Valued Constraint Satisfaction Problems over Infinite Domains

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt

dem Bereich Mathematik und Naturwissenschaften
der Technischen Universität Dresden

von

M.Sc. Caterina Viola

Eingereicht am 18. Februar 2020
Verteidigt am 17. Juni 2020

Gutachter:
Prof. Dr.rer.nat. Manuel Bodirsky
Prof. Ph.D. Andrei Krokhin

Die Dissertation wurde in der Zeit von März 2016 bis
Februar 2020 am Institut für Algebra angefertigt.

"Would you tell me, please, which way I ought to go from here?" "That depends a good deal on where you want to get to," said the Cat. "I don't much care where –" said Alice. "Then it doesn't matter which way you go," said the Cat. "– so long as I get somewhere," Alice added as an explanation. "Oh, you're sure to do that," said the Cat, "if you only walk long enough."

— Lewis Carroll, *Alice in Wonderland*

Acknowledgements

First and foremost, I want to express my deepest and most sincere gratitude to my supervisor, Manuel Bodirsky. He offered me his knowledge, guided my research, and rescued in many ways the scientist I eventually became. I am glad and honoured to have been one of his doctoral students.

I am very grateful to my collaborator and co-author Marcello Mamino, who was also a mentor during my first years in Dresden. His enthusiasm and eclectic mathematical knowledge have been crucial to the success of my doctoral studies.

I am also very grateful to Martin Schneider who kindly helped me to look at my doctoral project from a different mathematical perspective and agreed to collaborate with me.

Thanks to Johan Thapper for the hospitality and the several technical insights he gave me during my visit at the Laboratoire d'Informatique Gaspard-Monge (Université Paris-Est Marne-la-Vallée), in June 2018.

I thank Antoine and Jakub for our interesting discussions, for being very inspiring and kind colleagues, and, especially, for being my friends.

I am grateful to Johannes and to all the other people from the Institut für Algebra who provided me with a pleasant working environment.

Thanks to the DFG research training group QuantLA that financed my doctoral studies and gave me the unique chance to take part in a group of junior and senior researchers from different scientific communities. The atmosphere in QuantLA was the one that is possible to find in a group of friends and having shared the doctoral experience with the other junior members of the group has been an invaluable opportunity.

The last year of my doctoral studies has been funded from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 681988, "CSP-Infinity").

The presence of other women in my working environment and their solidarity have been very important to me and I am especially grateful to Shima's friendship and to the non-technical support I received from my second supervisor, Karin, who constantly checked on me and cared about how I felt during my studies.

Thanks to my mum and my dad for having taught me the importance of being an educated and independent woman, and to my sisters, Grazia and Gabriella, for their unconditional love, for inspiring me to be the best version of myself, and for making me learn how enriching diversity is.

I thank my soul-twin, Vanda, for always supporting me and for having somehow been present in my life every single day of the last 4 years, despite the long distance.

Last but not least, thanks to Claudio, my significant other, who encouraged me to follow my dreams and was with me in Dresden most of the time.

Contents

Introduction

In this chapter we informally introduce valued constraint satisfaction problems and the constraint satisfaction framework, we overview the state of the art, motivate our research, and, finally, we present examples of relevant computational problems that can be modelled as *valued constraint satisfaction problems*. All the preliminary concepts are formally defined in Chapter 1. At the end of this introduction we also describe the organisation of the thesis.

The content of this thesis mainly finds its background in two scientific disciplines: computational complexity theory and universal algebra. Computational complexity theory focuses on quantifying the amount of resources (e.g., running time or memory) required by the "best" (i.e., the fastest or the least memory consumptive) algorithm that solves a given task. Universal algebra studies properties of algebraic structures. The object of the thesis is the study of the computational complexity of certain combinatorial optimisation problems called *valued constraint satisfaction problems*, or *VCSPs* for short. The requirements and optimisation criteria of these problems are expressed by sums of *(valued) constraints* (also called *cost functions*). More precisely, the input of a VCSP consists of a finite set of variables, a finite set of cost functions that depend on these variables, and a cost u; the task is to find values for the variables such that the sum of the cost functions is at most u.[1] By restricting the set of possible cost functions in the input, a great variety of computational optimisation problems can be modelled as VCSPs. We study how the computational complexity of a VCSP depends on the universal algebraic properties of the (valued) structure determined by the set of allowed cost functions, in the case in which the range of values for the variables (the *domain*) is the set of rational numbers, $\mathbb{Q}$.

[1] We adopt the so-called *threshold* formulation of VCSPs. In the existing literature, VCSPs have also been studied using the *minimisation* formulation, in which no cost u is given as part of the instance, and the task is to find values for the variables that minimise the sum of the cost functions.

Constraint Satisfaction and Valued Constraint Satisfaction

Constraints occur in most fields of human endeavour, and they allow us to naturally and precisely formalise the interdependencies of physical-world objects as well as of their mathematical abstractions. In *Constraint Satisfaction* these dependencies are specified by logical relations among several variables. However, in many situations some constraints might be violated at some cost, or, even if all the constraints need be satisfied, there might be solutions which are preferable to others. These scenarios are captured by the *Valued Constraint Satisfaction* framework, in which dependencies and preferences are specified by cost functions of several variables.

Constraint Satisfaction Problems

An instance of a *Constraint Satisfaction Problem*, or *CSP* for short, states which relations must hold among some given variables. More formally, an instance of the CSP consists of a finite set of variables defined on a given domain and a set of relations on subsets of these variables; the computational task is to find an assignment of values to the variables that satisfies all the constraints. One of the strengths of the CSP framework is that it provides a unifying framework for various problems that have been independently studied before. By specifying properties of the domain and of the constraints (e.g., the domain can be finite or infinite, discrete or continuous), one can obtain different classes of problems. In fact, many problems arising from industry, business, manufacturing, and science can be formulated as CSP instances, and this is often the method of choice. Identifying restricted classes of CSPs that are solvable in polynomial time is of theoretical as well as practical interest in the design of constraint programming languages and efficient constraint solvers (for examples of practical applications of results on polynomial-time solvable classes of CSPs see, e.g., [36, 86]).

If the domain of values is a finite set, then it is well known that the CSP is NP-complete, in general. Therefore, the research on the computational complexity of finite-domain CSPs has typically focused on specific restrictions of the problem. In a *language-restricted problem*, the goal is to understand how the computational complexity depends on the structure (in this setting, the allowed constraints are fixed, but they can be combined in any way). In a *structurally-restricted problem* the goal is to understand how the computational complexity depends on the constraints hypergraph (in this setting the interaction of the constraints is restricted, but any sort of constraint can be used). Finally, in a *hybrid-restricted problem*, the goal is to understand how the computational complexity depends on some combination of the previous two restrictions of CSPs.

Language-restricted problems have been the focus of research on finite-domain CSPs for many years. Early results in this direction motivated Feder and Vardi [39] to conjecture that the CSP for structures over a finite domain is either polynomial-time solvable or NP-complete, namely they conjectured that the class of finite-domain CSPs satisfies a *complexity dichotomy*. By Ladner's Theorem [78], there are NP-intermediate problems[2] (unless P=NP), and the Feder-Vardi dichotomy conjecture would imply the impossibility of encoding them as finite-domain CSPs. The Feder-Vardi conjecture was the motivation of an intensive line of research over the last two decades, and, recently, it has been confirmed in two independent proofs by Bulatov [24] and Zhuk [101]. The success of the research program on the Feder-Vardi conjecture is based on the universal algebraic approach introduced by Bulatov, Krokhin, and Jeavons [25], which allows characterising the computational complexity of the CSP for a given structure by the algebraic properties of the structure itself.

If we allow the underlying domain of CSPs to be a set of arbitrary size, we obtain a more expressive framework capturing, besides all finite-domain CSPs, many computational problems that cannot be formulated in the finite-domain setting. Examples include the feasibility of linear programs over $\mathbb{Z}$ (the set of integer numbers), $\mathbb{Q}$ (the set of rational numbers), and $\mathbb{R}$ (the set of the real numbers) [91], the model-checking problem for Kozen's modal μ-calculus [72], and the solution of linear Diophantine equation systems [31, 62]. However, the increase in expressive power gained from considering CSPs over arbitrary domains is counterbalanced by some losses. Firstly, infinite-domain CSPs are not necessarily in NP. In fact, every computational decision problem over a finite alphabet is equivalent under polynomial-time Turing reduction to an infinite-domain CSP [3]. Secondly, even if many results of the universal algebraic theory for finite-domain CSPs can be extended to the arbitrary domain setting, there are many algebraic properties of finite-domain structures that do not hold for infinite-domain ones. Remarkably, the computational complexity of CSPs for structures over arbitrary domains is not fully characterised by the algebraic properties of the structure only.

In the last ten years, the research on infinite-domain CSPs has followed two specific directions. One of these directions is the identification of classes of structures that admit a complexity dichotomy (see [4, 6, 7, 17, 19, 71]). For such classes, it was shown that the delineation between polynomial-time solvability and NP-hardness can be described using algebraic and topological methods. The other direction is to study CSPs over some of the most basic

[2] An NP-intermediate problem is a problem that belongs to the class NP, and that is neither in P, nor NP-complete.

and well-known infinite domains, such as the numeric domains $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$ (the set of complex numbers); and to focus on constraint relations that are first-order definable from the usual order, addition, and multiplication on these domains (see [5, 10, 13, 61]).

Valued Constraint Satisfaction Problems

VCSPs may be considered a variation of (classical) CSPs that capture optimisation problems and that provide a common framework for well-know computational problems as CSPs, MAX CSPs (see [30, 32, 67]), *finite-valued* CSPs (see [97]), MAX ONES problems (see [37, 67]), and MINIMUM COST HOMOMORPHISM problems (see [51, 53, 95]).

While in the CSP setting (or classical constraint setting), the constraints are parametrised by relations on the variables expressing the feasibility of variable assignments, in the VCSP setting (or valued setting), the constraints are parametrised by cost functions applied to the variables expressing not only the feasibility but also the preferability of variable assignments. The non-feasibility of variable assignments is modelled by allowing the cost functions in the input to evaluate to $+\infty$. In fact, the class of (classical) CSPs is a subclass of the class of all VCSPs: CSP instances correspond to VCSP instances in which the cost functions take values in $\{0, +\infty\}$. Therefore, every instance of the CSP can be modelled as an instance of the VCSP. On the other hand, every VCSP instance induces a CSP instance: the problem of deciding the existence of an assignment with finite cost (the feasibility problem).

If we allow the cost functions to take only finite values, then we obtain instances of a *finite-valued* CSP, which is a mere optimisation problem. If we allow the cost functions to take only values in $\{0, 1\}$, we obtain instances of MAX-CSPs where the goal is to find an assignment maximising the number of satisfied constraints. Finally, if we only allow cost functions that are either $\{0, +\infty\}$-valued or finite-valued, we obtain problems like MAX-ONES problems, and MINIMUM COST HOMOMORPHISM problems, in which some feasibility constraints must be satisfied and some additional function of the variable assignments must be optimised.

Finite-Domain Valued Constraint Satisfaction Problems

Similarly to the classical constraint setting, if the domain of the cost functions is a finite set, then the VCSP is NP-complete[3], and, again similarly to the classical constraint setting, the research on the computational

[3] The problem is NP-complete in the threshold formulation. It is NP-hard in the minimisation formulation.

complexity of VCSPs has focused on the language-restricted problem, the structurally-restricted problem, and hybrid-restricted problems. In the case of the language-restricted problem, the algebraic approach was extended from finite-domain CSPs to finite-domain VCSPs (cf. [33, 34]) and it was shown that valued structures can be classified by studying their algebraic properties (cf. [33, 43]). The algebraic approach for finite-domain VCSPs led to several complexity classifications for special classes of valued structures: valued structures over a a 2-element (Boolean) domain [34], valued structures containing all $\{0,1\}$-valued cost functions [70], and finite-valued structures, in which the feasibility aspect is trivial and the cost functions only assume finite values, [97]. Moreover, a classification of valued structures with respect to their exact solvability by a specific algorithm was established for the *basic linear programming relaxation* [69], and the *Sherali-Adams relaxation* [98].

Similarly to the finite-domain classical constraint setting, the first complexity classifications for specific finite-domain valued structures had the form of a dichotomy between polynomial-time solvability and NP-hardness. This motivated Kozik and Ochremiak [73] to make an analogue of the finite-domains CSP dichotomy conjecture for finite-domain VCSPs, and proved the necessity of the algebraic condition that they conjectured to characterise polynomial-time solvable valued structures. It was proved later, by Kolmogorov, Krokhin, and Rolinek [68], that this algebraic condition is sufficient for polynomial-time solvability, assuming that the underlying feasibility problem is solvable in polynomial time. Since this further assumption follows from the Feder-Vardi dichotomy conjecture for classical CSPs, the recent proofs of the Feder-Vardi conjecture immediately settled the complexity dichotomy of finite-domain VCSPs.

Examples of finite-domain VCSPs

We present two examples of problems that can be modelled as VCSPs over a finite domain.

Example 1. (MAX CUT) Given a graph $G = (V, E)$, consider the problem of finding a subset S of vertices in V that maximises (or bound from below by a given threshold) the number of edges between S and its complement $V \setminus S$. This problem can be modelled as a VCSP instance over the domain $D = \{0, 1\}$, with objective function $\phi(v_1, \dots, v_n) = \sum_{(v_i,v_j)\in E} \mathrm{XOR}(v_i, v_j)$, where the variables $\{v_1, \dots, v_n\}$ correspond to the vertices in V, and where the cost function $\mathrm{XOR}\colon D^2 \to \mathbb{Q}$ is defined by

$$\mathrm{XOR}(x, y) := \begin{cases} 0 & \text{if } x \neq y \\ 1 & \text{if } x = y. \end{cases}$$

Assigning a variable to 1 is interpreted as the corresponding vertex belonging to the subset S. The MAX CUT problem is NP-hard[4] (see [45]). $\triangle$

Example 2. (MIN (s,t)-CUT) Let $G = (V, E)$ be a directed graph, let $w\colon E \to \mathbb{Q}_{\geq 0}$ be a weighting function defined on the edges of G, and let $s, t \in V$ be two distinct vertices. An (s,t)-*cut* of G is a subset C of vertices in V such that $s \in C$, and $t \notin C$. The weight of an (s,t)-cut C is defined as $\sum_{(u,v)\in E : u\in C, v\notin C} w(u,v)$. The goal of the MIN (s,t)-CUT problem is to find an (s,t)-cut C in G with minimum weight (or with weight bounded from above by a given threshold). Consider the valued structure Γ_{cut} with domain $D = \{0,1\}$, and cost functions $\eta_0, \eta_1 \colon D \to \mathbb{Q} \cup \{+\infty\}$, and $f^w_{cut}\colon D^2 \to \mathbb{Q} \cup \{+\infty\}$, for $w \in \mathbb{Q}_{\geq 0}$. Where, for $c \in \{0,1\}$, the function η_c is defined by

$$\eta_c(x) := \begin{cases} 0 & \text{if } x = c \\ +\infty & \text{otherwise,} \end{cases}$$

and f^w_{cut} is defined by

$$f^w_{cut}(x,y) = \begin{cases} w & \text{if } x = 0, \text{ and } y = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The MIN (s,t)-CUT problem can be modelled as an instance of the VCSP for Γ_{cut} with objective function

$$\phi(v_1, \ldots, v_n) = \eta_0(s) + \eta_1(t) + \sum_{(u,v)\in E} f^{w(u,v)}_{cut}(u,v),$$

where the variables $\{v_1, \ldots, v_n\}$ correspond to the vertices in V, and where assigning a variable to 0 is interpreted as the corresponding vertex belonging to the (s,t)-cut C. The unary cost functions η_0, η_1 ensure that s and t belong to C and $V \setminus C$, respectively. Any instance of the VCSP for Γ_{cut} can be polynomial-time many-one reduced to an instance of the MIN (s,t)-CUT problem, which can be solved in polynomial time (see [47, 75]). $\triangle$

As we already mentioned, in the finite-domain VCSP setting, there is another research line that focuses on the structurally-restricted problem, in which the set of admissible instances is restricted by specifying the allowed interaction of constraints, rather than by specifying a valued structure (see [29, 44, 48, 49, 81]). The computational complexity of the VCSP for a finite-domain valued structure may be reduced by considering hybrid restrictions of the VCSP, which combine the language restriction with some

[4]The problem is NP-hard in the minimisation formulation. It is NP-complete in the threshold formulation.

structural restrictions. Finally, we mention that there is a further research line on finite-domain VCSPs that takes into account extensions of VCSPs with various kinds of global constraints (see [26, 42, 82]), i.e., constraints that are applied to a non-fixed number of variables. In contrast to hybrid restrictions, extending a language-restricted finite-domain VCSP with some global constraint may increase the computational complexity.

Infinite-Domain Valued Constraint Satisfaction Problems

Analogously to the classical CSP setting, several combinatorial optimisation problems cannot be formulated as VCSPs over a finite domain, but they can be formulated as VCSPs over an infinite domain, e.g., $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{Z}$.

Example 3. (Linear Least Square Regression) Consider the problem of finding the linear function that best approaches how an outcome y depends on one or more explanatory parameters $x_0, x_1, \ldots, x_n$, by minimising the sum of the least square deviation from some (statistical or training) data $(y_j, x_{j0}, x_{j1}, \ldots, x_{jn})$, for $1 \leq j \leq m$ and some $m \in \mathbb{N}$, or by making it at most equal to a fixed error tolerance.

This problem can be modelled as an instance of a VCSP over the rationals, where the variables, $v_0, v_1, \ldots, v_n$, represent the coefficients of a linear relation constraining $x_0, x_1, \ldots, x_n$ and y. The cost functions, $f_j \colon \mathbb{Q}^{n+1} \to \mathbb{Q}$, arise from the objective function,

$$\phi(v_0, v_1, \ldots, v_n) = \sum_{j=1}^{m} f_j(v_0, v_1, \ldots, v_n),$$

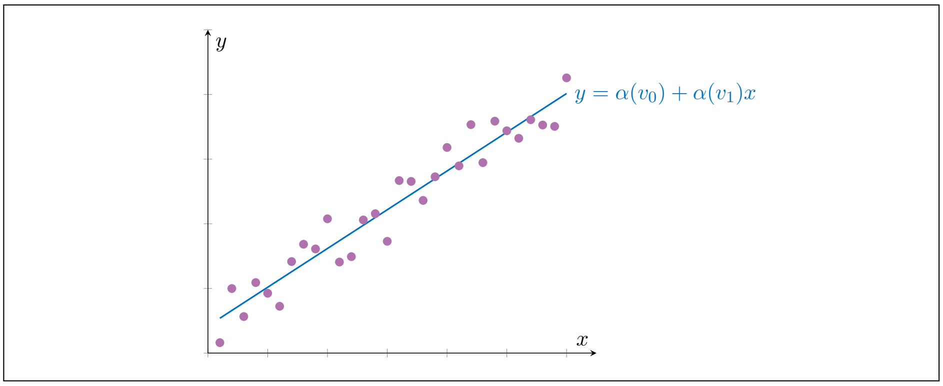
and are defined by

$$f_j(v_0, v_1, \ldots, v_n) := (v_0 + \sum_{i=1}^{n} v_i x_{ji} - y_j)^2.$$

These functions represent the least square deviation from the given data $(y_j, x_{j0}, x_{j1}, \ldots, x_{jn})$, for $1 \leq j \leq m$; and the threshold represents the error tolerance. Each assignment $\alpha \colon \{v_0, v_1, \ldots, v_n\} \to \mathbb{Q}$ corresponds to a linear function $y := \alpha(v_0) + \sum_{i=1}^{n} \alpha(v_i) x_i$.

The Linear Least Square Regression finds applications in areas of science and technology in which the goal is to predict, forecast, or reduce errors, e.g., economics, machine learning, biological sciences, and social sciences. Linear Least Square Regression can be solved in polynomial time (see, e.g., [21]). $\triangle$

Figure 1: Linear Least Square Regression



A set of (2-dimensional) points denoting the given data (x_i, y_i) and their corresponding Linear Least Square Regression $y = \alpha(v_0) + \alpha(v_1)x$.

Other examples include the Linear Programming problem, where the task is to optimise a linear function subject to linear inequalities (see Example 4), and the minimisation problem for sums of piecewise linear convex cost functions (see, e.g., [21]). Both of these problems can be solved in polynomial time, e.g. by the ellipsoid method (see, e.g., [50]).

Despite the great interest in such concrete VCSPs over the rational numbers, VCSPs over infinite domains have not yet been studied systematically. In order to obtain general results, we need to restrict the class of valued structures that we investigate, because without any restriction it is already hopeless to classify the complexity of the subclass of all infinite-domain CSPs.

One class that captures a variety of optimisation problems of theoretical and practical interest is that which arises from the restriction of the language to *piecewise linear (PL)* cost functions. A $\mathbb{Q}$-valued partial function is PL if it is first-order definable over $\mathbb{Q}$ with the usual order relation, the usual sum, and the identity in $\mathbb{Q}$.[5] The name is due to the fact that the representation of a piecewise linear function on a Cartesian coordinate system is the (disjoint) union of linear subspaces (pieces). An important example of VCSP for PL valued structures is Linear Programming.

Example 4. (Linear Programming) Linear Programming, or *LP* for short, is an optimisation problem with a linear objective function and a set of linear constraints imposed upon a given set of underlying variables. A

[5] In the PL setting the domains $\mathbb{Q}$ and $\mathbb{R}$ are interchangeable.

linear program has the form

$$\begin{aligned} &\text{minimise} && \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_j^i x_j \leq b^i, \text{ for } i \in \{1,\dots,m\}. \end{aligned}$$

This problem has n variables x_j (ranging over $\mathbb{Q}$ or $\mathbb{R}$) and m linear inequalities as constraints. The coefficients c_j, a_j^i, and b^i are rational numbers for all $j \in \{1,\dots,n\}$, and all $i \in \{1,\dots,m\}$. LP can be solved in polynomial time (see, e.g., [63, 65, 100]). $\triangle$

In general, VCSPs for PL valued structures are NP-complete (see Proposition 1.2.16). Therefore it is interesting to study how a further restriction to special classes of PL valued structures influences the computational complexity. However, studying the computational complexity of all VCSPs for all PL valued structures is a very ambitious goal. Already, the underlying CSP for PL (relational) structures, better known as *semilinear CSPs* has not been systematically studied, and only several partial results are known (see [5, 9, 10, 11, 61]). Results for semilinear CSPs suggest that the algebraic approach, so successful in the finite-domain case, is not sufficient to fully characterise the computational complexity of VCSPs, at least in the general case in which we allow the cost functions to assume $+\infty$ as a possible value. However, as in the classical constraint setting, studying the algebraic properties of PL valued structures might give interesting, though partial, results and led to prove the polynomial-time solvability for special PL valued structures.

Another important class of valued structures over the rationals is the class of *piecewise linear homogeneous (PLH)* valued structures. A valued structure is PLH if the cost functions thereof are PLH, i.e., first-order definable over $\mathbb{Q}$ using the usual total order, the identity in $\mathbb{Q}$, and the scalar multiplication by rational numbers. PLH functions arise in several scientific fields, e.g., some of the activation functions used in neural networks (as the ReLU and the Leaky ReLU activation functions) [80] are examples of PLH functions, and the functions arising in the piecewise linear subband coding schemes [93] used in image analysis and computer vision can be written as sums of PLH functions. Clearly, PLH valued structures form a subclass of the class of PL valued structures. Nevertheless the PLH setting is very expressive (e.g., every finite domain VCSP is equivalent to the VCSP for a suitable PLH valued structure) and captures many computational problems, as the examples below show. Finally, PLH valued structures have many interesting mathematical properties, making them a natural and reasonable intermediate step between finite-domain valued structures and valued structures over infinite domains.

Example 5. (LEAST CORRELATION CLUSTERING WITH PARTIAL INFORMATION) We consider the following problem: we are given an undirected graph on n vertices such that the set of edges E is partitioned in two classes, E_- and E_+. An edge $(x, y) \in E$ is either in E_+ or in E_- depending on whether x and y have been deemed to be similar or different. The goal is to (decide whether it is possible to) produce a partition of the vertices, namely a clustering, that agrees with the edge partition on at least l edges, where l is a given (rational) number between 0 and $|E|$. That is, we want a clustering that bounds the number of disagreements, i.e., the number of edges from E_+ between clusters plus the number of edges from E_- inside clusters. This problem can be seen as an instance of a VCSP with variables $x_1, \ldots, x_n$, objective function

$$\phi(x_1, \ldots, x_n) = \sum_{(x_i,x_j) \in E_+} f_1(x_i, x_j) + \sum_{(x_i,x_j) \in E_-} f_2(x_i, x_j),$$

and with $u := |E| - l$. The cost functions $f_1, f_2 \colon \mathbb{Q}^2 \to \mathbb{Q}$ are defined by

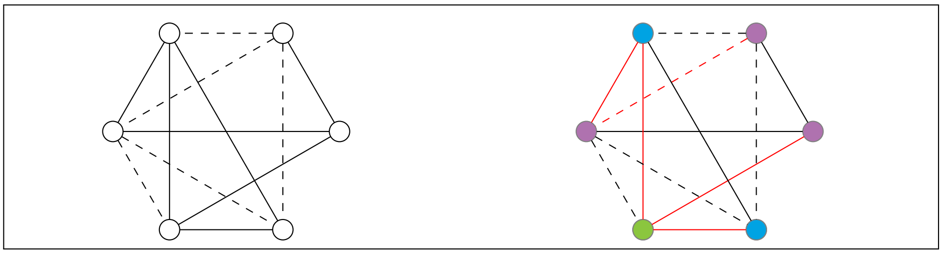
$$f_1(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 & \text{otherwise,} \end{cases} \quad \text{and} \quad f_2(x_i, x_j) = \begin{cases} 1 & \text{if } x_i = x_j \\ 0 & \text{otherwise.} \end{cases}$$

Observe that this problem cannot be modelled as a VCSP over a finite domain, as we do not want to bound the possible number of clusters and we want to allow graphs with any finite number of vertices as input.

The LEAST CORRELATION CLUSTERING WITH PARTIAL INFORMATION problem that we defined above can model the well-known MIN-CORRELATION CLUSTERING PROBLEM (see [1]). Since MIN-CORRELATION CLUSTERING PROBLEM is NP-complete [6] (see [45, 64]), so is LEAST CORRELATION CLUSTERING WITH PARTIAL INFORMATION. △

[6] The problem is NP-complete in the threshold formulation. It is NP-hard in the minimisation formulation.

Figure 2: Least Correlation Clustering with Partial Information



Left: an instance of the Least Correlation Clustering with Partial Information problem. The continuous and dashed segments denote edges in E_+ and in E_-, respectively. Right: an assignment with cost 5; different colours of the vertices denote different assignment and red edges denote occurrences of the cost functions evaluating to 1.

Example 6. (Minimum Feedback Arc Set) Let $G = (V, E)$ be a directed graph, and assume we are required to remove some of the edges in E in such a way to obtain an acyclic graph $G' := (V, E')$ while minimising the number of removed edges, $|E \setminus E'|$, or bounding it by a given threshold u. This problem can be seen as an instance of a VCSP with variables $x_1, \ldots, x_n$ such that each variable corresponds to a vertex in V, with objective function

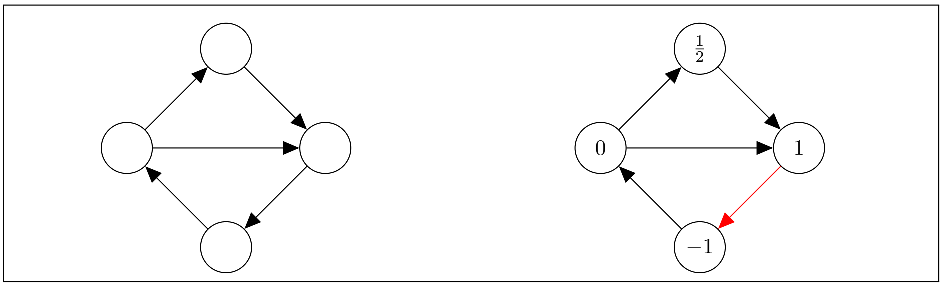
$$\phi(x_1, \ldots, x_n) = \sum_{(x_i, x_j) \in E} f(x_i, x_j),$$

and with threshold u, where the cost function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ is defined by

$$f(x_i, x_j) = \begin{cases} 0 & \text{if } x_i < x_j \\ 1 & \text{if } x_i \geq x_j. \end{cases}$$

The Minimum Feedback Arc Set problem is known to be NP-complete [6] (see [45, 64]). △

Figure 3: Minimum Feedback Arc Set



Left: an instance of the Minimum Feedback Arc Set problem. Right: an assignment with cost 1; the red edge denotes the occurrence of the cost function evaluating to 1.

Example 7. (Min (s,t)-Cut with Hard Order Constraints) Let $G = (V, E)$ be a directed graph, let $w\colon E \to \mathbb{Q}_{\geq 0}$ be a weighting function defined on the edges of G, and let $s, t \in V$ be two distinct vertices. The Min (s,t)-Cut with Hard Order Constraints problem consist of deciding whether G is acyclic and, in this case, finding an (s,t)-cut C of G with minimum weight (or with weight bounded from above by a threshold), such that every edge transversal to C and its complement is oriented from C to $V \setminus C$.

Consider the valued structure $\Gamma_{<cut}$ with domain $\mathbb{Q}$ and cost functions $\eta_-, \eta_+\colon D \to \mathbb{Q} \cup \{+\infty\}$, and $f^w_{<cut}\colon \mathbb{Q}^2 \to \mathbb{Q} \cup \{+\infty\}$, for $w \in \mathbb{Q}_{\geq 0}$, defined by

$$\eta_-(x) := \begin{cases} 0 & \text{if } x < 0 \\ +\infty & \text{otherwise,} \end{cases} \qquad \eta_+(x) := \begin{cases} 0 & \text{if } x > 0 \\ +\infty & \text{otherwise,} \end{cases}$$

and

$$f^w_{<cut}(x,y) = \begin{cases} 0 & \text{if } x < y < 0, \text{ or } 0 < x < y \\ w & \text{if } x < 0 < y \\ +\infty & \text{otherwise.} \end{cases}$$

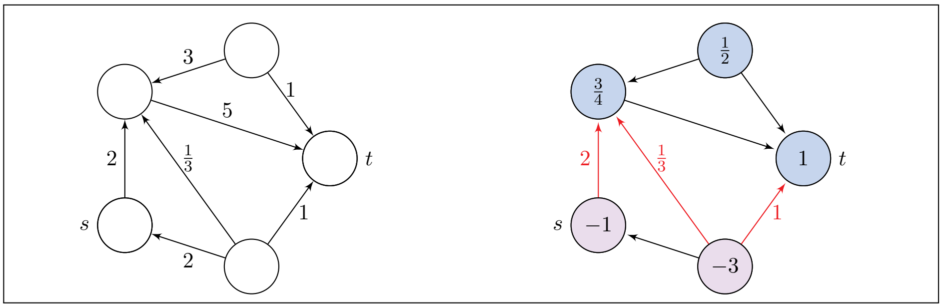
An instance of the Min (s,t)-Cut with Hard Order Constraints can be modelled as an instance of the VCSP for $\Gamma_{<cut}$ with objective function

$$\phi(v_1, \ldots, v_n) = \eta_-(s) + \eta_+(t) + \sum_{(u,v) \in E} f^{w(u,v)}_{<cut}(u,v),$$

where the variables $\{v_1, \ldots, v_n\}$ correspond to the vertices in V. Assigning a variables to a negative number is interpreted as the corresponding vertex belonging to the (s,t)-cut C.

Every instance of the VCSP for $\Gamma_{<cut}$ and therefore, in particular the Min (s,t)-Cut with Hard Order Constraints problem, can be solved in polynomial time, since the cost functions are PLH and *submodular* (see Chapter 6). $\triangle$

Figure 4: MIN (s,t)-CUT WITH HARD ORDER CONSTRAINTS



Left: an instance of the MIN (s,t)-CUT WITH HARD ORDER CONSTRAINTS problem. Right: an assignment with cost $\frac{10}{3}$; red edges denote occurrences of the cost functions assuming non-zero values.

In this thesis, we initiate the systematic investigation of the computational complexity of VCSPs for PL and PLH valued structures.

Organisation of the thesis

In Chapter 1, we give the formal definition of the VCSP over arbitrary domains together with some preliminary notions and results that we will need throughout the thesis. In Chapter 2, we show a polynomial-time algorithm solving the VCSP for the valued structures containing *all* the convex cost functions, we show a class of valued structure with domain $\mathbb{Q}$ for which the VCSP is polynomial-time equivalent to the associated CSP, and we present a family of PL valued structures whose VCSP is NP-hard. In Chapter 3, we introduce the notion of *sampling algorithm* for valued structures and provide an efficient sampling algorithm for PLH valued we present a sampling technique to solve VCSPs for PLH cost functions in polynomial time. The technique consists of a polynomial-time many-one reduction from the VCSP for a finite set of PLH cost functions to the VCSP for a *sample*, i.e., the same set of cost functions interpreted over a suitable finite domain. In Chapter 4, we present a sufficient condition under which the VCSP for infinite-domain valued structures that admit an efficient sampling algorithm can be solved in polynomial time using a linear programming relaxation. In Chapter 5, we apply the results of Chapters 3 and 4 to classify the computational complexity of the VCSP for special classes of PLH valued structures. Chapter 6 is devoted to submodular PLH valued structures. We provide two different approaches to solve VCSPs for these valued structures, and we show that submodularity defines a *maximally tractable class* of PLH valued structures. Informally, this means that adding any cost function that is not submodular leads to an NP-hard VCSP. In Chapter 7, we discuss how concepts and results from the algebraic theory for finite-domain VCSPs can be transferred or extended to the infinite-domain setting. In Chapter 8, we present a polynomial-time algorithm solving the restriction of the VCSP for *all* PL cost functions to instances with a fixed number of variables. Finally, Chapter 9 contains conclusions and some open problems.

Most of the results in this thesis have appeared in [14, 15, 16]. The content of Chapter 7 is the unpublished result of a collaboration with Friedrich Martin Schneider (Technische Universität Dresden).

Chapter 1

Preliminaries

In this chapter, we present the necessary background on valued constraint satisfaction problems and on the cost functions over the rational numbers that are studied in the thesis. In Section 1.1, we define valued structures and valued constraint satisfaction problems. In Section 1.2, we give some logic preliminaries and introduce the classes of piecewise linear cost functions and of piecewise linear homogeneous cost functions. In Section 1.3, we show the connection between valued constraint satisfaction problems and (classical) constraint satisfaction problems. In Section 1.4, we present the tools from universal algebra that we use to deal with valued constraint satisfaction problems. In Section 1.5, we define submodular cost functions. In Section 1.6, we define convex cost functions. Section 1.7 is about the Linear Programming problem. Finally, in Section 1.8, we discuss the generalisation of valued constraint satisfaction problems to the case in which the cost functions take values in arbitrary totally ordered commutative rings.

Throughout the thesis we adopt the following notation: we denote by x_i the i-th component of a tuple x. We denote by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$, respectively, the set of natural numbers, integers, rational numbers, and real numbers. We also use $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ to denote the set of nonnegative rational numbers and the set of nonnegative real numbers.

1.1 Valued Constraint Satisfaction Problems

We introduce the notions of valued structure and valued constraint language. The notion of valued structure has been used in [96]; however, the existing literature has more often referred to valued constraint languages. Throughout the thesis, we refer to valued structures. The choice of referring to valued structures rather than valued constraint languages allows us to use the established terminology from model theory (e.g., substructure, reduct, extension, expansion) and avoid to generate confusion (e.g., when considering subsets or supersets of the domain or the signature).

Definition 1.1.1. A *valued structure* Γ consists of

- a signature τ consisting of function symbols f, each equipped with an arity, $\mathrm{ar}(f)$,
- a (possibly infinite) set $D = \mathrm{dom}(\Gamma)$ (the *domain*),
- for each $f \in \tau$ a *cost function*, i.e., a function $f^\Gamma \colon D^{\mathrm{ar}(f)} \to \mathbb{Q} \cup \{+\infty\}$.

Here, $+\infty$ is an extra element with the expected properties that for all $c \in \mathbb{Q} \cup \{+\infty\}$

$$(+\infty) + c = c + (+\infty) = +\infty$$
$$\text{and } c < +\infty \text{ iff } c \in \mathbb{Q}.$$

Let D be a (possibly infinite) set.

Definition 1.1.2. A *valued constraint language (over D)* (or simply *valued language*) is the set of cost functions of a valued structure with domain D.

Throughout the thesis, we identify a valued language with the underlying valued structure. By the terminology 'valued structures' we refer to valued structures whose cost functions take values in $\mathbb{Q} \cup \{+\infty\}$ (or in $\mathbb{R} \cup \{+\infty\}$). However, it is possible to define valued structures whose cost functions take values in another totally ordered commutative ring with unit R (see Section 1.8). In this case, we use the terminology *R-valued structure*.[1]

Let Γ be a valued structure with signature τ. The *valued constraint satisfaction problem for* Γ, denoted by $\mathrm{VCSP}(\Gamma)$, is the following computational problem.

Definition 1.1.3. An *instance* I of $\mathrm{VCSP}(\Gamma)$ consists of

- a finite set of variables V_I,
- an expression ϕ_I of the form
$$\sum_{i=1}^{m} f_i(x_1^i, \ldots, x_{\mathrm{ar}(f_i)}^i)$$
where $f_1, \ldots, f_m \in \tau$ and all the x_j^i are variables from V_I, and
- a value $u_I \in \mathbb{Q}$.

[1]We mention that, in the finite-domain setting, valued structures with cost functions taking values in more general valuation structures, called linearly-ordered integral Abelian pomonoid, have also been taken into account (see [55]).

The task is to decide whether there exists an assignment $\alpha\colon V_I \to \mathrm{dom}(\Gamma)$ whose *cost*, defined as

$$\sum_{i=1}^{m} f_i^{\Gamma}(\alpha(x_1^i), \dots, \alpha(x_{\mathrm{ar}(f_i)}^i))$$

is finite, and if so, whether there is one whose cost is at most u_I.

The function $\phi_I^{\Gamma}\colon \mathrm{dom}(\Gamma) \to \mathbb{Q} \cup \{+\infty\}$ described by the expression ϕ_I is also called the *objective function*. The problem of deciding whether there exists an assignment with a finite cost is called the *feasibility problem*, which can also be modelled as a (classical) constraint satisfaction problem (cf. Section 1.3). The choice of defining VCSPs as decision problems, rather than as optimisation problems, as it is common for VCSPs over finite domains, is motivated by two major issues that do not occur in the finite-domain case. First, in the infinite-domain setting, one needs to capture the difference between a proper minimum and an infimum value that the cost of the assignment can be arbitrarily close to but never reach. Second, this choice allows us to model the case in which the infimum is $-\infty$, i.e., when there are assignments for the variables of arbitrarily small cost.

Remark 1.1.4. Sometimes, to prove the polynomial-time solvability of specific VCSPs, we exhibit algorithms that take as input only the set of variables and the objective function, and compute the infimum while specifying whether it is attained, i.e., whether it is a proper minimum. Such algorithms can be easily adapted to deal with instances in the form of Definition 1.1.3, where a threshold $u_I \in \mathbb{Q}$ is given as part of the input, and the task is to decide whether there exists an assignment of cost smaller or equal to u_I.

By the terminologies *VCSP over an infinite domain* and *infinite-domain VCSP* (*VCSP over a finite domain*, and *finite-domain VCSP*, respectively) we will refer to the VCSP for a valued structure with an infinite domain (to the VCSP for a valued structure with a finite domain, respectively).

Note that, given a valued structure Γ, if the signature τ of Γ is finite, it is inessential for the computational complexity of VCSP(Γ) how the function symbols in ϕ_I are represented. When considering valued structures with a (countably) infinite signature, the computational complexity of VCSP(Γ) depends on how the symbols in the signature are represented in the input instances. VCSPs for valued structures with infinite signatures have been studied in the case in which the domain is finite. In which case, the typical choice for the representation of cost functions is to list all their values in correspondence of every tuple in the domain (usually, when the domain of Γ is infinite this is no longer an option) or to assume a value-giving oracle (in this case the problem is tractable if it can be solved in polynomial time by using only polynomially many queries to the oracle, see [57]).

Because all the concrete infinite-domain valued structures with infinite signatures that we consider in this thesis are *piecewise linear*, in Section 1.2.4 we fix a representation of these cost functions that is strictly related both to the mathematical properties of piecewise linear functions and to the algorithmic procedures and mathematical tools that we want to use to deal with them.

1.2 Cost Functions over the Rationals

The class of all valued structures with arbitrary infinite domains is too large to allow for complete complexity classifications (see [3]: in general, every computational decision problem over a finite alphabet is polynomial-time Turing-equivalent to a CSP over an infinite domain, and therefore to a VCSP over an infinite domain with values in $\{0, +\infty\}$), so we have to restrict our focus to subclasses of infinite-domain VCSPs. In this section, we describe two natural and large classes of cost functions over the domain $D = \mathbb{Q}$, the rational numbers. These classes are most naturally introduced using first-order definability, so we briefly fix the necessary logic concepts.

1.2.1 Logic Preliminaries

We fix some standard logic terminology (see, e.g., [54]).

Definition 1.2.1. A *signature* is a set τ of function and relation symbols. Each function symbol f and each relation symbol R is equipped with an arity $\mathrm{ar}(f)$, $\mathrm{ar}(R) \in \mathbb{N}$. A *$\tau$-structure* $\mathfrak{A}$ consists of

- a set $A = \mathrm{dom}(\mathfrak{A})$, called the *domain* of $\mathfrak{A}$, whose elements are called the *elements* of the τ-structure;
- for each relation symbol $R \in \tau$ a relation $R^{\mathfrak{A}} \subseteq A^{\mathrm{ar}(R)}$;
- for each function symbol $f \in \tau$ a function $f^{\mathfrak{A}} \colon A^{\mathrm{ar}(f)} \to A$.

Function symbols of arity 0 are allowed and are called constant symbols. A *relational structure* (or structure with *relational signature*) is a structure whose signature contains only relation symbols. A *functional structure* (or structure with a *functional signature*) is a structure whose signature contains only function symbols. We want to point out that the definition of valued structure given in Section 1.1 (Definition 1.1.1) is different from the definition of structure with a functional signature, in general. In fact, the cost functions of a valued structure need not take values in the domain of the valued structure and might take value $+\infty$, i.e., might be not defined, in correspondence of some tuples of elements in the domain. However, the definitions of valued structure and of functional structure coincide whenever the interpretations of the function symbols assume only finite values ranging over a set which coincides with the domain.

We give two examples of structures that play an important role in this thesis.

