposing on them the binary equality or disequality relation respectively. To implement a twist, we introduce a new variable x_i' in the outer face, connect it with x_i by two parallel edges, impose the binary disequality relation on x_i and x_i', and hide vertex x_i by adding edges $x_{i-1} - x_i'$ and $x_{i+1} - x_i'$. Pinnings belong to $\mathrm{wClone_p}(\Gamma)$ by Lemma 9. $\square$

Lemma 20. *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 $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$.*

Proof. If Γ does not admit $\langle c_0 \rangle$, it contains a weighted relation assigning to the zero tuple a value larger than the optimum. Applying Opt, we have that $\mathrm{wClone_p}(\Gamma)$ contains a *relation* that is not invariant under c_0. We denote by ρ such a relation of minimum arity and by r its arity. Relation ρ is non-empty, but $\mathbf{0}^r \notin \rho$. If $r = 1$, then $\rho = \rho_1 \in \mathrm{wClone_p}(\Gamma)$.

Otherwise, $\mathbf{e}_i^r \in \rho$ for all i, because the minimisation of ρ over coordinate i produces a non-empty relation invariant under c_0 (by the choice of ρ) and hence containing $\mathbf{0}^{r-1}$. If $r \geq 3$, the $=$-restriction of ρ at coordinate 2 followed by the minimisation results in an $(r-1)$-ary relation ρ' with $\mathbf{e}_1^{r-1} \in \rho'$ and $\mathbf{0}^{r-1} \notin \rho'$, which contradicts the choice of ρ. Therefore, $r = 2$. If $(1,1) \in \rho$, we would again get a contradiction by applying the $=$-restriction and minimisation at coordinate 1. Hence we have $\rho = \rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$.

By the analogous argument for multimorphism $\langle c_1 \rangle$ we get $\rho_0 \in \mathrm{wClone_p}(\Gamma)$ or $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$. $\square$

Lemma 21. *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 $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$.*

Proof. If $\min \notin \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$, we choose a minimum-arity relation $\rho'_{\vee} \in \mathrm{wClone_p}(\Gamma)$ that is not invariant under min; its arity r is at least 2. Let $\mathbf{x}, \mathbf{y} \in \rho'_{\vee}$ be r-tuples such that $\min(\mathbf{x}, \mathbf{y}) \notin \rho'_{\vee}$. Tuples $\mathbf{x}, \mathbf{y}$ differ at every coordinate, otherwise we would obtain a contradiction with the choice of $\rho'_{\vee}$ by taking a pinning instead. Therefore, $\min(\mathbf{x}, \mathbf{y}) = \mathbf{0}^r \notin \rho'_{\vee}$ and, by the same argument as in Lemma 20, we have $\mathbf{e}_i^r \in \rho'_{\vee}$ for all i. But then $r = 2$, otherwise we could take as $\mathbf{x}, \mathbf{y}$ tuples $\mathbf{e}_2^r, \mathbf{e}_3^r$ which agree at the first coordinate, and obtain a smaller counterexample by pinning. Hence we have $\rho_{\neq} \subseteq \rho'_{\vee} \subseteq \rho_{\neq} \cup \{(1,1)\}$.

If $\min \in \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$, then we choose a minimum-arity weighted relation $\gamma \in \mathrm{wClone_p}(\Gamma)$ that does not admit multimorphism $\langle \min, \min \rangle$ and denote its arity by r. Let $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ be r-tuples such that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < 2 \cdot \gamma(\min(\mathbf{x}, \mathbf{y}))$. Without loss of generality, we have $\gamma(\mathbf{x}) < \gamma(\min(\mathbf{x}, \mathbf{y}))$ and may assume that $\mathbf{y} = \min(\mathbf{x}, \mathbf{y})$. Again, $\mathbf{x}$ and $\mathbf{y}$ must differ at every coordinate, which implies $\mathbf{x} = \mathbf{1}^r, \mathbf{y} = \mathbf{0}^r$. If $r \geq 2$, we would obtain a contradiction by applying the $=$-restriction and minimisation at coordinate 1. Hence, $r = 1$ and by scaling and adding a constant to γ we get $\gamma_1 \in \mathrm{wClone_p}(\Gamma)$.

Analogously, if $\max \notin \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$, we get $\rho'_{\uparrow} \in \mathrm{wClone_p}(\Gamma)$ where $\rho'_{\uparrow}$ is a binary relation such that $\rho_{\neq} \subseteq \rho'_{\uparrow} \subseteq \rho_{\neq} \cup \{(0,0)\}$. Otherwise, $\gamma_0 \in \mathrm{wClone_p}(\Gamma)$. It holds

$$\begin{aligned} \rho_{\neq}(x,y) &= \rho'_{\vee}(x,y) + \rho'_{\uparrow}(x,y) & (8) \\ &= \mathrm{Opt}\left(\rho'_{\vee}(x,y) + \gamma_0(x) + \gamma_0(y)\right) & (9) \\ &= \mathrm{Opt}\left(\rho'_{\uparrow}(x,y) + \gamma_1(x) + \gamma_1(y)\right), & (10) \end{aligned}$$

so $\rho_{\neq}$ can be constructed with a planar gadget if at least one of min, max is not a polymorphism of $\mathrm{wClone_p}(\Gamma)$.

Finally, consider the case when $\min, \max \in \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$ and hence $\gamma_0, \gamma_1 \in \mathrm{wClone_p}(\Gamma)$. Set $\mathrm{wClone_p}(\Gamma)$ is then a conservative language, so we have $\mathrm{wClone_p}(\Gamma)^* = \mathrm{wClone_p}(\Gamma)$. We choose a minimum-arity weighted relation $\gamma \in \mathrm{wClone_p}(\Gamma)$ that does not admit multimorphism $\langle \min, \max \rangle$ and denote its arity by r. Let $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ be tuples such that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\min(\mathbf{x}, \mathbf{y})) + \gamma(\max(\mathbf{x}, \mathbf{y}))$. Note that $\min(\mathbf{x}, \mathbf{y}), \max(\mathbf{x}, \mathbf{y}) \in \mathrm{Feas}(\gamma)$. By the choice of γ, tuples $\mathbf{x}, \mathbf{y}$ must differ at every coordinate, and hence $\mathbf{y} = \overline{\mathbf{x}}$, $\min(\mathbf{x}, \mathbf{y}) = \mathbf{0}^r$, $\max(\mathbf{x}, \mathbf{y}) = \mathbf{1}^r$. We partition coordinates $\{1, \ldots, r\}$ into $I = \{i \mid x_i = 0\}$ and $J = \{j \mid x_j = 1\}$. By Lemma 12, $\{\gamma\}^* \subseteq \mathrm{wClone_p}(\Gamma)$ contains a *binary* weighted relation that does not admit multimorphism $\langle \min, \max \rangle$, and hence $r = 2$. It holds $\gamma(0,1) + \gamma(1,0) < \gamma(0,0) + \gamma(1,1)$, where all the values are finite. We may assume that $\gamma(0,0) + \gamma(1,1) - \gamma(0,1) - \gamma(1,0) = 2$ and $\gamma(0,0) = 1$ (this can be achieved by scaling and adding a constant). We define unary weighted relations $\mu_1, \mu_2 \in \mathrm{wClone_p}(\Gamma)$ as $\mu_1(0) = \mu_2(0) = 0$, $\mu_1(1) = -\gamma(1,0)$, $\mu_2(1) = -\gamma(0,1)$. By adding μ_1 and μ_2 to γ at the first and second coordinate respectively we get $\gamma_{\neq}$, and therefore $\rho_{\neq} = \mathrm{Opt}(\gamma_{\neq}) \in \mathrm{wClone_p}(\Gamma)$. $\square$

Lemma 22. *Let Γ be a valued constraint language that does not admit multimorphism $\langle \neg \rangle$. If $\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$, then $\rho_0, \rho_1 \in \mathrm{wClone_p}(\Gamma)$.*

Proof. We choose a minimum-arity weighted relation $\gamma \in \mathrm{wClone_p}(\Gamma)$ that does not admit multimorphism $\langle \neg \rangle$ and denote its arity by r. Let $\mathbf{x} \in \mathrm{Feas}(\gamma)$ be an r-tuple such that $\gamma(\mathbf{x}) \neq \gamma(\overline{\mathbf{x}})$. It must hold $r = 1$, otherwise we would get a smaller counterexample by applying the $=$-restriction or $\neq$-restriction at the first coordinate (depending on whether $x_1 = x_2$ or $x_1 \neq x_2$) followed by minimisation. Hence, $\mathrm{Opt}(\gamma) = \rho_0$ or $\mathrm{Opt}(\gamma) = \rho_1$. Say $\mathrm{Opt}(\gamma) = \rho_0$, the other case is analogous. Then the twist $\gamma'(x) = \gamma(x \oplus 1)$ of γ satisfies $\mathrm{Opt}(\gamma') = \rho_1$. $\square$

Lemma 23. *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 $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$.*

Proof. If $\mathrm{Mn} \notin \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$, we choose a minimum-arity relation $\rho \in \mathrm{wClone_p}(\Gamma)$ that is not invariant under Mn. Its arity r must be at least 2; let us first assume $r \geq 3$. For any triple of r-tuples from ρ that agree at some coordinate, the r-tuple obtained by applying Mn to them also belongs to ρ (otherwise we would get a contradiction with the choice of ρ by taking a pinning instead). Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \rho$ be r-tuples such that $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \notin \rho$. Without loss of generality, we may assume that $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0}^r$ (this can be achieved with twists). By the same argument as in Lemma 20, we have $\mathbf{e}_i^r \in \rho$ for all i. Let $\mathbf{w} \in \rho$ be a tuple with the minimum even number of ones (such a tuple exists as at least one of $\mathbf{x}, \mathbf{y}, \mathbf{z}$ contains an even number of ones). If $\mathbf{w} \neq \mathbf{1}^r$, there are distinct coordinates i, j, k with $w_i = w_j = 1, w_k = 0$. Because $\mathbf{w}, \mathbf{e}_i^r, \mathbf{e}_j^r$ agree at coordinate k, tuple $\mathrm{Mn}(\mathbf{w}, \mathbf{e}_i^r, \mathbf{e}_j^r)$ belongs to ρ. However, it has two fewer ones than $\mathbf{w}$, which is a contradiction. Hence, $\mathbf{w} = \mathbf{1}^r$ and $r \geq 4$. But then $\mathrm{Mn}(\mathbf{1}^r, \mathbf{e}_3^r, \mathbf{e}_4^r) \notin \rho$ (as it contains an even number of ones), and we obtain a smaller counterexample by taking the $=$-restriction of ρ at the first coordinate followed by minimisation. Therefore, $r = 2$ and $|\rho| = 3$. Using twists, we can get from ρ relation $\rho_{\uparrow} = \{(0,0), (0,1), (1,0)\} \in \mathrm{wClone_p}(\Gamma)$.