Example 1.2.2. Let $\mathfrak{S}$ be the structure with domain $\mathbb{Q}$ and signature $\sigma := \{+, 1, \leq\}$ where

- $+$ is a binary function symbol that denotes the usual addition over $\mathbb{Q}$,
- 1 is a constant symbol that denotes $1 \in \mathbb{Q}$, and
- $\leq$ is a binary relation symbol that denotes the usual linear order of the rationals. $\triangle$

Example 1.2.3. Let $\mathfrak{L}$ be the structure with domain $\mathbb{Q}$ and a (countably infinite) signature $\tau_0 := \{<, 1\} \cup \{c\cdot\}_{c \in \mathbb{Q}}$ where

- $<$ is a relation symbol of arity 2 and $<^{\mathfrak{L}}$ is the strict linear order of $\mathbb{Q}$,
- 1 is a constant symbol and $1^{\mathfrak{L}} := 1 \in \mathbb{Q}$, and
- $c\cdot$ is a unary function symbol for every $c \in \mathbb{Q}$ such that $(c\cdot)^{\mathfrak{L}}$ is the function $x \mapsto cx$ (multiplication by c). $\triangle$

We use the following terminology from model theory for structures and extend it to valued structures.

Definition 1.2.4. Let Γ and Δ be (valued) structures over the same domain, i.e., $\mathrm{dom}(\Gamma) = \mathrm{dom}(\Delta)$, with signature σ and τ, respectively. We say that Δ is a *(valued) reduct* of Γ if $\tau \subseteq \sigma$ and for every symbol $s \in \tau$ the interpretation s^Γ coincides with the interpretation s^Δ. In this case, we also say that Γ is an *expansion* of Δ. We say that a (valued) reduct Δ of a (valued) structure Γ is a *(valued) finite reduct* of Γ if Δ has finite signature.

Definition 1.2.5. Let Γ and Δ be (valued) structures with the same signature τ. We say that Δ is a *(valued) substructure* of Γ if $\mathrm{dom}(\Delta) \subseteq \mathrm{dom}(\Gamma)$ and for every symbol $s \in \tau$, the interpretation s^Δ coincides with the restriction to $\mathrm{dom}(\Delta)$ of the interpretation s^Γ. In this case, we also say that Γ is an *extension* of Δ. We say that a (valued) substructure Δ of a (valued) structure Γ is a *(valued) finite substructure* of Γ if Δ has finite domain.

1.2.2 Quantifier Elimination

We adopt the usual definition of first-order logic. A formula is *atomic* if it does not contain logical symbols (connectives or quantifiers). By convention, we have two special atomic formulas, $\top$ and $\bot$, to denote truth and falsity.

Definition 1.2.6. Let τ be a signature. We say that a τ-structure $\mathfrak{A}$ has *quantifier elimination* if every first-order τ-formula is equivalent to a quantifier-free τ-formula over $\mathfrak{A}$.

Theorem 1.2.7 (Ferrante and Rackoff, [40], Section 3, Theorem 1)**.** *The structure $\mathfrak{S}$ from Example 1.2.2 has quantifier elimination.*

Theorem 1.2.8. *The structure $\mathfrak{L}$ from Example 1.2.3 has quantifier elimination.*

Observe that every atomic τ_0-formula ϕ has at most two variables:

- if ϕ has no variables, then it is equivalent to $\top$ or $\bot$,
- if ϕ has only one variable, say x, then it is equivalent to $c{\cdot}x\,\sigma\,d{\cdot}1$ or to $d{\cdot}1\,\sigma\,c{\cdot}x$ for $\sigma \in \{<,=\}$ and $c, d \in \mathbb{Q}$. Moreover, if $c = 0$ then ϕ is equivalent to a formula without variables, and otherwise ϕ is equivalent to $x\,\sigma\,\frac{d}{c}{\cdot}1$ or to $\frac{d}{c}{\cdot}1\,\sigma\,x$ for $\sigma \in \{<,=\}$, which we abbreviate by the more common $x < \frac{d}{c}$, $x = \frac{d}{c}$, and $\frac{d}{c} < x$, respectively.
- if ϕ has two variables, say x and y, then ϕ is equivalent to $c{\cdot}x\,\sigma\,d{\cdot}y$ or $c{\cdot}x\,\sigma\,d{\cdot}y$ for $\sigma \in \{<,=\}$. Moreover, if $c = 0$ or $d = 0$ then the formula ϕ is equivalent to a formula with at most one variable, and otherwise ϕ is equivalent to $x\,\sigma\,\frac{d}{c}{\cdot}y$ or to $\frac{d}{c}{\cdot}y\,\sigma\,x$.

To prove Theorem 1.2.8, it suffices to prove the following lemma.

Lemma 1.2.9. *For every quantifier-free τ_0-formula φ there exists a quantifier-free τ_0-formula ψ such that $\exists x.\varphi$ is equivalent to ψ over $\mathfrak{L}$.*

Proof. We define ψ in five steps.

1. Rewrite φ, using De Morgan's laws, in such a way that all the negations are applied to atomic formulas.
2. Replace
   - $\neg(s = t)$ by $s < t \vee t < s$, and
   - $\neg(s < t)$ by $t < s \vee s = t$,

   where s and t are τ_0-terms.

3. Write φ in disjunctive normal form in such a way that each of the clauses is a conjunction of non-negated atomic τ_0-formulas (this can be done by distributivity).

4. Observe that $\exists x \bigvee_i \bigwedge_j \chi_{i,j}$, where the $\chi_{i,j}$ are atomic τ_0-formulas, is equivalent to $\bigvee_i \exists x \bigwedge_j \chi_{i,j}$. Therefore, it is sufficient to prove the lemma for $\varphi = \bigwedge_j \chi_j$ where the χ_j are atomic τ_0-formulas. As explained above, we can assume without loss of generality that the χ_j are of the form $\top$, $\bot$, $x\,\sigma\,c$, $c\,\sigma\,x$, or $x\,\sigma\,cy$, for $c \in \mathbb{Q}$ and $\sigma \in \{<, =\}$. If χ_j equals $\bot$, then φ is equivalent to $\bot$ and there is nothing to be shown. If χ_j equals $\top$ then it can simply be removed from φ. If χ_j equals $x = c$ or $x = cy$ then replace every occurrence of x by $c \cdot 1$ or by $c \cdot y$, respectively. Then φ does not contain the variable x anymore and thus $\exists x.\varphi$ is equivalent to φ.

5. We are left with the case that all atomic τ_0-formulas involving x are (strict) inequalities, that is, $\varphi = \bigwedge_i \chi_i \wedge \bigwedge_i \chi'_i \wedge \bigwedge_i \chi''_i$, where

   - the χ_i are atomic formulas not containing x,
   - the χ'_i are atomic formulas of the form $x > u_i$,
   - the χ''_i are atomic formulas of the form $x < v_i$.

   Then $\exists x.\varphi$ is equivalent to $\bigwedge_i \chi_i \wedge \bigwedge_{i,j}(u_i < v_j)$.

Each step of this procedure preserves the satisfying assignments for φ and the resulting formula is in the required form; this is obvious for all but the last step, and for the last step follows from the correctness of Fourier-Motzkin elimination for systems of linear inequalities (see, e.g., [91], Section 12.2). Therefore, the procedure is correct. □

Proof (of Theorem 1.2.8). Let φ be a τ_0-formula. We prove that it is equivalent to a quantifier-free τ_0-formula by induction on the number n of quantifiers of φ. For $n = 1$ we have two cases:

- If φ is of the form $\exists x.\varphi'$ (with φ' quantifier-free) then, by Lemma 1.2.9, it is equivalent to a quantifier-free τ_0-formula ψ.

- If φ is of the form $\forall x.\varphi'$ (with φ' quantifier-free), then it is equivalent to $\neg\exists x.\neg\varphi'$. By Lemma 1.2.9, $\exists x.\neg\varphi'$ is equivalent to a quantifier-free τ_0-formula ψ. Therefore, φ is equivalent to the quantifier-free τ_0-formula $\neg\psi$.

Now suppose that φ is of the form $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n.\varphi'$ for $n \geq 2$ and $Q_1, \ldots, Q_n \in \{\forall, \exists\}$, and suppose that the statement is true for τ_0-formulas with at most $n-1$ quantifiers. In particular, $Q_2 x_2 \cdots Q_n x_n.\varphi'$ is equivalent to a quantifier -free τ_0-formula ψ. Therefore, φ is equivalent to $Q_1 x_1.\psi$, that is, a τ_0-formula with one quantifier that is equivalent to a quantifier-free τ_0-formula, again by the inductive hypothesis. $\square$

1.2.3 PL and PLH Cost Functions

A *partial function* of arity $n \in \mathbb{N}$ over a set A is a function

$$f\colon \operatorname{dom}(f) \to A \text{ for some } \operatorname{dom}(f) \subseteq A^n.$$

Let $\mathfrak{A}$ be a τ-structure. A partial function over A is called *first-order definable over* $\mathfrak{A}$ if there exists a first-order τ-formula $\phi(x_0, x_1, \ldots, x_n)$ such that for all $a_1, \ldots, a_n \in A$

- if $(a_1, \ldots, a_n) \in \operatorname{dom}(f)$ then $\mathfrak{A} \models \phi(a_0, a_1, \ldots, a_n)$ if and only if $a_0 = f(a_1, \ldots, a_n)$, and
- if $f(a_1, \ldots, a_n) \notin \operatorname{dom}(f)$ then there is no $a_0 \in A$ such that $\mathfrak{A} \models \phi(a_0, a_1, \ldots, a_n)$.

In the following, we consider *cost functions over* $\mathbb{Q}$, which are functions from $\mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$. It is sometimes convenient to view a cost function as a partial function over $\mathbb{Q}$. We interpret $f(t) = +\infty$ as $t \in \mathbb{Q}^{\operatorname{ar}(f)} \setminus \operatorname{dom}(f)$.

Definition 1.2.10. A cost function $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ (viewed as a partial function) is called

- *piecewise linear (PL)* if it is first-order definable over $\mathfrak{S}$, piecewise linear functions are sometimes called *semilinear* functions;
- *piecewise linear homogeneous (PLH)* if it is first-order definable over $\mathfrak{L}$ (viewed as a partial function).

A valued structure Γ is called *piecewise linear* (*piecewise linear homogeneous*) if every cost function in Γ is PL (or PLH, respectively).

Every PLH cost function is also PL since all functions of the structure $\mathfrak{L}$ are first-order definable in $\mathfrak{S}$. The cost functions in the valued structures from Examples 5-7 are PLH. The Linear Programming problem (cf. Example 4) can be modelled as a VCSP with PL cost functions (cf. Section 1.7), but it cannot be expressed as a VCSP with PLH cost functions.

We would like to point out that already the class of PLH cost functions is very large. In particular, one can view it as a generalisation of the class of all sets of cost functions over a finite domain D. Indeed, every VCSP for a valued structure with a finite domain is equivalent to a VCSP for a PLH valued structure: by identifying the finite domain D with a subset of $\mathbb{Q}$ in an arbitrary way, for every cost function $f\colon D^d \to \mathbb{Q} \cup \{+\infty\}$ we obtain the PLH cost function $f'\colon \mathbb{Q}^d \to \mathbb{Q} \cup \{+\infty\}$ defined by $f'(x_1, \dots, x_n) := f(x_1, \dots, x_n)$ if $x_1, \dots, x_n \in D$, and $f'(x_1, \dots, x_n) = +\infty$.

Remark 1.2.11. In both the PL and the PLH context the sets $\mathbb{Q}$ and $\mathbb{R}$ are interchangeable (both when considered as the domain of the functions and when considered as the set of finite values of the functions). However, when considering PL or PLH valued structures with an infinite signature, we might assume the variables to range over $\mathbb{R}$, but we always require all the coefficients to be rational as we need to manipulate them computationally.

1.2.4 Representation of PL Cost Functions

When considering valued structures with a (countably) infinite signature, the computational complexity of VCSP(Γ) depends on how the symbols in the signature are represented in the input instances. To represent a PL cost function, we use the fact that the structure $\mathfrak{S}$ has quantifier elimination (cf. [40]).

Definition 1.2.12. A set $C \subseteq \mathbb{Q}^n$ is a *polyhedral set* if it is the intersection of finitely many (open or closed) halfspaces, i.e., it can be specified by a conjunction of finitely many linear constraints, i.e., for some $r \in \mathbb{N}$ there exist linear functions $f_i\colon \mathbb{Q}^n \to \mathbb{Q}$, for $1 \le i \le r$, such that

$$C = \left\{ x \in \mathbb{Q}^n \mid \bigwedge_{i=1}^{p} (f_i(x) \le 0) \wedge \bigwedge_{i=p+1}^{q} (f_i(x) < 0) \wedge \bigwedge_{i=q+1}^{r} (f_i(x) = 0) \right\}.$$

A polyhedral set $C \subseteq \mathbb{Q}^n$ is *open* if it is the intersection of finitely many open halfspaces, i.e., for some $p \in \mathbb{N}$ there exist $f_i\colon \mathbb{Q}^n \to \mathbb{Q}$ linear functions, $1 \le i \le p$, such that $C = \{x \in \mathbb{Q}^n \mid \bigwedge_{i=1}^{p} (f_i(x) < 0)\}$. Similarly, a polyhedral set $C \subseteq \mathbb{Q}^n$ is *closed* if it is the intersection of finitely many closed halfspaces, i.e., for some $q \in \mathbb{N}$ there exist $f_i\colon \mathbb{Q}^n \to \mathbb{Q}$ linear functions, $1 \le i \le p$, such that $C = \{x \in \mathbb{Q}^n \mid \bigwedge_{i=1}^{p} (f_i(x) \le 0) \wedge \bigwedge_{i=p+1}^{q} (f_i(x) = 0)\}$. A polyhedral set $C \subseteq \mathbb{Q}^n$ is *bounded* if it is bounded as a subset of $\mathbb{Q}^n$. We remark that the infimum of a linear function in a closed and bounded polyhedral set is a proper minimum; while the infimum of a linear function in an open or unbounded polyhedral set is attained only if the linear function is constant.

Since the structure $\mathfrak{S}$ has quantifier elimination, a PL cost function can be written in the following form.

Remark 1.2.13 ([88], Definition 2.47). Let $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ be a PL function, then its domain dom(f) defined by R_f can be written as the union of finitely many polyhedral sets, relative to each of which $f(x)$ is given by a linear expression, i.e., there exist finitely many mutually disjoint $C_1, \dots, C_m$ polyhedral sets such that $\bigcup_{i=1}^m C_i = \mathrm{dom}(f) \subseteq \mathbb{Q}^n$, and

$$f(x_1, \dots, x_n) = \begin{cases} a_0^i + a_1^i x_1 + \dots + a_n^i x_n & \text{if } (x_1, \dots, x_n) \in C_i \\ +\infty & \text{if } (x_1, \dots, x_n) \in \mathbb{Q}^n \setminus \mathrm{dom}(f) \end{cases}$$

where $a^i = (a_0^i, a_1^i, \dots, a_n^i) \in \mathbb{Q}^{n+1}$, for $1 \leq i \leq m$.[2]

Sometimes, given a PL cost function f, we refer to the formulas defining the polyhedral sets in the input representation of f as the *case distinctions* of f. Given a PL function $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$, we call *regions of linearity* the finitely many subset $S_1, \dots, S_k \subseteq \mathrm{dom}(f)$ such that

- $\bigcup_{i=1}^k S_i = \mathrm{dom}(f)$,
- $f_{|S_i}$ is a linear polynomial for every $1 \leq i \leq k$, and
- k is minimal.

Every region of linearity is the union of finitely many polyhedral sets.

We fix the following representation of PL cost functions, which we use throughout the thesis, whenever we consider PL valued structures with an infinite signature.

Definition 1.2.14. Representation of PL cost functions. We assume that a PL cost function is represented by a list of linear constraints, specifying finitely many pairwise disjoint polyhedral sets, and a list of linear polynomials and $+\infty$s, defining the value of the function in each polyhedral set. The linear constraints and the linear polynomials are encoded by the list of their rational coefficients, and $+\infty$ is represented by a special symbol. The constants for (numerators and denominators of) rational coefficients for linear constraints and linear polynomials are represented in binary.

Definition 1.2.15. Given a PL valued structure Γ with an infinite signature, the *size of an instance I* of VCSP(Γ) is defined as the number of bits required to represent I as in Definition 1.2.14.

We show now that the VCSP for the valued structure containing *all* the PL cost functions is NP-complete.

Proposition 1.2.16. *For every valued PL valued structure Γ the problem* VCSP(Γ) *is in NP.*

[2]In [88] it was observed that perhaps piecewise linear functions should be called piecewise affine functions.

Proof. An instance I of VCSP(Γ) is given by a set of variables V, by an objective function given as the sum of finitely many PL cost functions $\sum_{i=1}^{m} f_i(x)$, with $x^i \in V^{\mathrm{ar}(f_i)}$, and by a threshold $u \in \mathbb{Q}$. By Remark 1.2.13, we can assume every cost function showing up in the instance is given in the form

$$f_i(x^i) = l_{i,j}(x^i) \quad \text{if } C_{i,j}(x^i)$$

for $1 \leq j \leq k_i$, where $l_{i,j}(x^i)$ is either a linear polynomial or the symbol $+\infty$ and $C_{i,j}(x^i)$ is a conjunction of linear constraints. For every $1 \leq i \leq m$ we define the new variable t_i and the formula

$$\psi_i(x^i, t_i) := \bigvee_{j=1}^{k_i} \left[(t_i = l_{i,j}(x^i)) \wedge C_{i,j}(x^i) \right].$$

We map the instance I to the formula

$$\exists x_1. \cdots \exists x_n. \bigwedge_{j=1}^{m} \psi_i(x^i, t_i) \wedge (t_1 + \cdots + t_m \leq u).$$

The map we defined is a polynomial-time many-one reduction from the VCSP for a PL valued structure to the existential theory of $\mathfrak{S} = (\mathbb{Q}; \leq, +, 1)$, which is in NP as explained in [10]. $\square$

Remark 1.2.17. The NP-hardness of the VCSP for the valued structure containing all the PL cost functions follows from the fact that there exist NP-hard problems, e.g., LEAST CORRELATION CLUSTERING WITH PARTIAL INFORMATION and MINIMUM FEEDBACK ARC SET (see Examples 5 and 6), which can be formulated as VCSPs for PL cost functions.

1.3 Constraint Satisfaction Problems

The question of whether an instance of VCSP(Γ) is *feasible*, that is, whether it admits an assignment with a finite cost, can be viewed as a (classical) constraint satisfaction problem. Formally, the constraint satisfaction problem for a relational structure $\mathfrak{A}$ with relational signature τ is the following computational problem, denoted by CSP($\mathfrak{A}$):

- the input is a finite conjunction ψ of atomic τ-formulas (*constraints*), and
- the question is whether ψ is satisfiable in $\mathfrak{A}$.

Let Γ be a valued structure with signature τ, we can associate Γ with the following relational structure Feas(Γ): for every function symbol f of arity n

from τ the signature of Feas(Γ) contains a relation symbol R_f of arity n such that $R_f^{\text{Feas}(\Gamma)} = \text{dom}(f^\Gamma)$. Every polynomial-time algorithm for VCSP(Γ) has to solve CSP(Feas(Γ)), in particular. In fact, an instance of VCSP(Γ) with objective function

$$\phi(x) = \sum_{i=1}^{m} f_i(x^i),$$

can be translated into an instance ψ of CSP(Feas(Γ)) by replacing subexpressions of the form $f(x_1, \ldots, x_n)$ in ϕ by $R_f(x_1, \ldots, x_n)$ and by replacing $+$ by $\wedge$, i.e.,

$$\psi(x) = \text{Feas}(\phi)(x) := \bigwedge_{i=1}^{m} R_{f_i}(x^i).$$

It is easy to see that ϕ is a feasible instance of VCSP(Γ) if, and only if, ψ is satisfiable in Feas(Γ).

Definition 1.3.1. Let Γ be a valued structure with domain D, and let $\phi(x)$ be an objective function for VCSP(Γ) with n free variables. We call *feasibility region* the set $\text{dom}(\phi) \subseteq D^n$, defined by $\psi(x) = \text{Feas}(\phi)(x)$.

Definition 1.3.2. By *finite-valued structures* we refer to a valued structure whose cost functions are finite-valued, i.e., they take a finite value on every point with rational coordinates.

The VCSP for a finite-valued structure is a mere optimisation problem, and in solving it, one does not have to care about the feasibility of the optimal solutions.

1.3.1 Semilinear Constraint Satisfaction Problems

Definition 1.3.3. A set that is first-order definable over $\mathfrak{S}$ (with parameters from $\mathbb{Q}$) is called *semilinear* (or *PL*).

The feasibility region of a PL function is a semilinear set; this is why PL functions are also called semilinear. The regions of linearity of a PL function are semilinear sets.

Definition 1.3.4. A relational structure $\mathfrak{A}$ with domain $\mathbb{Q}$ and signature τ is called *semilinear* if, for all $R \in \tau$, the interpretation $R^{\mathfrak{A}}$ is semilinear.

Definition 1.3.5. A set that is first-order definable over $\mathfrak{L}$ (with parameters from $\mathbb{Q}$) is called *piecewise linear homogeneous (PLH)*. A relational structure $\mathfrak{A}$ with domain $\mathbb{Q}$ and relational signature τ is called *piecewise linear homogeneous (PLH)* if, for all $R \in \tau$, the interpretation $R^{\mathfrak{A}}$ is PLH.

Clearly, if Γ is a PL valued structure, then $\mathrm{Feas}(\Gamma)$ is a semilinear relational structure; and if Γ is a PLH valued structure, then $\mathrm{Feas}(\Gamma)$ is a PLH relational structure. The computational complexity of semilinear CSPs has been studied for semilinear expansions of Linear Programming in [5], semilinear expansions of $(\mathbb{R};+)$ in [61], max-closed semilinear structures in [9], and median-closed semilinear structures in [11]. We also refer the reader to Section 4 of [10] for a recent survey on semilinear CSPs.

1.4 Universal Algebraic Tools

Let D be a set. If $x^1,\dots,x^k \in D^n$ and $g\colon D^k \to D$ is a function, then $g(x^1,\dots,x^k)$ denotes the n-tuple obtained by applying g *componentwise*, i.e.,

$$g(x^1,\dots,x^k) := (g(x^1_1,\dots,x^k_1),\dots,g(x^1_n,\dots,x^k_n)).$$

Given a set X, and a function $\omega\colon X \to \mathbb{Q}_{\geq 0}$. We define the *support* of ω as the set

$$\mathrm{Supp}(\omega) := \{x \in X \mid \omega(x) \neq 0\}.$$

An important and well-known concept from universal algebra used in the constraint satisfaction framework is the notion of a *homomorphism* between (relational) structures.

Definition 1.4.1. Let $\mathfrak{A}$ and $\mathfrak{B}$ be (relational) structures with the same signature τ and with domain A and B, respectively. A *homomorphism* from $\mathfrak{A}$ to $\mathfrak{B}$ is a function $h\colon A \to B$ such that for every relation symbol $R \in \tau$ and tuple $a \in D^{\mathrm{ar}(R)}$

$$R^{\mathfrak{A}}(a) \text{ implies } R^{\mathfrak{B}}\left(g(a)\right).$$

We say that $\mathfrak{A}$ is *homomorphic* to $\mathfrak{B}$ and write $\mathfrak{A} \to \mathfrak{B}$ to indicate the existence of a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$.

A generalisation of the universal algebraic notion of homomorphism to the valued constraint satisfaction framework has to capture, not only the satisfiability of the feasibility relations, but also the property of not increasing the value of cost functions. These requirements are fulfilled by the notion of *fractional homomorphism* (cf. [96]).

Definition 1.4.2. Let Δ and Γ be valued structures with the same signature τ and with domain D and C, respectively. Let C^D denote the set of all functions $g\colon D \to C$. A *fractional homomorphism* from Δ to Γ is a function $\omega\colon C^D \to Q_{\geq 0}$ *with a finite support*, $\mathrm{Supp}(\omega) := \{g \in C^D \mid \omega(g) > 0\}$, such that $\sum_{g\in C^D} \omega(g) = 1$, and such that for every function symbol $f \in \tau$ and tuple $a \in D^{\mathrm{ar}(f)}$, it holds that

$$\sum_{g \in C^D} \omega(g) f^{\Gamma}(g(a)) \leq f^{\Delta}(a).$$

We say that Δ is *fractionally homomorphic* to Γ and write $\Delta \to_f \Gamma$ to indicate the existence of a fractional homomorphism from Δ to Γ.

Trivially, if Γ is a valued structure and Δ is a valued substructure of Γ, then Δ is fractionally homomorphic to Γ.

The following proposition is adapted from [96], Proposition 2.1, where it is stated for valued structures with finite domains.

Proposition 1.4.3. *Let Δ and Γ be valued structures over the same signature τ and with domain D and C, respectively. Assume that $\Delta \to_f \Gamma$. Let I be an instance of* VCSP(Γ) *having variables $V_I = \{x_1, \dots, x_n\}$, objective function $\phi_I(x_1, \dots, x_n)$, and threshold $u_I \in \mathbb{Q}$. If there exists an assignment $h\colon V_I \to D$ with cost $\phi_I^{\Delta}(h(x_1, \dots, h(x_n)) \leq u_I$, then there exists an assignment $h'\colon V_I \to C$ with cost $\phi_I^{\Gamma}(h'(x_1), \dots, h'(x_n)) \leq u_I$. In particular, it holds that*

$$\inf_{c \in C^n} \phi_I^{\Gamma}(c) \leq \inf_{d \in D^n} \phi_I^{\Delta}(d).$$

Proof. Let I be an instance of VCSP(Γ) with variables $V_I = \{x_1, \dots, x_n\}$, objective function

$$\phi_I(x_1, \dots, x_n) = \sum_{j \in J} f_j(x^j),$$

with $f_j \in \tau$, $x^j \in V_I^{\mathrm{ar}(f_j)}$, for all $j \in J$, and threshold $u_I \in \mathbb{Q}$. Let ω be a fractional homomorphism from Δ to Γ, and let $h\colon V_I \to C$ be an arbitrary assignment. Then

$$\sum_{j \in J} f_j^{\Delta}(h(x^j)) \geq \sum_{j \in J} \sum_{g \in C^D} \omega(g) f_j^{\Gamma}(g(h(x^j))) = \sum_{g \in C^D} \omega(g) \sum_{j \in J} f_j^{\Gamma}(g(h(x^j))).$$

Therefore, there exists $g \in C^D$ such that the cost of the assignment $g \circ h$ to the variables of V_I when I is considered as an instance of VCSP(Γ) is at most the cost of the assignment h to variables of V_I when I is considered as an instance of VCSP(Δ). In particular, if the cost of h is at most u_I, then the cost of $g \circ h$ is at most u_I, too. □

We give now the notions of *polymorphism* and *fractional polymorphism*, which played a key role in the computational complexity classification of CSPs and VCSPs over finite domains.

Definition 1.4.4. A k-ary *operation* on a set D is a function $g\colon D^k \to D$. For $k \in \mathbb{N}$, we denote by $\mathcal{O}_D^{(k)}$ the set of all k-*ary operations* on D and we let $\mathcal{O}_D := \bigcup_{k \in \mathbb{N}} \mathcal{O}_D^{(k)}$.

Definition 1.4.5. Let $\mathfrak{A}$ be a structure with relational signature τ and domain D. Then a k-ary operation $g\colon D^k \to D$ is called a *polymorphism* of $\mathfrak{A}$ if for all $R \in \tau$ we have that $g(x^1, \ldots, x^k) \in R^{\mathfrak{A}}$ for all $x^1, \ldots, x^k \in R^{\mathfrak{A}}$, namely $R^{\mathfrak{A}}$ is *preserved* by g (where g is applied componentwise).

Observe that a k-ary polymorphism of a structure $\mathfrak{A}$ is a homomorphism from $\mathfrak{A}^k$ to $\mathfrak{A}$. The intuition behind the notion of polymorphism is that it enables to combine many feasible assignments into a new feasible assignment.

Definition 1.4.6. For $k \in \mathbb{N}$ and $1 \leq i \leq k$, the ith k-ary *projection* on a set D is the function $e_i^{(k)}\colon D^k \to D$ of the form $e_i^{(k)}(x^1, \ldots, x^k) = x^i$. For $k \in \mathbb{N}$, we denote by $\mathscr{J}_D^{(k)}$ the set of all *k-ary projections* on D and we let $\mathscr{J}_D := \bigcup_{k\in\mathbb{N}} \mathscr{J}_D^{(k)}$.

The projections on a set D are polymorphisms of every relational structure with domain D. In the valued constraint satisfaction framework, the notion of polymorphism is generalised by the notion of fractional polymorphism.

Definition 1.4.7. Let D be a set. An *m-ary fractional operation* on D *with a finite support* is a function $\omega\colon \mathscr{O}_D^{(m)} \to \mathbb{Q}_{\geq 0}$ with a finite support such that $\sum_{g\in\mathrm{Supp}(\omega)} \omega(g) = 1$.

Definition 1.4.8. Given a cost function $\gamma\colon D^n \to \mathbb{Q} \cup \{+\infty\}$, a k-ary fractional operation ω on D with a finite support is a *fractional polymorphism* of γ if, for every tuple $a^1, \ldots, a^k \in \mathbb{Q}^{\mathrm{ar}(\gamma)}$, it holds that

$$\sum_{g\in\mathrm{Supp}(\omega)} \omega(g)\gamma(g(a^1, \ldots, a^k)) \leq \frac{1}{k}\sum_{i=1}^{k} \gamma(a^i).$$

If ω is a fractional polymorphism of a cost function $\gamma\colon D^n \to \mathbb{Q} \cup \{+\infty\}$, we say that γ is *improved* by ω.

We mention that it is possible to define *fractional polymorphisms with arbitrary supports* (see Section 7.4). However, throughout the thesis we always refer to fractional operations with finite supports, unless it is explicitly indicated.

Remark 1.4.9. Fractional polymorphisms can be equivalently seen in a probabilistic setting. A fractional operation $\omega\colon \mathscr{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$ with a finite support is a fractional polymorphism of a function $\gamma\colon D^n \to \mathbb{Q} \cup \{+\infty\}$ if ω is a probability distribution over $\mathscr{O}_D^{(k)}$ (with a finite support) and it satisfies

$$\mathbb{E}_{g\sim\omega}\gamma(g(a^1, \ldots, a^k)) \leq \frac{1}{k}\sum_{i=1}^{k} \gamma(a^i),$$

for every $a^1, \ldots, a^k \in D^n$, where $\mathbb{E}_{g\sim\omega}\gamma(g(a^1, \ldots, a^k))$ is the expected value of γ associated with ω.

The intuition behind the notion of fractional polymorphism is that it enables to combine many feasible assignments into new feasible assignments such that the expected cost of a new assignment is improved, i.e., it is at most the average cost of the original assignments.

Example 1.4.10. Let us fix $k \in \mathbb{N}$, and $c \in \mathbb{Q}$. We define $\tilde{c}\colon \mathbb{Q}^k \to \mathbb{Q}$ to be the operation such that $\tilde{c}(a^1, \ldots, a^k) := c$, for every $a^1, \ldots, a^k \in \mathbb{Q}$. Let us consider the fractional operation $\chi_c\colon \mathcal{O}_{\mathbb{Q}}^{(k)} \to \mathbb{Q}_{\geq 0}$ defined by setting

$$\chi_c(g) := \begin{cases} 1 & \text{if } g = \tilde{c} \\ 0 & \text{otherwise.} \end{cases}$$

Then, an n-ary cost function f is improved by χ_c if, for every $a^1, \ldots, a^k \in \mathbb{Q}^n$, it holds that

$$f(c, \ldots, c) \leq \frac{1}{k}\sum_{i=1}^{k} f(a_1^i, \ldots, a_n^i),$$

which means that the tuple $(c, \ldots, c) \in \mathbb{Q}^n$ is expected to have a cost which is not greater than the average cost of other tuples of $\mathbb{Q}^n$. △

Definition 1.4.11. Given a valued structure Γ, we say that ω is a *fractional polymorphism* for Γ if ω is a fractional polymorphism of f^Γ for every function symbol f in the signature τ of Γ. The set of all fractional polymorphisms of a valued structure Γ is denoted by $\mathrm{fPol}(\Gamma)$. The set of all functions that are improved by a given set of fractional operations Ω is denoted by $\mathrm{Imp}(\Omega)$.

Definition 1.4.12. Let Γ be a valued structure. We define the *support* of Γ by setting

$$\mathrm{Supp}(\Gamma) := \bigcup_{\omega\in\mathrm{fPol}(\Gamma)} \mathrm{Supp}(\omega).$$

It is easy to observe that if ω is a fractional polymorphism of a given valued structure Γ, then every operation in $\mathrm{Supp}(\omega)$ is a polymorphism of $\mathrm{Feas}(\Gamma)$, i.e.,

$$\mathrm{Supp}(\Gamma) \subseteq \mathrm{Pol}(\mathrm{Feas}(\Gamma)).$$

Example 1.4.13. For every $k \in \mathbb{N}$, the fractional operation $\omega\colon \mathcal{O}_{\mathbb{Q}}^{(k)} \to \mathbb{Q}_{\geq 0}$ defined by

$$\omega(g) := \begin{cases} \frac{1}{k} & \text{if } g = e_i^{(k)}, \text{ for some } i \in \{1, \ldots, k\} \\ 0 & \text{otherwise} \end{cases}$$

is a fractional polymorphism of every valued structure with domain $\mathbb{Q}$. △

A k-ary operation g on a set D is *idempotent* if it holds that

$$g(x, \dots, x) = x \text{ for every } x \in D.$$

Example 1.4.14. Let D be a totally ordered set. The binary operations min and max on D giving, respectively, the smallest and the largest among two arguments are idempotent. △

A fractional operation is *idempotent* if every operation in its support is idempotent.

Remark 1.4.15. Let ω be an idempotent fractional operation on a set D, let $n \in \mathbb{N}$, and let f be an n-ary cost function with domain D. It is easy to see that if ω improves f, then it also improves the $(n-1)$-ary cost function $f(x_1, \dots, x_{i-1}, \cdot, x_{i+1} \dots, x_n)$ with domain D defined for $y \in D$ by

$$f(x_1, \dots, x_{i-1}, \cdot, x_{i+1} \dots, x_n)(y) = f(x_1, \dots, x_{i-1}, y, x_{i+1} \dots, x_n)$$

is improved by ω for all $1 \leq i \leq n$, and for all $x_1, \dots, x_{i-1}, x_{i+1} \dots, x_n \in D^{n-1}$.

The next example shows that Remark 1.4.15 becomes false, in general, if the fractional operation ω is not idempotent.

Example 1.4.16. Let $D = \{0, 1\}$ and let $\text{XOR}\colon D^2 \to \mathbb{Q}$ be the cost function defined for $x_1, x_2 \in D$ by

$$\text{XOR}(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 \neq x_2 \\ 1 & \text{if } x_1 = x_2. \end{cases}$$

Let $\omega_\neg$ be the unary fractional operation whose support contains a unique operation g on D defined by $g(x) = 1 - x$. Clearly, $\omega_\neg$ is not idempotent. It is easy to verify that $\omega_\neg$ improves XOR. However, $\omega_\neg$ does not improve the unary cost function $f\colon D \to \mathbb{Q}$ such that $f(x) = \text{XOR}(0, x)$. For instance, $f(g(1)) = 1 \not\leq 0 = f(1)$. △

Let $\mathcal{S}_k$ be the symmetric group on $\{1, \dots, k\}$. A k-ary operation g is *fully symmetric* if, for every permutation $\pi \in \mathcal{S}_k$, it holds that

$$g(x^1, \dots, x^k) = g(x^{\pi(1)}, \dots, x^{\pi(k)}).$$

A k-ary operation g is *totally symmetric* if it holds that

$$g(x^1, \dots, x^k) = g(y^1, \dots, y^k) \text{ whenever } \{x^1, \dots, x^k\} = \{y^1, \dots, y^k\}.$$

A fractional operation is *totally symmetric* (*fully symmetric*, respectively) if every operation in its support is totally symmetric (fully symmetric, respectively). As every totally symmetric operation is fully symmetric, every totally symmetric fractional operation is also fully symmetric. The fractional operations in the two following examples are totally symmetric.

Example 1.4.17. Let D be a totally ordered set. The fractional operation $\omega_{\text{sub}} \colon \mathcal{O}_D^{(2)} \to [0,1]$, defined by

$$\omega_{\text{sub}}(g) := \begin{cases} \frac{1}{2} & \text{if } g = \min \\ \frac{1}{2} & \text{if } g = \max \\ 0 & \text{otherwise} \end{cases}$$

is a binary totally symmetric fractional operation. △

Example 1.4.18. Let us define, for every $k \geq 2$, the fractional operations $\omega_{\min}^{(k)} \colon \mathcal{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$, and $\omega_{\max}^{(k)} \colon \mathcal{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$ by setting, respectively,

$$\omega_{\min}^{(k)}(g) := \begin{cases} 1 & \text{if } g = \min^{(k)} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\omega_{\max}^{(k)}(g) := \begin{cases} 1 & \text{if } g = \max^{(k)} \\ 0 & \text{otherwise.} \end{cases}$$

The fractional operations $\omega_{\min}^{(k)}$ and $\omega_{\max}^{(k)}$ are totally symmetric. △

We give now an example of a fractional operation that is fully symmetric but not totally symmetric.

Example 1.4.19. Let D be a connected set and let $\operatorname{avg}^{(k)} \colon D^k \to D$ be the k-ary arithmetic average operation defined, for every $(x^1, \dots, x^k) \in D^k$, by

$$\operatorname{avg}(x^1, \dots, x^k) := \frac{1}{k} \sum_{i=1}^{k} x^i.$$

Let us define, for every $k \geq 2$, the fractional operation $\omega_{conv}^{(k)} \colon \mathcal{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$ by setting

$$\omega_{conv}^{(k)}(g) := \begin{cases} 1 & \text{if } g = \operatorname{avg}^{(k)} \\ 0 & \text{otherwise.} \end{cases}$$

The fractional operations $\omega_{conv}^{(k)}$ are fully symmetric; however, for $k \geq 3$ they are not totally symmetric, because, e.g.,

$$\{1,1,2\} = \{1,2,2\}, \text{ but } \frac{1+1+2}{3} \neq \frac{1+2+2}{3}. \qquad \triangle$$

Definition 1.4.20. A k-ary operation $g\colon D^k \to D$ is said to be *conservative* if for every $(x^1,\dots,x^k) \in D^k$

$$g(x^1,\dots,x^k) \in \{x^1,\dots,x^k\}.$$

Example 1.4.21. The k-ary operations $\min^{(k)}$ and $\max^{(k)}$ giving the smallest and the largest, respectively, among k arguments are conservative. △

Definition 1.4.22. A k-ary fractional operation $\omega\colon \mathcal{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$ is a *conservative fractional operation* if every operation in its support is conservative.

Example 1.4.23. The fractional operation ω_{sub} introduced in Example 1.4.17 is conservative. △

Example 1.4.24. The k-ary arithmetic average operation $\text{avg}^{(k)}\colon D^k \to D$ (see Example 1.4.19) is not conservative, because, e.g., $\text{avg}^{(2)}(1,2) \notin \{1,2\}$. Therefore, the fractional operations $\omega_{\text{conv}}^{(k)}$ are not conservative. △

Definition 1.4.25. The *superposition* $h[g_1,\dots,g_m]$ of an m-ary operation h with m k-ary operations $g_1,\dots,g_m$ is the k-ary operation defined by

$$h[g_1,\dots,g_m](x^1,\dots,x^k) = h(g_1(x^1,\dots,x^k),\dots,g_m(x^1,\dots,x^k)).$$

The superposition of an m-ary operation h with m k-ary operations $g_1,\dots,g_m$ can also be seen as the composition $h \circ (g_1,\dots,g_m)\colon D^k \to D$ of the operation $h\colon D^m \to D$ with the map $(g_1,\dots,g_m)\colon D^k \to D^m$. The following is a standard definition from universal algebra.

Definition 1.4.26. A set $\mathcal{O}$ of operations is said to *generate* an operation g if g can be obtained by superposition of operations from $\mathcal{O}$ and projections.

Example 1.4.27. Let D be a totally ordered set. If $\mathcal{O} \subseteq \mathcal{O}_D$ contains the binary operation $\max\colon D^2 \to D$, then $\mathcal{O}$ generates the k-ary operation $\max^{(k)}$ that returns the largest of its k arguments by

$$\max{}^{(k)}(x^1,\dots,x^k) = \max(x^1,\max(x^2,\dots,\max(x^{k-1},x^k)\dots)). \qquad \triangle$$

Remark 1.4.28. Observe that if a relational structure $\mathfrak{A}$ is preserved by a set of operations $\mathcal{O} \subseteq \mathcal{O}_A$ then $\mathfrak{A}$ is preserved by all operations generated by $\mathcal{O}$.

Definition 1.4.29. The *superposition* $\omega[g_1,\dots,g_m]$ of an m-ary fractional operation ω with m k-ary operations $g_1,\dots,g_m$ is the k-ary fractional operation defined by

$$\omega[g_1,\dots,g_m](h) := \sum_{h' \mid h'[g_1,\dots,g_m]=h} \omega(h').$$

1.5 Submodularity

Submodular cost functions naturally appear in several scientific fields such as, for example, economics, game theory, machine learning, social network, and computer vision, and play a key role in operational research and combinatorial optimisation (see, e.g., [41, 74]). Submodularity also played an important role for the study of the computational complexity of finite-domain VCSPs, and guided the research on VCSPs for some time (see, e.g., [34, 60]), even though this might no longer be visible in the final classification obtained in [68, 69, 73].

Definition 1.5.1. Let D be a totally ordered set and let G be a totally ordered Abelian group. A partial function $f\colon D^n \to G$ is called *submodular* if for all $x,\, y \in D^n$

$$f(\min(x,y)) + f(\max(x,y)) \leq f(x) + f(y).$$

Definition 1.5.2. A valued structure Γ is *submodular* if all cost functions in Γ are submodular.

Remark 1.5.3. It is easy to see that a function over a totally ordered set D is submodular if, and only if, it is improved by the binary fractional operation ω_{sub} introduced in Example 1.4.17.

Note that if f is a submodular partial function and if $x, y \in \text{dom}(f)$, then $\min(x,y) \in \text{dom}(f)$ and $\max(x,y) \in \text{dom}(f)$. Therefore, if Γ is a submodular valued structure then the corresponding relational structure $\text{Feas}(\Gamma)$ (see Section 1.3) is preserved by the binary operations min and max.

1.6 Convexity

In Section 1.5, we introduced the notion of submodularity, which gives rise to the arguably most important property of cost functions in discrete optimisation. When moving to the context of continuous optimisation, the undoubtedly most important property of cost functions is convexity. Convexity is a basic notion from geometry which appears in most areas of mathematics (e.g., functional analysis, algebraic geometry, graph theory, crystallography, coding theory). It was formally defined for the first time by Archimedes of Syracuse (see [52]). However, the notion of convexity was used even before and it is impossible to say who considered it first (see [38]). Convex functions have a number of properties which are convenient in optimisation problems (e.g., every local minimum is a global minimum). Furthermore, convex cost functions arise in many problems from economics, engineering, computer science, and other sciences (see, e.g., [85]).