If $\mathrm{Mj} \notin \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$, we choose a minimum-arity relation $\rho'_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$ that is not invariant under Mj. Its arity r must be at least 3 since every unary and binary relation admits Mj as a polymorphism. By the same argument as for Mn, we may assume $\mathbf{0}^r \notin \rho'_{\text{1-in-3}}$, and it can be shown that $\mathbf{e}_i^r \in \rho'_{\text{1-in-3}}$ for all i. If $r \geq 4$, tuples $\mathbf{e}_1^r, \mathbf{e}_2^r, \mathbf{e}_3^r$ and $\mathrm{Mj}(\mathbf{e}_1^r, \mathbf{e}_2^r, \mathbf{e}_3^r) = \mathbf{0}^r$ agree at coordinate 4; we then obtain a smaller counterexample by pinning. Therefore, $r = 3$.

If neither of Mn, Mj is a polymorphism of $\mathrm{wClone_p}(\Gamma)$, we have

$$\rho_{\text{1-in-3}}(x, y, z) = \rho'_{\text{1-in-3}}(x, y, z) + \rho_{\uparrow}(x, y) + \rho_{\uparrow}(y, z) + \rho_{\uparrow}(z, x)\,, \tag{11}$$

which can be implemented in a planar way, and hence $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$. Otherwise, Γ is not a crisp language (i.e. not a $\{0, \infty\}$-valued language), because that would make it admit multimorphism $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$ or $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$. Let $\mu \in \mathrm{wClone_p}(\Gamma)$ be a minimum-arity non-crisp weighted relation and $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\mu)$ tuples such that $\mu(\mathbf{x}) \neq \mu(\mathbf{y})$. Tuples $\mathbf{x}, \mathbf{y}$ differ at every coordinate (otherwise we could obtain a smaller counterexample by pinning), and hence $\mathbf{y} = \overline{\mathbf{x}}$. Moreover, μ is unary, otherwise we could apply the $=$-restriction or $\neq$-restriction at the first coordinate (depending on whether $x_1 = x_2$ or $x_1 \neq x_2$) followed by minimisation to obtain a smaller counterexample. If $\mu(0) < \mu(1)$, we get $\gamma_0 \in \mathrm{wClone_p}(\Gamma)$ by scaling μ and adding a constant, and $\gamma_1 \in \mathrm{wClone_p}(\Gamma)$ by twisting γ_0; the case $\mu(0) > \mu(1)$ is symmetric. It holds

$$\begin{aligned}\rho_{\text{1-in-3}}(x, y, z) &= \mathrm{Opt}\left(\rho_{\uparrow}(x, y) + \rho_{\uparrow}(y, z) + \rho_{\uparrow}(z, x) + \gamma_1(x) + \gamma_1(y) + \gamma_1(z)\right) &(12)\\ &= \mathrm{Opt}\left(\rho'_{\text{1-in-3}}(x, y, z) + \gamma_0(x) + \gamma_0(y) + \gamma_0(z)\right)\,. &(13)\end{aligned}$$

Both can be implemented planarly, and therefore $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$ if exactly one of Mn, Mj is a polymorphism of $\mathrm{wClone_p}(\Gamma)$.

Finally, we consider the case when both $\mathrm{Mn}, \mathrm{Mj} \in \mathrm{Pol}(\mathrm{wClone_p}(\Gamma))$. Let $\gamma \in \mathrm{wClone_p}(\Gamma)$ be an r-ary weighted relation of the minimum arity for which Inequality (7) does not hold as *equality* for multimorphism $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathrm{Feas}(\gamma)$ be r-tuples that violate the equality. They do not agree at any coordinate (otherwise we could obtain a smaller counterexample by pinning), and hence $\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ differ everywhere. Without loss of generality, we may assume that $\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0}^r$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{1}^r$ (this can be achieved with twists) and $\mathbf{z} \neq \mathbf{0}^r$. Note that $\mathbf{0}^r, \mathbf{1}^r \in \mathrm{Feas}(\gamma)$ because Mj, Mn are polymorphisms of γ. Tuples $\mathbf{x}, \mathbf{y}, \mathbf{0}^r$ agree at all coordinates i where $z_i = 1$, and hence they satisfy (7) as equality, i.e.

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) + \gamma(\mathbf{0}^r) = 2 \cdot \gamma(\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r)) + \gamma(\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r)) = 2 \cdot \gamma(\mathbf{0}^r) + \gamma(\overline{\mathbf{z}})\,. \tag{14}$$

Because $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) + \gamma(\mathbf{z}) \neq 2 \cdot \gamma(\mathbf{0}^r) + \gamma(\mathbf{1}^r)$, this implies $\gamma(\mathbf{z}) + \gamma(\overline{\mathbf{z}}) \neq \gamma(\mathbf{0}^r) + \gamma(\mathbf{1}^r)$. We are going to apply Lemma 12 for this disequality. Language $\mathrm{wClone_p}(\Gamma)$ is conservative (as it contains both γ_0, γ_1), and hence $\mathrm{wClone_p}(\Gamma)^* = \mathrm{wClone_p}(\Gamma)$. It admits a majority polymorphism, therefore every relation in $\mathrm{wClone_p}(\Gamma)$ is 2-decomposable [41]. We partition coordinates $\{1, \ldots, r\}$ into $I = \{i \mid z_i = 0\}$ and $J = \{j \mid z_j = 1\}$.

By Lemma 12, there is a binary weighted relation $\gamma' \in \{\gamma\}^* \subseteq \mathrm{wClone_p}(\Gamma)$ with $\mathrm{Feas}(\gamma') = D^2$ and $\gamma'(0,1)+\gamma'(1,0) \neq \gamma'(0,0)+\gamma'(1,1)$. We may assume that $\gamma'(0,1)+\gamma'(1,0) < \gamma'(0,0)+\gamma'(1,1)$, otherwise we apply a twist. As in the proof of Lemma 21, weighted relation $\gamma_{\neq}$ can be obtained from γ'. Then we planarly construct $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$ as

$$\rho_{\text{1-in-3}}(x,y,z) = \mathrm{Opt}\left(\gamma_{\neq}(x,y)+\gamma_{\neq}(y,z)+\gamma_{\neq}(z,x)+\gamma_0(x)+\gamma_0(y)+\gamma_0(z)\right) . \tag{15}$$

□

Proof (of Theorem 16). Since Γ is intractable we know, by [4, 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 20, 21, and 22, we have $\rho_0,\rho_1,\rho_{\neq} \in \mathrm{wClone_p}(\Gamma)$ and hence by Lemma 23 $\rho_{\text{1-in-3}} \in \mathrm{wClone_p}(\Gamma)$. Planar 1-in-3 Positive 3-Sat problem is NP-complete [42], and therefore Γ is planarly-intractable by Theorem 7. □

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 24. *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 [10, 11]. As mentioned in Section 1, for Boolean domains Theorem 24 follows from Theorem 16. Thus, the only tractable Boolean conservative languages in the planar setting are given by the multimorphisms $\langle \min,\max\rangle$ and $\langle \mathrm{Mj},\mathrm{Mj},\mathrm{Mn}\rangle$ [4].

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 25. 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 [43] that languages admitting an STP multimorphism are tractable.

Definition 26. A ternary multimorphism $\langle f,g,h\rangle$ is called an MJN if f and g are (possibly equal) majority operations and h is a minority operation.

It was shown in [10] that languages admitting an MJN multimorphism are tractable.

Theorem 27 ([10])**.** *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 27 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 $\gamma \in \text{wClone}(\Gamma)$ 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 [10] that relies on [43], or by an LP relaxation [11].

Our approach is to follow the above-described proof and adapt it to the planar setting. We remark that similar graphs to G_Γ have been important in other studies of (V)CSPs. In particular, in the classification of conservative CSPs [13] and in the classification of Minimum Cost Homomorphism problems [44]. In [13], the graph has labels as vertices and three types of edges depending on three types of polymorphisms. In [44], the graph has, as in our case, pairs of labels as vertices but the edges of the graph are defined, informally, via a min/max polymorphism rather than a $\langle \min, \max \rangle$ multimorphism. Also, edges in [44] are not classified as soft or hard.

It is natural to replace $\text{wClone}(\Gamma)$ by $\text{wClone}_\text{p}(\Gamma)$ 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 27 from [10] heavily relies on [44], which guarantees, unless in an NP-hard case, the existence of a majority polymorphism and hence that the language is 2-decomposable. Second, some of the gadgets (and in particular the "i-expansion" from [10, Section 6.4]) are not necessarily planar. In more detail, [44] builds a similar graph to ours (as described above) and argues that, unless in an NP-hard case, this graph is bipartite (part of our G_Γ will also be bipartite). This property is then used in [44] to argue about the existence of a majority polymorphism. However, this is proved in [44] using clones and depends on the Galois connection between clones and relational co-clones; such a connection is not known for planar expressibility!

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 define our graph based on a language closure Γ^*, which 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 [44] but instead prove directly 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 MJN multimorphism satisfies Inequality (7) 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 [10].

We remark that it is not clear how to derive non-trivial properties of graph G_Γ used in our proofs from the related graph defined in [10] apart from the obvious fact that our graph is a subgraph of the graph from [10]. We believe that with more work one can derive that the two graphs are in fact the same using techniques and proofs from this paper, but have not done so since our goal was to obtain a complexity classification.