Definition 1.6.1. A set $S \subseteq \mathbb{Q}^n$ is said to be *convex* if for any two points x, $y \in S$ every point between them is still in S, i.e., for any $\lambda \in \mathbb{Q}$, $0 \leq \lambda \leq 1$, it holds $\lambda x + (1-\lambda)y \in S$.

Example 1.6.2. Halfspaces and hyperplanes are convex. The set of solutions to an atomic semilinear (i.e., PL) formula is a halfspace or a hyperplane, and therefore it is convex. △

Remark 1.6.3. It is easy to see that the intersection of convex sets is a convex set, and the projection of a convex set onto some of its coordinates is a convex set.

Example 1.6.4. Polyhedral sets are intersections of halfspaces; therefore, they are convex. △

Definition 1.6.5. A function $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ is said to be *convex* if for any two points x, $y \in \mathbb{Q}^n$ and for any $\lambda \in \mathbb{Q}$, $0 \leq \lambda \leq 1$, it holds

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

Remark 1.6.6. It is easy to see that a conic combination, i.e., a linear combination with nonnegative coefficients, of convex functions is a convex function. The minimum or infimum over some of the coordinates of a convex function is a convex function (see, e.g., [21], Section 3.2.5).

Definition 1.6.7. A valued structure Γ with domain $\mathbb{Q}$ is *convex* if all cost functions in Γ are convex. A (relational) structure $\mathfrak{A}$ with domain $\mathbb{Q}$ is *convex* if all relations in $\mathfrak{A}$ are convex.

Clearly, if Γ is a convex valued structure then the relational structure $\mathrm{Feas}(\Gamma)$ defined in Section 1.3 is a convex relational structure.

Proposition 1.6.8. *Let $D \subseteq \mathbb{Q}$ be a convex set. Let Γ be a convex valued structure with domain D. Then, for every $k \geq 2$, the valued structure Γ is improved by the fully symmetric fractional operation $\omega^{(k)}_{conv}\colon \mathcal{O}^{(k)}_D \to \mathbb{Q}_{\geq 0}$ (see Example 1.4.19) such that*

$$\omega^{(k)}_{conv}(g) = \begin{cases} 1 & \text{if } g = \mathrm{avg}^{(k)} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Let $f\colon D^n \to \mathbb{Q} \cup \{+\infty\}$ be a convex cost function. Jensen's Inequality (cf. [58]) implies that for all $k \geq 2$ and for all $x^1, \dots, x^k \in D^n$

$$f\left(\frac{1}{k}\sum_{i=1}^{k} x^i\right) \leq \frac{1}{k}\sum_{i=1}^{k} f(x^i).$$

Therefore, for every $k \geq 2$, the function f is improved by the fully symmetric fractional operation $\omega_{conv}^{(k)}$. □

We remark that for PL functions, there is no correlation between the notions of submodularity and convexity, as the following example shows.

Example 1.6.9. It is easy to check that:

- the function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by $f(x, y) := \max(ax, by)$ is submodular and convex, for all $a, b \in \mathbb{Q}_{>0}$;
- the function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by $f(x, y) := \min(ax, -by)$ is submodular and *not* convex, for all $a, b \in \mathbb{Q}_{>0}$;
- the function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by $f(x, y) := |x + y|$ is convex and *not* submodular;
- the function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by

  $$f(x, y) := \begin{cases} a & \text{if } (x \leq 0 \leq y) \text{ or } (y < 0 < x) \\ b & \text{otherwise} \end{cases}$$

  is *not* submodular and *not* convex, for all $a, b \in \mathbb{Q}_{\geq 0}$ with $a < b$. △

Observe that all the PL functions exhibited in Example 1.6.9 are also PLH.

1.7 Linear Programming

LINEAR PROGRAMMING, or *LP* for short, is an optimisation problem with a linear objective function and a set of linear constraints imposed upon a given set of underlying variables. The importance of LP in the valued constraint satisfaction framework relies not only on the fact that it is the most famous example of infinite-domain VCSPs (it can be actually modelled as a PL VCSP) but also on the fact that LP is a powerful tool for studying the computational complexity of VCSPs. *Linear programming relaxations*, i.e., relaxations to an LP instance, have been used to show the polynomial-time solvability of specific classes of finite-domain VCSPs and finite-domain (classical) CSPs (see, [69, 76]).

A linear program has the form

$$\text{minimise} \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \qquad \sum_{j=1}^{n} a_j^i x_j \leq b^i, \text{ for } i \in \{1, \dots, m\}.$$

This problem has n variables x_j (ranging over the rationals or over the real numbers) and m linear inequalities constraints. The coefficients c_j, a_j^i, and b^i are rational numbers, for all $j \in \{1, \dots, n\}$, and all $i \in \{1, \dots, m\}$.

The linear constraints, $\sum_{j=1}^{n} a_j^i x_j \leq b^i$, specify a polyhedral set (see Definition 1.2.12), namely the *feasibility polytope* over which the objective function has to be optimised. An LP instance is *inconsistent* if no feasible solution exists; in this case, the feasibility polytope is empty, and the LP instance is said to be *infeasible*. The feasibility polytope can be bounded or not, depending on whether so are all the linear constraints defining it. When the feasibility polytope is unbounded in the opposite direction to the gradient of the objective function (the gradient of the objective function is the vector of the coefficients of the objective function), then no optimal value is attained, and we say that the infimum of the objective function is $-\infty$.

An algorithm solving LP either finds a point in the feasibility polytope where the objective function has the smallest value if such a point exists, or it reports that the instance is infeasible (in this case, we assume that the output of the algorithm is $+\infty$), or it reports that the infimum of of the objective function is $-\infty$ (in this case, we assume that the output of the algorithm is $-\infty$). In the case the objective function has a minimum, it is attained in one of the extreme point solutions, i.e., in one of the vertices of the feasibility polytope. The reason for this is that the objective function is linear, and therefore, in particular, convex; and for a convex function defined on a convex closed domain, every local minimum is also a global minimum.

The LINEAR PROGRAM FEASIBILITY *problem*, *LPF*, is a decision problem having the form of a standard linear program but without any objective function to minimise. The output of an algorithm solving LPF is "no" or "yes", respectively, depending on whether the polyhedral set defined by the linear constraints is empty or not. LPF can be modelled as CSP(Δ), for a suitable semilinear relational structure Δ (see [5, 10]). Similarly, LINEAR PROGRAMMING can be modelled as a VCSP. For $n \in \mathbb{N}$, let

$$\mathbb{Q}[X_1, \dots, X_n] := \{a_0 + a_1 X_1 + \dots + a_n X_n \mid a_i \in \mathbb{Q} \text{ for } 0 \leq i \leq n\}$$

i.e., the set of linear polynomials in n variables with rational coefficients. For $n, k \in \mathbb{N}$, and for $p, q_1, \dots, q_k \in \mathbb{Q}[X_1, \dots, X_n]$ let

$$f^{(n)}_{p,q_1,\dots,q_k} \colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$$

be the function such that

$$f^{(n)}_{p,q_1,\dots,q_k}(x) = \begin{cases} p(x) & \text{if } \bigwedge_{i=1}^{k} (q_i(x) \leq 0) \\ +\infty & \text{otherwise.} \end{cases}$$

We define

$$\Gamma_{\mathrm{LP}} := \left\{ f^{(n)}_{p,q_1,\dots,q_k} \mid p, q_1, \dots, q_k \in \mathbb{Q}[X_1, \dots, X_n], \text{ and } n, k \in \mathbb{N} \right\}.$$

The valued language Γ_{LP} (with countably infinite signature) is piecewise linear and every instance of VCSP(Γ_{LP}) is easily seen to be a linear program, and vice versa (in particular, CSP(Feas(Γ_{LP})) =LPF).

In 1979, Khachian [65] showed that Linear Program Feasibility can be solved in polynomial time using the *Ellipsoid method*. This ground-breaking result was followed by several other polynomial-time algorithms addressing the optimisation version of LP, e.g., Karmarkar's interior point projective method [63], and barrier-function interior point methods (see,e.g., [100]). All the known polynomial-time algorithms solving LP rely on infinite approximation procedures.

1.8 VCSPs on Totally Ordered Commutative Rings

We already noticed that in the PL (and PLH) context the sets $\mathbb{Q}$ and $\mathbb{R}$ are interchangeable both when considered as the domain of the valued structure and when considered as the set of finite values taken by the cost functions. However, we fixed a representation of numeric coefficients (see Definition 1.2.14) that we use throughout the thesis whenever we deal with PL valued structures with infinite signatures. When fixing such a representation, we required the coefficients to be rational, because we need to computationally represent and manipulate them.

However, there are situations in which we would like to abstract from the details of number representation. For example, on some occasions it is convenient to extend the domain (and the interpretation) of a PL valued structure to a subset of a totally ordered commutative ring strictly containing $\mathbb{Q}$ (e.g., the set of the reals or, even a non-Archimedean real closed field), to get properties that are not shared with the original domain (e.g., completeness, closure, or boundedness). We point out that extending a PL valued structure to a subset of a totally ordered commutative ring strictly containing $\mathbb{Q}$ would affect the values taken by the cost functions, which are no longer values from $\mathbb{Q} \cup \{+\infty\}$.

Therefore, in such a situation we can adopt a more general definition of valued structure in which the cost functions are allowed to take values in a totally ordered commutative ring with unit. When working with this more general definition of valued structure, we can no longer assume the Turing (or bit) machine model. Instead, we adopt the *Blum-Chucker-Shub-Smale, BCSS, machine model* (cf. [2]). Such a machine operates on strings of symbols that represent elements of a totally ordered commutative ring (with unit), rather than on bits as in the classical Turing machine and algorithms have access to an oracle that performs a certain set of basic operations on them. We remark that in a BCSS machine there are no machine constants except 1 (the ring identity element).

An algorithm is *polynomial-time in the BCSS model* if it performs a number of fundamental ring operations which is bounded by a polynomial in the number of input integers. In general, a polynomial-time algorithm in the Turing model is not polynomial-time in the BCSS model, nor the vice versa holds (see [50] for a discussion of this topic).

Definition 1.8.1. An algorithm runs in *strongly polynomial time* if it is polynomial-time in the BCSS model and the space needed by the algorithm is polynomial in the length of the input.

Any strongly polynomial-time algorithm can be converted to a polynomial-time algorithm in the Turing model by replacing the fundamental ring operations by suitable algorithms performing these operations on a Turing machine. Therefore, a strongly polynomial-time algorithm is polynomial-time in both the BCSS and the Turing models. We remark that none of the known polynomial-time algorithms solving LP is strongly polynomial-time. A special class of strongly polynomial-time algorithms is the class of fully combinatorial algorithms.

Definition 1.8.2. Let R be a totally ordered commutative ring with unit. A problem over R can be solved in *fully combinatorial polynomial-time* if there exists a polynomial-time (uniform) machine on R in the sense of [2] (see Chapters 3-4) solving it by performing only additions and comparisons of elements in R as fundamental operations.

Chapter 2

Piecewise Linear Valued Structures

In this chapter, we show some simple initial results on PL valued structures. In Section 2.1, we show that the VCSP for the valued structure containing *all* convex PL cost functions is polynomial-time solvable. In Section 2.2, we show a class of PL valued structure for which solving the VCSP reduces to solve the corresponding feasibility problem. Finally, in Section 2.3, we introduce a sufficient condition for a PL VCSP to be NP-hard[1].

2.1 Convex PL Valued Structures

To study the computational complexity of the VCSP for convex PL valued structures (see Section 1.6) we adopt the minimisation formulation of the problem, that is, we assume that the input is given as a finite set of variables and a sum of PL cost functions (the objective function) applied to the given finite set of variables. The cost functions are represented as in Definition 1.2.14. For the class of convex PL valued structures, there is a polynomial-time algorithm that finds the infimum (if it exists) and decides whether it is attained. The polynomial-time solvability of the decision version of the problem, where a threshold is given as part of the instance, is an immediate consequence (see Remark 1.1.4).

It is well known that the minimisation of a convex PL cost function can be modelled as a linear program (see, e.g., [21]). We use this idea to deal with the minimisation of sums of convex PL cost functions whose domains are given as unions of finitely many polyhedral sets each specified by a conjunction of strict or weak linear inequalities.

[1]The problem is NP-hard in the minimisation formulation. It is NP-complete in the threshold formulation.

Lemma 2.1.1 ([88], Lemma 2.50)**.** *Let $C_1, \dots, C_r$ be a finite collection of convex and polyhedral sets in $\mathbb{Q}^n$. If the set $C = \bigcup_{k=1}^r C_k \subseteq \mathbb{Q}^n$ is convex then it is a polyhedral set.*

It follows that for a convex PL function the feasibility region is not only a union of polyhedral sets but it is itself a polyhedral set.

Theorem 2.1.2 ([88], Theorem 2.49)**.** *Let $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ be a function. Then the following are equivalent:*

1. *f is convex and PL;*

2. *for every $x \in \mathbb{Q}^n$,*

$$f(x) = \begin{cases} \max(l_1(x), \dots, l_p(x)) & \text{if } x \in D \\ +\infty & \text{if } x \notin D\ , \end{cases}$$

where $D = \mathrm{dom}(f)$ is a polyhedral set and $l_i\colon \mathbb{Q}^n \to \mathbb{Q}$ are linear functions, for $1 \leq i \leq p$.

From Theorem 2.1.2 it follows that a convex PL cost function is continuous in its domain.

Lemma 2.1.3. *Let us consider f_1, f_2, …, $f_m\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$, such that*

$$f_i(x) = \begin{cases} \max(a_{i,1}^T x + b_{i,1}, \dots, a_{i,k_i}^T x + b_{i,k_i}) & \text{if } x \in D_i \\ +\infty & \text{otherwise,} \end{cases}$$

where D_i is a closed polyhedral subset of $\mathbb{Q}^n$, the coefficients $a_{i,j}^T \in \mathbb{Q}^n$, and $b_{i,j} \in \mathbb{Q}$, for $1 \leq j \leq k_i$, and for $1 \leq i \leq m$. Let us define the following linear program J with rational variables t_1, t_2,…, t_m, t, $x_1, \dots, x_n$

$$\begin{array}{lll} & LP(J) := \min t & \\ \text{subject to} & t \ = t_1 + \cdots + t_m & \\ & t_1 \geq a_{1,j}^T x + b_{1,j} & \text{for } x \in D_1, \text{ for } 1 \leq j \leq k_1, \\ & \vdots & \vdots \\ & t_m \geq a_{m,j}^T x + b_{m,j} & \text{for } x \in D_m, \text{ for } 1 \leq j \leq k_m, \end{array}$$

where $x := (x_1, \dots, x_n)$. If there exists $x \in D = \bigcap_{i=1}^m D_i$, then

$$\min_{x \in \mathbb{Q}^n} (f_1(x) + f_2(x) + \cdots + f_m(x)) = \mathrm{LP}(J).$$

Proof. For every $i \in \{1, \dots, m\}$ and every $x \in D$ let us define the following set:

$$T_i(x) = \{t_i \in \mathbb{Q} \mid t_i \geq a_{i,j}^T x + b_{i,j}, \text{ , for all } j \in \{1, \dots, k_i\}\}$$

and let

$$T(x) = \{t = t_1 + t_2 + \dots + t_m \mid t_i \in T_i(x), \ i = 1, \dots, m\}.$$

Observe that for $1 \leq i \leq m$ and for every $x_0 \in D^n$ it holds that $t_i \geq f_i(x_0)$, for all $t_i \in T_i(x_0)$. It follows that $t \geq f_1(x_0) + f_2(x_0) + \dots + f_m(x_0)$, for all $t \in T(x_0)$. Let us observe, also, that for every $x_0 \in D^n$,

$$\min_{t \in \mathbb{Q}} T(x_0) = f_1(x_0) + f_2(x_0) + \dots + f_m(x_0).$$

Then

$$\mathrm{LP}(J) = \min_{x \in D}(\min_{t \in \mathbb{Q}} T(x)) = \min_{x \in D}(f_1(x) + f_2(x) + \dots + f_m(x)). \qquad \square$$

Theorem 2.1.4. *Let Γ be a PL valued structure (with a signature of arbitrary cardinality) such that every cost function in Γ is convex. Then* $\mathrm{VCSP}(\Gamma)$ *is polynomial-time solvable.*

Proof. Let τ be the signature of Γ. An instance of $\mathrm{VCSP}(\Gamma)$ consists of a set of variables $x_1, \dots, x_d$, and an objective function of the form

$$\phi(x) = f_1(x) + \dots + f_m(x),$$

with $f_1, \dots, f_m \in \tau$, where $x = (x_1, \dots, x_d)$. Every $f_i^\Gamma \colon \mathbb{Q}^{\mathrm{ar}(f_i)} \to \mathbb{Q} \cup \{+\infty\}$ has the form

$$f_i^\Gamma(x) = \begin{cases} a_{i1}^T x + b_{i1} & \text{if } x \in D_{i1} \\ \vdots & \vdots \\ a_{ik_i}^T x + b_{ik_i} & \text{if } x \in D_{ik_i} \\ +\infty & \text{otherwise} \end{cases},$$

for some $a_{i1}^T, \dots, a_{ik_i}^T \in \mathbb{Q}^{\mathrm{ar}(f)}$, $b_{i1}, \dots, b_{ik_i} \in \mathbb{Q}$ and with D_{ij} polyhedral sets such that the interior of D_{ij} is non-empty and for every j and $D_{ij} \cap D_{ij} = \emptyset$ for every $j \neq j'$.

Using Theorem 2.1.2 we can rewrite every f_i^Γ as

$$f_i(x)^\Gamma = \begin{cases} \max(a_{i1}^T x + b_{i1}, \dots, a_{ik_i}^T x + b_{ik_i}) & \text{if } x \in D_i = \bigcup_{j=1}^{k_i} D_{ij} \\ +\infty & \text{otherwise,} \end{cases}$$

(see also [21], Exercise 3.29).

For every $i \in \{1, \dots, m\}$, the union D_i is convex, therefore it is a polyhedral set (Lemma 2.1.1) and it is specified by a finite conjunction of linear constraints of this form

$$\bigwedge_{j=1}^{p_i}(l_{i,j}(x) \leq 0) \wedge \bigwedge_{j=p_i+1}^{q_i}(l_{i,j}(x) < 0) \wedge \bigwedge_{j=q_i+1}^{r_i}(l_{i,j}(x) = 0)$$

where $l_{i,j}$ is a linear polynomial with rational coefficients for $1 \leq j \leq r_i$. The list of linear constraints whose conjunction defines D_i can be found in polynomial time by solving a polynomial number of LPF instances (see Section 1.7): let L be the list of all linear constraints defining some D_{ij}, for $1 \leq j \leq k_i$. For every $l \in L$, and for every $j \in \{1, \ldots, k_i\}$, if the LPF instance defined by the linear constraints defining D_{ij} and l is not feasible, then we remove l from L.

Observe that this procedure is correct. In fact, since the D_{ij}s are pairwise disjoint (see Definition 1.2.14), if for $l \in L$ there exists some D_{ij} that does not intersect the halfspace H defined by l, then H does not contain the union D_i and we have to remove l from the list L. After this step L contains the linear constraints defining the smallest polyhedral set containing D_i, which coincides with D_i

By Lemma 2.1.3 we can decide the feasibility of ϕ and find its infimum by solving the following instance I of Linear Programming:

$$\begin{array}{lll}
 & \min t & \\
\text{subject to} & t = t_1 + \cdots + t_m & \\
 & t_i \geq a_{i,j}^T x + b_{i,j} & \text{for } 1 \geq j \leq k_i,\ 1 \leq i \leq m \\
 & l_{i,j}(x) \leq 0 & \text{for } 1 \leq j \leq p_i,\ 1 \leq i \leq m \\
 & l_{i,j}(x) \leq 0 & \text{for } p_i + 1 \leq j \leq q_i,\ 1 \leq i \leq m \\
 & l_{i,j}(x) = 0 & \text{for } q_i + 1 \leq j \leq r_i,\ 1 \leq i \leq m
\end{array}$$

(we substituted strict inequalities by their corresponding weak inequality), in this way we can find the minimum of our objective function in the closure of the feasibility region.

To decide whether the found infimum, $\min t$, is a proper minimum (in the case it exists and is not $-\infty$) we have to check whether there exists any point $y \in D$ such that $\min t = \phi(y)$. To do this we solve the following LPF instance with variables $y = (y_1, \ldots, y_d), z = (z_1, \ldots, z_m)$, which contains strict and weak linear inequalities:

$$\begin{aligned}
&z_1 + \cdots + z_m = \min t && \\
&z_i \geq a_{i,j}^T y + b_{i,j} && \text{for } 1 \geq j \leq k_i,\ 1 \leq i \leq m \\
&l_{i,j}(y) \leq 0 && \text{for } 1 \leq j \leq p_i,\ 1 \leq i \leq m \\
&l_{i,j}(y) < 0 && \text{for } p_i + 1 \leq j \leq q_i,\ 1 \leq i \leq m \\
&l_{i,j}(x) = 0 && \text{for } q_i + 1 \leq j \leq r_i,\ 1 \leq i \leq m.
\end{aligned}$$

Observe that LPF for a finite set of strict or weak linear inequalities can be solved in polynomial time (see Proposition 8.0.3). If the LPF instance above has a solution then $\min t$ is a proper minimum, otherwise $\min t$ is an infimum that is not attained in D. □

2.2 Median-Improved PL Valued Structures

In this section, we present a class of valued structures with rational domains that are *essentially crisp*, i.e., valued structures Γ with domain $\mathbb{Q}$ such that any solution to the problem CSP(Feas(Γ)) is a solution to VCSP(Γ). As a consequence of this fact, we obtain that the VCSP for a PL valued structure with a finite signature that is improved by a ternary fractional operation whose support contains only the *median* operation is polynomial-time solvable.

Definition 2.2.1. Let g be a ternary operation on $\mathbb{Q}$. We say that g is a *majority operation* if, for all x, $y \in \mathbb{Q}$, it holds that

$$g(x, x, y) = g(x, y, x) = g(y, x, x) = x.$$

Similarly, we say that g is a *minority operation* if for all x, $y \in \mathbb{Q}$, it holds that $g(x, y, y) = g(y, x, y) = g(y, y, x) = x$.

Example 2.2.2. The ternary operation *median*, $\text{med}\colon \mathbb{Q}^3 \to \mathbb{Q}$, is defined by

$$\begin{aligned}
\text{med}(x, y, z) &:= \max(\min(x, y), \min(y, z), \min(z, x)) \\
&= \min(\max(x, y), \max(y, z), \max(z, x)),
\end{aligned}$$

is is a majority operation. △

Proposition 2.2.3. *If a valued structure Γ with domain $\mathbb{Q}$ is improved by a fractional operation ω_{maj} whose support contains a single majority operation g, then every solution to* CSP(Feas(Γ)) *is a solution to* VCSP(Γ).

The proof of Proposition 2.2.3 directly follows from the next lemma, and was given in the finite-domain case in [34][2].

Lemma 2.2.4. *A function* $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ *is improved by a fractional operation* ω_{maj} *whose support contains a single majority operation* g *if, and only if,* f *is constant, i.e., there exists* $c \in \mathbb{Q}$ *such that* $f(x) = c$ *for every* $x \in \mathbb{Q}^n$*.*

Proof. By definition, ω_{maj} improves f if for every $x, y, z \in \mathbb{Q}^n$

$$f(g(x, y, z)) \leq \frac{1}{3}\left(f(x) + f(y) + f(z)\right). \tag{2.1}$$

We prove our statement by induction on the arity n of f. Obviously, if f is a constant function Inequality (2.1) is satisfied. Let $n = 1$ and let $x, y \in \mathbb{Q}$ be such that $x < y$. Since f is improved by ω_{maj} and $g(x, y, y) = y$, we get $3f(y) \leq f(x) + 2f(y)$, i.e. $f(y) \leq f(x)$. Similarly, since $g(x, x, y) = x$, we also get $f(x) \leq f(y)$. Therefore, f is a constant function.

Let $n \geq 2$, let us assume that every function of arity at most $n - 1$ that is improved by ω_{maj} is constant, and let f be an n-ary function improved by ω_{maj}. Let $x = (x_1, \ldots, x_{n-1}, x_n)$, $y = (y_1, \ldots, y_{n-1}, y_n) \in \mathbb{Q}^n$ be such that $x \neq y$. Since ω_{maj} is idempotent, it improves the unary function $f(y_1, \ldots, y_{n-1}, \cdot)\colon \mathbb{Q} \to \mathbb{Q} \cup \{+\infty\}$ (see Remark 1.4.15), and therefore $f(y_1, \ldots, y_{n-1}, x_n) = f(y_1, \ldots, y_{n-1}, y_n)$. Furthermore, the $(n-1)$-ary function $f(\cdot, x_n)\colon \mathbb{Q}^{n-1} \to \mathbb{Q} \cup \{+\infty\}$ is improved by ω_{maj}, then by the inductive hypothesis we obtain

$$f(x_1, \ldots, x_{n-1}, x_n) = f(y_1, \ldots, y_{n-1}, x_n) = f(y_1, \ldots, y_{n-1}, y_n).$$

Because of the arbitrary choice of x and y, the function f is constant. $\square$

Using a similar proof as for Lemma 2.2.4, it is easy to prove that if a PL valued structure Γ is improved by a fractional operation whose support contains a single minority operation g, then VCSP(Γ) is equivalent to CSP(Feas(Γ)). An easy consequence of Proposition 2.2.3 is the polynomial-time solvability of the VCSP for *median-improved* PL valued structures.

Definition 2.2.5. We define the fractional operation $\omega_{\text{med}}\colon \mathcal{O}_{\mathbb{Q}}^{(3)} \to \mathbb{Q}_{\geq 0}$ by setting

$$\omega_{\text{med}}(g) := \begin{cases} 1 & \text{if } g = \text{med} \\ 0 & \text{otherwise.} \end{cases}$$

We say that a valued structure Γ is *median-improved* if it is improved by the fractional operation ω_{med}.

[2] In [34], the authors use the notion of multimorphism rather than the notion of fractional polymorphism.

From Proposition 2.2.3, it trivially follows that given a PL valued structure Γ improved by ω_{med}, solving VCSP(Γ) is equivalent to solve the associated feasibility problem, i.e., to solve CSP(Feas(Γ)). In this case, the relational structure Feas(Γ) is preserved by the operation median, and therefore it can be solved in polynomial time.

Theorem 2.2.6 ([11], Corollary 5.5)**.** *Let $\mathfrak{A}$ be a semilinear relational structure with a finite signature. If the operation median preserves $\mathfrak{A}$ then* CSP($\mathfrak{A}$) *can be solved in polynomial time.*

As an immediate consequence of the previous theorem, we obtain the following result.

Corollary 2.2.7. *Let Γ be a PL valued structure with a finite signature that is improved by the fractional operation ω_{med}. Then* VCSP(Γ) *can be solved in polynomial time.*

2.3 A Family of NP-hard PL VCSPs

In this section, we generalise to the case of PL valued structures the following sufficient condition for finite-domain valued structures to have an NP-hard VCSP.

Proposition 2.3.1 ([34], Proposition 5.1)**.** *Let Γ be a valued structure with a finite domain D and costs in $\mathbb{Q} \cup \{+\infty\}$ (or $\mathbb{R} \cup \{+\infty\}$). If there exist $a, b \in D$, and there exist $\alpha, \beta \in \mathbb{Q}$, with $\alpha < \beta$, such that the cost function $f \colon D^2 \to \mathbb{Q} \cup \{+\infty\}$ by*

$$f(x,y) := \begin{cases} \alpha & \text{if } x \neq y \wedge x, y \in \{a,b\} \\ \beta & \text{if } x = y \wedge x, y \in \{a,b\} \\ +\infty & \text{otherwise,} \end{cases}$$

is in Γ, then VCSP(Γ) *is NP-hard.*

We fix the following notation

$$\begin{aligned}]a,b[&:= \{x \in \mathbb{Q} \mid a < x < b\} \\ [a,b[&:= \{x \in \mathbb{Q} \mid a \leq x < b\} \\ ]a,b] &:= \{x \in \mathbb{Q} \mid a < x \leq b\} \\ [a,b] &:= \{x \in \mathbb{Q} \mid a \leq x \leq b\} \end{aligned}$$

(a,b) denotes a set from $\{]a,b[, [a,b[,]a,b], [a,b]\}$.

Proposition 2.3.2. *Let $(a, b), (c, d)$ be two disjoint subsets of $\subset \mathbb{Q}$ (where a and d, respectively, can be the symbol $-\infty$ and $+\infty$, respectively), and $\alpha, \beta \in \mathbb{Q}$, with $\alpha < \beta$. Let Γ be a PL valued structure containing the cost function $f^{\alpha,\beta}_{(a,b),(c,d)} \colon D^2 \to \mathbb{Q} \cup \{+\infty\}$ defined by*

$$f(x, y)^{\alpha,\beta}_{(a,b),(c,d)} := \begin{cases} \alpha & \text{if } (x, y) \in ((a, b) \times (c, d)) \cup ((c, d) \times (a, b)) \\ \beta & \text{if } (x, y) \in (a, b)^2 \cup (c, d)^2 \\ +\infty & \text{otherwise.} \end{cases}$$

Then VCSP(Γ) *is NP-hard.*

Proof. We provide a polynomial-time many-one reduction from Max Cut (see Example 1) to VCSP(Γ). Let $I := (V, \phi, u)$ be an instance of Max Cut, i.e., an instance of VCSP(Γ_{XOR}), with set of variables $V := \{v_1, \ldots, v_d\}$, objective function

$$\phi(v_1, \ldots, v_d) := \sum_{j=1}^{m} \text{XOR}(v_{j_1}, v_{j_2}),$$

and threshold $u \in \mathbb{Q}$ (such that $0 \leq u \leq m$). Our reduction maps I to the instance $I' := (V, \psi, u')$ of VCSP(Γ) defined by setting

$$\psi(v_1, \ldots, v_d) := \sum_{j=1}^{m} f(v_{j_1}, v_{j_2}),$$

and $u' := m\alpha + u(\beta - \alpha) \in \mathbb{Q}$. We claim that there exists an assignment $s \colon V \to \{0, 1\}$ with cost $\phi^{\Gamma_{\text{XOR}}}(s(v_1), \ldots, s(v_d)) \leq u$ if, and only if, there exists an assignment $h \colon V \to \mathbb{Q}$ with cost $\psi^{\Gamma}(h(v_1), \ldots, h(v_d)) \leq u'$. Let $s \colon V \to \{0, 1\}$ be an assignment with cost $\phi^{\Gamma_{\text{XOR}}}(s(v_1), \ldots, s(v_d)) \leq u$, we define the assignment $h \colon V \to \mathbb{Q}$ by

$$h(v) := \begin{cases} \frac{a+b}{2} & \text{if } s(v) = 0 \\ \frac{c+d}{2} & \text{if } s(v) = 1. \end{cases}$$

For every $j \in \{1, \leq, m\}$ it holds that:

- if $\text{XOR}^{\Gamma_{\text{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 0$, then $s(v_{j_1}) \neq s(v_{j_2})$. Therefore, the couple $(h(v_{j_1}), h(v_{j_2}))$ is equal to either $\left(\frac{a+b}{2}, \frac{c+d}{2}\right)$, or $\left(\frac{c+d}{2}, \frac{a+b}{2}\right)$. In both cases $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \alpha$.

- If $\text{XOR}^{\Gamma_{\text{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 1$, then $s(v_{j_1}) = s(v_{j_2})$. Therefore, the couple $(h(v_{j_1}), h(v_{j_2}))$ is equal to either $\left(\frac{a+b}{2}, \frac{a+b}{2}\right)$, or $\left(\frac{c+d}{2}, \frac{c+d}{2}\right)$. In both cases $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \beta$.

Since $\phi^{\Gamma_{\mathrm{XOR}}}(s(v_1), \dots, s(v_d)) \leq u$, there are at most u indices $j \in \{1, \dots, m\}$ for which $\mathrm{XOR}^{\Gamma_{\mathrm{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 1$, and there are at least $m - u$ indices $j \in \{1, \dots, m\}$ for which $\mathrm{XOR}^{\Gamma_{\mathrm{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 0$. Consequently, there are at most u indices $j \in \{1, \dots, m\}$ for which $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \beta$, and at least $m - u$ indices $j \in \{1, \dots, m\}$ for which $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \alpha$. It follows that

$$\psi^{\Gamma}(h(v_1), \dots, h(v_d)) \leq (m-u)\alpha + u\beta = u'.$$

Vice versa, let $h \colon V \to \mathbb{Q}$ be an assignment for the variables in V with cost $\psi^{\Gamma}(h(v_1), \dots, h(v_d)) \leq u'$. Observe that, since $u' \in \mathbb{Q}$, every occurrence of $f^{\Gamma}(h(v_{j_1}), h(v_{j_2}))$ in the sum $\psi^{\Gamma}(h(v_1), \dots, h(v_d))$ has a finite value, that is, $h(v) \in (a, b) \cup (c, d)$ for every $v \in V$. We define the assignment $s \colon V \to \{0, 1\}$ by

$$s(v) := \begin{cases} 0 & \text{if } h(v) \in (a,b) \\ 1 & \text{if } h(v) \in (c,d). \end{cases}$$

For every $j \in \{1, \leq, m\}$ it holds that:

- if $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \alpha$, then $(h(v_{j_1}), h(v_{j_2}))$ belongs to the subset $((a, b) \times (c, d)) \cup ((c, d) \times (a, b))$. Therefore, $s(v_{j_1}) \neq s(v_{j_2})$, which implies $\mathrm{XOR}^{\Gamma_{\mathrm{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 0$.
- if $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \beta$, then $(h(v_{j_1}), h(v_{j_2})) \in (a, b)^2 \cup (c, d)^2$. Therefore, $s(v_{j_1}) = s(v_{j_2})$, and consequently $\mathrm{XOR}^{\Gamma_{\mathrm{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 0$.

Since $\psi^{\Gamma}(h(v_1), \dots, h(v_d)) \leq u'$ and $u' = (m - u)\alpha + u\beta$, there are at most u indices $j \in \{1, \dots, m\}$ for which $f^{\Gamma}(h(v_{j_1}), h(v_{j_2})) = \beta$. Consequently, there are at most u indices $j \in \{1, \dots, m\}$ for which $\mathrm{XOR}^{\Gamma_{\mathrm{XOR}}}(s(v_{j_1}), s(v_{j_2})) = 1$, which implies that

$$\phi^{\Gamma_{\mathrm{XOR}}}(s(v_1), \dots, s(v_d)) \leq u.$$

□

Proposition 2.3.2 gives rise to a family of PL valued structures whose VCSP is NP-hard. In Corollary 7.1.13, we combine Proposition 2.3.2 and the notion of *expressibility*. We observe that the PL cost functions $f^{\alpha,\beta}_{(a,b),(c,d)}$ defined in Proposition 2.3.2 are, in particular, PLH.

Summary and Outlook

We have seen some simple yet interesting results for PL valued structures. Some of these are generalisations of known results from the theory of finite-domain VCSPs. In the next chapter, we focus on the class of PLH valued structures whose investigation is a natural intermediate step between the study of finite-domain valued structures and the study of PL valued structures.

Chapter 3

Piecewise Linear Homogeneous Valued Structures

In this chapter, we focus on PLH valued structures. PLH valued structures form a subclass of the class of PL valued structures; however, the PLH setting is still very expressive: every finite-domain VCSP is equivalent to the VCSP for a suitable PLH valued structure. PLH valued structures have many mathematical properties (that are not shared, in general, with other PL valued structures) that make the study of VCSPs for PLH valued structures a natural intermediate step between finite-domain VCSPs and VCSPs for PL valued structures.

We prove that the VCSP for a PLH valued structure with a finite signature is polynomial-time many-one equivalent to the VCSP for a valued structure over a suitable finite domain. We present this technique in two steps: firstly we show the reduction for the feasibility problem, i.e., we prove such a reduction for PLH relational structure; secondly, we extend this method to solve the *optimisation problem*, i.e., to find a solution of cost at most the given input threshold.

We use the following result for PLH valued structures from Section 1.2.2.

Theorem 1.2.8. *The structure $\mathfrak{L}$ from Example 1.2.3 has quantifier elimination.*

3.1 Efficient Sampling for PLH Relational Structures

We present an *efficient sampling algorithm* for PLH relational structures. Let us formally introduce the notion of a sampling algorithm for a relational structure.

Definition 3.1.1. Let $\mathfrak{C}$ be a structure with a finite relational signature τ. A *sampling algorithm for* $\mathfrak{C}$ takes as input a positive integer d and computes a finite-domain structure $\mathfrak{D}$ homomorphic to $\mathfrak{C}$ such that every finite conjunction of atomic τ-formulas having at most d distinct free variables is satisfiable in $\mathfrak{C}$ if, and only if, it is satisfiable in $\mathfrak{D}$. A sampling algorithm is called *efficient* if its running time is bounded by a polynomial in d. We refer to the output of a sampling algorithm by calling it the *sample*.

The definition above is a slight reformulation of Definition 2.2 from [8] and it is easily seen to give the same results using the same proofs. We decided to bound the number of variables instead of the size of the conjunction of atomic τ-formulas because this is more natural in our context. These two quantities are polynomially related by the assumption that the signature τ is finite.

Throughout this section, we refer to a fixed PLH relational structure $\mathfrak{A}$ with a finite signature τ. We give a formal definition of the *numerical data* in $\mathfrak{A}$; we need it later on. By quantifier elimination (Theorem 1.2.8), each of the finitely many relations $R^{\mathfrak{A}}$ for $R \in \tau$ has a quantifier-free τ_0-formula ϕ_R over $\mathfrak{L}$. As in the proof of Theorem 1.2.8, we can assume that all formulas ϕ_R are positive (that is, they contain no negations). We fix one such representation that we use from now on. Let $\mathrm{At}(\phi_R)$ denote the set of atomic subformulas of ϕ_R. Each atomic τ_0-formula is of the form $t_1 \overset{<}{=} t_2$, where t_1 and t_2 are terms. We call an atomic formula *trivial* if it is equivalent to $\perp$ or $\top$, and *non-trivial* otherwise. As in the proof of Theorem 1.2.8, we make the assumption that atomic formulas are of the form $\perp$ or $\top$ if they are trivial and, otherwise, of the form either $c_1 \cdot 1 \overset{<}{=} x_i$ or $x_i \overset{<}{=} c_2 \cdot 1$ or $c_1 \cdot x_i \overset{<}{=} c_2 \cdot x_j$, with constants c_1 and c_2 not both negative and where function symbols $c_i\cdot$ are never composed. This assumption can be made without loss of generality (again, see the proof of Theorem 1.2.8).

Given a set of non-trivial atomic formulas Φ, we define

$$H(\Phi) = \left\{ \frac{c_1}{c_2} \;\middle|\; t_1 = c_1 \cdot x_i,\ t_2 = c_2 \cdot x_j,\ \text{for some } t_1 \overset{<}{=} t_2 \text{ in } \Phi \right\},$$

$$\begin{aligned} K(\Phi) = & \left\{ \frac{c_2}{c_1} \;\middle|\; t_1 = c_1 \cdot x_i,\ t_2 = c_2 \cdot 1,\ \text{for some } t_1 \overset{<}{=} t_2 \text{ in } \Phi \right\} \\ & \cup \left\{ \frac{c_1}{c_2} \;\middle|\; t_1 = c_1 \cdot 1,\ t_2 = c_2 \cdot x_j,\ \text{for some } t_1 \overset{<}{=} t_2 \text{ in } \Phi \right\} \cup \{1\}. \end{aligned}$$

The efficient sampling algorithm for $\mathfrak{A}$ works in two steps. First, the problem $\mathrm{CSP}(\mathfrak{A})$ is transferred to the equivalent CSP for a suitable structure

$\mathfrak{A}^\star$ that is an extension of an expansion (or an expansion of an extension) of $\mathfrak{A}$; second, we provide an efficient sampling algorithm for $\mathfrak{A}^\star$, which is also an efficient sampling algorithm for $\mathfrak{A}$.

Definition 3.1.2. The ordered $\mathbb{Q}$-vector space $\mathbb{Q}^\star$ is defined as

$$\mathbb{Q}^\star = \{x + y\boldsymbol{\epsilon} \mid x, y \in \mathbb{Q}\}$$

where $\boldsymbol{\epsilon}$ is merely a formal device, namely $x + y\boldsymbol{\epsilon}$ represents the pair (x, y). We define addition and multiplication by a scalar componentwise

$$\begin{gathered}(x_1 + y_1\boldsymbol{\epsilon}) \ + \ (x_2 + y_2\boldsymbol{\epsilon}) = (x_1 + x_2) + (y_1 + y_2)\boldsymbol{\epsilon} \\ c \cdot (x + y\boldsymbol{\epsilon}) = (cx) + (cy)\boldsymbol{\epsilon}.\end{gathered}$$

Clearly, $\mathbb{Q}$ is embedded in $\mathbb{Q}^\star$, by mapping every rational number x into $x + 0\boldsymbol{\epsilon}$. The order is induced by $\mathbb{Q}$ extended with $0 < \boldsymbol{\epsilon} \ll 1$, namely the lexicographical order of the components x and y

$$(x_1 + y_1\boldsymbol{\epsilon}) \ < \ (x_2 + y_2\boldsymbol{\epsilon}) \quad \text{iff} \quad \begin{cases} x_1 < x_2 \quad \text{or} \\ x_1 = x_2 \ \wedge \ y_1 < y_2. \end{cases}$$

Any τ_0-formula has an obvious interpretation in any ordered $\mathbb{Q}$-vector space Q extending $\mathbb{Q}$, and, in particular, in $\mathbb{Q}^\star$.

Theorem 3.1.3. *The first-order theory of ordered $\mathbb{Q}$-vector spaces in the signature $\tau_0 \cup \{+, -\}$ is complete*

Proof. The proof follows from [99, Chapter 1, Remark 7.9]. □

Proposition 3.1.4. *Let $\phi(x_1, \dots, x_d)$ and $\psi(x_1, \dots, x_d)$ be τ_0-formulas. Then ϕ and ψ are equivalent in $\mathbb{Q}$ if, and only if, they are equivalent in any ordered $\mathbb{Q}$-vector space Q extending $\mathbb{Q}$ (for instance $Q = \mathbb{Q}^\star$).*

Proof. By Theorem 3.1.3, the τ_0-sentence

$$\forall x_1, \dots, x_d \phi(x_1, \dots, x_d) \Longleftrightarrow \psi(x_1, \dots, x_d)$$

holds in $\mathbb{Q}$ if, and only if, it holds in Q. □

Proposition 3.1.4 gives us a natural extension $\mathfrak{A}^\star$ of $\mathfrak{A}$ to the domain $\mathbb{Q}^\star$, that is, the τ-structure obtained by interpreting each relation symbol $R \in \tau$ by the relation $R^{\mathfrak{A}^\star}$ defined on $\mathbb{Q}^\star$ by the same (quantifier-free) τ_0-formula ϕ_R that defines $R^{\mathfrak{A}}$ over $\mathbb{Q}$ (by the proposition, the choice of equivalent τ_0-formulas is immaterial).

In the following corollary, we see that there is no difference between $\mathfrak{A}$ and $\mathfrak{A}^\star$ as long as feasibility is concerned.

Corollary 3.1.5. *Let ϕ be an instance of* $\mathrm{CSP}(\mathfrak{A})$*, and let $\phi^\star$ be the corresponding instance of* $\mathrm{CSP}(\mathfrak{A}^\star)$*. Then ϕ is satisfiable if and only if $\phi^\star$ is.*

Proof. The proof immediately follows from Proposition 3.1.4 by observing that ϕ (respectively $\phi^\star$) is unsatisfiable if and only if it is equivalent to $\perp$. □

As a consequence of Corollary 3.1.5, we can work in the extended structure $\mathfrak{A}^\star$. Our goal is to prove the following theorem.

Theorem 3.1.6. *There is an efficient sampling algorithm for $\mathfrak{A}^\star$.*

Before giving the proof of Theorem 3.1.6, we present some preliminary lemmas in which we explicitly define the domain of the wanted sample.

Let ϕ be an atomic τ_0-formula. We write $\bar{\phi}$ for the formula $t_1 \leq t_2$ if ϕ is of the form $t_1 < t_2$, and for the formula $t_1 = t_2$ if ϕ is of the form $t_1 = t_2$. We call $\bar{\phi}$ the *closure* of the formula ϕ. First, we investigate the positive solutions to the closures of finitely many atomic τ_0-formulas. Then, in a second step that builds on the first one, we investigate the solutions to finitely many atomic τ_0-formulas.

Lemma 3.1.7. *Let Φ be a finite set of atomic τ_0-formulas having free variables in $\{v_1, \ldots, v_d\}$. Assume that $\bar{\Phi} := \bigcup_{\phi \in \Phi} \bar{\phi}$ has a simultaneous solution $(x_1, \ldots, x_d) \in \mathbb{Q}_{>0}$ in positive numbers. Then $\bar{\Phi}$ has a solution taking values in the set $C_{\Phi,d} \subset \mathbb{Q}$ defined as follows*

$$C_{\Phi,d} = \left\{ |k| \prod_{i=1}^{s} |h_i|^{e_i} \;\middle|\; k \in K(\Phi),\ e_1, \ldots, e_s \in \mathbb{Z},\ \sum_{r=1}^{s} |e_r| < d \right\}$$

where $h_1, \ldots, h_s$ is an enumeration of the (finitely many) elements of $H(\Phi)$.

Proof. Let $\gamma \leq \beta$ be maximal such that there are Ψ_1, Ψ_2, Ψ_3 with

$$\begin{aligned}
\bar{\Phi} \ &= \{s_1 = s'_1, \ldots, s_\alpha = s'_\alpha\} \cup \{t_1 \leq t'_1, \ldots, t_\beta \leq t'_\beta\} \\
\Psi_1 &= \{s_1 = s'_1, \ldots, s_\alpha = s'_\alpha\} \\
\Psi_2 &= \{t_1 = t'_1, \ldots, t_\gamma = t'_\gamma\} \\
\Psi_3 &= \{t_{\gamma+1} \leq t'_{\gamma+1}, \ldots, t_\beta \leq t'_\beta\},
\end{aligned}$$

where s_i, s'_i, t_j, t'_j are τ_0-terms for all i, j, and $\Psi_1 \cup \Psi_2 \cup \Psi_3$ is satisfiable in positive numbers. Clearly, the space of positive solutions of $\Psi_1 \cup \Psi_2$ must be contained in the space of positive solutions of Ψ_3. In fact, by construction, they intersect: consider any straight line segment connecting a solution of $\Psi_1 \cup \Psi_2 \cup \Psi_3$ and a solution to $\Psi_1 \cup \Psi_2$ not satisfying Ψ_3. On this segment, there must be a solution of $\Psi_1 \cup \Psi_2 \cup \Psi_3$ lying on the boundary of one of the inequalities of Ψ_3, contradicting the maximality of γ. By the last observation, it suffices to prove that there is a solution of $\Psi_1 \cup \Psi_2$ taking values in $C_{\Phi,d}$. Put an edge between two variables x_i and x_j when they appear in the same formula of $\Psi_1 \cup \Psi_2$. For each connected component of the graph thus defined, either it contains at least one variable v_i such that there is a constraint of the form $h \cdot v_i = k \cdot 1$, or all constraints are of the form $h \cdot v_i = h' \cdot v_j$. In the first case, assign $v_i = \frac{k}{h}$; in the second case, assign one of the variables v_i arbitrarily to 1. Then, in any case, since the diameter of the connected component is smaller than d, all the variables in this connected component are forced to take values in $C_{\Phi,d}$, by simple propagation of v_i. $\square$

Lemma 3.1.8. *Let Φ be a finite set of atomic τ_0-formulas having free variables in the set $\{v_1, \dots, v_d\}$. Assume that the formulas in Φ are simultaneously satisfiable in $\mathbb{Q}$. Then they are simultaneously satisfiable in*

$$D_{\Phi,d} := -C^\star_{\Phi,d} \cup \{0\} \cup C^\star_{\Phi,d} \subseteq \mathbb{Q}^\star,$$

where

$$C^\star_{\Phi,d} = \{x + nx\boldsymbol{\epsilon} \mid x \in C_{\Phi,d},\ n \in \mathbb{Z},\ -d \leq n \leq d\} \subseteq \mathbb{Q}^\star,$$

$C_{\Phi,d}$ is defined as in Lemma 3.1.7, and $-C^\star_{\Phi,d}$ denotes the set $\{-x \mid x \in C^\star_{\Phi,d}\}$.

Proof. We first fix a solution $v_i = a_i$ for $i = 1 \dots d$ of Φ. In general, some of the values a_i are positive, some are 0, and some others are negative: we look for a new solution $z_1, \dots, z_d \in D_{\Phi,d}$ such that z_i is positive, respectively 0 or negative, if and only if a_i is. To this aim we rewrite the formulas in Φ replacing each variable v_i with either y_i, or 0 (formally $0 \cdot 1$), or $-y_i$ (formally $-1 \cdot y_i$). We call Φ^+ the new set of formulas, which, by construction, is satisfiable in positive numbers $y_i = b_i$. To establish the lemma, it suffices to find a solution of Φ^+ taking values in $C^\star_{\Phi,d}$.

By Lemma 3.1.7, we have an assignment $y_i = c_i$ of values $c_1, \dots, c_d$ in $C_{\Phi^+,d} \subseteq C_{\Phi,d}$ that satisfies simultaneously all formulas $\bar{\phi}$ with $\phi \in \Phi^+$. Let $n_1, \dots, n_d \in \{n \in \mathbb{Z} \mid -d \leq n \leq d\}$ be such that for all i, j

$$\begin{aligned} n_i < n_j &\quad \text{if, and only if,} \quad \tfrac{b_i}{c_i} < \tfrac{b_j}{c_j}, \\ 0 < n_i &\quad \text{if, and only if,} \quad 1 < \tfrac{b_i}{c_i}, \\ n_i < 0 &\quad \text{if, and only if,} \quad \tfrac{b_i}{c_i} < 1. \end{aligned}$$

Such numbers exist: simply sort the set $\{1\} \cup \left\{ \frac{b_i}{c_i} \mid i = 1, \dots, d \right\}$ and consider the positions in the sorted sequence counting from that of 1. We claim that the assignment $y_i = c_i + n_i c_i \boldsymbol{\epsilon} \in \mathbb{Q}^\star$, for $1 \leq i \leq d$, satisfies all formulas of Φ^+. To prove this claim, we consider the different cases for atomic formulas

- $k \cdot y_i < h \cdot y_j$: if $kc_i < hc_j$ the formula is obviously satisfied. Otherwise, $kc_i = hc_j$; in this case, k and h are positive and the constraint
$$kc_i + kn_ic_i\boldsymbol{\epsilon} < hc_j + hn_jc_j\boldsymbol{\epsilon}$$
is equivalent to $n_i < n_j$. This, in turn, is equivalent by construction to $\frac{b_i}{c_i} < \frac{b_j}{c_j}$, which we get by observing that $b_ihc_j = b_ikc_i < b_jhc_i$.

- $k \cdot y_i = h \cdot y_j$: obviously, it holds that $kb_i = hb_j$ and $kc_i = hc_j$; therefore, it holds that $\frac{b_i}{c_i} = \frac{b_j}{c_j}$, and, as a consequence, also $n_i = n_j$, from which it follows that the formula is satisfied by the chosen assignment.

- $k \cdot 1 < h \cdot y_j$: similarly to the first case, if $k < hc_j$ the formula is obviously satisfied. Otherwise, $k = hc_j$; therefore, k and h are positive and then the constraint
$$k \cdot 1 < hc_j + hn_jc_j\boldsymbol{\epsilon}$$
is equivalent to $0 < n_j$. Then we obtain $1 < \frac{b_j}{c_j}$, by observing that $hc_j = k < hb_j$.

- $k \cdot y_i < h \cdot 1$: as the case above.

- $k \cdot 1 = h \cdot y_j$: obviously, $k \cdot 1 = hb_j = hc_j$; therefore, $\frac{b_j}{c_j} = 1$ and $n_j = 0$, from which it follows that the formula is satisfied by the chosen assignment.

- $k \cdot y_i = h \cdot 1$: as the case above. $\square$

Proof of Theorem 3.1.6. The sampling algorithm produces the finite substructure $\mathfrak{A}^\star_{\mathrm{At}(\tau),d}$ of $\mathfrak{A}^\star$ having domain $D_{\mathrm{At}(\tau),d}$ where $\mathrm{At}(\tau) := \bigcup_{R\in\tau} \mathrm{At}(\phi_R)$, that is, the τ-structure with domain $D_{\mathrm{At}(\tau),d}$ in which each relation symbol $R \in \tau$ denotes the restriction of $R^{\mathfrak{A}^\star}$ to $D_{\mathrm{At}(\tau),d}$. It is immediate to observe that this structure can be computed in polynomial time in d. Since $\mathfrak{A}^\star_{\mathrm{At}(\tau),d}$ is a substructure of $\mathfrak{A}^\star$, it is clear that if an instance is satisfiable in $\mathfrak{A}^\star_{\mathrm{At}(\tau),d}$, then it is a fortiori satisfiable in $\mathfrak{A}^\star$.

The vice versa follows from Lemma 3.1.8. Consider a set Ψ of atomic τ-formulas having free variables $x_1, \ldots, x_d$. Assume that Ψ is satisfied in $\mathfrak{A}^\star$ by the assignment $x_i = a_i$ for $i \in \{1, \ldots, d\}$. For each $R \in \Psi$, let $\Phi_R \subset \mathrm{At}(\phi_R)$ be the set of atomic subformulas of ϕ_R which are satisfied by our assignment $x_i = a_i$. Clearly, the atomic τ_0-formulas $\Phi := \bigcup_{R\in\Psi} \Phi_R$ are simultaneously satisfiable. Since the formulas ϕ_R contain no negations by construction, any simultaneous solution of Φ must also satisfy Ψ. By Lemma 3.1.8,

Φ has a solution in the set $D_{\Phi,d}$ defined therein. We can observe that $C_{\Phi,d} \subset C_{\mathrm{At}(\tau),d}$, hence $D_{\Phi,d} \subset D_{\mathrm{At}(\tau),d}$ and the claim follows. □

Corollary 3.1.9. *There exists an efficient sampling algorithm for* $\mathfrak{A}$*.*

Proof. On input d, the sampling algorithm produces the τ_0-reduct of the sample for $\mathfrak{A}^\star$ (on input d) described in the proof of Theorem 3.1.6. By Corollary 3.1.5 and Theorem 3.1.6, this is an efficient sampling algorithm for $\mathfrak{A}$. □

3.1.1 The Tractability of Max-Closed PLH CSPs

A relational structure $\mathfrak{A}$ is max-*closed* if it is preserved by the binary operation max.

In this section, we apply the results from Section 3.1 to prove the following result:

Theorem 3.1.10. *Let* $\mathfrak{A}$ *be a structure having finite relational signature* τ*. Assume that for every* $R \in \tau$*, the interpretation* $R^{\mathfrak{A}}$ *is PLH and preserved by* max*. Then* CSP($\mathfrak{A}$) *is polynomial-time solvable.*

This result is incomparable to known results about *max-closed* semilinear relations [12], i.e. semilinear relations that are preserved by the operation max. In particular, in that case, the weaker bound NP $\cap$ co-NP has been shown for a larger class, and the polynomial-time solvability only for a smaller class (which does not contain many max-closed PLH relations, for instance $x \geq \max(y, z)$). To prove Theorem 3.1.10, we use the notion of totally symmetric polymorphism and a result from [8].

Theorem 3.1.11 ([8], Theorem 2.5)**.** *Let* $\mathfrak{A}$ *be a structure over a finite relational signature with totally symmetric polymorphisms of all arities. If there exists an efficient sampling algorithm for* $\mathfrak{A}$*, then* CSP($\mathfrak{A}$) *is in P.*

Proof of Theorem 3.1.10. Since $\mathfrak{A}$ is preserved by max, the k-ary operation $\max^{(k)}$ is a polymorphism of $\mathfrak{A}$, for all $k \geq 1$ (see Example 1.4.27 and Remark 1.4.28). Furthermore, by Corollary 3.1.9 there exists an efficient sampling algorithm for $\mathfrak{A}$. Therefore, from Theorem 3.1.11, it follows that CSP($\mathfrak{A}$) can be solved in polynomial time. □

3.2 Efficient Sampling for PLH Valued Structures

In this section, we introduce the notion of a *sampling algorithm for a valued structure* and exhibit an *efficient* sampling algorithm for PLH valued structures.

Definition 3.2.1. Let Γ be a valued structure with domain C and a finite signature τ. A *sampling algorithm for* Γ takes as input a positive integer d and computes a finite-domain valued τ-structure Δ fractionally homomorphic to Γ such that, for every finite sum ϕ of τ-terms having at most d distinct variables, $V = \{x_1, \ldots, x_d\}$, and every $u \in \mathbb{Q}$, there exists a solution $h \colon V \to C$ with $\phi^\Gamma(h(x_1), \ldots, h(x_d)) \leq u$ if, and only if, there exists a solution $h' \colon V \to D$ with $\phi^\Delta(h'(x_1), \ldots, h'(x_d)) \leq u$. A sampling algorithm is called *efficient* if its running time is bounded by a polynomial in d. We refer to the output of a sampling algorithm by calling it the *sample*.

Remark 3.2.2. Observe that the output Δ_d of a sampling algorithm for a given valued structure Γ with a finite signature τ does not depend on the rational threshold u. Therefore, given a finite sum ϕ of function symbols from τ with variables $V := \{x_1, \ldots, x_n\}$, it holds that

$$\inf_{h \colon V \to \mathrm{dom}(\Gamma)} \phi^\Gamma(h(x_1), \ldots, h(x_n)) = \inf_{h' \colon V \to \mathrm{dom}(\Delta_d)} \phi^{\Delta_d}(h'(x_1), \ldots, h'(x_n)).$$

3.2.1 The Ring of Formal Laurent Power Series

In the present section, we extend the method developed in Section 3.1 to the treatment of PLH VCSPs. To better highlight the analogy with Section 3.1, so that the reader already familiar with it may quickly get an intuition of the arguments here, we use identical notation to represent corresponding objects. This choice has the drawback that some symbols, notably $\mathbb{Q}^\star$, need to be re-defined (the new $\mathbb{Q}^\star$ contains the old one). In this section, we sometimes skip details that can be borrowed unchanged from Section 3.1.

Definition 3.2.3. Let $\mathbb{Q}^\star$ denote the ring $\mathbb{Q}((\boldsymbol{\epsilon}))$ of formal Laurent power series in the indeterminate $\boldsymbol{\epsilon}$, that is, $\mathbb{Q}^\star$ is the set of formal expressions

$$\sum_{i=-\infty}^{+\infty} a_i \boldsymbol{\epsilon}^i$$

where $a_i \neq 0$ for only finitely many *negative* values of i. Clearly, $\mathbb{Q}$ is embedded in $\mathbb{Q}^\star$ (the embedding is defined similarly to its corresponding in Section 3.1). The ring operations on $\mathbb{Q}^\star$ are defined as usual

$$\sum_{i=-\infty}^{+\infty} a_i \boldsymbol{\epsilon}^i + \sum_{i=-\infty}^{+\infty} b_i \boldsymbol{\epsilon}^i = \sum_{i=-\infty}^{+\infty} (a_i + b_i) \boldsymbol{\epsilon}^i,$$

$$\sum_{i=-\infty}^{+\infty} a_i \boldsymbol{\epsilon}^i \cdot \sum_{i=-\infty}^{+\infty} b_i \boldsymbol{\epsilon}^i = \sum_{i=-\infty}^{+\infty} \left(\sum_{j=-\infty}^{+\infty} a_j b_{i-j} \right) \boldsymbol{\epsilon}^i,$$

where the sum in the product definition is always finite by the hypothesis on a_i, b_i with negative index i. The order is the lexicographical order induced by $\mathbb{Q}$ extended with $0 < \boldsymbol{\epsilon} \ll 1$, i.e.,

$$\sum_{i=-\infty}^{+\infty} a_i \boldsymbol{\epsilon}^i < \sum_{i=-\infty}^{+\infty} b_i \boldsymbol{\epsilon}^i \quad \text{iff} \quad \exists i \; a_i < b_i \land \forall j < i \; a_j = b_j.$$

It is well known that $\mathbb{Q}^\star$ is an ordered field, that is, all non-zero elements have a multiplicative inverse and the order is compatible with the field operations. We define the following subsets of $\mathbb{Q}^\star$ for $m \leq n$

$$\mathbb{Q}^\star_{m,n} := \left\{ \sum_{i=m}^{n} \boldsymbol{\epsilon}^i a_i \;\middle|\; a_i \in \mathbb{Q} \right\} \subset \mathbb{Q}^\star .$$

Clearly $\mathbb{Q}$ is embedded in $\mathbb{Q}^\star$, by mapping every rational number a into $\sum_{i=-\infty}^{+\infty} a_i \boldsymbol{\epsilon}^i$ such that $a_i = 0$, for every $i \neq 0$, and $a_0 = a$.

Definition 3.2.4. We define a new structure $\mathfrak{L}^\star$, which is an extension of an expansion (or an expansion of an extension) of $\mathfrak{L}$, having domain $\mathbb{Q}^\star$, signature $\tau_1 := \tau_0 \cup \{k\}_{k \in \mathbb{Q}^\star_{-1,1}}$, and such that the interpretation of symbols in τ_0 is formally the same as for $\mathfrak{L}$ and the symbols $k \in \mathbb{Q}^\star_{-1,1}$ denote constants (i.e., functions with arity 0).

Notice that, for technical reasons, we allow only constants from $\mathbb{Q}^\star_{-1,1}$. In the remainder of this section, τ_1-formulas are interpreted in the structure $\mathfrak{L}^\star$. We make for τ_1-formulas the same assumptions as in Section 3.1 (that atomic formulas are of the form $\perp$ or $\top$ if they are trivial, and otherwise of the form either $c_1 \cdot 1 \overset{<}{=} x_i$, or $x_i \overset{<}{=} c_2 \cdot 1$, or $c_1 \cdot x_i \overset{<}{=} c_2 \cdot x_j$ with constants c_1 and c_2 not both negative and where function symbols $c_i \cdot$ are never composed). Also $H(\Phi)$ and $K(\Phi)$, where Φ is a set of atomic τ_1-formulas are defined similarly to Section 3.1. Observe that the reduct of $\mathfrak{L}^\star$ obtained by restricting the signature to τ_0 is *elementarily equivalent* to $\mathfrak{L}$, i.e., it satisfies the same first-order sentences.

Similarly as in Section 3.1, for every PLH valued structure Γ and every positive integer number d, we explicitly give a τ_1-structure with a finite domain $D^\star$ such that every instance of VCSP(Γ) with at most d distinct free variables has a solution with values in $\mathbb{Q}$ if, and only if, it has a solution with values in $D^\star \subseteq \mathbb{Q}^\star$ (Lemma 3.2.7). We need two preliminary results: Lemma 3.2.5 and Lemma 3.2.6, which are analogues of Lemma 3.1.7 and Lemma 3.1.8 from Section 3.1. More specifically, in Lemma 3.2.5 we consider the positive solutions to the closures of finitely many τ_1-formulas, and in Lemma 3.2.6 we consider the positive solutions to finitely many τ_1-formulas.

Lemma 3.2.5. *Let Φ be a finite set of atomic τ_1-formulas having free variables in $\{v_1, \ldots, v_d\}$ and let $\bar{\Phi}$ be the set $\bigcup_{\phi \in \Phi} \bar{\phi}$. Suppose that there is $0 < r \in \mathbb{Q}^\star$ such that all satisfying assignments of $\bar{\Phi}$ with values $(x_1, \ldots, x_d)$ in $\mathbb{Q}^\star$ also satisfy $0 < x_i \leq r$, for all $i \in \{1, \ldots, d\}$. Let u, $\alpha_1, \ldots, \alpha_d$ be elements of $\mathbb{Q}^\star$. Assume that the formulas in Φ are simultaneously satisfiable by a point $(x_1, \ldots, x_d) \in (\mathbb{Q}^\star)^d$ such that $\sum_{i=1}^d \alpha_i x_i < u$. Let us define the set*

$$C_{\Phi,d} = \left\{ |k| \prod_{i=1}^{s} |h_i|^{e_i} \;\middle|\; k \in K(\Phi),\ e_1, \ldots, e_s \in \mathbb{Z},\ \sum_{r=1}^{s} |e_r| < d \right\} \subseteq \mathbb{Q}^\star_{-1,1},$$

where $h_1, \ldots, h_s$ is an enumeration of the (finitely many) elements of $H(\Phi)$. Then there is a point in $(x'_1, \ldots, x'_d) \in (C_{\Phi,d})^d \subseteq (\mathbb{Q}^\star)^d$ with $\sum_{i=1}^d \alpha_i x'_i < u$ that simultaneously satisfies all $\bar{\phi}$, for $\phi \in \Phi$.

Proof. As in the proof of Lemma 3.1.7 (to which we direct the reader for many details) we take a maximal $\gamma \leq \beta$ such that there are Ψ_1, Ψ_2, Ψ_3 with

$$\begin{aligned}
\bar{\Phi} &= \{s_1 = s'_1, \ldots, s_\alpha = s'_\alpha\} \cup \{t_1 \leq t'_1, \ldots, t_\beta \leq t'_\beta\} \\
\Psi_1 &= \{s_1 = s'_1, \ldots, s_\alpha = s'_\alpha\} \\
\Psi_2 &= \{t_1 = t'_1, \ldots, t_\gamma = t'_\gamma\} \\
\Psi_3 &= \{t_{\gamma+1} \leq t'_{\gamma+1}, \ldots, t_\beta \leq t'_\beta\}
\end{aligned}$$

and $\Psi_1 \cup \Psi_2 \cup \Psi_3$ is satisfiable by an assignment $(x_1, \ldots, x_d)$ such that $\sum_{i=1}^d \alpha_i x_i < u$. As in the proof of Lemma 3.1.7 the set of solutions of $\Psi_1 \cup \Psi_2$ satisfying $\sum_{i=1}^d \alpha_i x_i < u$ is contained in the solutions of Ψ_3. So, here too, it suffices to show that there is a solution of $\Psi_1 \cup \Psi_2$ with $\sum_{i=1}^d \alpha_i v_i < u$ taking values in $C_{\Phi,d}$. The proof of Lemma 3.1.7 shows that there is a solution of $\Psi_1 \cup \Psi_2$ taking values $(x_1, \ldots, x_d)$ in $C_{\Phi,d}$ without necessarily meeting the requirement that $\sum_{i=1}^d \alpha_i x_i < u$. We prove that, in fact, any such solution meets the additional constraint.

Let $x_i = a_i, b_i$, be two distinct satisfying assignments for $\Psi_1 \cup \Psi_2$ such that $\sum_{i=1}^d \alpha_i a_i < u$ and $\sum_{i=1}^d \alpha_i b_i \geq u$. We know that the first assignment, $x_i = a_i$, exists and we assume the existence of the second assignment, $x_i = b_i$, towards a contradiction. The two assignments must differ; then, without loss of generality, we can assume that $a_1 \neq b_1$. For $t \in \mathbb{Q}^\star$, with $t \geq 0$, define the assignment $x_i(t) = (1+t)a_i - tb_i$. Since all constraints in $\Psi_1 \cup \Psi_2$ are equalities, it is clear that the new assignment $x_i(t)$ satisfies $\Psi_1 \cup \Psi_2$ for all $t \in \mathbb{Q}^\star$. Moreover, if $t \geq 0$ we obtain

$$\sum_i \alpha_i x_i(t) \leq \sum_i \alpha_i a_i - t \left(\sum_i \alpha_i b_i - \sum_i \alpha_i a_i \right) < u.$$

Let $t = \frac{2r}{|b_1 - a_1|}$. Then

$$x_1(t) = a_1 + \frac{2r}{|b_1 - a_1|}(a_1 - b_1)$$

is either not smaller than $2r$ or smaller than 0, depending on the sign of $(a_1 - b_1)$. In either case, we have a solution $x_i = x_i(t)$ of $\Psi_1 \cup \Psi_2$ satisfying $\sum_i \alpha_i x_i(t) < u$, which must, therefore, be a solution of Φ that does not satisfy $0 < x_i \leq r$. □

Lemma 3.2.6. *Let Φ be a finite set of atomic τ_1-formulas having free variables in $\{v_1, \ldots, v_d\}$. Suppose that there are $0 < l < r \in \mathbb{Q}^\star$ such that all the assignments $(x_1, \ldots, x_d)$ satisfying Φ in the domain $\mathbb{Q}^\star$ also satisfy $l < x_i < r$, for all $i \in \{1, \ldots, d\}$. Let $\alpha_1, \ldots, \alpha_d$ be rational numbers and $u \in \mathbb{Q}^\star_{-1,1}$. Assume that the formulas in Φ are simultaneously satisfiable by a point $(x_1, \ldots, x_d) \in (\mathbb{Q}^\star)^d$ such that $\sum_{i=1}^d \alpha_i x_i \leq u$. Then the same formulas are simultaneously satisfiable by a point $(x'_1, \ldots, x'_d) \in (C^\star_{\Phi,d})^d \subseteq (\mathbb{Q}^\star)^d$ such that $\sum_i \alpha_{i=1}^d x'_i \leq u$, where*

$$C^\star_{\Phi,d} = \{x + nx\boldsymbol{\epsilon}^3 \mid x \in C_{\Phi,d},\ n \in \mathbb{Z},\ -d \leq n \leq d\} \subseteq \mathbb{Q}^\star_{-1,4}\,.$$

Proof. We consider two cases: either all satisfying assignments with values $(x_1, \ldots, x_d)$ in $(\mathbb{Q}^\star)^d$ satisfy the inequality $\sum_i \alpha_{i=1}^d x_i \geq u$ or there is a satisfying assignment $(x_1, \ldots, x_d)$ for Φ such that $\sum_i \alpha_{i=1}^d x_i < u$.

We claim that, in the first case, all satisfying assignments, in fact, satisfy $\sum_{i=1}^d \alpha_i x_i = u$. To prove our claim, assume that $x_i = a_i, b_i$ are two satisfying assignments such that $\sum_{i=1}^d \alpha_i a_i = u$ and $v := \sum_{i=1}^d \alpha_i b_i > u$. As in the proof of Lemma 3.2.5, let us consider assignments having the form $x_i(t) = (1+t)a_i - tb_i$, for all $t \in \mathbb{Q}^\star$. Clearly, for all $t > 0$ it holds $\sum_{i=1}^d \alpha_i x_i(t) = u - t(v-u) < u$. As in Lemma 3.2.5, the new assignment must satisfy all the equality constraints in Φ. Each inequality constraint implies a strict inequality on t (remember that Φ only has strict inequalities). Since all of these implied strict inequalities must be satisfied by $t = 0$, there is an open interval of acceptable values of t around 0, and, in particular, an acceptable value $t > 0$. Our claim is thus established. Therefore, in this case, it suffices to find any satisfying assignment for Φ taking values in $C^\star_{\Phi,d}$. The assignment is now constructed as in the proof of Lemma 3.1.8, replacing the formal symbol $\boldsymbol{\epsilon}$ in that proof by $\boldsymbol{\epsilon}^3$. More precisely, take a satisfying assignment $x_i = b_i$ for Φ and, by Lemma 3.2.5, a satisfying assignment $x_i = c_i$ for $\bar{\Phi}$ taking values in $C_{\Phi,d}$. Observe that the hypothesis that all solutions of Φ satisfy $l < x_i$ for all i is used here to ensure that all solutions of $\bar{\Phi}$ assign positive values to the variables, which is required by

Lemma 3.2.5. Let $-d \leq n_1, \ldots, n_d \leq d$ be integers such that for all i, j

$$\begin{aligned} n_i < n_j &\quad \text{if, and only if,} \quad \tfrac{b_i}{c_i} < \tfrac{b_j}{c_j}, \\ 0 < n_i &\quad \text{if, and only if,} \quad 1 < \tfrac{b_i}{c_i}, \\ n_i < 0 &\quad \text{if, and only if,} \quad \tfrac{b_i}{c_i} < 1. \end{aligned}$$

The assignment $y_i = c_i + n_i c_i \boldsymbol{\epsilon}^3$ can be seen to satisfy all formulas of Φ by the same check as in the proof of Lemma 3.1.8. Observe that we have to replace $\boldsymbol{\epsilon}$ appearing in Lemma 3.1.8 by $\boldsymbol{\epsilon}^3$ here, so that $\mathbb{Q}^\star_{-1,1} \cap \boldsymbol{\epsilon}^3\, \mathbb{Q}^\star_{-1,1} = \emptyset$.

For the second case, fix a satisfying assignment $x_i = b_i$. By Lemma 3.2.5 there is an assignment $x_i = c_i \in C_{\Phi,d}$ such that $\sum_{i=1}^d \alpha_i c_i < u$ and this assignment satisfies $\bar{\phi}$ for all $\phi \in \Phi$. From these two assignments construct the numbers n_i and then the assignment $y_i = c_i + n_i c_i \boldsymbol{\epsilon}^3$ as before. For the same reason it is clear that the new assignment satisfies Φ. To conclude that $\sum_{i=1}^d \alpha_i y_i < u$ we write

$$\sum_{i=1}^{d} \alpha_i y_i = \sum_{i=1}^{d} \alpha_i c_i + \boldsymbol{\epsilon}^3 \sum_{i=1}^{d} \alpha_i n_i c_i < u$$

because the first summand is in $\mathbb{Q}^\star_{-1,1}$ and smaller than u, therefore the second summand is neglected in the lexicographical order. $\square$

Lemma 3.2.7. *Let Φ be a finite set of atomic τ_0-formulas having free variables in $\{v_1, \ldots, v_d\}$. Let u, $\alpha_1, \ldots, \alpha_d$ be rational numbers. Then the following are equivalent:*

1. *The formulas in Φ are simultaneously satisfiable in $\mathbb{Q}$, by a point $(x_1, \ldots, x_d) \in \mathbb{Q}^d$ such that $\sum_{i=1}^d \alpha_i x_i \leq u$.*
2. *The formulas in Φ are simultaneously satisfiable in $D_{\Phi,d} \subseteq \mathbb{Q}^\star$, by a point $(x'_1, \ldots, x'_d) \in D^d_{\Phi,d}$ such that $\sum_{i=1}^d \alpha_i x'_i \leq u$, where the set $D_{\Phi,d}$ is defined as follows*

$$\begin{aligned} D_{\Phi,d} &:= -C^\star_{\Phi',d} \cup \{0\} \cup C^\star_{\Phi',d} \subseteq \mathbb{Q}^\star_{-1,4} \\ \Phi' &:= \Phi \cup \{x > \boldsymbol{\epsilon},\ x < -\boldsymbol{\epsilon},\ x > -\boldsymbol{\epsilon}^{-1},\ x < \boldsymbol{\epsilon}^{-1}\}. \end{aligned}$$

Remark 3.2.8. Observe that the set $D_{\Phi,d}$ depends neither on the α_i's nor on u. In fact, $D_{\Phi,d}$ only depends on the set of formulas Φ and on the number d of free variables.

Proof. The implication *2* $\rightarrow$ *1* is immediate observing that the conditions Φ and $\sum_{i=1}^d \alpha_i x_i \leq u$ are first-order definable in $\mathfrak{S}$. In fact, any assignments with values in $D_{\Phi,d}$ satisfying the conditions is, in particular, an assignment in $\mathbb{Q}^\star$, and, by the completeness of the first-order theory of ordered $\mathbb{Q}$-vector spaces, we have an assignment taking values in $\mathbb{Q}$.

For the vice versa, fix any assignment $x_i = a_i$ with $a_i \in \mathbb{Q}$, for every $i \in \{1, \dots, d\}$. We pre-process the formulas in Φ producing a new set of atomic formulas Φ' as follows. We replace every variable x_i such that $a_i = 0$ with the constant $0 = 0 \cdot 1$. Then we replace each of the remaining variables x_i with either y_i or $-y_i$ accordingly to the sign of a_i. Finally, we add the constraints $\boldsymbol{\epsilon} < y_i$ and $y_i < \boldsymbol{\epsilon}^{-1}$ for each of these variables. Similarly, we produce the new coefficients $\alpha_i' = \operatorname{sign}(a_i)\alpha_i$. It is clear that the new set of formulas Φ' has a satisfying assignment in the set of positive rational numbers with $\sum_{i=1}^{d} \alpha_i' y_i \leq u$. Observing that a positive rational number x always satisfies $\boldsymbol{\epsilon} < x < \boldsymbol{\epsilon}^{-1}$, we see that Φ' satisfies the hypothesis of Lemma 3.2.6 with $l = \boldsymbol{\epsilon}$ and $r = \boldsymbol{\epsilon}^{-1}$. Hence the statement. $\square$

Remark 3.2.9. Lemma 3.2.7 provides a polynomial-time many-one reduction of the VCSP for a PLH valued structure Γ with a finite signature to the VCSP for a valued structure $\Delta^\star$ with a finite signature having as domain a finite subset of $\mathbb{Q}^\star$. We want to point out that, however, Lemma 3.2.7 does not give rise to a sampling algorithm: firstly, the cost functions in $\Delta^\star$ take values in $\mathbb{Q}^\star$ rather than in $\mathbb{Q} \cup \{+\infty\}$; secondly, the signature of $\Delta^\star$ is strictly larger than the signature of Γ. In the next section, we show how to obtain an efficient sampling algorithm for PLH valued structures using Lemma 3.2.7.

Once we have reduced the VCSP for a PLH valued structure with a finite signature to a $\mathbb{Q}^\star$-valued constraint satisfaction problem over a finite domain $D \subseteq \mathbb{Q}^\star$ (with a finite signature), there are two possible approaches. The formulas Φ in Lemma 3.2.7 are going to define a subset of the domain of a PLH function in $\mathbb{Q}^\star$, while the coefficients α_i define the function on that subset. The first approach is to interpret our PLH functions over the domain $\mathbb{Q}^\star$; the second one is to substitute a suitably small rational value of $\boldsymbol{\epsilon}$ in the formal expression of $D_{\Phi,d}$ and thereby map the problem to $\mathbb{Q}$. In the first case, we have to *transfer* the known approaches for $\mathbb{Q}$ to the new domain; in the second case, we can *use* them (after having computed a suitable $\boldsymbol{\epsilon}$). (For a comparison of the two approaches, we direct the reader to Section 6.3 of Chapter 6, where we apply both approaches to study the complexity of VCSPs for submodular PLH valued structures.)

If we interpret our PLH functions in the domain $\mathbb{Q}^\star$, the algorithm solving the obtained finite-domain VCSP is required to run in strongly polynomial time (see Chapter 1, Section 1.8). However, it is natural to think about applying the techniques known for finite-domain VCSPs to solve the VCSP for PLH valued structures that, after having been sampled have the algebraic conditions that guarantee the polynomial-time solvability (over finite domains). On the other hand, these techniques for finite-domain VCSPs rely on linear programming relaxations, and none of the known polynomial-time algorithms solving LP is known to be strongly polynomial-time. Therefore, if we want to use the characterisation of polynomial-time solvable finite-domain VCSPs, we have first to map the finite domain $D_{\Phi,d}$ into (a subset of) $\mathbb{Q}$.

3.2.2 Reduction to a VCSP over a Finite Rational Domain

In this section, we use the results achieved in Section 3.2.1 to provide an efficient sampling algorithm for PLH valued structures.

Let Φ be a finite set of τ_0-formulas having at most d distinct free variables. By Lemma 3.2.7, for every $\alpha_1, \ldots, \alpha_d, u \in \mathbb{Q}$ the formulas in Φ are simultaneously satisfiable in $\mathbb{Q}$ by a point $(x_1, \ldots, x_d)$ such that $\sum_{i=1}^{d} \alpha_i x_i \leq u$ if, and only if, they are simultaneously satisfiable in $D_{\Phi,d} \subseteq \mathbb{Q}^{\star}_{-1,4}$ by a point $(x'_1, \ldots, x'_d)$ such that $\sum_{i=1}^{d} \alpha_i x'_i \leq u$. The elements in $D_{\Phi,d} \subseteq \mathbb{Q}^{\star}_{-1,4}$ are of the form

$$\sum_{i=-1}^{4} x_i \boldsymbol{\epsilon}^i \text{ where } x_i \in -C_{\Phi,d} \cup \{0\} \cup C_{\Phi,d}$$

where $C_{\Phi,d}$ is as in Lemma 3.1.7. In $\mathfrak{L}^{\star}$ it holds that

$$-\boldsymbol{\epsilon}^{-1} < x < -\boldsymbol{\epsilon} \qquad \text{for every } x \in -C_{\Phi,d},$$
$$\boldsymbol{\epsilon} < x < \boldsymbol{\epsilon}^{-1} \qquad \text{for every } x \in C_{\Phi,d}.$$

(See Lemmas 3.2.5, 3.2.6, 3.2.7, and Lemmas 3.1.7, 3.1.8.)

Lemma 3.2.10. *Let Φ be a finite set of τ_0-formulas having at most d distinct free variables. Let $D_{\Phi,d}$ be defined as in Lemma 3.2.7 (2.) and let $\mathfrak{D}^{\star}$ be the finite substructure of $\mathfrak{L}^{\star}$ with domain $D_{\Phi,d} \subseteq \mathbb{Q}^{\star}_{-1,4}$. Then there exists a positive rational ε with $-\varepsilon^{-1} < x < -\varepsilon$ for every $x \in -C_{\Phi,d}$, and $\varepsilon < x < \varepsilon^{-1}$ for every $x \in C_{\Phi,d}$ such that the map $\eta \colon D_{\Phi,d} \to \mathbb{Q}$ defined by*

$$\eta\left(\sum_{j=-1}^{4} x_j \epsilon^j\right) = \sum_{j=-1}^{4} x_j \varepsilon^j$$

is a homomorphism from the τ_0-reduct of $\mathfrak{D}^{\star}$ to $\mathfrak{L}$. Moreover, ε and η are computable in polynomial time in d.

Proof. Let $C_{\Phi,d} \subseteq \mathbb{Q}$ be as in Lemma 3.1.7 and let $-C_{\Phi,d} := \{-x \mid x \in C_{\Phi,d}\}$. We define $C := \{x - y \mid x, y \in C_{\Phi,d} \cup \{0\} \text{ and } x - y > 0\}$. Observe that it holds $C \supseteq C_{\Phi,d}$. Define

$$\varepsilon := \frac{1}{6} \frac{\min C}{\max C}.$$

The number ε is positive and rational and it can be computed in polynomial time in d. Furthermore,

$$\begin{array}{ll} 0<\varepsilon<1, & \\ \varepsilon<\min C<\min C_{\Phi,d} & \text{implying } \max(-C_{\Phi,d})<-\varepsilon, \text{ and} \\ \max C=\max C_{\Phi,d}<\varepsilon^{-1} & \text{implying } -\varepsilon^{-1}<\min(-C_{\Phi,d}). \end{array}$$

It follows that $-\varepsilon^{-1}<x<-\varepsilon$ for every $x \in -C_{\Phi,d}$, and $\varepsilon<x<\varepsilon^{-1}$ for every $x \in C_{\Phi,d}$.

It is easy to see that η preserves the scalar multiplication by rational elements and the identity in $\mathbb{Q}$. We prove now that η is order-preserving (and therefore injective). Let us consider

$$x:=\sum_{j=-1}^{4} x_j \boldsymbol{\epsilon}^j, \text{ and } y:=\sum_{j=-1}^{4} y_j \boldsymbol{\epsilon}^j \in D_{\Phi,d}$$

such that x is smaller than y in the lexicographic order induced by $\mathbb{Q}^{\star}_{-1,4}$. This means that there exists an index $i \in \{-1,\ldots,4\}$ such that $x_j=y_j$ for $-1 \leq j<i$ and $x_i<y_i$. Since $\varepsilon>0$, it holds that $x_i\varepsilon^i<y_i\varepsilon^i$ and, consequently,

$$\sum_{j=-1}^{i} x_j\varepsilon^j<\sum_{j=-1}^{i} y_j\varepsilon^j.$$

Moreover, if $i \neq 4$, then for all $j \in \{i+1,\ldots,4\}$ it holds

$$(x_j-y_j)\varepsilon^{j-i} \leq (x_j-y_j)\varepsilon \leq (\max C)\varepsilon=\frac{\min C}{6} \leq \frac{y_i-x_i}{6}$$

because $\varepsilon<1$ and $x_j-y_j \leq \max C$ (indeed, if $x_j-y_j>0$ then $x_j-y_j \in C$, otherwise x_j-y_j is smaller than any element in C). Therefore, we obtain $(x_j-y_j)\varepsilon^j \leq \frac{y_i-x_i}{6}\varepsilon^i$ and

$$\sum_{j=i+1}^{4}(x_j-y_j)\varepsilon^j \leq \sum_{j=i+1}^{4}\frac{y_i-x_i}{6}\varepsilon^i \leq \frac{5}{6}(y_i-x_i)\varepsilon^i<(y_i-x_i)\varepsilon^i.$$

It follows that

$$\sum_{j=i}^{4} x_j\varepsilon^j<\sum_{j=i}^{4} y_j\varepsilon^j,$$

and, since $x_j=y_j$ for $-1 \leq j \leq i-1$,

$$\sum_{j=-1}^{4} x_j\varepsilon^j<\sum_{j=-1}^{4} y_j\varepsilon^j,$$

i.e., η preserves the order. $\square$

Let Φ be a finite set of τ_0-formulas with at most d variables, and let η be the map introduced in Lemma 3.2.10, we set $E_{\Phi,d} := \eta(D_{\Phi,d}) \subseteq \mathbb{Q}$. Lemma 3.2.10 implies the following corollary.

Corollary 3.2.11. *Let Γ be a PLH valued structure with a finite signature τ. Then there exists an efficient sampling algorithm for Γ.*

Proof. For every cost function f in τ let us consider the quantifier-free τ_0-formula ϕ_f defining $R_f^\Gamma = \mathrm{dom}(f)$ over $\mathbb{Q}$. Let $\mathrm{At}(\phi_f)$ denote the set of atomic subformulas of ϕ_f and let $\mathrm{At}(\tau) := \bigcup_{f\in\tau} \mathrm{At}(\phi_f)$.