The rest of this section is devoted to proving Theorem 24.

Definition 28. 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 $(a_1, b_2), (b_1, a_2) \in \text{Feas}(\gamma)$ and

$$\gamma(a_1, b_2) + \gamma(b_1, a_2) < \gamma(a_1, a_2) + \gamma(b_1, b_2)\,. \qquad (16)$$

If there exists such a weighted relation γ with at least one of $(a_1, a_2), (b_1, b_2)$ belonging to $\text{Feas}(\gamma)$, 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.

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

Lemma 29. *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^*.*

Proof. Both *if* implications follow directly from the definition of G_Γ; we need to prove the *only if* part. Let γ be a weighted relation establishing edge $(a_1, b_1) - (a_2, b_2)$ such that $\mathrm{Feas}(\gamma) \subseteq \{a_1, b_1\} \times \{a_2, b_2\}$ (this can be always achieved by domain restriction). Note that we may add to γ any unary finite-valued weighted relation without invalidating (16). We choose any $\lambda \in \mathbb{Q}$ such that $\lambda < \gamma(b_1, b_2)$ and $\gamma(a_1, b_2) + \gamma(b_1, a_2) - \lambda < \gamma(a_1, a_2)$. Note that such λ exists due to (16). We define unary weighted relations γ_1, γ_2 such that $\gamma_1(a_1) = \lambda - \gamma(a_1, b_2)$, $\gamma_2(a_2) = \lambda - \gamma(b_1, a_2)$, and $\gamma_1(x) = \gamma_2(x) = 0$ otherwise. Now consider binary weighted relation γ' defined as $\gamma'(x, y) = \gamma(x, y) + \gamma_1(x) + \gamma_2(y)$. We have $\gamma'(a_1, b_2) = \gamma'(b_1, a_2) = \lambda$ and $\lambda < \gamma'(a_1, a_2), \gamma'(b_1, b_2)$, so then $\mathrm{Opt}(\gamma') = \{(a_1, b_2), (b_1, a_2)\} \in \Gamma^*$.

If the edge is soft and $(a_1, a_2), (b_1, b_2) \in \mathrm{Feas}(\gamma)$, we proceed as above with $\lambda = \gamma(b_1, b_2)$, so that $\mathrm{Opt}(\gamma') = \{(b_1, b_2), (a_1, b_2), (b_1, a_2)\} \in \Gamma^*$. Otherwise we simply take $\mathrm{Feas}(\gamma) \in \Gamma^*$. ☐

We show that the absence of soft self-loops is a necessary condition for planar tractability.

Theorem 30. *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 that $\rho = \{(a, a), (a, b), (b, a)\} \in \Gamma^*$ by Lemma 29. 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 $\{a, b\}$. By [4, Theorem 7.1], Γ' is intractable (in particular, Γ' does not fall into either of the two tractable cases for Boolean conservative valued constraint languages [4] corresponding to the $\langle \min, \max \rangle$ and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ multimorphisms). Observe that Γ' is not self-complementary since neither of its weighted relations is self-complementary. By Theorem 16, Γ' is planarly-intractable and thus, by Theorem 7, so is Γ. ☐

It remains to show that this condition is also sufficient.

Theorem 31. *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 $\overline{M}$.*

In order to prove Theorem 31, we now introduce several lemmas. From now on we will assume that G_Γ has no soft self-loop.

Lemma 32. *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 29, 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 33. *Every relation in Γ^* is 2-decomposable.*

Proof. Let $\rho \in \Gamma^*$ be an r-ary relation. By definition, $\mathbf{x} \in \rho$ implies $\bigwedge_{1 \le i,j \le r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho)$. We prove the converse implication by induction on r. If $r \le 2$, 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 $r \ge 4$. 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 $k \ge 4$ 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 an extension of the definition of an edge in G_Γ to non-binary weighted relations.

Lemma 34. *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*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{17}$$

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 12, 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 $\{\gamma\}^*$ is 2-decomposable by Lemma 33), and hence the edge is soft. ☐

Lemma 35. *Let $\gamma \in \Gamma^*$ be an r-ary weighted relation and $I, J \subseteq \{1, \ldots, r\}$ a partition of its coordinates. If $\mathbf{x}, \mathbf{y}, \mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J \in \mathrm{Feas}(\gamma)$ and, for all $i \in I$, $(x_i, y_i) \in \overline{M}$, then*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) = \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,. \tag{18}$$

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 34, graph G_Γ contains a soft edge incident to (x_i, y_i) for some $i \in I$, which contradicts Lemma 32. ☐

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:

$$\langle \sqcap, \sqcup \rangle(x, y) = \begin{cases} (x, y) & \text{if } (x, y) \in M_1, & \text{(19a)} \\ (y, x) & \text{if } (x, y) \in M_2, & \text{(19b)} \\ (x, y) & \text{otherwise.} & \text{(19c)} \end{cases}$$

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

Theorem 36. *$\langle \sqcap, \sqcup \rangle$ is a multimorphism of Γ.*

Proof. Let $\gamma \in \Gamma$ be an r-ary weighted relation and $\mathbf{x}, \mathbf{y} \in \text{Feas}(\gamma)$. Suppose for the sake of contradiction that (7) does not hold. We partition set $\{1, \ldots, r\}$ into I and J: Set J consists of all coordinates j such that case (19b) 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 34, graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$ for some $i \in I, j \in J$, which contradicts Lemma 32. $\square$

The following definition corresponds to the "μ function" from [10, Section 6].