On input d, the algorithm produces the valued finite substructure Δ of Γ having domain $\eta(D_{\mathrm{At}(\tau),d})$, where η is defined as in Lemma 3.2.10. It is immediate to see that the valued structure Δ has polynomial size in d and can be computed in polynomial time in d, because $D_{\mathrm{At}(\tau),d}$ and $\eta(D_{\mathrm{At}(\tau),d})$ can be computed in polynomial time in d (see Remark 3.2.8, and Lemma 3.2.10). Let ϕ be a finite sum of function symbols from τ with at most d variables from $V := \{v_1, \dots, v_d\}$, and let u be a rational number. By Lemma 3.2.7, there exists an assignment $h\colon V \to \mathbb{Q}$ such that $\phi^\Gamma(h) \leq u$ if, and only if, there exists an assignment $h'\colon V \to D_{\mathrm{At}(\tau),d}$ such that $\phi^{\Gamma^\star}(h) \leq u$. Furthermore, by Lemma 3.2.10 there exists a positive rational number ε such that the map $\eta : D_{\mathrm{At}(\tau),d} \to \mathbb{Q}$ is a homomorphism of τ_0-structures from the τ_0-reduct of $\mathfrak{D}^\star$ to $\mathfrak{L}$. Since η is injective, the assignment $\eta \circ h'\colon V \to \eta(D_{\mathrm{At}(\tau),d})$ has cost $\phi^\Delta(\eta \circ h') \leq u$. $\square$

Summary and Outlook

We have shown a polynomial-time many-one reduction of the VCSP for a PLH valued structure to a finite-domain VCSP. We remark that, in order to prove the polynomial-time solvability of (a class of) PLH valued structures, we need to show the polynomial-time solvability of the obtained finite-domain VCSP. To accomplish with this task, we have two options: either use the efficient sampling algorithm given in Corollary 3.2.11 and apply the methods from the theory of finite-domain VCSPs, as explained in the next chapter; or (polynomial-time many-one) map the PLH VCSP onto a finite-domain VCSP with costs in $\mathbb{Q}^\star$ (see Lemma 3.2.7) and provide a strongly polynomial-time algorithm (see Section 1.8) that solves the finite-domain problem in $\mathbb{Q}^\star$. In Chapter 6, we apply both approaches to prove the polynomial-time solvability of submodular PLH valued structures and discuss the differences between the two approaches.

Chapter 4

The Power of LP for Infinite-Domain VCSPs

In this chapter, we give a sufficient algebraic condition under which valued structures that admit an efficient sampling algorithm can be solved in polynomial time using a linear programming relaxation.

4.1 The Basic Linear Programming Relaxation

Every VCSP over a finite domain has a natural linear programming relaxation. Let Γ be a valued structure over a finite domain D with signature τ, such that the cost functions take values in $\mathbb{Q} \cup \{+\infty\}$. Let I be an instance of VCSP(Γ) with set of free variables $V_I = \{x_1, \dots, x_d\}$, and objective function

$$\phi_I(x_1, \dots, x_d) = \sum_{j \in J} f_j(x_1^j, \dots, x_{n_j}^j),$$

with $f_j \in \Gamma$, $x^j = (x_1^j, \dots, x_{n_j}^j) \in V^{n_j}$, for all $j \in J$ (the set J is finite and indexing the cost functions that are summands of ϕ_I). Define the sets W_1, W_2, and W of variables $\lambda_j(t)$, $\mu_{x_i}(a)$, for $j \in J$, $t \in D^{n_j}$, $x_i \in V$, and $a \in D$, as follows.

$$\begin{aligned} W_1 &:= \{\lambda_j(t) \mid j \in J \text{ and } t \in D^{n_j}\}, \\ W_2 &:= \{\mu_{x_i}(a) \mid x_i \in V \text{ and } a \in D\}, \\ W &:= W_1 \cup W_2. \end{aligned}$$

The *basic linear programming, BLP, relaxation for* Γ *associated to* I (see [69, 96], and references therein) is a linear program with variables in W and is defined as follows

$$\mathrm{BLP}(I,\Gamma) := \min_{\lambda_j(t)\in\mathbb{Q}} \sum_{j\in J} \sum_{t\in D^{n_j}} \lambda_j(t) f_j^\Gamma(t)$$

such that

$$\sum_{t\in D^{n_j}:t_l=a} \lambda_j(t) = \mu_{x_l^j}(a) \qquad \text{for all } j\in J,\ l\in\{1,\dots,n_j\},\ a\in D \tag{4.1}$$

$$\sum_{a\in D} \mu_{x_i}(a) = 1 \qquad \text{for all } x_i\in V \tag{4.2}$$

$$\lambda_j(t) = 0 \qquad \text{for all } j\in J,\ t\notin \mathrm{dom}(f_j) \tag{4.3}$$

$$0 \le \lambda_j(t), \mu_{x_i}(a) \le 1 \qquad \text{for all } \lambda_j(t)\in W_1,\ \mu_{x_i}(a)\in W_2. \tag{4.4}$$

The variables $\lambda_j(t)$, $\mu_{x_i}(a) \in W$ can assume rational (or real) values in $[0,1]$.

For every $j \in J$, we can interpret λ_j as a distribution of probabilities on tuples in D^{n_j} i.e., $\lambda_j(t)$ can be seen as the probability to assign the label t to $(x_1^j,\dots,x_{n_j}^j)$ (observe that $\sum_{t\in D^{n_j}} \lambda_j(t) = 1$ follows from Conditions (4.1) and (4.2)). In this interpretation, $\mu_{x_i}(a)$ is the probability of assigning the value a to the variable x_i and therefore it is the marginal probability of the distribution λ_j for the variable v_i (as expressed by Condition (4.1)).

If there is no feasible solution to this linear program, then we obtain $\mathrm{BLP}(I,\Gamma) = +\infty$. We say that the BLP relaxation *solves* the instance I of VCSP(Γ) if $\mathrm{BLP}(I,\Gamma) = \min_{x\in D^d} \phi_I(x)$. We say that the BLP relaxation solves the VCSP for Γ if it solves all instances of VCSP(Γ). For a given instance of a finite-domain VCSP, the corresponding BLP relaxation can be computed in polynomial time. Therefore, if the VCSP for a valued structure Γ is solved by the BLP relaxation, then VCSP(Γ) can be solved in polynomial time.

The valued structures with finite domains that can be solved by the BLP relaxation have been characterised by Kolmogorov, Thapper, and Živný in [69].

Theorem 4.1.1 ([69], Theorem 1)**.** *Let Δ be a valued structure with a finite signature and a finite domain. Then the* BLP *relaxation solves* VCSP(Δ) *if, and only if, Δ has fully symmetric fractional polymorphisms of all arities.*

4.2 The Sampling + BLP Algorithm

Let Γ be a valued structure that admits an efficient sampling algorithm. We may solve VCSP(Γ) by using the following algorithm that on instance I computes a sample Δ of Γ and then solves the BLP relaxation for Δ associated to I.

ALGORITHM 1: Sampling + BLP Algorithm

Input: $I := (V_I, \phi_I, u_I)$.
Output: accepts if there exists an assignment $h\colon V_I \to \mathrm{dom}(\Gamma)$ such that $\phi_I(h(x_1, \ldots, x_{|V_I|})) \leq u_I$.
$\Delta := \mathrm{Sampling}_\Gamma(|V_I|)$;
$\mathrm{BLP}(I, \Delta)$;
if $\mathrm{BLP}(I, \Delta) \leq u_I$ **then**
  accept;
else
  reject;
end

Note that Algorithm 1 runs in polynomial time in $|V_I|$ and if it rejects, then indeed the answer to VCSP(Γ) is no, without further assumptions.

Lemma 4.2.1. *Let Γ be a valued structure admitting an efficient sampling algorithm. Then Algorithm 1 runs in polynomial time in $|V_I|$.*

Proof. Let $I := (V_I, \phi_I, u_I)$ be an instance of VCSP(Γ) and let Δ be the finite-domain valued structure computed by the sampling algorithm for Γ on input $|V_I|$. The sampling algorithm runs in polynomial time in $|V_I|$, so the size of Δ is polynomial in $|V_I|$. Since $\mathrm{BLP}(I, \Delta)$ can be implemented to run in polynomial time in $|V_I| + |\Delta|$, it follows that the entire algorithm runs in polynomial time in $|V_I|$. □

In the following section, we present a sufficient condition under which Algorithm 1 correctly solves VCSP(Γ).

4.3 Fully Symmetric Fractional Polymorphisms

The main result of this section, Theorem 4.3.5, states that if Γ is improved by fully symmetric fractional operations of all arities, then Algorithm 1 correctly solves VCSP(Γ) in polynomial time. Note that there are valued structures that have fully symmetric fractional polymorphisms of all arities which are not inherited by the valued structures computed by sampling, as the next example shows.

Example 4.3.1. Let Γ be a convex PLH valued structure with a finite signature. For all $k \geq 2$, the valued structure Γ is improved by the fully symmetric fractional operation $\omega_{conv}^{(k)}$ defined in Example 1.4.19 (see Proposition 1.6.8). Observe that the finite-domain valued structure computed by a sampling algorithm for Γ might not have fully symmetric fractional polymorphisms of all arities. In fact, the fractional polymorphisms $\omega_{conv}^{(k)}$ are not even inherited by valued finite substructures of Γ whose domain contains more than one element. However, (by Theorem 4.3.5) Algorithm 1 solves VCSP(Γ). $\triangle$

Definition 4.3.2. Let Δ be a valued τ-structure with domain D and let $m \geq 1$. The *multiset-structure* $\mathcal{P}^m(\Delta)$ [96] is the valued structure with domain $\left(\binom{D}{m}\right)$, i.e., the set of multisets of elements from D of size m, and for every k-ary function symbol $f \in \tau$, and $\alpha_1, \dots, \alpha_k \in \left(\binom{D}{m}\right)$ the function $f^{\mathcal{P}^m(\Delta)}$ is defined as follows

$$f^{\mathcal{P}^m(\Delta)}(\alpha_1, \dots, \alpha_k) := \frac{1}{m} \min_{t^1, \dots, t^k \in D^m : \{t^l\} = \alpha_l} \sum_{i=1}^{m} f^{\Delta}(t_i^1, \dots, t_i^k).$$

(Here we denote by $\{t^l\}$ the multiset whose elements are the coordinates of t^l.)

Lemma 4.3.3 ([96], Lemma 2.2)**.** *Let Δ be a valued structure with a finite domain, and $m \geq 2$. Then $\mathcal{P}^m(\Delta) \to_f \Delta$ if, and only if, Δ has an m-ary fully symmetric fractional polymorphism.*

Lemma 4.3.4. *Let Γ be a valued structure (with a finite signature τ), and let Δ be a valued structure with a finite domain such that $\Delta \to_f \Gamma$, and $m \geq 2$. If Γ has an m-ary fully symmetric fractional polymorphism, then $\mathcal{P}^m(\Delta) \to_f \Gamma$.*

Lemma 4.3.4 is a generalisation of Lemma 4.3.3 to valued structures with arbitrary domains. However, while Lemma 4.3.3 follows directly from the definition of $\mathcal{P}^m(\Delta)$, our proof of Lemma 4.3.4 is inspired by the proof of [8], Lemma 2.4.

Proof of Lemma 4.3.4. Let C and D be, respectively, the domain of Γ and the domain of Δ, respectively. Let χ be a fractional homomorphism from Δ to Γ and let ω be an m-ary fully symmetric fractional polymorphism of Γ. For every $g \in \mathrm{Supp}(\omega) \subseteq \mathcal{O}_C^{(m)}$ and every $h \in \mathrm{Supp}(\chi) \subseteq C^D$, we define

$$(g \circ h) \colon \left(\binom{D}{m}\right) \to C$$

by setting, for $\alpha = \{a^1, \dots, a^m\} \in \left(\binom{D}{m}\right)$,

$$(g \circ h)(\alpha) = g(h(a^1), \dots, h(a^m)).$$

Observe that $(g \circ h)$ is well defined as g is fully symmetric (the order of $h(a^1), \dots, h(a^m)$ does not matter). We define the function $\omega' \colon C^{\left(\binom{D}{m}\right)} \to \mathbb{Q}_{\geq 0}$ by setting, for every $g \in C^{\left(\binom{D}{m}\right)}$,

$$\omega'(g') = \sum_{g \in \mathrm{Supp}(\omega)} \sum_{h \in \mathrm{Supp}(\chi) : g \circ h = g'} \omega(g)\chi(h).$$

We claim that ω' is a fractional homomorphism from $\mathcal{P}^m(\Delta)$ to Γ. Indeed, the support $\mathrm{Supp}(\omega') = \{(g \circ h) \mid g \in \mathrm{Supp}(\omega), h \in \mathrm{Supp}(\chi)\}$, is finite as the support of ω and the support of χ are finite. It also holds that

$$\sum_{g' \in \mathrm{Supp}(\omega')} \omega'(g') = \sum_{g \in \mathrm{Supp}(\omega)} \omega(g) \sum_{h \in \mathrm{Supp}(\chi)} \chi(h) = \sum_{g \in \mathrm{Supp}(\omega)} \omega(g) = 1.$$

Furthermore, for every k-ary $f \in \tau$ and tuple $(\alpha_1, \ldots, \alpha_k) \in \left(\binom{D}{m}\right)^k$, with $\alpha_i = \{\alpha_i^1, \ldots, \alpha_i^m\}$, it holds that

$$\begin{aligned}
&\sum_{g' \in C^{\left(\binom{D}{m}\right)}} \omega'(g') f^{\Gamma}(g'(\alpha_1, \ldots, \alpha_k)) \\
=&\sum_{g \in \mathrm{Supp}(\omega)} \sum_{h \in \mathrm{Supp}(\chi)} \omega(g)\chi(h) f^{\Gamma}(g(h(\alpha_1^1), \ldots, h(\alpha_1^m)), \ldots, g(h(\alpha_k^1), \ldots, h(\alpha_k^m))) \\
\leq&\sum_{h \in \mathrm{Supp}(\chi)} \chi(h) \left(\frac{1}{m} \sum_{i=1}^{m} f^{\Gamma}\left(h(\alpha_1^{\pi_1(i)}), \ldots, h(\alpha_1^{\pi_1(i)}) \right) \right) \qquad (4.5) \\
=&\frac{1}{m} \sum_{i=1}^{m} \sum_{h \in \mathrm{Supp}(\chi)} \chi(h) f^{\Gamma}\left(h(\alpha_1^{\pi_1(i)}), \ldots, h(\alpha_1^{\pi_1(i)}) \right) \\
\leq&\frac{1}{m} \sum_{i=1}^{m} f^{\Delta}(\alpha_1^{\pi_1(i)}, \ldots, \alpha_k^{\pi_k(i)}), \qquad (4.6)
\end{aligned}$$

for every $\pi_1, \ldots, \pi_k \in S_m$. Inequality (4.5) holds because ω is a fully symmetric fractional polymorphism of Γ and Inequality (4.6) holds because χ is a fractional homomorphism from Δ to Γ. Then, in particular, it holds that

$$\begin{aligned}
&\sum_{g' \in C^{\left(\binom{D}{m}\right)}} \omega'(g') f^{\Gamma}(g'(\alpha_1, \ldots, \alpha_k)) \leq \frac{1}{m} \min_{t^1, \ldots, t^k \in D^m : \{t^l\} = \alpha_l} \sum_{i=1}^{m} f^{\Delta}(t_i^1, \ldots t_i^k) \\
=&f^{\mathcal{P}^m(\Delta)}(\alpha_1, \ldots, \alpha_k),
\end{aligned}$$

which concludes the proof. □

Theorem 4.3.5. *Let Γ be a valued structure with a finite signature having fully symmetric fractional polymorphisms of all arities. If there exists an efficient sampling algorithm for Γ, then Algorithm 1 correctly solves* VCSP(Γ) *(in polynomial time).*

Proof. Let I be an instance of VCSP(Γ) with variables $V_I = \{x_1, \ldots, x_n\}$, objective function $\phi_I(x_1, \ldots, x_n) = \sum_{j \in J} \gamma_j(x^j)$ where J is a finite set of indices, $\gamma_j \in \Gamma$, and $x^j \in V_I^{\mathrm{ar}(f_j)}$, and threshold u_I. Let Δ be the finite-domain valued structure computed by the sampling algorithm for Γ on input $|V_I|$.

Let C be the (possibly infinite) domain of Γ and $D \subseteq C$ the finite domain of Δ. Note that if $\mathrm{BLP}(I, \Delta) \not\leq u_I$ then $\min_{d \in D^n} \phi_I(d) \not\leq u_I$ which implies $\inf_{c \in C^n} \phi_I(c) \not\leq u_I$ since D was produced by the sampling algorithm for Γ on input $|V_I|$. We may, therefore, safely reject. Otherwise, if $\mathrm{BLP}(I, \Delta) \leq u_I$, then $\min_{\alpha \in (\binom{D}{m})^n} \phi_I^{\mathcal{P}^m(\Delta)}(\alpha) \leq \mathrm{BLP}(I, \Delta)$. The proof of this previous statement is contained in the first part of the proof of Theorem 3.2 in [96]; we report it here for completeness. Let $(\lambda^\star, \mu^\star)$ be an optimal solution to $\mathrm{BLP}(I, \Delta)$ and let M be a positive integer such that $M \cdot \lambda^\star$, and $M \cdot \mu^\star$ are both integral. Let $\nu \colon V_I \to \left(\binom{D}{M}\right)$ be defined by mapping the variable x_i to the multiset in which the elements are distributed accordingly to $\mu^\star_{x_i}$, i.e., for every $a \in D$ the number of occurrences of a in $\nu(x_i)$ is equal to $M\mu^\star_{x_i}(a)$. Let f_j be a k-ary function symbol in τ that occurs in a term $f_j(x^j)$ of the objective function ϕ_I. Now we write

$$M \cdot \sum_{t \in D^k} \lambda^\star_j(t) f_j^\Delta(t) = f_j^\Delta(\alpha^1) + \cdots + f_j^\Delta(\alpha^M),$$

where the $\alpha^i \in D^k$ are such that $\lambda^\star_j(t)$-fractions are equal to t. Let us define $\alpha'_l := (\alpha^1_i, \ldots, \alpha^M_i)$ for $1 \leq i \leq k$. We get

$$\begin{aligned}
&\sum_{t \in D^k} \lambda^\star_j(t) f_j^\Delta(t) = \frac{1}{M} \sum_{i=1}^{M} f_j^\Delta(\alpha^i) = \frac{1}{M} \sum_{i=1}^{M} f_j^\Delta(\alpha^i_1, \ldots, \alpha^i_k) \\
\geq &\frac{1}{M} \min_{t^1, \ldots, t^k \in D^M : \{t^l\} = \{\alpha'_l\}} \sum_{i=1}^{M} f_j^\Delta(t^1_i, \ldots, t^k_i) = f_j^{\mathcal{P}^M(\Delta)}(\alpha'_1, \ldots, \alpha'_k) \\
= &f_j^{\mathcal{P}^M(\Delta)}(\nu(x)),
\end{aligned}$$

where the last equality follows as the number of a's in α'_i is

$$M \cdot \sum_{t \in D^k : t^i = a} \lambda^\star_j(t) = M \cdot \mu^\star_{x_i}(a).$$

Then

$$\begin{aligned}
\mathrm{BLP}(I, \Delta) = &\sum_{j \in J} \sum_{t \in D^{\mathrm{ar}(f_j)}} \lambda^\star_j(t) f_j^\Delta(t) = \sum_{j \in J} \left(\sum_{t \in D^{\mathrm{ar}(f_j)}} \lambda^\star_j(t) f_j^\Delta(t) \right) \\
\geq &\sum_{j \in J} \left(f_j^{\mathcal{P}^M(\Delta)}(\nu(x)) \right) \geq \min_{\alpha \in (\binom{D}{m})^n} \phi_I^{\mathcal{P}^m(\Delta)}(\alpha).
\end{aligned}$$

Since we assumed $\mathrm{BLP}(I, \Delta) \leq u_I$, we obtain

$$\min_{\alpha \in \left(\binom{D}{m}\right)^n} \phi_I^{\mathcal{P}^m(\Delta)}(\alpha) \leq u_I.$$

Moreover, since Γ has fully symmetric fractional polymorphisms of all arities, Lemma 4.3.4 implies the existence of a fractional homomorphism

$$\omega \colon C^{\left(\binom{D}{M}\right)} \to \mathbb{Q}_{\geq 0}\,.$$

From Proposition 1.4.3 it follows that

$$\inf_{c \in C^n} \phi_I^{\Gamma}(c) \leq \min_{\alpha \in \left(\binom{D}{m}\right)^n} \phi_I^{\mathcal{P}^m(\Delta)}(\alpha) \leq \mathrm{BLP}(I, \Delta) \leq u_I.$$

□

We remark that Theorem 4.3.5 generalises the following known result (that we used in Section 3.1.1 to prove the polynomial-time complexity of the CSP for max-closed PLH relational structures).

Theorem 3.1.11 ([8], Theorem 2.5)**.** *Let $\mathfrak{A}$ be a structure over a finite relational signature with totally symmetric polymorphisms of all arities. If there exists an efficient sampling algorithm for $\mathfrak{A}$, then* $\mathrm{CSP}(\mathfrak{A})$ *is in P.*

More precisely, we extended Theorem 3.1.11 to valued structures and, at the same time, to the weaker assumption of having fully symmetric fractional polymorphisms of all arities rather than totally symmetric fractional polymorphisms of all arities. The following example is adapted from [77] (Example 99) and exhibits a PL valued structure having fully symmetric polymorphisms of all arities but having no totally symmetric polymorphism of arity 3.

Example 4.3.6. Let us consider the PL valued structure Γ with domain $\mathbb{Q}$ and signature $\{f_+, f_-\}$ such that $f_+^\Gamma, f_-^\Gamma \colon \mathbb{Q}^3 \to \mathbb{Q} \cup \{+\infty\}$ are defined by

$$f_+^\Gamma(x_1, x_2, x_3) := \begin{cases} x_1 + x_2 + x_3 & \text{if } x_1 + x_2 + x_3 \geq 1 \\ +\infty & \text{otherwise,} \end{cases}$$

and

$$f_-^\Gamma(x_1, x_2, x_3) := \begin{cases} x_1 + x_2 + x_3 & \text{if } x_1 + x_2 + x_3 \leq -1 \\ +\infty & \text{otherwise.} \end{cases}$$

Clearly, the cost functions f_+^Γ, and f_-^Γ are PL and it is easy to see that they are convex. As all cost functions in Γ are PL and convex, by Proposition 1.6.8, the valued structure Γ is improved by the fully symmetric fractional operations $\omega^{(k)}_{conv}$ for every $k \geq 2$, i.e., Γ has fully symmetric fractional polymorphisms of all arities. We already observed that the fractional operations $\omega^{(k)}_{conv}$ are not totally symmetric for $k \geq 3$ (see Example 1.4.19).

Assume now that ω is a ternary totally symmetric fractional polymorphism of Γ and let $t\colon\ \mathbb{Q}^3 \to \mathbb{Q}$ be a totally symmetric operation in $\mathrm{Supp}(\omega)$, then, in particular, t is a polymorphism of $\mathrm{Feas}(\Gamma)$, i.e., t preserves the feasibility relations

$$\mathrm{dom}(f^{\Gamma}_{+}) = \{(x_1, x_2, x_3) \in \mathbb{Q}^3 \mid x_1 + x_2 + x_3 \geq 1\}, \text{ and}$$
$$\mathrm{dom}(f^{\Gamma}_{-}) = \{(x_1, x_2, x_3) \in \mathbb{Q}^3 \mid x_1 + x_2 + x_3 \leq -1\}.$$

By the total symmetry of t, there exists $a \in \mathbb{Q}$ such that

$$\begin{aligned} &t(1,1,-1) = t(1,-1,1) = t(-1,1,1) \\ =&t(-1,-1,1) = t(-1,1,-1) = t(1,-1,-1) = a. \end{aligned}$$

Observe that $(1,1,-1),(1,-1,1),(-1,1,1) \in \mathrm{dom}(f^{\Gamma}_{+})$, then, by applying t componentwise we get $(a,a,a) \in \mathrm{dom}(f^{\Gamma}_{+})$, i.e., $a \geq \frac{1}{3}$; on the other hand, $(-1,-1,1),(-1,1,-1),(1,-1,-1) \in \mathrm{dom}(f^{\Gamma}_{-})$, then, by applying t componentwise we get $(a,a,a) \in \mathrm{dom}(f^{\Gamma}_{-})$, i.e., $a \leq -\frac{1}{3}$, which is a contradiction. △

Summary and Outlook

We have given a sufficient condition for valued structures admitting an efficient sampling algorithm to be polynomial-time tractable: if the valued structure has fully symmetric fractional polymorphisms of all arities, then the VCSP is correctly solved in polynomial time by a combination of the sampling algorithm with the basic linear programming relaxation. In the next chapter, we apply Theorem 4.3.5 to prove the polynomial-time tractability of VCSPs for some classes of PLH valued structures.

Chapter 5

Complexity Results for PLH Valued Structures

In this chapter, we apply the results of Chapters 3 and 4 to study the computational complexity of the VCSP for concrete PLH valued structures (Sections 5.1, and 5.2), and we give a sufficient condition for the *maximal tractability* of classes of PLH valued structures (Section 5.3). Submodular PLH valued structures are not among the classes of valued structures that we examine in the present chapter because we decided to dedicate them a whole chapter (Chapter 6).

The following corollary is an immediate consequence of Theorem 4.3.5 and the existence of an efficient sampling algorithm for PLH valued structures with finite signatures (Corollary 3.2.11).

Corollary 5.0.1. *Let Γ be a PLH valued structure with a finite signature that is improved by fully symmetric fractional operations of all arities. Then* VCSP(Γ) *can be solved in polynomial time.*

In the next sections, we survey classes of PLH valued structures satisfying the hypothesis of Proposition 5.0.1.

5.1 Convex PLH Valued Structures

In this section, we study the computational complexity of convex PLH valued structures. Since PLH valued structures are in particular PL, we already know that the VCSP for a convex PLH valued structure can be solved in polynomial time (see Chapter 2; we show that the polynomial-time solvability of the VCSP for convex PLH valued structures can also be obtained from an application of Corollary 5.0.1.

We have already seen that a convex valued structure has fully symmetric fractional polymorphisms of all arities (see Proposition 1.6.8). Therefore, the next corollary directly follows from Proposition 1.6.8 and Corollary 5.0.1.

Corollary 5.1.1. *Let Γ be a PLH valued structure with a finite signature such that the cost functions in Γ are convex. Then* VCSP(Γ) *can be solved in polynomial time.*

5.2 Componentwise Decreasing PLH Valued Structures

In this section, we study the computational complexity of componentwise decreasing (and componentwise increasing) PLH valued structures.

Definition 5.2.1. Let $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ be an n-ary function. We say that

- f is *componentwise decreasing* if
$$f(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n) \geq f(x_1, \dots, x_{i-1}, z_i, x_{i+1}, \dots, x_n),$$
for every $y_i < z_i$, $1 \leq i \leq k$, and $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \in \mathbb{Q}$;

- f is *componentwise increasing* if
$$f(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n) \leq f(x_1, \dots, x_{i-1}, z_i, x_{i+1}, \dots, x_n),$$
for every $y_i < z_i$, $1 \leq i \leq k$, and $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \in \mathbb{Q}$.

A valued structure Γ is said to be *componentwise decreasing* (*componentwise increasing*, respectively) if all the cost functions in Γ are componentwise decreasing (componentwise increasing, respectively).

Example 5.2.2. It is easy to see that the PLH function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by $f(x, y) := -\max(x, y)$ is a componentwise decreasing function, which is *not* submodular and *not* convex. $\triangle$

In [34], componentwise decreasing functions, and componentwise increasing functions are respectively called *antitone* functions and *monotone* functions.

Corollary 5.2.3. *Let Γ be a componentwise decreasing PLH valued structure with a finite signature. Then* VCSP(Γ) *can be solved in polynomial time.*

Proof. The proof follows from Proposition 5.0.1, since for every $k \geq 2$ the fully symmetric fractional operation $\omega^{(k)}_{\max}$, defined in Example 1.4.18, improves Γ as the next lemma shows. □

Lemma 5.2.4. *A function $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ is componentwise decreasing if, and only if, it is improved by $\omega^{(k)}_{\max}$, for every $k \geq 2$.*

Proof. Let $k \geq 2$ and let us first assume that f is a componentwise decreasing function with arity 1. Let us consider $x_1, \ldots, x_k \in \mathbb{Q}$ and let us assume without loss of generality that $x_1 = \max(x_1, \ldots, x_k)$. Then we have that

$$f(\max(x_1,\ldots,x_k)) = f(x_1) = \frac{\overbrace{f(x_1)+\cdots+f(x_1)}^{k \text{ times}}}{k} \leq \frac{f(x_1)+\cdots+f(x_k)}{k},$$

i.e., $\omega^{(k)}_{\max}$ is a fractional polymorphism of f. Assume now $n \geq 2$ and that every $(n-1)$-ary componentwise decreasing function is improved by $\omega^{(k)}_{\max}$. We prove that for every $n \in \mathbb{N}$, an n-ary componentwise decreasing function f is improved by $\omega^{(k)}_{\max}$. Let us fix $x^1, \ldots, x^k \in \mathbb{Q}^n$. The restricted function

$$f(\cdot, \max(x^1_n, \ldots, x^k_n))\colon \mathbb{Q}^{n-1} \to \mathbb{Q} \cup \{+\infty\},$$

which maps every $(z_1, \ldots, z_{n-1}) \in \mathbb{Q}^{n-1}$ to $f(z_1, \ldots, z_{n-1}, \max(x^1_n, \ldots, x^k_n))$, is clearly componentwise decreasing and therefore, by the inductive hypothesis, it is improved by $\omega^{(k)}_{\max}$, that is,

$$\begin{aligned} & f(\max(x^1_1,\ldots,x^k_1),\ldots,\max(x^1_{n-1},\ldots,x^k_{n-1}),\max(x^1_n,\ldots,x^k_n)) \\ \leq\ & \frac{f(x^1_1,\ldots,x^1_{n-1},\max(x^1_n,\ldots,x^k_n))+\cdots+f(x^k_1,\ldots,x^k_{n-1},\max(x^1_n,\ldots,x^k_n))}{k}. \end{aligned}$$

Again by the fact that f is componentwise decreasing we get that the right-hand side of the last inequality is

$$\leq \frac{1}{k}(f(x^1_1,\ldots,x^1_{n-1},x^1_n)+\cdots+f(x^k_1,\ldots,x^k_{n-1},x^k_n)),$$

i.e., f is improved by $\omega^{(k)}_{\max}$.

Conversely, if $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ is improved by $\omega^{(k)}_{\max}$ for all $k \geq 2$, then, in particular, it is improved by $\omega_{\max} = \omega^{(2)}_{\max}$. For all $i \in \{1, \ldots, n\}$ and $x_1, \ldots, x_{i-1}, y_i, z_i, x_{i+1}, \ldots, x_n \in \mathbb{Q}$ such that $y_i < z_i$ it holds that

$$\begin{aligned} & f(x_1,\ldots,x_{i-1},z_i,x_{i+1},\ldots,x_n) \\ \leq\ & \frac{f(x_1,\ldots,x_{i-1},y_i,x_{i+1},\ldots,x_n)}{2} + \frac{f(x_1,\ldots x_{i-1},z_i,x_{i+1},\ldots,x_n)}{2}. \end{aligned}$$

It follows that

$$f(x_1, \ldots, x_{i-1}, z_i, x_{i+1}, \ldots, x_n) \leq f(x_1, \ldots x_{i-1}, y_i, x_{i+1}, \ldots, x_n),$$

i.e., f is componentwise decreasing. $\square$

Lemma 5.2.4 and Corollary 5.2.3 may be shown to have a dual form that holds in the case of componentwise increasing cost functions.

5.3 A Condition for Maximal Tractability of PLH Valued Structures

In this section, we give a sufficient condition for the maximal tractability of a specific subclass of valued structures within the class of all PLH valued structures. We apply this result to prove that the valued structures containing all componentwise decreasing PLH cost functions, and all componentwise increasing PLH cost functions, respectively, are maximally tractable within the class of all PLH valued structures.

Definition 5.3.1. Let us fix a set D. Let $\mathcal{V}$ be a class of valued structures with domain D and let Γ be a valued structure in $\mathcal{V}$. We say that Γ is *maximally tractable within* $\mathcal{V}$ if

- VCSP(Γ') can be solved in polynomial time, for every valued finite reduct Γ' of Γ; and
- for every valued structure Δ in $\mathcal{V}$ that is an expansion of Γ, there exists a value finite reduct Δ' of Δ such that VCSP(Δ') is NP-hard.

Definition 5.3.2. Given a finite domain $D \subset \mathbb{Q}$ and a partial function $f \colon D^n \to \mathbb{Q}$ we define the *canonical extension* of f as the PLH function $\hat{f} \colon \mathbb{Q}^n \to \mathbb{Q}$, by

$$\hat{f}(x) = \begin{cases} f(x) & x \in D^n \\ +\infty & \text{otherwise.} \end{cases}$$

We need the following lemma.

Lemma 5.3.3. *Let $\omega \colon \mathcal{O}_{\mathbb{Q}}^{(k)} \to \mathbb{Q}_{\geq 0}$ be a conservative fractional operation with a finite support, for some $k \geq 1$. Assume that*

- *for every PLH valued structure $\Gamma \subset \mathrm{Imp}(\omega)$ with a finite signature, the problem* VCSP(Γ) *can be solved in polynomial time;*
- *the class of finite-domain valued structures that are improved by ω is maximally tractable within the class of finite-domain valued structures.*

Then the class of all PLH valued structures that are improved by ω is maximally tractable within the class PLH valued structures.

Proof. Let $f^{\mathbb{Q}}\colon \mathbb{Q}^m \to \mathbb{Q} \cup \{+\infty\}$ be a PLH cost function that is not improved by ω, i.e., there exist k many tuples $a^1, \dots, a^k \in \mathbb{Q}^m$ such that

$$\frac{1}{k}\sum_{i=1}^{k} f^{\mathbb{Q}}(a^i) < \sum_{g \in \operatorname{Supp}(\omega)} f^{\mathbb{Q}}(g(a^1, \dots, a^k)).$$

Let

$$D := \{a_1^1, \dots, a_m^1, a_1^2, \dots, a_m^{k-1}, a_1^k, \dots, a_m^k\} \subset \mathbb{Q},$$

and let Δ be the valued structure with domain D, such that its signature τ contains a function symbol for every cost functions on D that is improved by ω. Notice that the restriction $f^{\mathbb{Q}}|_D$ is not improved by ω, for our choice of D. Therefore, by hypothesis, there exists a valued structure Δ' having domain D and signature $\tau' \cup \{f\}$, where $\tau' \subseteq \tau$ is finite and the cost function $f^{\Delta'} = f^{\mathbb{Q}}_{|}D$, such that VCSP$(\Delta')$ NP-hard.

We define Γ' to be the valued structure with domain D and signature $\tau' \cup \{f, \chi_D\}$ such that the functions symbols in the signature are interpreted as follows:

- for every $g \in \tau'$, the cost function $g^{\Gamma'}$ is the canonical extension of $g^{\Delta'}$,
- the cost function $f^{\Gamma'}$ is $f^{\mathbb{Q}}$, and
- the unary cost function $\chi_D^{\Gamma'}\colon \mathbb{Q} \to \mathbb{Q} \cup \{+\infty\}$ is defined, for every $x \in \mathbb{Q}$ as
$$\chi_D^{\Gamma'}(x) = \begin{cases} 0 & \text{if } x \in D \\ +\infty & \text{if } x \in \mathbb{Q} \backslash D. \end{cases}$$

Observe that, for every $g \in \tau'$, since ω is a fractional polymorphism of $g^{\Delta'}$ it is also a fractional polymorphism of $g^{\Gamma'}$. Moreover, $\chi_D^{\Gamma'}$ is PLH and it is improved by ω, since ω is conservative.

The valued structure Γ' is both an extension and an expansion of Δ'. We claim that VCSP(Γ') in NP-hard. Indeed, for every instance $I := (V, \phi_I, u)$ of VCSP(Δ'), with $V := \{v_1, \dots, v_n\}$, we define the instance $J := (V, \phi_J, u)$ of VCSP(Γ') such that

$$\phi_J(v_1, \dots, v_n) := \phi_I(v_1, \dots, v_n) + \sum_{i=1}^{n} \chi_D(v_i).$$

Because of the terms involving χ_D, an assignment $h\colon V \to \mathbb{Q}$ is such that $\phi_J^{\Gamma'}(h(v_1), \dots, h(v_n))$ is smaller than $+\infty$ if, and only if, $h(v_i) \in D$ for all

$v_i \in V$. In this case,

$$\phi_J^{\Gamma'}(h(v_1), \dots, h(v_n)) = \phi_I^{\Delta'}(h(v_1), \dots, h(v_n)).$$

Therefore, deciding whether there exists an assignment $h\colon V \to \mathbb{Q}$ such that

$$\phi_J^{\Gamma'}(h(v_1), \dots, h(v_n)) \le u$$

is equivalent to decide whether there exists an assignment $h'\colon V \to D$ such that

$$\phi_I^{\Delta'}(h'(v_1), \dots, h'(v_n)) \le u.$$

Since J is computable in polynomial time from I, the NP-hardness of VCSP(Γ') follows from the NP-hardness of VCSP(Δ'). □

To prove the maximal tractability of the valued structure containing all componentwise decreasing PLH cost function within the class of PLH valued structures, we make use of the following result.

Theorem 5.3.4 (Cohen-Cooper-Jeavons-Krokhin, [34], Theorem 6.15)**.** *Let D be a finite and totally ordered set. Then the valued structure containing all componentwise decreasing cost functions over D is maximally tractable within the class of all valued structures with domain D.*

Theorem 5.3.5. *The valued structure containing all componentwise decreasing PLH cost functions is maximally tractable within the class of PLH valued structures.*

Proof. Componentwise decreasing cost functions over $\mathbb{Q}$ are characterised by a conservative fractional polymorphism (see Lemma 5.2.4) and for every componentwise decreasing PLH valued structure Γ with a finite signature, VCSP(Γ) can be solved in polynomial time (Corollary 5.2.3). Then the proof follows from Lemma 5.3.3 and Theorem 5.3.4. □

Theorem 5.3.5 may be shown to have a dual form that holds in the case of componentwise increasing cost functions.

Summary and Outlook

We have shown the polynomial-time solvability of the VCSP for convex PLH valued structures, and componentwise decreasing PLH valued structures. We also have given a condition for the maximal tractability of a PLH valued structure. Such maximal tractability results are of particular importance for the more ambitious goal to classify the complexity of the VCSP for all classes of PLH cost functions: to prove a complexity dichotomy it suffices to identify all maximally tractable classes.

Chapter 6

The Complexity of Submodular PLH VCSPs

Submodular functions are arguably the most important class of cost functions in combinatorial optimisation and operational research. They naturally appear in several scientific fields such as, for example, economics, game theory, machine learning, social network, and computer vision (see, e.g., [41, 74]), and guided the research on finite-domain VCSPs for some time (see, e.g., [34, 60]). We gave the definition of submodular functions in Section 1.5. In the present section, we prove that the VCSP for submodular PLH valued structures is polynomial-time solvable and that submodularity is, indeed, a condition of maximal tractability for PLH valued structures: adding any cost function that is not submodular leads to an NP-hard VCSP.

We aim to prove the following result.

Theorem 6.0.1. *Let* Γ *be a PLH valued structure with a finite signature. Assume that all cost functions in* Γ *are submodular. Then* $\mathrm{VCSP}(\Gamma)$ *can be solved in polynomial time.*

To prove Theorem 6.0.1, we exhibit two different polynomial-time algorithms, both relying on the results for PLH valued structures with finite signatures presented in Chapter 3. In Section 6.1, we show that submodular valued structures have fully symmetric fractional polymorphisms of all arities. Therefore submodular PLH valued structures with finite signatures satisfy the hypothesis of Theorem 5.0.1, that is, the VCSP for these valued structures can be solved in polynomial time. In Section 6.2, we interpret our PLH cost functions in the domain $\mathbb{Q}^\star$ and provide a fully combinatorial polynomial-time algorithm solving the VCSP for the obtained $\mathbb{Q}^\star$-valued finite-domain valued structure. In Section 6.3, we discuss the differences between the two algorithms presented and, finally, in Section 6.4 we show that the subclass of submodular PLH valued structures is maximal with respect to polynomial-time solvability within the class of PLH valued structures with finite signatures.

6.1 The Algorithm in $\mathbb{Q}$

We already observed, in Section 1.5, that a function over a totally ordered set D is submodular if, and only if, it is improved by the binary symmetric fractional operation $\omega_{\text{sub}} \colon \mathcal{O}_D^2 \to \mathbb{Q}_{\geq 0}$ such that

$$\omega_{\text{sub}}(g) = \begin{cases} \frac{1}{2} & \text{if } g = \max \\ \frac{1}{2} & \text{if } g = \min \\ 0 & \text{otherwise.} \end{cases}$$

In fact, there is another equivalent characterisation of submodularity based on fractional polymorphisms. For every $k \geq 2$ and every $i \in \{1, \dots, k\}$ we define $s_i^{(k)} \colon D^k \to D$ to be the operation returning the i-th smallest of its arguments with respect to the total order on D. Observe that for $k = 2$, we have $s_1^{(2)} = \min$ and $s_2^{(2)} = \max$, and $\omega_{\text{sub}}^{(2)}$ is the fractional operation ω_{sub} characterising submodular functions. We define, for every $k \geq 2$, the k-ary fractional operation $\omega_{\text{sub}}^{(k)} \colon \mathcal{O}_D^{(k)} \to \mathbb{Q}_{\geq 0}$ having support

$$\text{Supp}(\omega_{\text{sub}}^{(k)}) = \{s_i^{(k)} \mid 1 \leq i \leq k\}$$

by setting

$$\omega_{\text{sub}}^{(k)}(g) := \begin{cases} \frac{1}{k} & \text{if } g \in \text{Supp}(\omega_{\text{sub}}^{(k)}) \\ 0 & \text{otherwise.} \end{cases}$$

The operations $s_i^{(k)}(x^1, \dots, x^k)$ are totally symmetric for all $i \in \{1, \dots, k\}$ and all $k \geq 2$; therefore, the fractional operations $\omega_{\text{sub}}^{(k)}$ are totally symmetric for all $k \geq 2$.