Definition 37. 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 in order to be a polymorphism of Γ.

Lemma 38. *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 their join 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 $(b, c) \in \overline{M}$. $\square$

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

$$\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle(x, y, z) = \begin{cases} (x, y, z) & \text{if } x = y \wedge (y, z) \in \overline{M} \text{ or } xy|z, & \text{(20a)} \\ (z, x, y) & \text{if } z = x \wedge (x, y) \in \overline{M} \text{ or } zx|y, & \text{(20b)} \\ (y, z, x) & \text{if } y = z \wedge (z, x) \in \overline{M} \text{ or } yz|x, & \text{(20c)} \\ (x, y, z) & \text{otherwise.} & \text{(20d)} \end{cases}$$

Note that the operations of $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ are majorities and a minority on $\overline{M}$. Also note that in the subcase $x = y \wedge (y, z) \in \overline{M}$ of case (20a), the output has to be (x, y, z) for $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ to be an MJN multimorphism of Γ on $\overline{M}$ (and similarly for the first subcase of case (20b) and case (20c)). It is the other cases where there is some freedom and where we differ from [10].

Theorem 39. *$\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ is a multimorphism of Γ.*

We will actually prove that (7) in Definition 14 holds with *equality.*

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

First we show that case (20b) does not occur. Let I be the set of coordinates i such that case (20b) 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 38), and either $z_i = x_i$ or $z_i x_i | y_i$.

We claim that $\{x_i, y_i, z_i\} \times \{x_j, y_j, z_j\} \subseteq \text{Pr}_{i,j}(\text{Feas}(\gamma))$ for all $i \in I, j \in J$. Note that we already have $(x_i, x_j), (y_i, y_j), (z_i, z_j) \in \text{Pr}_{i,j}(\text{Feas}(\gamma))$. It holds

$$(x_i, y_j) \in \text{Pr}_{i,j}(\text{Feas}(\gamma)) \iff (y_i, x_j) \in \text{Pr}_{i,j}(\text{Feas}(\gamma)), \quad (21)$$

$$(z_i, y_j) \in \text{Pr}_{i,j}(\text{Feas}(\gamma)) \iff (y_i, z_j) \in \text{Pr}_{i,j}(\text{Feas}(\gamma)), \quad (22)$$

otherwise there would be a soft edge in $\overline{M}$ (i.e. soft edge $(x_i, y_i) - (y_j, x_j)$ and $(z_i, y_i) - (y_j, z_j)$ respectively).

If $(x_i, y_j), (z_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, then there are edges $(x_i, y_i) - (y_j, x_j), (z_i, y_i) - (y_j, z_j)$, and hence $(x_j, y_j), (z_j, y_j) \in \overline{M}$. Because case (20b) does not apply at coordinate j, it holds $z_j \neq x_j$, and therefore labels x_j, y_j, z_j are all distinct. But then $(x_i, z_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, as otherwise we would have that $\{(x_i, z_j), (x_i, x_j), (y_i, y_j)\} \in \Gamma^*$ (obtained by domain restriction of $\mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$), and thus $z_j x_j | y_j$ would hold. Analogously, we have $(z_i, x_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$. This implies $z_i \neq x_i$, and hence $z_i x_i | y_i$ holds. By domain restriction of $\mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ we obtain bijection relation $\{(x_i, x_j), (y_i, y_j), (z_i, z_j)\} \in \Gamma^*$; joining it with a binary relation showing that $z_i x_i | y_i$ gives us $z_j x_j | y_j$, which is a contradiction.

If $(x_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ and $(z_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, then we have $z_i \neq x_i$, $z_i x_i | y_i$, and $(z_j, y_j) \in \overline{M}$. It must also hold $(x_i, z_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, otherwise there would be a soft edge incident to vertex (z_j, y_j). But then we have $\{(x_i, y_j), (y_i, y_j), (z_i, z_j)\} \in \Gamma^*$, which implies $x_i y_i | z_i$ and contradicts Lemma 38. The case when $(x_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ and $(z_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ can be ruled out by an analogous argument.

Therefore, we have $(x_i, y_j), (z_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$. It must also hold $(x_i, z_j), (z_i, x_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, otherwise there would be a soft edge in $\overline{M}$ (incident to vertex (x_i, y_i) and (z_i, y_i) respectively). Hence, we have shown that $\{x_i, y_i, z_i\} \times \{x_j, y_j, z_j\} \subseteq \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$.

Because $\mathrm{Feas}(\gamma)$ is 2-decomposable by Lemma 33, 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

$$\gamma(\mathbf{y}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J) = \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{h}_J)\,, \tag{23}$$

otherwise we would obtain a smaller counterexample by pinning γ at every coordinate $i \in I$ to its respective label y_i. This gives $\mathbf{y}_I \cdot \mathbf{f}_J, \mathbf{y}_I \cdot \mathbf{g}_J, \mathbf{y}_I \cdot \mathbf{h}_J \in \mathrm{Feas}(\gamma)$; by an analogous argument we get $\mathbf{u}_I \cdot \mathbf{v}_J \in \mathrm{Feas}(\gamma)$ for any $\mathbf{u} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ and $\mathbf{v} \in \{\mathbf{f}, \mathbf{g}, \mathbf{h}\}$. By Lemma 35, it holds

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) = \gamma(\mathbf{x}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{24}$$

$$\gamma(\mathbf{z}_I \cdot \mathbf{z}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) = \gamma(\mathbf{z}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J)\,. \tag{25}$$

By adding (23), (24), and (25) we get

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{z}_I \cdot \mathbf{z}_J) = \gamma(\mathbf{z}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{x}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{h}_J)\,, \tag{26}$$

and hence (7) holds as equality (note that $\mathbf{f}_I = \mathbf{z}_I, \mathbf{g}_I = \mathbf{x}_I, \mathbf{h}_I = \mathbf{y}_I$). This is a contradiction; therefore case (20b) does not apply at any coordinate.

Suppose that case (20c) 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 (20b), which was proved impossible.

Finally, we have that only cases (20a) and (20d) 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. $\square$

5. Conclusions

We have studied the computational complexity of planar VCSPs. For conservative valued constraint languages on arbitrary finite domains, we have given a complete complexity classification. For valued constraint language on Boolean domains, we have given a necessary condition for tractability. The obvious open problem is to give a complexity classification of Boolean valued constraint languages, following a classification of crisp Boolean constraint languages [9, 34]. Another line of work is to consider larger domains in the non-conservative setting. As discussed in Section 1, this might be difficult given the Four Colour Theorem. Finally, planar restrictions correspond to forbidding K_5 and $K_{3,3}$ as minors. A possible avenue of research is to consider other forbidden minors in the incidence graph of the VCSP instance.

Acknowledgments

The authors were supported by a Royal Society Research Grant. The work was partly 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. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

[1] P. Fulla, S. Živný, On Planar Valued CSPs, in: Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS'16), 2016, pp. 39:1–39:14. doi:http://dx.doi.org/10.4230/LIPIcs.MFCS.2016.39.
URL http://zivny.cz/publications/fz16mfcs-preprint.pdf

[2] P. Hell, J. Nešetřil, Colouring, constraint satisfaction, and complexity, Computer Science Review 2 (3) (2008) 143–163. doi:10.1016/j.cosrev.2008.10.003.

[3] A. Krokhin, S. Živný, The complexity of valued CSPs, in: A. Krokhin, S. Živný(Eds.), Complexity and approximability of Constraint Satisfaction Problems, Vol. 7 of Dagstuhl Follow-Ups, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, pp. 233–266. doi:10.4230/DFU.Vol7.15301.233.
URL http://zivny.cz/publications/kz17survey.pdf

[4] D. A. Cohen, M. C. Cooper, P. G. Jeavons, A. A. Krokhin, The Complexity of Soft Constraint Satisfaction, Artificial Intelligence 170 (11) (2006) 983–1016. doi:10.1016/j.artint.2006.04.002.

[5] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.

[6] B. M. E. Moret, Planar NAE3SAT is in P, SIGACT News 19 (2) (1988) 51–54.

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

[8] M. R. Garey, D. S. Johnson, The rectilinear steiner tree problem in NP complete, SIAM Journal of Applied Mathematics 32 (1977) 826–834.

[9] Z. Dvořák, M. Kupec, On Planar Boolean CSP, in: Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15), Vol. 9134 of Lecture Notes in Computer Science, Springer, 2015, pp. 432–443.

[10] V. Kolmogorov, S. Živný, The complexity of conservative valued CSPs, Journal of the ACM 60 (2), article No. 10. doi:10.1145/2450142.2450146.
URL http://zivny.cz/publications/kz13jacm-preprint.pdf

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

[12] J. Thapper, S. Živný, The power of Sherali-Adams relaxations for general-valued CSPs, SIAM Journal on ComputingTo appear, arXiv:1606.02577.

[13] A. A. Bulatov, Complexity of conservative constraint satisfaction problems, ACM Transactions on Computational Logic 12 (4), article 24. doi:10.1145/1970398.1970400.

[14] F. Rossi, P. van Beek, T. Walsh (Eds.), The Handbook of Constraint Programming, Elsevier, 2006.

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

[16] D. A. Cohen, M. C. Cooper, P. Creed, P. Jeavons, S. Živný, An algebraic theory of complexity for discrete optimisation, SIAM Journal on Computing 42 (5) (2013) 915–1939. doi:10.1137/130906398.
URL http://zivny.cz/publications/cccjz13sicomp-preprint.pdf

[17] P. Fulla, S. Živný, A Galois Connection for Valued Constraint Languages of Infinite Size, ACM Transactions on Computation Theory 8 (3), article No. 9. doi:10.1145/2898438.
URL http://www.cs.ox.ac.uk/Stanislav.Zivny/homepage/publications/fz16toct-preprint.pdf

[18] T. Feder, M. 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) (1998) 57–104. doi:10.1137/S0097539794266766.