Proposition 6.1.1. *Let D be a totally ordered set and $f \colon D^n \to \mathbb{Q} \cup \{+\infty\}$ be a submodular function. Then the fractional operation $\omega_{\text{sub}}^{(k)}$ improves the function f, for all $k \geq 2$.*

Proof. Clearly,

$$\sum_{g \in \text{Supp}(\omega_{\text{sub}}^{(k)})} \omega_{\text{sub}}^{(k)}(g) = 1.$$

We want to prove that for all $k \geq 2$ and for all $x^1, \dots, x^k \in D^n$ it holds that

$$\frac{1}{k} \sum_{i=1}^{k} f(s_i^{(k)}(x^1, \dots, x^k)) \leq \frac{1}{k} \left(f(x^1) + \dots + f(x^k) \right). \tag{6.1}$$

By using the submodularity of f we can write

$$f(x^1) + \cdots + f(x^k) = \frac{1}{k-1} \sum_{1 \le i < j \le k} \big(f(x^i) + f(x^j)\big)$$

$$\ge \frac{1}{k-1} \sum_{1 \le i < j \le k} \big(f(\min(x^i, x^j)) + f(\max(x^i, x^j))\big)$$

$$= \sum_{1 \le i < j \le k} \frac{f(\min(x_1^i, x_1^j), \ldots, \min(x_n^i, x_n^j)) + f(\max(x_1^i, x_1^j), \ldots, \max(x_n^i, x_n^j))}{k-1}$$

$$\ge \sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \ldots, x_1^k), \ldots, s_i^{(k)}(x_n^1, \ldots, x_n^k)\right), \tag{6.2}$$

from which Inequality (6.1) follows. We prove Inequality (6.2) by induction on the arity n of f. Observe that for every coordinate $1 \le l \le n$, the following equality between multisets holds:

$$\begin{aligned} &\left\{\min(x_l^i, x_l^j) \mid 1 \le i < j \le k\right\} \cup \left\{\max(x_l^i, x_l^j) \mid 1 \le i < j \le k\right\} \\ &= \Big\{ \underbrace{s_1^{(k)}(x_l^1, \ldots, x_l^k)}_{k-1 \text{ occurrences}}, \underbrace{s_2^{(k)}(x_l^1, \ldots, x_l^k)}_{k-1 \text{ occurrences}}, \ldots, \underbrace{s_k^{(k)}(x_l^1, \ldots, x_l^k)}_{k-1 \text{ occurrences}} \Big\}. \end{aligned} \tag{6.3}$$

If f has arity $n = 1$, then Inequality (6.2) immediately follows from Equality (6.3). Let $n \ge 2$, assume that Inequality (6.2) is true for submodular functions of arity at most $n-1$, and let us prove it for submodular functions of arity n. From Equality (6.3) and the inductive hypothesis, it follows that there exist $(k-1)$-many permutations $\pi_1, \ldots, \pi_{k-1} \in S_k$ such that

$$\sum_{1 \le i < j \le k} \Big(f(\min(x_1^i, x_1^j), \ldots, \min(x_n^i, x_n^j)) + f(\max(x_1^i, x_1^j), \ldots, \max(x_n^i, x_n^j))\Big)$$

$$\ge \sum_{p=1}^{k-1} \sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \ldots, x_1^k), \ldots, s_i^{(k)}(x_{n-1}^1, \ldots, x_{n-1}^k), s_{\pi_p(i)}^{(k)}(x_n^1, \ldots, x_n^k)\right). \tag{6.4}$$

We claim that for every $p \in \{1, \ldots, k-1\}$ it holds that

$$\sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \ldots, x_1^k), \ldots, s_i^{(k)}(x_{n-1}^1, \ldots, x_{n-1}^k), s_{\pi_p(i)}^{(k)}(x_n^1, \ldots, x_n^k)\right)$$

$$\ge \sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \ldots, x_1^k), \ldots, s_i^{(k)}(x_{n-1}^1, \ldots, x_{n-1}^k), s_i^{(k)}(x_n^1, \ldots, x_n^k)\right). \tag{6.5}$$

To prove Inequality (6.5), let $j := \max\{i \in \{1, \ldots, k\} \mid \pi_p(i) \ne i\}$. Then there exists $l \in \{1, \dots, j-1\}$ such that $\pi_p(l) = j$. By the submodularity of f we have that

$$
\begin{aligned}
& f\left(s_j^{(k)}(x_1^1, \dots, x_1^k), \dots, s_j^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_{\pi_p(j)}^{(k)}(x_n^1, \dots, x_n^k)\right) \\
+ & f\left(s_l^{(k)}(x_1^1, \dots, x_1^k), \dots, s_l^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_j^{(k)}(x_n^1, \dots, x_n^k)\right) \\
\geq & f\left(s_j^{(k)}(x_1^1, \dots, x_1^k), \dots, s_j^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_j^{(k)}(x_n^1, \dots, x_n^k)\right) \\
+ & f\left(s_l^{(k)}(x_1^1, \dots, x_1^k), \dots, s_l^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_{\pi_p(j)}^{(k)}(x_n^1, \dots, x_n^k)\right).
\end{aligned}
$$

After this step

$$
\begin{aligned}
& \sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \dots, x_1^k), \dots, s_i^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_{\pi_p(i)}^{(k)}(x_n^1, \dots, x_n^k)\right) \\
\geq & \sum_{i=j}^{k} f\left(s_i^{(k)}(x_1^1, \dots, x_1^k), \dots, s_i^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_i^{(k)}(x_n^1, \dots, x_n^k)\right) \\
+ & \sum_{i=1}^{j-1} f\left(s_i^{(k)}(x_1^1, \dots, x_1^k), \dots, s_i^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_{\pi'_p(i)}^{(k)}(x_n^1, \dots, x_n^k)\right)
\end{aligned}
$$

where $\pi'_p \in S_k$ is the permutation defined by

$$
\pi'_p(i) = \begin{cases} \pi_p(j) & \text{if } i = l \\ j & \text{if } i = j \\ \pi_p(i) & \text{otherwise.} \end{cases}
$$

By reiterating the described procedure at most $j - 1 \leq k$ times for every $p \in \{1, \dots, k-1\}$, we get the claim. By Inequality (6.5), we can rewrite Inequality (6.4) as follows

$$
\begin{aligned}
& \sum_{1 \leq i < j \leq k} \left(f(\min(x_1^i, x_1^j), \dots, \min(x_n^i, x_n^j)) + f(\max(x_1^i, x_1^j), \dots, \max(x_n^i, x_n^j))\right) \\
\geq & \sum_{p=1}^{k-1} \sum_{i=1}^{k} f\left(s_i^{(k)}(x_1^1, \dots, x_1^k), \dots, s_i^{(k)}(x_{n-1}^1, \dots, x_{n-1}^k), s_i^{(k)}(x_n^1, \dots, x_n^k)\right) \\
= & \sum_{p=1}^{k-1} \sum_{i=1}^{k} f\left(s_i^{(k)}(x^1, \dots, x^k)\right) = (k-1) \sum_{i=1}^{k} f\left(s_i^{(k)}(x^1, \dots, x^k)\right),
\end{aligned}
$$

that is, Inequality (6.2) holds and this concludes the proof. □

The next corollary immediately follows from the total symmetry of fractional operations $\omega_{\text{sub}}^{(k)}$ and from Proposition 6.1.1.

Corollary 6.1.2. *Let Γ be a submodular PL (or PLH) valued structure. Then Γ has totally symmetric fractional polymorphisms of all arities.*

Let us recall from Chapter 5 the following result for PLH valued structures with finite signatures having fully symmetric fractional polymorphism of all arities.

Corollary 5.0.1. *Let Γ be a PLH valued structure with a finite signature that is improved by fully symmetric fractional operations of all arities. Then* $\mathrm{VCSP}(\Gamma)$ *can be solved in polynomial time.*

If Γ is a submodular PLH valued structure with a finite signature then by Corollary 6.1.2 we can solve the VCSP for the computed sample (Section 3.2) using the BLP, after running an efficient sampling algorithm for PLH valued structures (see Section 3.2.2).

Proof of Corollary 6.0.1. The statement is an immediate consequence of Corollary 5.0.1, since every submodular PLH valued structure has fully symmetric fractional polymorphisms of all arities (Corollary 6.1.2) and every totally symmetric fractional operation is fully symmetric. □

6.2 The Algorithm in $\mathbb{Q}^\star$

Another way to show the polynomial-time solvability of submodular PLH valued structures with finite signatures is by using Lemma 3.2.7 and interpreting the function symbols in the signature in $\mathbb{Q}^\star$.

We already observed that for the polynomial-time solvability of $\mathrm{VCSP}(\Gamma)$ we need, in particular, $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ be polynomial-time solvable. If Γ is a submodular PLH valued structure, then $\mathrm{Feas}(\Gamma)$ is a semilinear relational structure whose relations are

- first-order definable over $\mathfrak{L}$, and
- preserved by the polymorphisms max and min.

Therefore, the tractability of $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$, for a submodular PLH valued structure Γ with a finite signature, follows from Theorem 3.1.10 that we restate hereunder.

Theorem 3.1.10. *Let $\mathfrak{A}$ be a structure having finite relational signature τ. Assume that for every $R \in \tau$, the interpretation $R^{\mathfrak{A}}$ is PLH and preserved by* max. *Then* $\mathrm{CSP}(\mathfrak{A})$ *is polynomial-time solvable.*

As we already discussed in Section 1.8, one can extend Definition 1.1.3 considering VCSPs whose cost functions take values in any totally ordered ring containing $\mathbb{Q}$, and in particular in $\mathbb{Q}^\star$. We need to establish the basics of such extended VCSPs. More precisely, we need to prove Corollary 6.2.3 hereafter, that builds on a fully combinatorial algorithm (for definition see Section 1.8) due to Iwata and Orlin [56] (Theorem 6.2.2).

Definition 6.2.1. A collection $\mathcal{C}$ of subsets of a given set Q is said to be a *ring family* if it is closed under union and intersection.

Equivalently, a ring family is a distributive sublattice of $\mathcal{P}(Q)$ with respect to union and intersection, notably every distributive lattice can be represented in this form (Birkhoff's representation theorem). Computationally, we represent a ring family following Section 6 of [92]. Namely, fixed a representation for the elements of Q, the ring family $\mathcal{C}$ is represented by the smallest set $M \subseteq Q$ in $\mathcal{C}$, and an oracle that given an element of $v \in Q$ returns the smallest $M_v \subset Q$ in $\mathcal{C}$ such that $v \in M_v$. The construction in Section 6 of [92] proves that any algorithm capable of minimising submodular set functions can be used to minimise submodular set functions defined on a ring family represented in this way. Observe that this construction is fully combinatorial.

Theorem 6.2.2 (Iwata-Orlin [56] + Schrijver [92])**.** *There exists a fully combinatorial polynomial-time algorithm over $\mathbb{Q}$ that*

- *taking as input a finite set $Q = \{1, \ldots, n\}$ and a ring family, $\mathcal{C} \subseteq 2^Q$, represented as in [92, Section 6] (namely as above),*
- *having access to an oracle that computes a submodular set-function $\psi \colon \mathcal{C} \to \mathbb{Q}$,*

computes an element $S \in \mathcal{C}$ such that $\psi(S) = \min_{A \in \mathcal{C}} \psi(A)$ in time bounded by a polynomial $p(n)$ in the size n of the domain.

Corollary 6.2.3. *Let R be a totally ordered commutative ring with unit (for instance $\mathbb{Q}^\star$). Then, there exists a fully combinatorial polynomial-time algorithm over R that*

- *taking as input a finite set $Q = \{1, \ldots, n\}$ and a ring family, $\mathcal{C} \subseteq 2^Q$, represented as in Theorem 6.2.2,*
- *having access to an oracle that computes a submodular set-function $\psi \colon \mathcal{C} \to R$,*

computes an element $S \in \mathcal{C}$ such that $\psi(S) = \min_{A \in \mathcal{C}} \psi(A)$ in time bounded by a polynomial $p(n)$ in the size n of the domain.

Proof. Theorem 6.2.2 provides a fully combinatorial algorithm to minimise submodular functions that, over $\mathbb{Q}$, runs in polynomial time and computes a correct result. We claim that any such algorithm must be correct and run in polynomial time over R as well. To show this, we prove the following:

1. The algorithm terminates in time $p(n)$, where $p(n)$ is as in Theorem 6.2.2.
2. The output of the algorithm coincides with the minimum of ψ.

Let R_ψ denote the subgroup of the additive group $(R, +)$ generated by $\psi(\mathcal{C})$, and let $E_\psi := \{g_1, \dots, g_m\}$ be a set of free generators of R_ψ. For any tuple $r = (r_1, \dots, r_m) \in \mathbb{Q}^m$, we define a group homomorphism $h_r \colon R_\psi \to \mathbb{Q}$, by $h_r(g_i) = r_i$. Let $R_N := N \cdot (E_\psi \cup \{0\} \cup -E_\psi)$ be the subset of R consisting of the elements having the form $\pm x_1 \pm x_2 \dots \pm x_k$, with $k \leq N$ and $x_1, x_2, \dots, x_k \in E_\psi$.

In general, the group homomorphisms h_r are not order-preserving. We claim that for all N, there exists $r \in \mathbb{Q}^m$ such that $h_r|_{R_N}$ is order-preserving. Assume that no such tuple r exists. The inequalities denoting that $h_r|_{R_N}$ is order-preserving are expressed by a finite linear program P in the variables $r_1, \dots, r_m$. By the assumption and Farkas' lemma there is a linear combination (with coefficients in $\mathbb{Z}$) of the inequalities of P which is contradictory. Therefore, P is contradictory in any ordered ring, and, in particular, in R. However, $r_i = g_i$, for all $i \in \{1, \dots, m\}$, is a valid solution of P in R.

Fix $N := \hat{N} \cdot 2^{p(n)}$, where $\hat{N}$ is such that $\psi(S) \in R_{\hat{N}}$ for all $S \in \mathcal{C}$. For this N, let r be a tuple satisfying the claim. We run two parallel instances of the algorithm, one over R with input ψ, and the other in $\mathbb{Q}$ with input $h_r \circ \psi$. We can prove that the two runs are *exactly parallel* for at least $p(n)$ steps; therefore, since the second run stops within these $p(n)$ steps, also the first one must do so. Formally, we prove, in a register machine model, that, at each step $i \leq p(n)$, if a register contains the value g in the first run, it must contain the value $h_r(g)$ in the second. This is easily established proving by induction on i that a value computed at step i must be in $R_{\hat{N}\cdot 2^i}$. Point 1 is thus established.

For point 2, let $\min_R$ and $\min_\mathbb{Q}$ be the output of the algorithm over (ψ, R) and $(h_r \circ \psi, \mathbb{Q})$, respectively. The induction above shows, in particular, that $\min_\mathbb{Q} = h_r(\min_R)$. We know that $h_r(\min_R) = \min_\mathbb{Q} = h_r \circ \psi(S_0)$ for some S_0 and $h_r \circ \psi(S) \geq \min_\mathbb{Q} = h_r(\min_R)$ for each element S of $\mathcal{C}$. By our choice of N, the corresponding relations, $\min_R = \psi(S_0)$ and $\psi(S) \geq \min_R$ for each element S of $\mathcal{C}$, must hold in R. □

The following lemma is essentially contained in [34, Theorem 6.7], except that we replace the set of values by an arbitrary totally ordered commutative ring with unit R. To state the lemma properly, we need to observe that, given a submodular function f defined on Q^d, where $Q = \{1, \ldots, n\}$, we can associate to it the following ring family $\mathcal{C}_f \subseteq \mathcal{P}(Q \times \{1, \ldots, d\})$. For every $x = (x_1, \ldots, x_d) \in Q^d$ define

$$\mathcal{C}_x := \{(q, i) \mid q \in Q,\, q \leq x_i\} \subseteq Q \times \{1, \ldots, d\}$$

then we let C_f be the union of C_x for all x such that $f(x) < +\infty$.

Lemma 6.2.4. *Let $R \supseteq \mathbb{Q}$ be a totally ordered ring. There exists a fully combinatorial polynomial-time algorithm over R that*

- *taking as input a finite set $Q = \{1, \ldots, n\}$ and an integer d,*
- *having access to an oracle computing a partial submodular $f \colon Q^d \to R$,*
- *given the representation of $\mathcal{C}_f$ as in Theorem 6.2.2,*

computes an $x \in Q^d$ such that $f(x)$ is minimal, in time polynomial in n and d.

Proof. The problem reduces to minimising a submodular set-function on the ring family $\mathcal{C}_f$, for the details see the proof of Theorem 6.7 in [34]. $\square$

Proof of Theorem 6.0.1. Similarly to the proof of Theorem 3.1.10, we use a sampling technique. Namely, given an instance I of VCSP(Γ), we employ Lemma 3.2.7 to fix a valued finite substructure Γ_I of Γ, whose domain's cardinality is polynomial in $|V_I|$, having a subset $\mathbb{Q}_I^\star$ of $\mathbb{Q}_{-1,4}^\star$ as domain, such that the variables V_I of I have an assignment having cost at most u_I in $\mathbb{Q}$ if, and only if, they have one in $\mathbb{Q}_I^\star$. Once we have Γ_I, we conclude by Lemma 6.2.4.

The valued structure Γ_I obviously needs to have the same signature τ as Γ. For each function symbol $f \in \tau$ we consider a τ_0-formula ϕ_f defining f^Γ and we let f^{Γ_I} be the function defined in $\mathbb{Q}^\star$ by the same formula. By Proposition 3.1.4, the choice of ϕ_f is immaterial. Remains to define the domain $\mathbb{Q}_I^\star \subset \mathbb{Q}^\star$.

By quantifier-elimination (Theorem 1.2.8), any piecewise linear homogeneous cost function $f \colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ can be written as

$$f(x_1, \ldots, x_{\mathrm{ar}(f)}) \;=\; \begin{cases} t_{f,1} & \text{if } \chi_{f,1} \\ \cdots & \\ t_{f,m_f} & \text{if } \chi_{f,m_f} \\ +\infty & \text{otherwise} \end{cases}$$

where $t_{f,1}, \ldots, t_{f,m_f}$ are τ_0-terms, $\chi_{f,1}, \ldots, \chi_{f,m_f}$ are conjunctions of atomic τ_0-formulas with variables from $\{x_1, \ldots, x_{\mathrm{ar}(f)}\}$, and $\chi_{f,1}, \ldots, \chi_{f,m_f}$ define disjoint subsets of $\mathbb{Q}^n$. We fix such a representation for each of the cost functions in Γ, and we collect all the atomic formulas appearing in every one of the conjunctions $\chi_{f,i}$, for $f \in \Gamma$ and $1 \leq i \leq m_f$, into the set Φ. Clearly, Φ is finite and depends only on the fixed valued structure Γ. Finally, $\mathbb{Q}^\star_I := D_{\Phi,|V_I|}$, as defined in Lemma 3.2.7.

The size of $\mathbb{Q}^\star_I$ is clearly polynomial by simple inspection of the definition. Its representation has polynomial size as well if the numbers are represented in binary, and, with this representation, the evaluation of f^{Γ_I} for $f \in \tau$ takes polynomial time.

Given an assignment $\alpha\colon V_I \to \mathbb{Q}^\star_I$ of value $\leq u_I$ we have, a fortiori, an assignment $\colon V_I \to \mathbb{Q}^\star$ of value $\leq u_I$, hence, by the usual completeness of the first-order theory of ordered $\mathbb{Q}$-vector spaces, there is an assignment $\colon V_I \to \mathbb{Q}$ with the same property.

Finally, let $\beta\colon V_I \to \mathbb{Q}$ be an assignment having value $\leq u_I$. We need to find an assignment $\beta'\colon V_I \to \mathbb{Q}^\star_I$ with value $\leq u_I$. Let

$$\phi_I = \sum_{i=1}^{m} f_i(x_1^i, \dots, x_{\mathrm{ar}(f_i)}^i)$$

(cf. Definition 1.1.3). For each $i \in \{1, \dots, m\}$ select the formula χ_i among $\chi_{f_i,1}, \dots, \chi_{f_i,m_{f_i}}$ that is satisfied by the assignment β. Clearly, the conjunction of atomic τ_0-formulas $\chi := \bigwedge_{i=1}^m \chi_i$ is satisfiable. Moreover, ϕ_I restricted to the subset of $(\mathbb{Q}^\star)^{|V_I|}$ where χ holds is obviously linear. Then we can apply Lemma 3.2.7, and we get an assignment β' whose values are in $D_{\chi,|V_I|}$ (where, by a slight abuse of notation, we wrote χ for the set of conjuncts of χ). We conclude observing that $D_{\chi,|V_I|} \subseteq D_{\Phi,|V_I|} = \mathbb{Q}^\star_I$.

It remains to check that Lemma 6.2.4 applies to our situation. Clearly $R = \mathbb{Q}^\star$, the function f is the objective function described by ϕ_I, and we let $n = |\,\mathbb{Q}^\star_I\,|$ so that we identify Q with an enumeration of $\mathbb{Q}^\star_I$ in increasing order (which can be computed in polynomial time). The oracle computing f is straightforward to implement since sums and comparisons in $\mathbb{Q}^\star$ merely reduce to the corresponding componentwise operations on the coefficients. The representation of the ring family $\mathcal{C}_f$ requires a moment of attention. To construct the oracle, as well as to find the minimal element M, we need an algorithm that, given a variable $x \in V_I$ and a value $q \in \mathbb{Q}^\star_I$, finds the componentwise minimal feasible assignment $\alpha_x\colon V_I \to \mathbb{Q}^\star_I$ that gives to x a value $\geq q$ (which is easily seen to exist by observing that the set of feasible assignments is preserved by the operation min). This algorithm is easy to construct, observing that the associate feasibility problem is a *min-closed* CSP, i.e., a CSP for a relational structure that is invariant under the operation min. We describe how to find M; the procedure for M_v is essentially the same.

Suppose that for each variable $x \in V_I$ we can find the smallest element $\beta(x) \in \mathbb{Q}_I^\star$ such that there is a feasible assignment $\gamma_x \colon V_I \to \mathbb{Q}_I^\star$ such that $\gamma_x(x) = \beta(x)$, then, by the min-closure, $\beta = \min_{x \in V_I} \gamma_x$ is the minimal assignment. To find $\beta(x)$ it is sufficient to solve the feasibility problem, using Theorem 3.1.10, adding a constraint $x \geq k$ for increasing values of $k \in \mathbb{Q}_I^\star$. □

6.3 $\mathbb{Q}$ versus $\mathbb{Q}^\star$

Our special focus on submodularity is justified not only by the important role of submodularity in many scientific fields and, especially, in optimisation, but also by the fact that the class of submodular valued structures has been our training ground in studying the VCSP for PLH valued structures. Indeed, even if they are used to deal with submodular PLH cost functions, the two approaches presented in this chapter are quite generic, as much as the results. We think that both approaches are interesting and worthy of being presented here. While we do not have a preferred approach and we do not know which one of the two polynomial-time algorithms is more efficient, we give the reader some argument to compare the mathematical features of the two algorithms.

Both approaches rely on Lemma 3.2.7. The first approach (Section 6.1) consists in performing a polynomial-time many-one reduction that maps our VCSP to a ($\mathbb{Q}$-valued) finite-domain one and then *using* an approach known for ($\mathbb{Q}$-valued) finite-domain VCSPs. The second approach (Section 6.2) consists in interpreting our PLH functions over the domain $\mathbb{Q}^\star$, performing a polynomial-time many-one reduction that maps our VCSP to a ($\mathbb{Q}^\star$-valued) finite-domain one, and then *transferring* the known approaches for $\mathbb{Q}$ to the new domain.

By using the algorithm in $\mathbb{Q}$, we dispense with non-Archimedean extensions entirely, and we need not the existence of a fully combinatorial polynomial-time algorithm (nor a strongly polynomial-time one) that solves the finite-domain VCSP computed by our polynomial-time reduction. On the other hand, even if limiting our horizon to $\mathbb{Q}^\star_{-1,4} \simeq \mathbb{Q}^6$ might seem a simplification, in practice it makes things more complicated. For example, on several occasions in Section 6.2, we used the fact that $\mathbb{Q}^\star$ has a field structure that makes proofs more direct and intuitive. Also, explicitly choosing an $\boldsymbol{\epsilon}$ small enough obfuscates the ideas in the arguments, which are converted in an unsightly bureaucracy of inequalities. Even computationally, mapping everything to $\mathbb{Q}$ is tantamount to converting arrays of small integers into big numbers by concatenation, and therefore, it is hardly an improvement. Furthermore, in the general case, the existence of a polynomial-time computable rational value of ϵ that works is not necessary for the algorithm in $\mathbb{Q}^\star$, even though, in the PLH case, such an ϵ exists.

6.4 The Maximal Tractability of Submodular PLH Valued Structures

We show that submodular PLH valued structures are maximally tractable within the class of PLH valued structures. As in the case of componentwise decreasing (and componentwise increasing) PLH valued structures, the maximal tractability of submodular PLH valued structures is an application of Lemma 5.3.3. We employ the following result for submodular finite-domain valued structures.

Theorem 6.4.1 (Cohen-Cooper-Jeavons-Krokhin, [34], Theorem 6.7)**.** *Let D be a finite and totally ordered set. Then the valued structure containing all submodular cost functions over D is maximally tractable within the class of all valued structures with domain D.*

Theorem 6.4.2. *The valued structure containing all submodular PLH cost functions is maximally tractable within the class of PLH valued structures.*

Proof. Submodularity is characterised by a conservative fractional polymorphism (see Remark 1.5.3) and for every submodular PLH valued structure Γ with a finite signature, VCSP(Γ) can be solved in polynomial time (Theorem 6.0.1). Therefore, the proof follows from Lemma 5.3.3 and Theorem 6.4.1. $\square$

Summary and Outlook

We have proved that the VCSP for submodular PLH cost functions is polynomial-time solvable, by providing two algorithms. We remark that our algorithms not only decide the feasibility problem and whether there exists a solution of cost at most u, but can also be adapted to efficiently compute the infimum of the costs of all solutions (which might be $-\infty$), and decide whether the infimum is attained. The modification is straightforward, observing that, for both algorithms, the finite-domain valued structure computed does not depend on the threshold u. We have also shown that submodular PLH cost functions are maximally tractable within the class of PLH cost functions.

Chapter 7

Expressive Power

In the present chapter, we discuss how concepts and results from the algebraic theory for finite-domain VCSPs (see [33, 43]) can be transferred or extended to the infinite-domain setting.

The content of this chapter is the result of a collaboration with Friedrich Martin Schneider (Institut für Algebra, Technische Universität Dresden). We extend the notion of *expressive power* to valued structures over arbitrary domains (Section 7.1) and discuss the relationship between the expressive power and the set of fractional polymorphisms of the same infinite-domain valued structure (Sections 7.2 and 7.3). More precisely, in Section 7.3, we characterise the set of cost functions that are improved by all fractional polymorphisms of a valued structure in terms of *local expressive power*. Finally, in Section 7.4, we introduce fractional polymorphisms with supports of arbitrary cardinality.

The notion of expressive power of a valued structure Γ captures all cost functions that can be simulated in a specific way, as explained in Section 7.1, by cost functions from Γ.

Example 7.0.1. Let Δ be the PL valued structure containing the only cost function $\delta\colon\ \mathbb{Q}^2 \to \mathbb{Q}$ defined, for all $(x, y) \in \mathbb{Q}^2$, by

$$\delta(x, y) := \begin{cases} x - 1 & \text{if } y \leq 1 \\ y & \text{if } y > 1. \end{cases}$$

Although δ is not PLH, it is easy to observe that, for all $(x, y) \in \mathbb{Q}^2$, it can be written (in polynomial time) as the sum $\gamma_1(x, y) + \gamma_2(y)$ of the two cost functions $\gamma_1, :\ \mathbb{Q}^2 \to \mathbb{Q}$, and $\gamma_2\colon\ \mathbb{Q} \to \mathbb{Q}$ defined, respectively, by

$$\gamma_1(t, u) := \begin{cases} t & \text{if } u \leq 1 \\ 0 & \text{if } u > 1 \end{cases}, \text{ and } \quad \gamma_2(v) := \begin{cases} -1 & \text{if } v \leq 1 \\ v & \text{if } v > 1 \end{cases},$$

for all $t, u, v \in \mathbb{Q}$. It is easy to check that γ_1, and γ_2 are submodular and PLH. Let Γ be the valued structure having γ_1, and γ_2 as only cost functions. Then, every instance of VCSP(Δ) can be written in polynomial time as an instance of VCSP(Γ). Since Γ is a submodular PLH valued structure, VCSP(Γ) is polynomial-time solvable (see Chapter 6) and then so is VCSP(Δ). △

In the finite-domain case, the expressive power of a valued structure is defined as follows.

Definition 7.0.2 ([33], Definition 2.3)**.** Let Γ be a valued structure with a finite domain D. We say that a cost function $\rho\colon D \to \mathbb{Q} \cup \{+\infty\}$ is *expressible in* Γ if there exists an instance I of VCSP(Γ) with variables $V = \{v_1, \ldots, v_n\}$ and objective function ϕ, and there exist a list $L = \{w_1, \ldots, w_r\}$ of variables from V such that for every $x_1, \ldots, x_r \in D$,

$$\rho(x_1, \ldots, x_r) = \min_{\substack{s\colon V \to D:\\ h(w_i)=x_i, 1\leq i\leq r}} \phi^\Gamma(h(v_1)), \ldots, h(v_n)).$$

The valued language $\langle\Gamma\rangle$ containing all cost functions with domain D that are expressible in Γ is called the *expressive power* of Γ.

Example 7.0.3 ([33], Example 2.3)**.** Let D be a finite set. Let us consider the set containing a single variable $V := \{v\}$, the list of variables $L := \{v, v\}$ from V, and the sum $f_\emptyset$ of no function symbol, namely the empty objective function. Then, the binary equality cost function $\rho_=\colon D^2 \to \{0, +\infty\}$, defined by

$$\rho_=(x, y) := \begin{cases} 0 & \text{if } x = y \\ +\infty & \text{otherwise,} \end{cases}$$

can be written as

$$\rho_=(x, y) = \min_{\substack{h\colon V \to D:\\ h(v)=x, h(v)=y}} f_\emptyset.$$

Therefore, the equality cost function is expressible in any valued structure with a finite domain. △

For any valued structure Γ with a finite domain, adding to Γ a cost function that is expressible in Γ does not change the computational complexity of VCSP(Γ) (see [33], Theorem 2.4).

7.1 The Expressive Power of Valued Structures with Infinite Domains

We extend the notion of expressibility to valued structures with arbitrary domains. We discuss the effect of adding to a valued structure Γ a cost function which is expressible in Γ on the computational complexity of VCSP(Γ).

Definition 7.1.1. Let Γ be a valued structure with an arbitrary domain D and a signature τ, and let $\rho\colon D^r \to \mathbb{Q} \cup \{+\infty\}$ be a cost function. We say that ρ is an *affine conical combination* over Γ if there exist

- function symbols $\gamma_1, \dots, \gamma_m \in \tau$,
- positive rational numbers $\lambda_1, \dots, \lambda_m$, and a rational number c,

such that for every $(x_1, \dots, x_r) \in D$, it holds

$$\rho(x_1, \dots, x_r) = \sum_{j=1}^{m} \lambda_j \gamma_j^\Gamma(x_1^j, \dots, x_{r_j}^j) + c,$$

where $x_1^j, \dots, x_{r_j}^j \in \{x_1, \dots, x_r\}$, for every $j \in \{1, \dots, m\}$. We call *affine convex cone*, or simply *cone*, of Γ, the valued structure Cone(Γ) containing all cost functions over $\mathbb{Q}$ that are affine conical combinations over Γ.

Remark 7.1.2. In the classical constraint satisfaction framework (over arbitrary domains), the notion of affine conical combination corresponds to expressibility using conjunctions (positive formulas).

Example 7.1.3. If Γ is a PL valued structure, then Cone(Γ) is a PL valued structure. If Γ is a PLH valued structure, then Cone(Γ) is a PL valued structure. $\triangle$

The next lemma says that solving VCSP(Cone(Γ)) is not harder than solving VCSP(Γ).

Lemma 7.1.4. *Let Γ be a valued structure with arbitrary domain D and a finite signature, and let Δ be a valued finite reduct of* Cone(Γ)*. Then there exists a polynomial-time many-one reduction from* VCSP(Δ) *to* VCSP(Γ).

Proof. We claim that, for every set of variables $V := \{v_1, \dots, v_n\}$ and every finite sum ϕ of function symbols from Δ with at most n free variables, there exists a finite sum ϕ' of function symbols from Γ such that for every $u \in \mathbb{Q}$ there exists $u' \in \mathbb{Q}$ such that the following holds:

$$\begin{aligned}&\text{there exists } h\colon V \to D \text{ with cost } \phi^\Delta(h(v_1), \dots, h(v_n) \leq u\\ &\text{if, and only if, there exists} \\ &h'\colon V' \to D \text{ with cost } {\phi'}^\Gamma(h'(v_1), \dots, h'(v_n)) \leq u'.\end{aligned} \tag{7.1}$$

Let σ be the (finite) signature of Γ and let $\tau \supseteq \sigma$ be the (finite) signature of Δ. Let $I := (V, \phi, u)$ be an instance of VCSP(Δ) such that $V := \{v_1, \dots, v_n\}$ and

$$\phi(v_1, \dots, v_n) := \sum_{j=1}^{m} \hat{\gamma}_j(v_1^j, \dots, v_{r_j}^j)$$

where $\hat{\gamma}_j \in \tau$ and $v_i^j \in V$, for $1 \leq j \leq m$ and for $1 \leq i \leq r_j$. By the definition of affine conical combination, for every $j \in \{1, \dots, m\}$ there exist

- function symbols $\gamma_{j,1}, \dots, \gamma_{j,m_j} \in \sigma$,
- numerical coefficients $\lambda_{j,1}, \dots, \lambda_{j,m_j} \in \mathbb{Q}_{\geq 0}$, and $c \in \mathbb{Q}$

such that for every $h \colon V \to D$

$$\hat{\gamma}_j^{\Delta}(h(v_1^j), \dots, h(v_{r_j}^j)) = \sum_{i=1}^{m_j} \lambda_{j,i} \gamma_{j,i}^{\Gamma}(h(v_1^{j,i}), \dots, h(v_{r_{j,i}}^{j,i})) + c_j,$$

where $v_l^{j,i} \in V$, for $1 \leq l \leq r_{j,i}$. We define an instance $I' := (V, \phi', u')$ of VCSP(Γ) such that Condition 7.1 holds, I' is computable in polynomial time in the size of $I := (V, \phi, u)$, and V and ϕ' do not depend on u. For every $1 \leq j \leq m$ and for every $1 \leq i \leq m_j$ there exist positive integers $\alpha_{j,i}$ and $\beta_{j,i}$ such that

$$\lambda_{j,i} = \frac{\alpha_{j,i}}{\beta_{j,i}} \text{ and } \gcd(\alpha_j, \beta_j) = 1.$$

For $1 \leq j \leq m$ and for $1 \leq i \leq m_j$, let us define

$$\begin{aligned} l_j &:= \operatorname{lcm}(\beta_{j,1}, \dots, \beta_{j,m_j}), \\ l &:= \operatorname{lcm}(l_1, \dots, l_m), \text{ and} \\ \mu_{j,i} &:= l\lambda_{j,i}. \end{aligned}$$

We define

$$\phi'(v_1, \dots, v_n) := \sum_{j=1}^{m} \sum_{i=1}^{m_j} \mu_{j,i} \gamma_{j,i}(v_1^{j,i}, \dots, v_{r_{j,i}}^{j,i}).$$

Let $u' := l\left(u - \sum_{j=1}^{m} c_j\right)$. Therefore, for every $h \colon V \to D$ it holds that

$$\phi^{\Delta}(h(v_1), \dots, h(v_r)) \leq u$$

if, and only if,

$$\sum_{j=1}^{m} \sum_{i=1}^{m_j} \mu_{j,i} \gamma_{j,i}(h(v_1^{j,i}), \dots, h(v_{r_{j,i}}^{j,i})) = \phi'^{\Gamma}(h(v_1), \dots, h(v_n)) \leq u'. \qquad \square$$

Definition 7.1.5. Let Γ be a valued structure with arbitrary domain D and let $\mu\colon D^r \to \mathbb{Q} \cup \{+\infty\}$ be a cost function. We say that μ is *expressible in* Γ if there exist

- a set of variables $V := \{v_1, \ldots, v_n\}$,
- a list $L := \{w_1, \ldots, w_r\}$ of variables from V, and
- an n-ary cost function ρ that is an affine conical combination over Γ,

such that for every $(x_1, \ldots, x_r) \in D$,

$$\mu(x_1, \ldots, x_r) = \inf_{\substack{h\colon V \to D: \\ h(w_i)=x_i, 1 \le i \le r}} \rho(h(v_1), \ldots, h(v_n)).$$

The valued language $\langle \Gamma \rangle$ containing all cost functions with domain D that are expressible in Γ is called the *expressive power* of Γ.

Remark 7.1.6. From Definition 7.1.5 it follows that the equality cost function over the rationals, $\rho_=\colon\ \mathbb{Q}^2 \to \{0, +\infty\}$ is expressible in any valued structure with rational domain (see Example 7.0.3).

Remark 7.1.7. In the classical constraint satisfaction framework (over arbitrary domains), the notion of expressive power corresponds to expressibility using conjunction and existential quantification (primitive positive formulas). For this reason, the relations that are expressible in a relational structure $\mathfrak{A}$ are also called *pp-definable in* $\mathfrak{A}$.

Clearly, we have that $\Gamma \subseteq \mathrm{Cone}(\Gamma) \subseteq \langle \Gamma \rangle$. We would like to know whether, as in the case of $\mathrm{Cone}(\Gamma)$, the problem $\mathrm{VCSP}(\langle \Gamma \rangle)$ can be (polynomial-time many-one) reduced to $\mathrm{VCSP}(\Gamma)$.

Let Γ be a valued structure with signature τ. The *strict valued constraint satisfaction problem for* Γ, denoted by $\mathrm{VCSP}_s(\Gamma)$, is the following computational problem.

Definition 7.1.8. Let Γ be a valued structure. An *instance* I of $\mathrm{VCSP}_s(\Gamma)$ consists of

- a finite set of variables V_I,
- an expression ϕ_I of the form

  $$\sum_{i=1}^{m} f_i(x_1^i, \ldots, x_{\mathrm{ar}(f_i)}^i)$$

  where $f_1, \ldots, f_m \in \tau$ and all the x_j^i are variables from V_I, and

- a value $u_I \in \mathbb{Q}$.

The task is to decide whether there exists an assignment $\alpha\colon V_I \to \mathrm{dom}(\Gamma)$ whose *cost*, defined as

$$\sum_{i=1}^{m} f_i^{\Gamma}(\alpha(x_1^i), \dots, \alpha(x^i_{\mathrm{ar}(f_i)}))$$

is finite, and if so, whether there is one whose cost is *strictly* smaller than u_I.

Remark 7.1.9. Observe that for a valued structure Γ with a finite domain, the problem $\mathrm{VCSP}(\Gamma)$ is polynomial-time many-one equivalent to $\mathrm{VCSP}_s(\Gamma)$.

Remark 7.1.10. It is easy to show, with a proof similar to the proof of Lemma 7.1.4, that, for every valued structure Γ, the problem $\mathrm{VCSP}_s(\mathrm{Cone}(\Gamma))$ is polynomial-time many-one reducible to $\mathrm{VCSP}_s(\Gamma)$.

Lemma 7.1.11. *Let Γ be a valued structure with a finite signature, and let Δ be a valued finite reduct of $\langle\Gamma\rangle$. Then there exists a polynomial-time many-one reduction from $\mathrm{VCSP}_s(\Delta)$ to $\mathrm{VCSP}_s(\Theta)$, for some valued finite reduct Θ of $\mathrm{Cone}(\Gamma)$.*

Proof. Let τ be the signature of Δ and let $\sigma \subseteq \tau$ be the signature of $\mathrm{Cone}(\Gamma)$. By the definition of expressive power (Definition 7.1.5), for every set of variables $V := \{x_1, \dots, x_r\}$ and every finite sum ϕ of terms over τ there exist a polynomial-time computable set of variables $V' := \{v_1, \dots, v_n\}$, a list $L := \{w_1, \dots, w_r\}$ of variables from V', and a finite sum ϕ' of terms over σ such that for every $u \in \mathbb{Q}$ there exists $u' \in \mathbb{Q}$ such that

$$\begin{aligned}
&\text{there exists } h\colon V \to D \text{ with cost } \phi^{\Delta}(h(x_1), \dots, h(x_r)) < u \\
&\text{if, and only if, there exists } h'\colon V' \to D \text{ such that} \qquad\qquad (7.2)\\
&h'(w_i) = h(x_i), 1 \le i \le r \text{ with cost } \phi'^{\mathrm{Cone}(\Gamma)}(h'(v_1), \dots, h(v_n)) < u'.
\end{aligned}$$

Let I be an instance of $\mathrm{VCSP}_s(\Delta)$ such that $V := \{x_1, \dots, x_r\}$ and

$$\phi(x_1, \dots, x_r) := \sum_{j=1}^{m} \hat{\gamma}_j(x_1^j, \dots, x_{r_j}^j)$$

where $\tilde{\gamma}_j \in \tau$ and $v_i^j \in V$, for $1 \le j \le m$ and for $1 \le i \le r_j$. By the definition of expressive power, for every $j \in \{1, \dots, m\}$ there exist

- a set of new variables $W_j := \{v^j_{r_j+1}, \dots, v^j_{n_j}\}$ such that $W_j \cap V = \emptyset$ and $W_j \cap W_{j'} = \emptyset$ for every $j' \ne j$, $1 \le j' \le m$ (it may be that $W_j = \emptyset$),

- an n_j-ary function symbol $\rho_j \in \sigma$ with $\rho_j^{\mathrm{Cone}(\Gamma)}$ an affine conic combination over Γ,

such that for every $h\colon V \to D$

$$\hat{\gamma}_j^{\Delta}(h(x_1^j), \dots, h(x_{r_j}^j)) = \inf_{\substack{h'\colon V \cup W_j \to D\colon \\ h'_{|V}=h}} \rho_j^{\mathrm{Cone}(\Gamma)}(h'(v_1^j), \dots, h'(v_{n_j}^j)),$$

where $v_l^j \in V \cup W_j$, for $1 \le l \le n_j$. We define an instance $I' := (V', \phi', u')$ of $\mathrm{VCSP}_s(\mathrm{Cone}(\Gamma))$ such that Condition 7.2 holds, I' is computable in polynomial time in the size of $I := (V, \phi, u)$, V' and ϕ' do not depend on u. Let us set $V' := V \cup \bigcup_{j=1}^m W_j = \{v_1, \dots, v_n\}$. We define

$$\phi'(v_1, \dots, v_n) := \sum_{j=1}^m \rho_j^{\mathrm{Cone}(\Gamma)}(v_1^j, \dots, v_{n_j}^j),$$

and let $L := \{x_1, \dots, x_r\}$ be an enumeration of the variables in V. Let $u' := u$. Observe that for every $h\colon V \to D$ it holds that

$$\phi^{\Delta}(h(x_1), \dots, h(x_r)) < u.$$

Observe that, since the sets of variables W_j are mutually disjoint,

$$\begin{aligned} &\sum_{j=1}^m \inf_{\substack{h'\colon V \cup W_j \to D\colon \\ h'_{|V}=h}} \rho_j^{\mathrm{Cone}(\Gamma)}(h'(v_1^j), \dots, h'(v_{n_j}^j)) \\ = &\inf_{\substack{h'\colon V' \to D\colon \\ h'(w_i)=h(x_i), 1 \le i \le r}} \phi'^{\mathrm{Cone}(\Gamma)}(h'(v_1), \dots, h'(v_n)). \end{aligned}$$

Let us assume that there exists an assignment $h^\star\colon V' \to D$ such that

$$\phi'^{\,\mathrm{Cone}(\Gamma)}(h^\star(v_1), \dots, h^\star(v_n)) < u,$$

then it follows immediately that

$$\inf_{\substack{h'\colon V' \to D\colon \\ h'(w_i)=h^\star(x_i), 1 \le i \le r}} \phi'^{\,\mathrm{Cone}(\Gamma)}(h'(v_1), \dots, h'(v_n)) < u,$$

and, consequently, $\phi^{\Delta}(h^\star(x_1), \dots, h^\star(x_r)) < u$.