[19] C. Carbonnel, M. C. Cooper, Tractability in constraint satisfaction problems: a survey, Constraints 21 (2) (2016) 115–144. doi:10.1007/s10601-015-9198-6.

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

[21] A. Bulatov, A. Krokhin, P. Jeavons, Classifying the Complexity of Constraints using Finite Algebras, SIAM Journal on Computing 34 (3) (2005) 720–742. doi:10.1137/S0097539700376676.

[22] J. Thapper, S. Živný, Necessary Conditions on Tractability of Valued Constraint Languages, SIAM Journal on Discrete Mathematics 29 (4) (2015) 2361–2384. doi:10.1137/140990346.
URL http://zivny.cz/publications/tz15sidma-preprint.pdf

[23] M. Kozik, J. Ochremiak, Algebraic properties of valued constraint satisfaction problem, in: Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15), Vol. 9134 of Lecture Notes in Computer Science, Springer, 2015, pp. 846–858. doi:10.1007/978-3-662-47672-7_69.

[24] A. Huber, A. Krokhin, R. Powell, Skew bisubmodularity and valued CSPs, SIAM Journal on Computing 43 (3) (2014) 1064–1084. doi:10.1137/120893549.

[25] V. Kolmogorov, J. Thapper, S. Živný, The power of linear programming for general-valued CSPs, SIAM Journal on Computing 44 (1) (2015) 1–36. doi:10.1137/130945648.

[26] J. Thapper, S. Živný, The complexity of finite-valued CSPs, Journal of the ACM 63 (4), article No. 37. doi:10.1145/2974019.

[27] V. Kolmogorov, A. A. Krokhin, M. 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, pp. 1246–1258.

[28] V. Kolmogorov, M. Rolínek, R. Takhanov, Effectiveness of structural restrictions for hybrid CSPs, in: Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC'15), Vol. 9472 of Lecture Notes in Computer Science, Springer, 2015, pp. 566–577. doi:10.1007/978-3-662-48971-0_48.

[29] R. Takhanov, Hybrid (V)CSPs and algebraic reductions, Tech. rep., arXiv:1506.06540 (June 2015). URL http://arxiv.org/abs/1506.06540

[30] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, Journal of the ACM 54 (1) (2007) 1–24. doi:10.1145/1206035.1206036.

[31] V. Dalmau, P. Jonsson, The complexity of counting homomorphisms seen from the other side, Theoretical Computer Science 329 (1-3) (2004) 315–323. doi:10.1016/j.tcs.2004.08.008.

[32] T. Färnqvist, P. Jonsson, Bounded tree-width and csp-related problems, in: Proceedings of the 18th International Symposium on Algorithms and Computation (ISAAC), Vol. 4835 of Lecture Notes in Computer Science, Springer, 2007, pp. 632–643. doi:10.1007/978-3-540-77120-3_55.

[33] K. Meeks, The challenges of unbounded treewidth in parameterised subgraph counting problems, Discrete Applied Mathematics 198 (2016) 170–194. doi:10.1016/j.dam.2015.06.019.

[34] A. Kazda, V. Kolmogorov, M. Rolínek, Even delta-matroids and the complexity of planar Boolean CSPs, in: Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17), 2017, pp. 307–326.

[35] J. Cai, P. Lu, M. 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), IEEE Computer Society, 2010, pp. 427–436.

[36] H. Guo, T. Williams, The complexity of planar Boolean #CSP with complex weights, in: Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13), Vol. 7965 of Lecture Notes in Computer Science, Springer, 2013, pp. 516–527.

[37] S. Khanna, R. Motwani, Towards a syntactic characterization of PTAS, in: Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96), 1996, pp. 329–337. doi:10.1145/237814.237979.

[38] N. Creignou, S. Khanna, M. Sudan, Complexity Classification of Boolean Constraint Satisfaction Problems, Vol. 7 of SIAM Monographs on Discrete Mathematics and Applications, SIAM, 2001.

[39] J. E. Hopcroft, R. E. Tarjan, Efficient planarity testing, Journal of the ACM 21 (4) (1974) 549–568. doi:10.1145/321850.321852.

[40] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Vol. 24 of Algorithms and Combinatorics, Springer, 2003.

[41] P. Jeavons, D. Cohen, M. C. Cooper, Constraints, Consistency and Closure, Artificial Intelligence 101 (1–2) (1998) 251–265. doi:10.1016/S0004-3702(98)00022-8.

[42] W. Mulzer, G. Rote, Minimum-weight Triangulation is NP-hard, Journal of the ACM 55 (2) (1998) 11:1–11:29. doi:10.1145/1346330.1346336.

[43] D. A. Cohen, M. C. Cooper, P. G. Jeavons, Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms, Theoretical Computer Science 401 (1-3) (2008) 36–51. doi:10.1016/j.tcs.2008.03.015.

[44] R. 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), 2010, pp. 657–668. doi:10.4230/LIPIcs.STACS.2010.2493.