Vice versa, let us assume that there exists an assignment $h\colon V \to D$ such that $\phi^{\Delta}(h(x_1), \dots, h(x_n)) < u$, and let

$$\varepsilon := u' - \inf_{\substack{h'\colon V' \to D\colon \\ h'(w_i)=h(x_i), 1 \le i \le r}} \phi'^{\,\mathrm{Cone}(\Gamma)}(h'(v_1), \dots, h'(v_n)),$$

then, by a property of the infimum, there exists $h^\star \colon V' \to D$ such that $h^\star(w_i) = h(x_i)$ for $1 \leq i \leq r$, and

$$\phi'^{\Delta}(h^\star(v_1), \ldots, h^\star(v_n)) < \inf_{\substack{h' \colon V' \to D: \\ h'(v_i) = h(x_i), 1 \leq i \leq r}} \phi'^{\Delta}(h'(v_1), \ldots, h'(v_n)) + \varepsilon = u'.$$

□

Proposition 7.1.12. *Let Γ be a valued structure with an arbitrary domain D and a finite signature. Then for every valued finite reduct Δ of $\langle \Gamma \rangle$ there exists a polynomial-time reduction from* $\mathrm{VCSP}_s(\Delta)$ *to* $\mathrm{VCSP}_s(\Gamma)$*.*

Proof. To prove Proposition 7.1.12, it is enough to compose the polynomial-time reductions provided in Remark 7.1.10 and in Lemma 7.1.11. □

We apply Proposition 7.1.12 to find a family of PL valued structures having an NP-hard VCSP_s.

Corollary 7.1.13. *Let Γ be a PL valued structure with a finite signature. If there exist intervals $(a,b), (c,d) \subset \mathbb{Q}$ (where a and d, respectively, can be the symbol $-\infty$ and $+\infty$, respectively), and $\alpha, \beta \in \mathbb{Q}$, with $\alpha < \beta$, such that the cost function $f^{\alpha,\beta}_{(a,b),(c,d)} \colon D^2 \to \mathbb{Q} \cup \{+\infty\}$ defined by*

$$f(x,y)^{\alpha,\beta}_{(a,b),(c,d)} := \begin{cases} \alpha & \text{if } (x,y) \in ((a,b) \times (c,d)) \cup ((c,d) \times (a,b)) \\ \beta & \text{if } (x,y) \in (a,b)^2 \cup (c,d)^2 \\ +\infty & \text{otherwise} \end{cases}$$

is expressible in Γ, then $\mathrm{VCSP}_s(\Gamma)$ *is NP-hard.*

Proof. The strict-threshold version[1] of Max Cut is polynomial-time many-one equivalent to its weak-threshold version. Therefore, it is easy to see, by using the same proof as for Proposition 2.3.2, that $\mathrm{VCSP}_s(\langle \Gamma \rangle)$ is NP-hard. Our statement then follows from Proposition 7.1.12. □

7.2 Expressive Power and Fractional Polymorphisms

We now prove that every cost function that is expressible in a valued structure Γ (with an arbitrary domain) is improved by every fractional polymorphism of Γ.

[1] Given a graph (V,E) and a rational number u, the strict-threshold version of Max Cut is the problem of deciding whether exist a subset S of V such that the number of edges between S and its complement $V \setminus S$ is strictly larger than u; the weak-threshold version of Max Cut is the problem of deciding whether exist a subset S of V such that the number of edges between S and its complement $V \setminus S$ is at least u.

Lemma 7.2.1. *Let Γ be a valued structure with an arbitrary domain D and values in $\mathbb{Q} \cup \{+\infty\}$ (or $\mathbb{R} \cup \{+\infty\}$). Every fractional polymorphism of Γ improves every cost function that is expressible in Γ, that is,*

$$\langle \Gamma \rangle \subseteq \mathrm{Imp}\left(\mathrm{fPol}(\Gamma)\right).$$

Proof. Let ω be a k-ary fractional polymorphism of Γ, and let us consider ρ be a cost function with domain D and values in $\mathbb{Q} \cup \{+\infty\}$ (or in $\mathbb{R} \cup \{+\infty\}$, depending on the set of values of Γ) that is expressible in Γ. Let $x^1, \ldots, x^k \in D^r$, we want to prove that

$$\sum_{g \in \mathrm{Supp}(\omega)} \omega(g)\rho(g(x^1, \ldots, x^k)) \leq \frac{1}{k} \sum_{l=1}^{k} \rho(x^l).$$

By definition of expressibility (see Definition 7.1.5), there exist a finite set of variables $V := \{v_1, \ldots, v_n\}$, a list $L := \{w_1, \ldots, w_r\}$ of variables from V, finitely many cost functions $\gamma_1, \ldots, \gamma_m \in \Gamma$, finitely many coefficients $\lambda_1, \ldots, \lambda_m \in \mathbb{Q}_{\geq 0}$, and a constant $c \in \mathbb{Q}$ such that, for every $l \in \{1, \ldots, k\}$

$$\rho(x^l) = \rho(x_1^l, \ldots, x_r^l) = \inf_{\substack{h\colon V \to D\colon \\ h(w_i) = x_i^l, 1 \leq i \leq r}} \sum_{j=1}^{m} \lambda_j \gamma_j(h(v^j)) + c,$$

where $v^j \in V^{\mathrm{ar}(\gamma_j)}$ for $1 \leq j \leq m$.

Let $\epsilon > 0$ be an arbitrarily small positive rational number. By definition of infimum, for every $l \in \{1, \ldots, k\}$ there exists an assignment $h_\epsilon^l\colon V \to D$ with $h_\epsilon^l(w_i) = x_i^l$ such that

$$\rho(x^l) \leq \sum_{j=1}^{m} \lambda_j \gamma_j(h_\epsilon^l(v^j)) + c, \text{ and} \tag{7.3}$$

$$\sum_{j=1}^{m} \lambda_j \gamma_j(h_\epsilon^l(v^j)) + c < \rho(x^l) + \epsilon. \tag{7.4}$$

From Inequality (7.3) we easily get that

$$\begin{aligned}
&\sum_{g \in \mathrm{Supp}(\omega)} \omega(g)\rho(g(x^1,\ldots,x^k)) \\
\leq\ &\sum_{g \in \mathrm{Supp}(\omega)} \omega(g) \left(\sum_{j=1}^{m} \lambda_j \gamma_j (g(h^1_\epsilon(v^j),\ldots,h^k_\epsilon(v^j))) + c \right) \qquad (7.5)\\
=\ &\sum_{j=1}^{m} \lambda_j \left(\sum_{g \in \mathrm{Supp}(\omega)} \omega(g)\gamma_j(g(h^1_\epsilon(v^j)),\ldots,h^k_\epsilon(v^j)) \right) + c.
\end{aligned}$$

Since ω is a fractional polymorphism of Γ, it improves $\gamma_1,\ldots,\gamma_m$, and therefore the last row of Inequality (7.5) is at most

$$\begin{aligned}
&\sum_{j=1}^{m} \lambda_j \left(\frac{1}{k} \sum_{l=1}^{k} \gamma_j(h^l_\epsilon(v^j)) \right) + c = \frac{1}{k} \sum_{l=1}^{k} \left(\sum_{j=1}^{m} \lambda_j \gamma_j(h^l_\epsilon(v^j)) + c \right) \\
< &\frac{1}{k} \sum_{l=1}^{k} (\rho(x^l) + \epsilon),
\end{aligned}$$

where the last inequality is justified by the choice of h^l_ϵ for $1 \leq l \leq k$. Summarising, we get that

$$\sum_{g \in \mathrm{Supp}(\omega)} \omega(g)\rho(g(x^1,\ldots,x^k)) < \frac{1}{k} \sum_{l=1}^{k} \rho(x^l) + \epsilon$$

for every rational $\epsilon > 0$, that is,

$$\sum_{g \in \mathrm{Supp}(\omega)} \omega(g)\rho(g(x^1,\ldots,x^k)) \leq \frac{1}{k} \sum_{l=1}^{k} \rho(x^l),$$

as we wanted to prove. □

Remark 7.2.2. Observe that the proof of Lemma 7.2.1 also works in the case in which the fractional operations considered take real (rather than rational) values, that is, for a valued structure Γ with an arbitrary domain D every real-valued fractional polymorphism of Γ improves every cost function that is expressible in Γ.

In the case of a valued structure Γ with a finite domain and a finite signature also the converse of Lemma 7.2.1 holds, i.e., $\langle\Gamma\rangle = \mathrm{Imp}\,(\mathrm{fPol}(\Gamma))$ (see [33]). In the case of a valued structure Γ with a finite domain and an infinite signature, it holds that $\overline{\langle\Gamma\rangle_{\mathrm{Opt}}} = \mathrm{Imp}_{\mathbb{R}}\,(\mathrm{fPol}_{\mathbb{R}}(\Gamma))$ (see [43]), where

$\overline{\langle\Gamma\rangle_{\mathrm{Opt}}}$ is the topological closure of the closure of $\langle\Gamma\rangle$ under the operator Opt[2]; and the index in $\mathrm{fPol}_{\mathbb{R}}$ and $\mathrm{Imp}_{\mathbb{R}}$ denotes the fact that the fractional operations considered take real (rather than rational) values. Since the computational complexity of VCSP(Γ) is invariant under adding to Γ cost functions that are expressible in Γ (and cost functions from $\overline{\langle\Gamma\rangle_{\mathrm{Opt}}}$), the finite-domain valued structures whose VCSP is polynomial-time solvable can be characterised in terms of fractional polymorphisms. This characterisation extends the characterisation of the expressive power of finite-domain relational structures, for which it holds that the set of formulas that are pp-definable in $\mathfrak{A}$ coincides with the set of relations that are preserved by all polymorphisms of $\mathfrak{A}$ (cf. [20, 46]).

We point out that in the classical constraint setting when moving to the infinite-domain case, the characterisation of polynomial-time solvable CSPs in terms of polymorphisms holds in particular cases (see [18]). However, it does not hold in general for semilinear, i.e., PL, relational structures (see [10], Section 4)[3]. This implies that, in general, fractional polymorphisms do not capture the computational complexity of the feasibility problem associated with an infinite-domain VCSP. It is natural to ask what happens if we restrict our investigation of infinite-domain VCSPs to the finite-valued case (that is, the case in which the cost functions take values $< +\infty$). In the next section, we show a local version of the converse of Lemma 7.2.1 for finite-valued structures with arbitrary domains and arbitrary signatures.

7.3 Applying Farkas' Lemma to Infinite-Domain Finite-Valued Structures

In this section, we show a universal algebraic local characterisation of the expressive power of finite-valued structures with arbitrary countable domains and arbitrary signatures.

In the remainder of the chapter, we assume that all the cost functions are defined in some power of a countable infinite set D and take values in $\mathbb{R}$ (rather than $\mathbb{R} \cup \{+\infty\}$). We also assume that the fractional operations we consider have values in $\mathbb{R}$ (rather than in $\mathbb{Q}$ as we assumed so far).

[2]Given a cost function $f\colon D^n \to \mathbb{Q} \cup \{+\infty\}$, the cost function $\mathrm{Opt}(f)\colon D^n \to \mathbb{Q} \cup \{+\infty\}$ is defined by $\mathrm{Opt}(f)(x) = 0$ if $f(x) = \min_{D^n} f$, and $\mathrm{Opt}(f)(x) = +\infty$ otherwise.

[3]The authors of [10] exhibit a class of semilinear relational structures (called *essentially convex*) that is not characterised by any polymorphism in $\mathbb{Q}$ and, nevertheless, it gives rise to a class of maximal tractable CSPs. However, this class of semilinear relational structures is characterised by a polymorphism in $\mathbb{Q}^\star = \{x + y\epsilon \mid x, y \in \mathbb{Q}\}$. Therefore, so far, there is no counterexample to a possible characterisation of the computational complexity of semilinear CSPs by polymorphisms in $\mathbb{Q}^\star$.

Let M be an arbitrary set. We define, as usual, the *support* of a function $\alpha\colon M \to \mathbb{R}$ as the set $\operatorname{Supp}(\alpha) := \{x \in M \mid \alpha(x) \neq 0\}$. We consider the vector space

$$\mathbb{R}[M] := \left\{\alpha \in \mathbb{R}^M \mid \operatorname{Supp}(\alpha) \text{ is finite}\right\}.$$

For any $x \in M$, let us define $\delta_x \in \mathbb{R}[M]$ by

$$\delta_x(y) := \begin{cases} 1 & \text{if } y = x, \\ 0 & \text{otherwise} \end{cases}$$

for $y \in M$. Let us also consider $\mathbb{R}_{\geq 0}[M] := \{\alpha \in \mathbb{R}[M] \mid \forall x \in M\colon\ \alpha(x) \geq 0\}$. For any set D we define $\mathrm{wRel}_D^{(r)} := \mathbb{R}^{D^r}$ for every $r \in \mathbb{N}$, and

$$\mathrm{wRel}_D := \bigcup_{r \in \mathbb{N}} \mathrm{wRel}_D^{(r)}.$$

7.3.1 Local Expressive Power

In this section, we introduce the notion of *local expressive power* of a valued structure. Informally, given a valued structure Γ with domain D, the local expressive power of Γ consists of all cost functions whose restrictions to any finite subset D' of D can be simulated by using the restrictions of cost functions from Γ to D'.

Definition 7.3.1. Let D be a set and let us consider $\Gamma \subseteq \mathrm{wRel}_D$. Let us set

$$\mathrm{M}_k(\Gamma) := \left\{(S, \gamma) \,\middle|\, \gamma \in \Gamma,\ S \in D^{\mathrm{ar}(\gamma) \times k}\right\}$$

for $k \in \mathbb{N}$. We say that $\rho\colon D^r \to \mathbb{R}$ belongs to the *local expressive power* of Γ and write $\rho \in \ell\mathrm{Expr}(\Gamma)$ if, for every $\epsilon > 0$ and every $k \in \mathbb{N}$, every $x^1, \dots, x^k \in D^r$, and every finite subset $\mathcal{F} \subseteq \mathcal{O}_D^{(k)}$ there exist $\lambda \in \mathbb{R}_{\geq 0}[\mathrm{M}_k(\Gamma)]$ and $c \in \mathbb{R}$ such that, for each $i \in \{1, \dots, k\}$,

$$\left|\rho(x^i) - \left(\sum_{(S,\gamma) \in \operatorname{Supp}(\lambda)} \lambda(S, \gamma)\gamma\Big(e_i^{(k)}(S)\Big) + c\right)\right| \leq \epsilon \tag{7.6}$$

and, for each $f \in \mathcal{F}$,

$$\rho(f(x^1, \dots, x^k)) \leq \sum_{(S,\gamma) \in \operatorname{Supp}(\lambda)} \lambda(S, \gamma)\gamma(f(S)) + c + \epsilon, \tag{7.7}$$

where $e_i^{(k)}$ and f are applied to the matrices S componentwise, i.e., to their rows.

Observe that for every $\Gamma \subseteq \mathrm{wRel}_D$ the local expressive power $\ell\mathrm{Expr}(\Gamma)$

is a topologically closed subseteq of wRel_D.

Proposition 7.3.2. *Let D be a set and let us consider $\Gamma \subseteq \mathrm{wRel}_D$. Then $\overline{\langle\Gamma\rangle} \subseteq \ell\mathrm{Expr}(\Gamma)$. Where $\overline{\langle\Gamma\rangle}$ denotes the topological closure of $\langle\Gamma\rangle$.*

Proof. Let us consider a cost function $\rho\colon D^r \to \mathbb{Q}$ such that $\rho \in \langle\Gamma\rangle$. By the definition of expressive power (Definition 7.1.5) there exist

- a set of variables $V := \{v_1, \ldots, v_n\}$,
- a list $L := \{w_1, \ldots, w_r\}$ of variables from V,
- cost function $\gamma_1, \ldots, \gamma_m$ from Γ,
- positive rational numbers $\lambda_1, \ldots, \lambda_m$, and a rational number c,

such that for every $(x_1, \ldots, x_r) \in D$ it holds

$$\rho(x_1, \ldots, x_r) = \inf_{\substack{h\colon V \to D:\\ h(w_l)=x_l, 1\leq l\leq r}} \sum_{j=1}^{m} \lambda_j \gamma_j^{\gamma}(v^j) + c,$$

where $v^j \in V$ for $1 \leq j \leq m$. Let us fix $\epsilon > 0$, a positive integer k, and $x^1, \ldots, x^k \in D^r$. From the definition of infimum, it follows that for every $i \in \{1, \ldots, k\}$ it holds that

1. for every $s\colon V \to D$ with $s(w_l) = x^i_l$ for $1 \leq l \leq r$
$$\rho(x^i) \leq \sum_{j=1}^{m} \lambda_j \gamma_j(s(v^j)) + c;$$
2. there exists $s^i\colon V \to D$ with $s^i(w_l) = x^i_l$ for $1 \leq l \leq r$ such that
$$\left|\rho(x^i) - \left(\sum_{j=1}^{m} \lambda_j \gamma_j(s^i(v^j)) + c\right)\right| \leq \epsilon.$$

For every $j \in \{1, \ldots, m\}$, let S_j be the matrix from $D^{\mathrm{ar}(\gamma_j)\times k}$ with columns $s^1(v^j), \ldots, s^k(v^j)$. For every $f \in \mathcal{O}^{(k)}_D$, let $s\colon V \to D$ be the assignment defined by $s(v) := f(s^1(v), \ldots, s^k(v))$, for all $v \in V$. Observe that,

$$(s(w_1), \ldots, s(w_r)) = f\left(\begin{pmatrix} s^1(w_1) & \ldots & s^k(w_1) \\ \vdots & & \vdots \\ s^1(w_r) & \ldots & s^k(w_r)\end{pmatrix}\right) = f(x^1, \ldots, x^k),$$

and for every $j \in \{1, \dots, m\}$

$$s(v^j) := (s(v_1^j), \dots, s(v_{\mathrm{ar}(\gamma_j)}^j)) = f\left(\begin{pmatrix} s^1(v_1^j) & \dots & s^k(v_1^j) \\ \vdots & & \vdots \\ s^1(v_{\mathrm{ar}(\gamma_j)}^j) & \dots & s^k(v_{ar(\gamma_j)}^j) \end{pmatrix}\right) = f(S_j).$$

Therefore, by Condition 1 we obtain $\rho(f(S_\rho)) \le \sum_{j=1}^m \lambda_j \gamma_j(f(S_j)) + c$. Let $\lambda \in \mathbb{R}_{\ge 0}[\mathrm{M}_k(\Gamma)]$ be such that

$$\lambda(S, \gamma) := \begin{cases} \lambda_j & \text{if } (S, \gamma) = (S_j, \gamma_j) \\ 0 & \text{otherwise.} \end{cases}$$

With this notation we can rewrite Conditions 1 and 2 as follows.

1. For every $f \in \mathcal{O}_D^{(k)}$,

$$\rho(f(x^1, \dots, x^k)) \le \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S, \gamma)\gamma(f(S)) + c;$$

2. for every $e_i^{(k)} \in \mathcal{J}_D^{(k)}$,

$$\left|\rho(e_i^{(k)}(x^1, \dots, x^k)) - \left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S, \gamma)\gamma(e_i^{(k)}(S)) + c\right)\right| \le \epsilon.$$

Therefore, $\rho \in \ell\mathrm{Expr}(\Gamma)$ and $\langle\Gamma\rangle \subseteq \ell\mathrm{Expr}(\Gamma)$. Since $\ell\mathrm{Expr}(\Gamma)$ is topologically closed, $\overline{\langle\Gamma\rangle} \subseteq \ell\mathrm{Expr}(\Gamma)$. □

We now show that for valued structures with finite domains, the local expressive power coincides with the topological closure of the expressive power.

Proposition 7.3.3. *Let D be a finite set. Let $\Gamma \subseteq \mathrm{wRel}_D$ and $\rho\colon D^r \to \mathbb{Q}$. Then $\rho \in \ell\mathrm{Expr}(\Gamma)$ if, and only if, $\rho \in \overline{\langle\Gamma\rangle}$, where $\overline{\langle\Gamma\rangle}$ is the topological closure of $\langle\Gamma\rangle$.*

Proof. We know that $\overline{\langle\Gamma\rangle} \subseteq \ell\mathrm{Expr}(\Gamma)$ (Proposition 7.3.2). Let us assume then that $\rho \in \ell\mathrm{Expr}(\Gamma)$. We claim that for every $\epsilon > 0$ there exists $\tilde{\rho} \in \langle\Gamma\rangle$ such that $|\rho(x) - \tilde{\rho}(x)| < \epsilon$ for every $x \in D^r$, that is, $\rho \in \overline{\langle\Gamma\rangle}$.

Let $k := |D|^r$ and let $x^1, \dots, x^k$ be an enumeration of all tuples from D^r. By the definition of local expressive power there exist $\lambda \in \mathbb{R}_{\ge 0}[M_k(\Gamma)]$, and $c \in \mathbb{R}$ such that the Conditions (7.6) and (7.7) of Definition 7.3.1 hold for $\mathcal{F} = \mathcal{O}_D^{(k)}$. Since $\lambda \in \mathbb{R}_{\ge 0}[M_k(\Gamma)]$ and Γ has finite domain and finite signature, by definition there exist cost functions $\gamma_1, \dots, \gamma_m \in \Gamma$ (not necessarily pairwise distinct), and matrices $S_1, \dots, S_m \in D^{k\times \mathrm{ar}(\gamma_j)}$ such that

$\mathrm{Supp}(\lambda) = \{(\gamma_j, S_j) \mid 1 \leq j \leq m\}$. Let us set $\lambda_j := \lambda(\gamma_j, S_j)$ for $1 \leq j \leq m$, and let $S_\rho \in D^{r\times k}$ be the matrix with columns $x^1, \ldots, x^k$. Then the following two conditions hold

- for all $e_i^{(k)} \in \mathcal{J}_D^{(k)}$,

$$\left| \rho(e_i^{(k)}(S_\rho)) - \left(\sum_{j=1}^{m} \lambda_j \gamma_j(e_i^{(k)}(S_j)) + c \right) \right| \leq \epsilon, \tag{7.8}$$

- for all $f \in \mathcal{O}_D^{(k)}$,

$$\rho(f(S_\rho)) - \left(\sum_{j=1}^{m} \lambda_j \gamma_j(f(S_j)) + c \right) \leq \epsilon. \tag{7.9}$$

Let us define the cost function $\tilde{\rho} \colon D^r \to \mathbb{R}$ such that for every $i \in \{1, \ldots, k\}$, let

$$\tilde{\rho}(x^i) := \min_{\substack{f \colon D^k \to D: \\ f(S_\rho) = x^i}} \sum_{j=1}^{m} \lambda_j \gamma_j(f(S_j)) + c.$$

The cost function $\tilde{\rho}$ is well defined since for every $x = (x_1, \ldots, x_r) \in D^r$ there exists an index $i \in \{1, \ldots, k\}$ such that $(x_1, \ldots, x_r) = (x_1^i, \ldots, x_r^i) = x^i$. Observe that $\tilde{\rho} \in \langle \Gamma \rangle$. To see this, let us associate the rows of S_ρ with fresh variables $\{v_1, \ldots, v_r\}$, and let us associate every row of every S_j with a variable v_l^j, for $1 \leq l \leq \mathrm{ar}(\gamma_j)$. Let V be the set of variables defined as

$$\begin{aligned} V =& \{v_1, \ldots, v_r, v_{r+1}, \ldots, v_n\} \\ :=& \{v_1, \ldots, v_r\} \cup \{v_l^j \mid 1 \leq j \leq m,\, 1 \leq l \leq \mathrm{ar}(\gamma_j)\}, \end{aligned}$$

where two variables of V are the same variable whenever they are associated with two rows that are equal as tuples of D^k. Then every assignment $s \colon V \to D$ corresponds to a function $f \colon D^k \to D$, and therefore, it holds that for every $x = (x_1 \ldots, x_r) \in D^r$

$$\rho(x) = \min_{\substack{s \colon V \to D: \\ s(v_i) = x_i,\, 1 \leq i \leq r}} \sum_{j=1}^{m} \lambda_j \gamma_j(s(v_1^j), \ldots, s(v_{\mathrm{ar}(\gamma_j)}^j)) + c,$$

that is, $\rho \in \langle \Gamma \rangle$.

Let $i \in \{1, \ldots, k\}$. We now observe that there exists $f^i \in \mathcal{O}_D^{(k)}$ such that

$f^i(S_\rho) = x^i$ and $\tilde{\rho}(x^i) = \sum_{j=1}^{m} \lambda_j \gamma_j(f^i(S_j)) + c$; therefore, we obtain

$$\rho(x^i) \leq \left(\sum_{j=1}^{m} \lambda_j \gamma_j(f^i(S_j)) + c\right) + \epsilon \leq \tilde{\rho}(x^i) + \epsilon. \tag{7.10}$$

On the other hand, from Condition (7.8), it follows that

$$\tilde{\rho}(x^i) - \epsilon \leq \left(\sum_{j=1}^{m} \lambda_j \gamma_j(e_i^{(k)}(S_j)) + c\right) - \epsilon \leq \rho(x^i). \tag{7.11}$$

Therefore, $|\rho(x^i) - \tilde{\rho}(x^i)| < \epsilon$ for every $i \in \{1, \ldots, k\}$, i.e., $|\rho(x) - \tilde{\rho}(x)| < \epsilon$ for every $x \in D^r$. This proves the claim. □

7.3.2 Weighted Polymorphisms

We now give the notion of *weighted polymorphism* which can be seen as a generalisation of the notion of fractional polymorphism. The notion of fractional polymorphism of a valued structure Γ is defined for fractional operations on $\mathcal{O}_{\mathrm{dom}(\Gamma)}$ having a finite support, taking nonnegative values, and summing up to one. The notion of weighted polymorphism, on the other hand, is defined for *weightings*, i.e., for real-valued functions on $\mathcal{O}_{\mathrm{dom}(\Gamma)}$ having a finite support, taking negative values only on projections, and summing up to zero.

Definition 7.3.4. Let D be an arbitrary set. For each $k \in \mathbb{N}$, we define the *set of k-ary weightings on $\mathcal{O}_D$* as

$$\mathrm{w}\mathcal{O}_D^{(k)} := \left\{ \omega \in \mathbb{R}\left[\mathcal{O}_D^{(k)}\right] \,\middle|\, \sum_{f \in \mathrm{Supp}(\omega)} \omega(f) = 0,\ \forall f \in \mathcal{O}_D^{(k)} \setminus \mathcal{J}_D^{(k)} \colon\ \omega(f) \geq 0 \right\}.$$

Moreover, let $\mathrm{w}\mathcal{O}_D := \bigcup_{k \in \mathbb{N}} \mathrm{w}\mathcal{O}^{(k)}$. Given $\omega \in \mathrm{w}\mathcal{O}_D^{(k)}$ and $\rho \in \mathrm{wRel}_D^{(r)}$ with $k, r \in \mathbb{N}$, we say that ω is a *weighted polymorphism* of ρ (resp., ρ is *weight-improved* by ω) if

$$\sum_{f \in \mathrm{Supp}(\omega)} \omega(f) \rho(f(x^1, \ldots, x^k)) \ \leq\ 0$$

for all $x^1, \ldots, x^k \in D^r$. For $\Gamma \subseteq \mathrm{wRel}_D$, we define

$$\mathrm{wPol}(\Gamma) := \{\omega \in \mathrm{w}\mathcal{O}_D \mid \forall \gamma \in \Gamma \colon\ \omega \text{ weighted polymorphism of } \gamma\}.$$

For $\Omega \subseteq \mathrm{w}\mathscr{O}_D$, we let

$$\mathrm{wImp}(\Omega) := \{\gamma \in \mathrm{wRel}_D \mid \forall \omega \in \Omega\colon \gamma \text{ is weight-improved by } \omega\}.$$

Example 7.3.5. We have seen in Remark 1.5.3 that a function on a totally ordered set D is submodular if, and only if, it is improved by the fractional operation ω_{sub} on $\mathscr{O}_D^{(2)}$ introduced in Example 1.4.17. It is trivial to see that we can also characterise submodular functions on D as those functions that are weight-improved by the weighting $\omega_{sub}^w\colon \mathscr{O}_D^{(2)} \to \mathbb{R}$ such that

$$\omega_{sub}^w(g) := \begin{cases} 1 & \text{if } g = \min \text{ or } g = \max \\ -1 & \text{if } g = e_1^{(2)} \text{ or } g = e_2^{(2)} \\ 0 & \text{otherwise.} \end{cases}$$

△

Remark 7.3.6. Fractional polymorphisms and weighted polymorphisms are closely related. More precisely, given a function $\rho\colon D \to \mathbb{R} \cup \{+\infty\}$, every k-ary fractional operation ω that improves ρ yields the k-ary weighting ω^w of ρ defined, for every $g \in \mathscr{O}_D^k$ by

$$\omega^w(g) := \begin{cases} \omega(g) & \text{if } g \in \mathscr{O}_D^{(k)} \setminus \mathscr{J}_D^{(k)} \\ \omega(g) - \frac{1}{k} & \text{otherwise,} \end{cases}$$

which weight-improves ρ. Also, it is easy to verify that, vice versa, every weighting that weight-improves ρ yields a fractional operation that improves ρ (see [102], Remark 2.7). From this observation, it is easy to derive that for every $\Gamma \subseteq \mathrm{wRel}_D$ it holds that $\mathrm{Imp}(\mathrm{fPol}(\Gamma)) = \mathrm{wImp}(\mathrm{wPol}(\Gamma))$; therefore, from Lemma 7.2.1 it follows that for every valued structure Γ over an arbitrary domain D and values in $\mathbb{R}$,

$$\langle \Gamma \rangle \subseteq \mathrm{wImp}(\mathrm{wPol}(\Gamma)).$$

In Theorem 7.3.7, we provide a characterisation of $\mathrm{wImp}(\mathrm{wPol}(\Gamma))$, in terms of local expressive power, which holds for finite-valued structures Γ with arbitrary countable domains.

Theorem 7.3.7. *Let D be an arbitrary countable set and let $\Gamma \subseteq \mathrm{wRel}_D$. Then*

$$\mathrm{wImp}(\mathrm{wPol}(\Gamma)) = \ell\mathrm{Expr}(\Gamma).$$

In order to prove Theorem 7.3.7, we need to set some preliminaries and to recall two constructions for locally convex topological vector spaces.

7.3.3 Locally Convex Spaces

In this section, we present some preliminary definitions and results for locally convex topological spaces.

Throughout the section, by *topological vector space* we always mean *Hausdorff topological space*[4] *over* $\mathbb{R}$. In order to agree on some further terminology and notation, let X be a topological vector space. As usual, we say that X is *locally convex* if $0 \in X$ admits a neighbourhood base consisting of convex (open) subsets of X. We denote by X^* the *topological dual of* X, i.e., the topological vector space of all continuous linear functionals on X equipped with the weak-$*$ topology, which is the initial topology on X^* generated by all maps of the form $T_x \colon X^* \to \mathbb{R}$ such that $T_x(\phi) = \phi(x)$, where $x \in X$. It is well known that X^* is a locally convex topological vector space (see, e.g., [23], Proposition 3.12). For a subset $S \subseteq X$, we define $S^+ := \{\phi \in X^* \mid \forall x \in S \colon \phi(x) \geq 0\}$. Furthermore, if Y is another topological vector space and the map $A \colon X \to Y$ is linear and continuous, then we define the continuous linear map $A^* \colon Y^* \to X^*$ by setting $A^*(\psi) := \psi \circ A$.

Lemma 7.3.8 (Abstract core of Farkas' lemma, [94], Lemma 2.1)**.** *Let X and Y be locally convex topological vector spaces. If S is a closed convex cone in Y and $A \colon X \to Y$ is a linear and continuous map, then it holds that*

$$(A^{-1}(S))^+ = \overline{A^*(S^+)}.$$

The following more familiar version of Farkas' lemma is a mere reformulation of Lemma 7.3.8.

Theorem 7.3.9 (Farkas' lemma)**.** *Let X and Y be locally convex topological vector spaces and let $A \colon X \to Y$ be a linear and continuous map. Let S be a closed convex cone in Y and let $\phi \in X^*$. The following are equivalent.*

(1) $\forall x \in X \colon\; A(x) \in S \implies \phi(x) \geq 0.$

(2) $\phi \in \overline{A^*(S^+)}.$

Proof. By Lemma 7.3.8, we deduce that

$$\begin{aligned}
\phi \in \overline{A^*(S^+)} \quad &\iff \quad \phi \in (A^{-1}(S))^+ \\
&\iff \quad \forall x \in A^{-1}(S) \colon\; \phi(x) \geq 0 \\
&\iff \quad \forall x \in X \colon\; (A(x) \in S \implies \phi(x) \geq 0)\,. \qquad \square
\end{aligned}$$

For our purposes, we need the following refinement of Theorem 7.3.9.

[4] A topological space is said to be *Hausdorff* if two distinct points always lie in two disjoint open sets.

Corollary 7.3.10. *Let X, Y, Z be locally convex topological vector spaces and let $A\colon X \to Y$, $B\colon X \to Z$ be linear and continuous maps. Let $S \subseteq Y$ and $T \subseteq Z$ be closed convex cones and let $\phi \in X^*$. The following are equivalent.*

(1) $\forall x \in X\colon\ (A(x) \in S \wedge B(x) \in T) \implies \phi(x) \geq 0.$

(2) $\phi \in \overline{\{A^*(\mu) + B^*(\nu) \mid \mu \in S^+,\, \nu \in T^+\}}.$

Proof. Consider the locally convex topological vector spaces $\tilde{X} := X$ and $\tilde{Y} := Y \times Z$, the continuous linear maps $\tilde{\phi} := \phi$ and $\tilde{A}\colon \tilde{X} \to \tilde{Y}$ such that $\tilde{A}(x) = (A(x), B(x))$, and the closed convex cone $\tilde{S} := S \times T \subseteq \tilde{Y}$. It is well known that the map $\Psi\colon Y^* \times Z^* \to (Y \times Z)^*$ given by

$$\Psi(\mu, \nu)(y, z) := \mu(y) + \nu(z) \qquad (\mu \in Y^*,\, \nu \in Z^*,\, y \in Y,\, z \in Z)$$

is a topological isomorphism. Note that if $\mu \in Y^*$ and $\nu \in Z^*$, then

$$\begin{aligned} \tilde{A}^*(\Psi(\mu, \nu))(x) &= \Psi(\mu, \nu)(\tilde{A}(x)) = \Psi(\mu, \nu)(A(x), B(x)) \\ &= \mu(A(x)) + \nu(B(x)) = A^*(\mu)(x) + B^*(\nu)(x) \\ &= (A^*(\mu) + B^*(\nu))(x) \end{aligned}$$

for all $x \in X$, i.e., $\tilde{A}^*(\Psi(\mu, \nu)) = A^*(\mu) + B^*(\nu)$. Furthermore, it is easy to see that for all $\mu \in Y^*$ and, for all $\nu \in Z^*$ it holds that

$$\Psi(\mu, \nu) \in (S \times T)^+ \quad \text{if, and only if,} \quad \mu \in S^+ \wedge \nu \in T^+.$$

Consequently,

$$\begin{aligned} \tilde{A}^*(\tilde{S}^+) &= \left\{ \tilde{A}^*(\Psi(\mu, \nu)) \,\middle|\, \mu \in Y^*,\, \nu \in Z^*,\, \Psi(\mu, \nu) \in \tilde{S}^+ \right\} \\ &= \left\{ A^*(\mu) + B^*(\nu) \,\middle|\, \mu \in S^+,\, \nu \in T^+ \right\}. \end{aligned}$$

Because of these observations, the statement of Corollary 7.3.10 for the convex topological vector spaces X, Y, Z, the functional ϕ, the maps A, B, and the closed convex cones S, T is an immediate consequence of Theorem 7.3.9 applied to $\tilde{X}, \tilde{Y}, \tilde{\phi}, \tilde{A}, \tilde{S}$. □

We now recall two constructions for locally convex topological vector spaces. First, let E be any vector space over $\mathbb{R}$. We define τ_E to be the topology generated by the set of all seminorms on E (see [90], Chapter II, Section 4), i.e., the finest locally convex topology on E. An additional argument shows that any linear map from E into a locally convex topological vector space is necessarily continuous with respect to τ_E.

For the second construction, let X and Y be two vector spaces over $\mathbb{R}$ and let $\beta\colon X \times Y \to \mathbb{R}$ be any bilinear map. We denote by $\sigma_\beta(X,Y)$ the initial topology on X generated by all maps of the form $X \to \mathbb{R}$, $x \mapsto \beta(x,y)$ with $y \in Y$. It is not difficult to see that $\sigma_\beta(X,Y)$ is a locally convex vector space topology on X. Moreover, $\sigma_\beta(X,Y)$ is Hausdorff if and only if β *separates* X, i.e.,

$$\forall x \in X\colon \quad (\forall y \in Y\colon\ \beta(x,y) = 0) \implies x = 0.$$

Furthermore, if β *separates* Y, i.e.,

$$\forall y \in Y\colon \quad (\forall x \in X\colon\ \beta(x,y) = 0) \implies y = 0,$$

then the map $\psi\colon Y \to (X, \sigma_\beta(X,Y))^*$ defined by $\psi(y) = (x \mapsto \beta(x,y))$ is an isomorphism between $(X, \sigma_\beta(X,Y))^*$ and Y.

7.3.4 Proof of Theorem 7.3.7

Everything is in place for stating and proving Theorem 7.3.7. Our argument runs by an application of Farkas' lemma in the form of Corollary 7.3.10.

Proof of Theorem 7.3.7. Let $\rho\colon D^r \to \mathbb{R}$, with $r \in \mathbb{N}$. For contradiction, we assume that $\rho \in \ell\mathrm{Expr}(\Gamma) \setminus \mathrm{wImp}(\mathrm{wPol}(\Gamma))$. Then there exist $k \in \mathbb{N}$ and $\omega \in \mathrm{w}\mathcal{O}_D^{(k)} \cap \mathrm{wPol}(\Gamma)$ such that ρ is not weight-improved by ω, that is, we can find $x^1, \dots, x^k \in D^r$ with $\epsilon := \sum_{f \in \mathrm{Supp}(\omega)} \omega(f)\rho(f(x^1, \dots, x^k)) > 0$. Let us define $\mathcal{F} := \mathrm{Supp}(\omega)$ and $L := (|\mathcal{F}| + 1)(\sup_{f \in \mathcal{F}} |\omega(f)| + 1)$. Since $\rho \in \ell\mathrm{Expr}(\Gamma)$, there exist $\lambda \in \mathbb{R}_{\geq 0}[\mathrm{M}_k(\Gamma)]$ and $c \in \mathbb{R}$ such that, for each $i \in \{1, \dots, k\}$,

$$\left| \rho(x^i) - \left(\sum_{(S,\gamma) \in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma\Big(e_i^{(k)}(S)\Big) + c \right) \right| \leq \frac{\epsilon}{2L}$$

and, for each $f \in \mathcal{F}$,

$$\rho(f(x^1, \dots, x^k)) \leq \sum_{(S,\gamma) \in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c + \frac{\epsilon}{2L}.$$

Now, if $f \in \mathcal{F} \setminus \mathcal{J}_D^{(k)}$, then $\omega(f) \geq 0$; thus

$$\begin{aligned}
&\omega(f)\rho(f(x^1, \ldots, x^k)) \\
&\leq \omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c + \frac{\epsilon}{2L}\right) \\
&\leq \omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c\right) + \frac{\epsilon}{2(|F|+1)}.
\end{aligned}$$

On the other hand, if $f \in \mathcal{F} \cap \mathcal{J}_D^{(k)}$, then $f = e_i^{(k)}$ for some $i \in \{1, \ldots, k\}$, and hence

$$\begin{aligned}
&\omega(f)\rho(f(x^1, \ldots, x^k)) = \omega(f)\rho(x^i) \\
&\leq \omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma\Big(e_i^{(k)}(S)\Big) + c\right) + \frac{\epsilon}{2(|F|+1)} \\
&= \omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c\right) + \frac{\epsilon}{2(|F|+1)}.
\end{aligned}$$

We conclude that

$$\begin{aligned}
&\sum_{f\in \mathrm{Supp}(\omega)} \omega(f)\rho(f(x^1, \ldots, x^k)) \\
&\leq \sum_{f\in \mathrm{Supp}(\omega)} \left(\omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c\right) + \frac{\epsilon}{2(|F|+1)}\right) \\
&\leq \frac{\epsilon}{2} + \sum_{f\in \mathrm{Supp}(\omega)} \omega(f)\left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c\right) \\
&= \frac{\epsilon}{2} + \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\left(\sum_{f\in \mathrm{Supp}(\omega)} \omega(f)\gamma(f(S))\right) \leq \frac{\epsilon}{2},
\end{aligned}$$

which gives the desired contradiction.

Vice versa, let us consider a cost function $\rho\colon D^r \to \mathbb{R}$ with $r \in \mathbb{N}$, and suppose that $\rho \in \mathrm{wImp}(\mathrm{wPol}(\Gamma))$. We show that $\rho \in \ell\mathrm{Expr}(\Gamma)$. In order to do this, let $k \in \mathbb{N}$ and $x^1, \ldots, x^k \in D^r$. Set $M := \mathrm{M}_k(\Gamma)$ and $N := \mathcal{O}_D^{(k)}$. Let us apply Corollary 7.3.10 to the following objects:

- the locally convex topological vector space $Z := \big(\mathbb{R}[N], \tau_{\mathbb{R}[N]}\big)$ (see Section 7.3.3),

- the closed linear subspace $X := \{\omega \in \mathbb{R}[N] \mid \sum_{f \in \mathrm{Supp}(\omega)} \omega(f) = 0\}$ of Z with the (clearly locally convex) subspace topology,
- the locally convex vector space $Y := (\mathbb{R}^M, \sigma_\beta(\mathbb{R}^M, \mathbb{R}[M]))$ for the bilinear map

$$\beta\colon \mathbb{R}^M \times \mathbb{R}[M] \to \mathbb{R}, \qquad \beta(x, \lambda) := \sum_{(S,\gamma) \in \mathrm{Supp}(\lambda)} \lambda(S, \gamma) x(S, \gamma),$$

  which separates both $\mathbb{R}^M$ and $\mathbb{R}[M]$ (see Section 7.3.3; note also that $\sigma_\beta(\mathbb{R}^M, \mathbb{R}[M])$ coincides with the product topology on $\mathbb{R}^M$),
- consider the following continuous linear maps:
  - $\phi\colon X \to \mathbb{R}$ defined by $\phi(\omega) := \sum_{f \in \mathrm{Supp}(\omega)} \omega(f) \rho(f(x^1, \dots, x^k))$,
  - $A\colon X \to Y$ defined by $A(\omega)(S, \gamma) := \sum_{f \in \mathrm{Supp}(\omega)} \omega(f) \gamma(f(S))$, and
  - $B\colon X \to Z$ defined by

$$B(\omega)(f) := \begin{cases} \omega(f) & \text{for } f \in \mathcal{O}_D^{(k)} \setminus \mathcal{J}_D^{(k)}, \\ 0 & \text{for } f \in \mathcal{J}_D^{(k)}. \end{cases}$$

- note that $S := \{x \in \mathbb{R}^M \mid \forall (S, \gamma) \in M\colon\ x(S, \gamma) \geq 0\}$ is a closed convex cone in $\mathbb{R}^M$, while $T := \{\omega \in \mathbb{R}[N] \mid \forall f \in N\colon\ \omega(f) \leq 0\}$ is a closed convex cone in $\mathbb{R}[N]$.

We now claim that Statement (1) of Corollary 7.3.10 is satisfied. Otherwise, there would exist some $x \in X$ such that $A(x) \in S$ and $B(x) \in T$, but $\phi(x) < 0$, i.e., there would be some $\omega \in \mathbb{R}\left[\mathcal{O}_D^{(k)}\right]$ with $\sum_{f \in \mathrm{Supp}(\omega)} \omega(f) = 0$ such that

(a) $\sum_{f \in \mathrm{Supp}(\omega)} \omega(f) \gamma(f(S)) \geq 0$ for all $(S, \gamma) \in M$,

(b) $\omega(f) \leq 0$ for all $f \in \mathcal{O}_D^{(k)} \setminus \mathcal{J}_D^{(k)}$, and

(c) $\sum_{f \in \mathrm{Supp}(\omega)} \omega(f) \rho(f(x^1, \dots, x^k)) < 0$,

so that $-\omega \in \mathrm{wPol}(\Gamma)$ (by (a) and (b)), while $-\omega$ would not weight-improve ρ (due to (c)), which would be a contradiction. Therefore, statement (1) of Corollary 7.3.10 holds, thus (2) as well.

To show that $\rho \in \ell\mathrm{Expr}(\Gamma)$, let $\epsilon > 0$ and let $\mathcal{F}$ be a finite subset of $\mathcal{O}_D^{(k)}$. Without loss of generality, we may and will assume that $\mathcal{J}_D^{(k)} \subseteq \mathcal{F}$. By statement (2) of Corollary 7.3.10, there exist $\mu \in S^+$ and $\nu \in T^+$ such that for all $f \in \mathcal{F}$, and for all $e \in \mathcal{J}_D^{(k)}$ it holds that

$$|\phi(\delta_f - \delta_e) - (\mu(A(\delta_f - \delta_e)) + \nu(B(\delta_f - \delta_e)))| \leq \frac{\epsilon}{2}. \qquad (7.12)$$

Note that $\phi(\delta_f - \delta_e) = \rho(f(x^1, \ldots, x^k)) - \rho(e(x^1, \ldots, x^k))$ whenever $f \in \mathcal{F}$ and $e \in \mathcal{J}_D^{(k)}$. Furthermore, since $(\mathbb{R}^M, \sigma(\mathbb{R}^M, \mathbb{R}[M]))^* \cong \mathbb{R}[M]$ (see Section 7.3.3), there exists $\lambda \in \mathbb{R}[M]$ such that

$$\text{for all } x \in R^M, \qquad \mu(x) = \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)x(S,\gamma).$$

Since $\mu \in S^+$, it follows that $\lambda(S,\gamma) = \mu(\delta_{(S,\gamma)}) \geq 0$, for all $(S,\gamma) \in M$. Furthermore, it holds that

$$\begin{aligned}\mu(A(\delta_f - \delta_e)) &= \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)(\gamma(f(S)) - \gamma(e(S))) \\ &= \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(s)) - \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(e(s)),\end{aligned}$$

for all $f \in \mathcal{F}$, $e \in \mathcal{J}_D^{(k)}$. Moreover, $B(\delta_f - \delta_e) = \delta_f$ for all $f \in \mathcal{F}$, $e \in \mathcal{J}_D^{(k)}$. Let us define

$$\begin{aligned}c_f &:= \rho(f(x^1, \ldots, x^k)) - \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(s)) - \nu(\delta_f), \\ K_e &:= \rho(e(x^1, \ldots, x^k)) - \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(e(s))\end{aligned}$$

for $f \in \mathcal{F}$, and $e \in \mathcal{J}_D^{(k)}$. Then, we can rewrite Inequality (7.12) as follows

$$\text{for all } f \in \mathcal{F}, \;\; \text{and for all } e \in \mathcal{J}_D^{(k)}, \quad |c_f - K_e| \leq \frac{\epsilon}{2}. \tag{7.13}$$

Let us define $c := \min_{f \in \mathcal{F}} c_f$. From Inequality (7.13), it follows that, for each $e \in \mathcal{J}_D^{(k)}$, $|c - K_e| \leq \frac{\epsilon}{2}$, which, by a further application of Inequality (7.13), entails $|c - c_f| \leq \epsilon$, for every $f \in \mathcal{F}$. Let us prove that for every $f \in \mathcal{F}$ it holds that

$$\rho(f(x^1, \ldots, x^k)) \leq \sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c + \epsilon. \tag{7.14}$$

Note that, for every $f \in N$, from $-\delta_f \in T$ and $\nu \in T^+$, it follows that $\nu(\delta_f) = -\nu(-\delta_f) \leq 0$. Consequently, for each $f \in \mathcal{F}$,

$$\begin{aligned}&\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + c \\ \geq &\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(f(S)) + \nu(\delta_f) + c \\ = &\,\rho(f(x^1, \ldots, x^k)) - c_f + c \geq \rho(f(x^1, \ldots, x^k)) - \epsilon,\end{aligned}$$

which proves (7.14). Finally, we prove that for every $i \in \{1, \dots, k\}$

$$\left| \rho(x^i) - \left(\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma\left(e_i^{(k)}(S)\right) + c \right) \right| \leq \epsilon. \tag{7.15}$$

In order to do this, let us consider $i \in \{1, \dots, k\}$ and set $e := e_i^{(k)}$. Then it holds that

$$\begin{aligned} &\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(e(S)) + c \\ \leq &\sum_{(S,\gamma)\in \mathrm{Supp}(\lambda)} \lambda(S,\gamma)\gamma(e(S)) + K_e + \frac{\epsilon}{3} \\ \leq &\,\rho(e(x^1, \dots, x^k)) + \frac{\epsilon}{2} \;=\; \rho(x^i) + \frac{\epsilon}{2}. \end{aligned}$$

The last inequality, together with Inequality (7.14) and the fact that $e \in \mathcal{F}$, implies Inequality (7.15). Hence, $\rho \in \ell\mathrm{Expr}(\Gamma)$. □

7.4 Fractional Polymorphisms with Arbitrary Supports

In this section, we introduce fractional polymorphisms with arbitrary supports. In the previous section, we have seen that in the case of a finite-valued structure Γ with an arbitrary domain the fractional polymorphisms (with a finite support) of Γ are only known to provide a *local* characterisation of the expressive power. A natural question to ask is whether we could obtain *global* information on the expressive power and, therefore, on the computational complexity of VCSPs for infinite-domain finite-valued structures by extending the definition of fractional polymorphisms to fractional operations with possibly infinite supports.

We start by formally defining the notions of a fractional operation and a fractional polymorphism with an arbitrary support over an arbitrary domain D.

Definition 7.4.1. Let D be an arbitrary set, and let k be a positive integer. A k-ary *fractional operation* over D is a probability measure on $\mathcal{O}_D^{(k)}$, i.e., a map $\mu\colon \mathcal{P}(\mathcal{O}_D^{(k)}) \to \mathbb{R}_{\geq 0}$ such that

- $\mu(\emptyset) = 0$, and $\mu(\mathcal{O}_D^{(k)}) = 1$;
- μ satisfies the countable additivity property, i.e., for all countable collections $\{X_n\}_{n\in\mathbb{N}}$ of pairwise disjoint subsets $X_n \subseteq \mathcal{O}_D^{(k)}$, it holds that $\mu(\bigcup_{n\in\mathbb{N}} X_n) = \sum_{n\in\mathbb{N}} \mu(X_n)$.

As usual, given a k-ary fractional operation μ over a domain D, we define its *support* as the set

$$\operatorname{Supp}(\mu) := \{x \in \mathcal{O}_D^{(k)} \mid \mu(\{x\}) > 0\}.$$

Definition 7.4.2. A k-ary fractional operation μ over an arbitrary domain D with an *arbitrary* support *improves* a function $\rho\colon D^n \to \mathbb{R} \cup \{+\infty\}$ if it satisfies

$$\mathbb{E}_{g\sim\mu}\rho(g(a^1, \ldots, a^k)) \leq \frac{1}{k}\sum_{i=1}^{k} \rho(a^i),$$

for every $a^1, \ldots, a^k \in D^n$, where $\mathbb{E}_{g\sim\mu}\rho(g(a^1, \ldots, a^k))$ is the expected value of ρ associated to μ.

The next proposition is a well-known result from probability measure theory and shows that the support of a fractional operation over an arbitrary domain is countable. We report the proof for completeness.

Proposition 7.4.3. *Let μ be a probability measure on an arbitrary space X. Then the support of μ is countable.*

Proof. Let us define, for all $n \in \mathbb{N}$, the subset $A_n := \left\{x \in X \mid \mu(\{x\}) \geq \frac{1}{n}\right\}$. We can write the support of μ as $\operatorname{Supp}(\mu) = \bigcup_{n\in\mathbb{N}} A_n$. Observe that, for every $n \in \mathbb{N}$, the set A_n is finite. If this was not the case, then there would exist a sequence $(x_i)_{i\in\mathbb{N}} \in (A_n)^{\mathbb{N}}$ of pairwise distinct elements and, by the countable additivity property, we would obtain that $\mu(\{x_i \mid i \in \mathbb{N}\})$ $= \sum_{i\in\mathbb{N}} \mu(\{x_i\}) \geq \sum_{i\in\mathbb{N}} \frac{1}{n} = +\infty$, which contradicts the assumption that $\mu(\{x_i \mid i \in \mathbb{N}\}) \leq \mu(X) = 1$. Therefore, $\operatorname{Supp}(\mu)$ is countable, because it is the countable union of finite sets. $\square$

Because of Proposition 7.4.3, we can give an alternative definition of fractional operations with arbitrary supports improving cost functions over arbitrary domains.

Given a k-ary fractional operation μ over an arbitrary set D, we define $\mu(g) := \mu(\{g\})$, for every $g \in \mathcal{O}_D^{(k)}$.

Definition 7.4.4. A k-ary fractional operation μ over an arbitrary domain D *improves* a function $\rho\colon D^n \to \mathbb{R} \cup \{+\infty\}$ if it satisfies

$$\sum_{g\in\operatorname{Supp}(\omega)} \mu(g)\rho(g(a^1, \ldots, a^k)) \leq \frac{1}{k}\sum_{i=1}^{k} \rho(a^i),$$

for every $a^1, \ldots, a^k \in D^n$.

Definition 7.4.5. Let D be an arbitrary set, and let Γ be a valued structure with domain D. We say that μ is a *fractional polymorphism with an arbitrary support* of Γ if μ improves every cost function of Γ. The set of all fractional polymorphisms (with arbitrary supports) of a valued structure Γ is denoted by $\mathrm{fPol}_\infty(\Gamma)$. The set of all functions that are improved by a given set of fractional operations (with arbitrary supports) Ω is denoted by $\mathrm{Imp}(\Omega)$.

If Γ is a valued structure with a finite domain, then $\mathrm{fPol}(\Gamma) = \mathrm{fPol}_\infty(\Gamma)$. If Γ is a valued structure with an infinite domain, then $\mathrm{fPol}(\Gamma) \subseteq \mathrm{fPol}_\infty(\Gamma)$, and therefore it holds that $\mathrm{Imp}(\mathrm{fPol}_\infty(\Gamma)) \subseteq \mathrm{Imp}(\mathrm{fPol}(\Gamma))$. The following result can be proved using the same proof as for Lemma 7.2.1 (in fact, in that case, the finiteness of the support of fractional polymorphisms did not play any role in the argument).

Proposition 7.4.6. *Let D be an infinite set, and let Γ be a valued structure with an arbitrary domain D. Then, every fractional polymorphism of Γ improves every cost function that is expressible in Γ, that is,*

$$\langle \Gamma \rangle \subseteq \mathrm{Imp}\left(\mathrm{fPol}_\infty(\Gamma)\right).$$

Summary and Outlook

In this chapter, we have presented some extension of concepts and results from the algebraic theory of finite-domain valued structures to the arbitrary-domain case. In particular, we have characterised the set of cost functions which are improved by all fractional polymorphisms of a finite-valued structure. This characterisation was given in terms of *local* expressive power of a finite-valued structure. Whether fractional polymorphisms characterise the expressive power of infinite-domain finite-valued structures remains an open question.

Chapter 8

PL VCSPs with Fixed Number of Variables

For valued structures with infinite signatures, it makes sense to consider the restricted version of the VCSP where only a fixed number of variables is allowed in the input. In general, the VCSP for *all* PL cost functions is NP-complete (see Proposition 1.2.16 and Remark 1.2.17). In this chapter, we prove that the restriction of the VCSP for *all* PL cost functions to instances with a fixed number of variables is polynomial-time solvable.

The restriction to a fixed number of variables has been studied for several problems in computational optimisation and, usually, this kind of restriction has led to an improvement in the computational complexity: two remarkable examples of this situation are the combinatorial polynomial-time algorithm of Megiddo [83] to solve the restriction to a fixed number of variables of LINEAR PROGRAMMING (in its full generality, LINEAR PROGRAMMING can be solved in polynomial time (see, e.g., [63, 65, 100]), but all the known algorithms rely on infinite approximation procedures); and the algorithm of Lenstra [79] to solve the restriction to a fixed number of variables of Integer Programming Feasibility (which is a NP-complete problem in its full generality).

Throughout the chapter, we assume that the input is given as a sum of PL cost functions (the objective function), which are represented as in Definition 1.2.14, i.e., the coefficients for linear polynomials and the coefficients for linear constraints showing up in the input are represented in binary.

We first formalise the problem that we want to focus on.

Definition 8.0.1. Let d be a positive integer, and let $V := \{x_1, \ldots, x_d\}$ be a set of variables. An *instance* I of the *valued constraint satisfaction problem (VCSP) for PL cost functions with variables in* V consists of an expression

ϕ of the form

$$\sum_{i=1}^{m} f_i(x_1^i, \ldots, x_{\mathrm{ar}(f_i)}^i),$$

where $f_1, \ldots, f_m$ are finitely many PL cost functions represented as in Definition 1.2.14 and all the x_j^i are variables from V. The task is to find the *infimum cost* of ϕ, defined as

$$\inf_{\alpha\colon V \to \mathbb{Q}} \sum_{i=1}^{m} f_i(\alpha(x_1^i), \ldots, \alpha(x_{\mathrm{ar}(f_i)}^i)),$$

and to decide whether it is attained, i.e., whether it is a proper minimum.

In the remainder of the chapter, we use the fact that Linear Program Feasibility for a set of linear constraints containing also strict inequalities can be solved in polynomial time. Given a set of linear constraints l and a linear expression obj, we denote by $LPF(l)$ the LPF instance defined by the linear constraints in l, and we denote by $LP(l, \mathrm{obj})$ the LP instance defined by the linear constraints in l and by the objective function obj. In LP and LPF the feasibility polytope is defined by weak linear inequalities, i.e., by linear constraints of the form $\sum_{j=1}^{n} a_j x_j \leq b$. The feasibility of a set of linear constraints containing also strict linear inequalities (i.e., of the form $\sum_{j=1}^{n} a_j x_j < b$) can be solved by solving a linear number of linear programs, as shown in [59], where the authors give a polynomial-time algorithm deciding the feasibility of a set of *Horn disjunctive linear constraints*. However, the feasibility of a set of linear constraints containing strict and weak linear inequalities can be decided by solving only one LP instance.

Lemma 8.0.2 (Motzkin Transposition Theorem [84, 89])**.** *Let $A \in \mathbb{Q}^{k_1 \times d}$, and $B \in \mathbb{Q}^{k_2 \times d}$ be matrices such that* $\max(k_1, d) \geq 1$. *The system*

$$\begin{cases} Ax < 0 \\ Bx \leq 0 \end{cases}$$

has a solution $x \in \mathbb{Q}^d$ if, and only, if the system

$$\begin{cases} A^T y + B^T z = 0 \\ y \geq 0,\ z \geq 0 \end{cases}$$

does not admit a solution $(y, z) \in \mathbb{Q}^{k_1 + k_2}$ such that $y \neq (0, \ldots, 0)$.

Proposition 8.0.3. *The* Linear Program Feasibility *problem (LPF) for a finite set of strict or weak linear inequalities is polynomial-time many-one reducible to LP, and therefore it can be solved in polynomial time.*

Proof. Let us assume that the linear constraints in the input consist of k_1 strict inequalities, and k_2 weak inequalities, i.e., we have to check the satisfiability of the following system

$$\begin{cases} \sum_{i=1}^{d} a_{j,i}x_i + a_{j,d+1} < 0 & \text{for } 1 \leq j \leq k_1 \\ \sum_{i=1}^{d} b_{j,i}x_i + b_{j,d+1} \leq 0 & \text{for } 1 \leq j \leq k_2. \end{cases} \tag{8.1}$$

Let us first observe that the system (8.1) is equivalent to the following one

$$\begin{cases} \sum_{i=1}^{d+1} a_{j,i}t_i < 0 & \text{for } 1 \leq j \leq k_1 \\ -t_{d+1} < 0 \\ \sum_{i=1}^{d+1} b_{j,i}t_i \leq 0 & \text{for } 1 \leq j \leq k_2. \end{cases} \tag{8.2}$$

Indeed, if $(t_1, \ldots, t_d, t_{d+1})$ is a solution for (8.2), then $(x_1, \ldots, x_d)$ with $x_i := \frac{t_i}{t_{d+1}}$ is a solution for (8.1); vice versa if $(x_1, \ldots, x_d)$ is a solution for (8.1), then $(x_1, \ldots, x_d, 1)$ is a solution for (8.2). Let us consider the following linear program

$$\begin{aligned} \text{minimise} \qquad & \sum_{j=1}^{k_1+1} (-y_j) \\ \text{subject to} \qquad & A^T y + B^T z = 0 \\ & -y \leq 0 \\ & -z \leq 0, \end{aligned} \tag{8.3}$$

with variables $y_1, \ldots, y_{k_1+1}, z_1, \ldots, z_{k_2}$,where $A \in \mathbb{Q}^{(k_1+1)\times(d+1)}$ is the matrix such that $(A)_{ji} = a_{ji}$ for $1 \leq j \leq k_1$ and $1 \leq i \leq d+1$ and the (k_1+1)-th row of A is $(0, \ldots, 0, -1)$; and the matrix $B \in \mathbb{Q}^{(k_2)\times(d+1)}$ is such that $(B)_{ji} = b_{ji}$.

Observe that the linear program (8.3) can be computed in polynomial time (in the size of the input). By Lemma 8.0.2, the system (8.2) is satisfiable if, and only if, the feasibility polytope determined by the linear constraints in (8.3) does not admit a solution $(y, z) \in \mathbb{Q}^{(k_1+1)+k_2}$ such that $y \neq (0, \ldots, 0)$. If the output of the algorithm for LP on the input instance (8.3) is $+\infty$ or a tuple having 0 in the first $k_1 + 1$ coordinates, then the system (8.1) is satisfiable, and therefore we accept. Otherwise, if the output is $-\infty$ or a tuple $(y, z) \in \mathbb{Q}^{(k_1+1)+k_2}$ such that $y \neq (0, \ldots, 0)$, then the system (8.1) is not satisfiable and we reject. □

We exhibit a polynomial-time algorithm that finds the infimum of the objective function while saying whether it is attained, i.e., whether it is a proper minimum. The polynomial-time solvability of the problem in the threshold formulation trivially follows (see Remark 1.1.4). The following theorem uses an idea that appeared in [10], Observation 17.

Theorem 8.0.4. *Let V be a finite set of variables. Then there is an algorithm that solves the* VCSP *for PL cost functions having variables in V in polynomial time.*

Proof. We prove that Algorithm 2 correctly solves the VCSP for PL cost functions with variables in V in polynomial time. An input of an instance of the VCSP is a representation of an objective function ϕ as the sum of a finite number of given cost functions, $f_1, \ldots, f_n$, applied to some of the variables in $V = \{x_1, \ldots, x_d\}$, that is,

$$\phi(x_1, \ldots, x_d) = \sum_{i=1}^{n} f_i(x^i),$$

where $x^i \in V^{\operatorname{ar}(f_i)}$ for $1 \leq i \leq d$. We want to point out that even if, a priori, every cost function f_i has a certain arity that does not depend on d, as it is applied to a tuple $x^i = (x^i_1, \ldots, x^i_{\operatorname{ar}(f_i)}) \in V^{\operatorname{ar}(f_i)}$ we can see f_i as a d-ary function. Therefore, for $1 \leq i \leq d$, we can assume that the cost function f_i is defined for every $x \in \mathbb{Q}^d$ by

$$f_i(x) = \begin{cases} \sum_{j=1}^{d} a_j^{i,l} x_j + b^{i,l} & \text{if } C_{i,l}(x), \text{ for some } 1 \leq l \leq m_i \\ +\infty & \text{otherwise.} \end{cases}$$

For every $1 \leq l \leq m_i$ the formulas $C_{i,l}(x)$ have the following form:

$$C_{i,l}(x) = \bigwedge_{j=1}^{p} (h_j^{i,l}(x) \leq 0) \wedge \bigwedge_{j=p+1}^{q} (h_j^{i,l}(x) < 0) \wedge \bigwedge_{j=q+1}^{r} (h_j^{i,l}(x) = 0),$$

for some $p, q, r \in \mathbb{N}$ and for some linear polynomials $h_j^{i,l} \colon \mathbb{Q}^d \to \mathbb{Q}$, where $1 \leq j \leq r$. We assume that the cost functions f_i are represented as in Definition 1.2.14.

Algorithm 2 first extracts the list of linear polynomials $p_1, \ldots, p_k$ that appear in the list of linear constraints defining some cost function f_i , i.e.,

$$\{p_1, \ldots, p_k\} := \bigcup_{i=1}^{n} \bigcup_{l=1}^{m_i} \bigcup_{j} \left\{ h_j^{i,l} \right\}.$$

Observe that the linear polynomials $p_1, \ldots, p_k$ decompose the space $\mathbb{Q}^d$ into σ polyhedral sets, where

$$\sigma \leq \tau_d(k) = \sum_{i=0}^{d} 2^i \binom{k}{i} \tag{8.4}$$

and that this bound is tight, i.e., $\sigma = \tau_d(k)$ whenever the hyperplanes defined by $p_i(x) = 0$, for $1 \le i \le k$, are in general position.

Inequality (8.4) can be verified by induction on the number, k, of hyperplanes. Clearly, for all $d \in \mathbb{N}$, one hyperplane divides $\mathbb{Q}^d$ into $3 = 2^0 + 2^1$ polyhedral sets. Suppose now that $k \ge 2$ and that Inequality (8.4) is true for every d and for at most $k-1$ hyperplanes. Suppose that the k hyperplanes are in general position (we get in this way the upper bound $\tau_d(k)$). Observe that, by adding the hyperplanes one-by-one, the k-th hyperplane intersects at most $\tau_{d-1}(k-1)$ of the polyhedral sets obtained until the previous step. In fact, this number is equal to the number of polyhedral sets in which a hyperplane (that is, a subspace of dimension $d-1$) is divided by $k-1$ subspaces of dimension $d-2$.

Suppose that we know how the space is decomposed into polyhedral sets by the hyperplanes $p_1(x) = 0, \ldots, p_{k-1}(x) = 0$. Adding $p_k(x) = 0$ to the list of hyperplanes decomposing the space, each one of the polyhedral sets intersecting it is divided into three polyhedral sets (corresponding to $p_k(x) < 0$, $p_k(x) = 0$, and $p_k(x) > 0$, respectively). Summing up, at every step we add to the "old polyhedral sets" (i.e., polyhedral sets obtained until the previous step) two more polyhedral sets for each of the old ones intersecting $p_k(x) = 0$, then it follows that

$$\tau_d(k) = \tau_d(k-1) + 2\tau_{d-1}(k-1).$$

Using this equality and the inductive hypothesis we obtain

$$\begin{aligned}
\tau_d(k) =& 2\sum_{i=0}^{d-1} 2^i \binom{k-1}{i} + \sum_{i=0}^{d} 2^i \binom{k-1}{i} \\
=& \sum_{i=1}^{d} 2^i \left(\binom{k-1}{i-1} + \binom{k-1}{i} \right) + 1 \\
=& \sum_{i=1}^{d} 2^i \binom{k}{i} + 1 = \sum_{i=0}^{d} 2^i \binom{k}{i}.
\end{aligned}$$

In particular, the number σ of polyhedral sets is bounded by a polynomial in k, and the Algorithm 2 produces a tree, which has 3^k branches a priori but actually has polynomially many branches.

The algorithm computes the list of all non-empty polyhedral sets by computing at most $\sum_{i=1}^{k-1} \tau_d(i)$ instances of linear program feasibility, and then it computes the infimum of the objective function in every non-empty polyhedral set by computing at most $3\tau_d(k)$ linear programs. Observe that the only closed and bounded non-empty polyhedral sets computed by Algorithm 2 are 0-dimensional subspaces, i.e., points, and all the other polyhedral sets computed are open or unbounded. Therefore, in order to check whether the infimum in a polyhedral set C is a proper minimum, it is enough to check whether the objective function is constant in C, that is, whether its infimum in C is equal to its supremum in C. This check is done by solving at most $3\tau_d(k)$ further linear programs. The linear expression of the objective function in a polyhedral set can be computed by running a number of Linear Program Feasibility instances that is polynomial in the size of the input instance. Globally, the running time of Algorithm 2 is polynomial in the size of the input instance. □

ALGORITHM 2: Algorithm for PL VCSPs with a Fixed Number of Variables

Input: $\phi(x) = f_1(x) + \cdots + f_n(x)$ with $f_i(x) = f_{ij}(x)$ if $x \in C_{ij}$, and the C_{ij}'s each given as a finite set of linear conditions, for $1 \leq j \leq n_i$, and $1 \leq i \leq n$.

Output: $(\text{val}, \text{attr})$ where val is the value of the infimum of the objective function, and attr is a string which specifies whether val is attained ($\text{attr} =$ "min") or not ($\text{attr} =$ "inf").

$\{p_1, \ldots, p_k\} :=$ the set of all the linear functions appearing in the C_{ij}'s;
$L := \{\{\}\}$ (the set of polyhedral sets in which the p_i's divide the space);
for $i = 1, ..., k$ **do**
    for *each* l *in* L **do**
        $l_{-1} := l \cup \{p_i < 0\}$;
        $l_0 := l \cup \{p_i = 0\}$;
        $l_1 := l \cup \{-p_i < 0\}$;
        for $j = -1, 0, 1$ **do**
            if *LPF*(l_j) $=$ *yes* **then**
                $L := (L \setminus \{l\}) \cup \{l_j\}$
            end
        end
    end
end
val $:= +\infty$;
attr $:=$" inf";
for *each* l *in* L **do**
    $l_c := \{\}$ (the closure of l);
    for *each* $c \in l$ **do**
        if c *is of the form* $(p < 0)$ **then**
            $l_c := l_c \cup \{p \leq 0\}$
        else
            $l_c := l_c \cup \{c\}$
        end
    end
    for $i = 1, ..., n$ **do**
        $g_i := +\infty$;
        for $j = 1, \ldots, n_i$ **do**
            if *LPF*($l \cup C_{ij}$) $=$ *yes* **then**
                $g_i(x) := f_{ij}(x)$
            end
        end
    end
    obj $:= \sum_{i=1}^{n} g_i(x)$ (the linear expression of the ϕ in l);
    $m :=$ LP(l_c, obj} (the infimum of ϕ in l);
    $M := -$LP(l_c, $-$obj) (the supremum of ϕ in l);
    if $m <$ val **then**
        if $m = M$ **then**
            attr $:=$ "min"(the infimum is attained iff ϕ is constant in l)
        else
            attr $:=$ "inf"
        end
    end
end
return (val, attr);

Chapter 9

Conclusion and Open Problems

In this final chapter, we summarise the results of the thesis and list some open problems.

With this thesis, we initiated the systematic research on VCSPs over infinite domains. We focussed on piecewise linear (PL) and piecewise linear homogeneous (PLH) valued structures, which provide a mathematically elegant framework for many interesting computational problems. In Chapter 2, we proved that the VCSP for all convex PL cost functions is polynomial-time solvable, exhibited a class of PL valued structures whose VCSP is equivalent to the corresponding feasibility problem and presented a family of PL valued structures whose VCSP is NP-hard. In Chapter 3, we provided an efficient sampling algorithm for PLH valued structures, i.e., a polynomial-time many-one reduction of their VCSPs to finite-domain ones. In Chapter 4, we gave a sufficient condition for the polynomial-time solvability of VCSPs for valued structures admitting an efficient sampling algorithm: if the valued structure has fully symmetric fractional polymorphisms of all arities, then the VCSP is correctly solved in polynomial time by a combination of the sampling algorithm and the basic linear programming relaxation. In Chapter 5, we applied results of Chapters 3, and 4 to show the polynomial-time solvability of convex PLH and componentwise decreasing PLH (and its dual class of componentwise increasing PLH valued structures) valued structures. We also showed that the class of componentwise decreasing valued structures (as its dual class of componentwise increasing valued structures) is maximally tractable within the class of PLH valued structures. Such maximal tractability results are of particular importance for the more ambitious goal to classify the complexity of the VCSP for all classes of PLH valued structures: to prove a dichotomy it suffices to identify all maximally tractable classes. Submodular PLH valued structures are the object of Chapter 6, in which we exhibited two different approaches to solve the VCSP for submodular PLH valued structures in polynomial time, and we showed that this class of valued structures is maximally tractable. Both approaches used to show the polynomial-time solvability of submodular PLH VCSPs rely on the efficient sampling algorithm for PLH valued structures: the first approach is a combination of the sampling algorithm and the basic linear programming relaxation; the second approach consists in transferring the problem in the ring of formal Laurent power series, $\mathbb{Q}^\star$, and using a fully combinatorial polynomial-time algorithm to solve the finite-domain $\mathbb{Q}^\star$-valued problem obtained by sampling. In Chapter 7, we extended some concepts from the algebraic theory of finite-domain VCSPs to the infinite-domain case and showed that the expressive power of finite-valued structures with arbitrary countable domains is *locally* characterised by its fractional polymorphisms. Finally, in Chapter 8, we provided a polynomial-time algorithm solving the restriction of the VCSP for *all* PL valued cost functions to a fixed number of variables.

9.1 Open Problems

We list some interesting open questions and challenges for future research on infinite-domain VCSPs.

Submodular PL Valued Structures

We have shown (in Chapter 6) that the VCSP for submodular PLH valued structures is polynomial-time solvable; this result relies on the existence of an efficient sampling algorithm for PLH valued structures. A challenge is to extend our tractability result to the class of all submodular PL VCSPs. We believe that submodular PL VCSPs are polynomial-time solvable. However, proving our conjecture would require an approach that is not based on a sampling technique. In fact, already the relational structure $(\mathbb{Q}; 0, S, D)$ where $S := \{(x, y) \mid y = x + 1\}$, and $D := \{(x, y) \mid y = 2x\}$ (which has both min and max as polymorphisms) does not admit an efficient sampling algorithm (it is easy to see that for $d \in \mathbb{N}$, every sample computed on input d must have exponentially many vertices in d).

The PLH Cost Function k

Let us consider the valued structure Γ with domain $\mathbb{Q}$ and a signature containing a unique function symbol k whose interpretation is the cost function

$k^\Gamma \colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by

$$k^\Gamma(x,y) = \begin{cases} y & \text{if } -x \leq y \leq x \\ x & \text{if } -y \leq x \leq y \\ -y & \text{if } x \leq y \leq -x \\ -x & \text{if } y \leq x \leq -y. \end{cases}$$

We aim to determine the computational complexity of VCSP(Γ). Observe that, since Γ is PLH, there is an efficient sampling algorithm for Γ (see Chapter 3). However, if Δ is the finite-domain sample for Γ_k computed on some input $n > 0$, then Δ satisfies the sufficient condition for the NP-hardness of the VCSP for finite-domain finite-valued structures (see [97]). To see this it is enough to observe that $x \in \mathrm{dom}(\Delta)$ implies $-x \in \mathrm{dom}(\Delta)$, and that $\arg\min(k^\Delta) = \{(-d,d),(d,-d)\}$, where $d := \max\{|d| \mid d \in \mathrm{dom}(\Delta)\}$.

Binary Fully Symmetric Fractional Polymorphisms

A finite-domain valued structure Γ has fully symmetric fractional polymorphisms of all arities if, and only if, it has a binary fully symmetric fractional polymorphism (cf. [69]). We ask whether this characterisation can be extended to valued structures with domains of arbitrary cardinality. Note that the proof in the finite-domain case relies on the finiteness of the domain, and therefore it cannot work in the infinite-domain case. We think, instead, that to answer our question, one should take into account fractional polymorphisms with arbitrary supports (see Section 7.4). Proving that the existence of binary fully symmetric fractional polymorphisms characterises of the existence of fully symmetric polymorphisms of all arities for valued structures with arbitrary domains would imply the polynomial-time solvability of VCSPs for PLH valued structures improved by a binary fractional polymorphism whose complexity is not known, e.g., *bisubmodular* PLH valued structures. The notion of bisubmodularity was defined for functions over domains with 3 element (see, e.g., [87]), however this definition can be extended to functions over $\mathbb{Q}$ [1]. A valued structure Γ with domain $\mathbb{Q}$ (or $\mathbb{R}$) is *bisubmodular* if every cost functions $f \colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ in Γ has the binary fractional polymorphism ω defined by

$$\omega(g) := \begin{cases} \frac{1}{2} & \text{if } g = \min_0 \text{, or } g = \max_0 \\ 0 & \text{otherwise,} \end{cases}$$

[1]The definition of bisubmodular functions with domain $\mathbb{Q}$ was suggested by Johan Thapper.

where

$$\min_0(x,y) := \begin{cases} \min(x,y) & \text{if } x \cdot y > 0 \\ 0 & \text{otherwise,} \end{cases}$$
$$\max_0(x,y) := \begin{cases} \max(x,y) & \text{if } x \cdot y > 0 \\ 0 & \text{otherwise,} \end{cases}$$

for every $x, y \in \mathbb{Q}$.

The Expressive Power of Infinite-Domain Finite-Valued CSPs

The computational complexity of the VCSP for a valued structure Γ with a finite domain can be determined by studying the expressive power of Γ which turns out to be equal to the set of cost functions that are improved by *all* fractional polymorphisms of Γ (see [33, 43]). In the infinite-domain setting, fractional polymorphisms already fail to characterise the computational complexity of the feasibility problem associated with the VCSP for a PL valued structure (see [10]). In Chapter 7, we have seen that in the finite-valued case (that is, the case in which the cost functions take values $< +\infty$), fractional polymorphisms characterise the *local expressive power* of valued structures (with an arbitrary signature) over an arbitrary countable domain.

However, we do not know whether there exist infinite-domain finite-valued structures that are not captured by any fractional polymorphism and nevertheless have a polynomial-time solvable VCSP. Deciding whether fractional polymorphisms provide a global characterisation of the computational complexity of the VCSP for infinite-domain finite-valued structures, or at least for PL finite-valued structures, remains an open challenge.

Promise VCSPs over Arbitrary Domains

Let $\mathfrak{A}$ and $\mathfrak{B}$ be two (relational) structures over the same signature τ and domain A and B, respectively, such that there exists a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. The *Promise Constraint Satisfaction Problem* (see [22, 27]) for $\mathfrak{A}$ and $\mathfrak{B}$, or *Promise CSP($\mathfrak{A}, \mathfrak{B}$)* for short, is the following computational problem.

Definition 9.1.1. An *instance* I of Promise CSP(Γ, Δ) consists of a finite conjunction ψ of atomic τ-formulas. The task is to output

- YES if ψ is satisfiable in $\mathfrak{A}$;
- NO if ψ is not satisfiable in $\mathfrak{B}$.

The notion of Promise CSPs can be extended to capture optimisation problems in a similar way as CSPs can be extended to VCSPs.

Let Γ and Δ be two valued structures over the same signature τ and domain C and D, respectively, such that there exists a fractional homomorphism from Γ to Δ. The *Promise Valued Constraint Satisfaction Problem*[2] for Γ and Δ, or *Promise VCSP(*Γ, Δ*)* for short, is the following computational problem.

Definition 9.1.2. An *instance* I of Promise VCSP(Γ, Δ) consists of

- a finite set of variables V_I,
- an expression ϕ_I of the form

$$\sum_{i=1}^{m} f_i(x_1^i, \dots, x_{\mathrm{ar}(f_i)}^i)$$

  where $f_1, \dots, f_m \in \tau$ and all the x_j^i are variables from V_I, and
- a value $u_I \in \mathbb{Q}$.

The task is to output

- YES if there exists an assignment $\alpha \colon V_I \to \mathrm{dom}(\Gamma)$ with cost

$$\sum_{i=1}^{m} f_i^{\Gamma}(\alpha(x_1^i), \dots, \alpha(x_{\mathrm{ar}(f_i)}^i)) \leq u;$$

- NO if *every* assignment $\beta \colon V_I \to \mathrm{dom}(\Delta)$ has cost

$$\sum_{i=1}^{m} f_i^{\Delta}(\beta(x_1^i), \dots, \beta(x_{\mathrm{ar}(f_i)}^i)) \not\leq u.$$

We think that the computational complexity of Promise VCSPs for valued structures Γ and Δ having, respectively, a finite and an infinite domain is an interesting topic for future research, which is closely related with the study of infinite-domain valued structures. For example, it is easy to prove using a similar argument as in Chapter 4 that if Δ has fully symmetric fractional polymorphisms of all arities, then Promise VCSP(Γ, Δ) is polynomial-time solvable.

[2] Promise VCSPs have been defined (for finite-domain valued structures) by Alexandr Kazda (personal communication).

Semialgebraic VCSPs with Fixed Number of Variables

We would like to continue the line of research on the computational complexity of VCSPs with a fixed number of variables by studying the computational complexity of the VCSP with a fixed number of variables and *semialgebraic* cost functions. A function $f\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called semialgebraic if its domain can be represented as the union of finitely many *basic semialgebraic sets* (see [10]) of the form $\{x \in \mathbb{R}^n \mid \chi(x)\}$ where χ is a conjunction of (weak or strict) polynomial inequalities with integer coefficients, relative to each of which $f(x)$ is given by a polynomial expression with integer coefficients.

The VCSP for all semialgebraic cost function is equivalent to the existential theory of the reals (see [10]), which is in PSpace (see [28]). The restriction of the feasibility problem associated with a semialgebraic VCSP to a fixed number of variables is polynomial-time solvable by cylindrical decomposition (cf. [35]). However, we do not know whether this approach can solve our optimisation problem in polynomial time. Another contribution related with our open problem was given in [66] by Khachiyan and Porkolab, who proved that the problem of minimising a convex polynomial objective function with integer coefficients over a fixed number of integer variables, subject to polynomial constraints with integer coefficients that define a convex region, can be solved in polynomial time in the size of the input.

Bibliography

[1] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. URL: https://doi.org/10.1023/B:MACH.0000033116.57574.95.

[2] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation*. Springer-Verlag, Berlin, Heidelberg, 1998. URL: https://doi.org/10.1007/978-1-4612-0701-6.

[3] Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 184 –196. Springer Verlag, 2008. URL: https://doi.org/10.1007/978-3-540-70583-3_16.

[4] Manuel Bodirsky, Peter Jonsson, and Trung Van Pham. The complexity of phylogeny constraint satisfaction problems. *ACM Transactions on Computational Logic (TOCL)*, 18(3), 2017. URL: https://doi.org/10.1145/3105907.

[5] Manuel Bodirsky, Peter Jonsson, and Timo Von Oertzen. Essential convexity and complexity of semi-algebraic constraints. *Logical Methods in Computer Science*, 8(4), 2012. URL: https://lmcs.episciences.org/1218.

[6] Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. *Theory of Computing Systems*, 3(2):136–158, 2008. URL: https://doi.org/10.1007/s00224-007-9083-9.

[7] Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2), February 2010. URL: https://doi.org/10.1145/1667053.1667058.

[8] Manuel Bodirsky, Dugald Macpherson, and Johan Thapper. Constraint satisfaction tractability from semi-lattice operations on infinite

sets. *ACM Transaction of Computational Logic (TOCL)*, 14(4):1–30, 2013. URL: `https://doi.org/10.1145/2528933`.

[9] Manuel Bodirsky and Marcello Mamino. Max-closed semilinear constraint satisfaction. In *Proceedings of the 11th International Computer Science Symposium on Computer Science (CSR)— Theory and Applications*, volume 9691, page 88–101, Berlin, Heidelberg, 2016. Springer-Verlag. URL: `https://doi.org/10.1007/978-3-319-34171-2_7`.

[10] Manuel Bodirsky and Marcello Mamino. Constraint satisfaction problems over numeric domains. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7, pages 79–111. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/6958`.

[11] Manuel Bodirsky and Marcello Mamino. A polynomial-time algorithm for median-closed semilinear constraints, 2018. URL: `https://arxiv.org/abs/1808.10068`, `arXiv:1808.10068`.

[12] Manuel Bodirsky and Marcello Mamino. Tropically convex constraint satisfaction. *Theory of Computing Systems*, 62(3):481–509, 2018. URL: `https://doi.org/10.1007/s00224-017-9762-0`.

[13] Manuel Bodirsky, Marcello Mamino, Barnaby Martin, and Antoine Mottet. The complexity of linear diophantine constraints. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 117, pages 33:1–33:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2018/9615`.

[14] Manuel Bodirsky, Marcello Mamino, and Caterina Viola. Submodular functions and valued constraint satisfaction problems over infinite domains. In *Proceedings of the 27th EACSL Annual Conference on Computer Science Logic (CSL)*, pages 12:1–12:22, 2018. URL: `http://drops.dagstuhl.de/opus/volltexte/2018/9679`.

[15] Manuel Bodirsky, Marcello Mamino, and Caterina Viola. Piecewise linear valued CSPs solvable by linear programming relaxation. 2019. Submitted for journal publication. Preprint available at arxiv.org/abs/1912.09298.

[16] Manuel Bodirsky, Marcello Mamino, and Caterina Viola. Piecewise Linear Valued Constraint Satisfaction Problems with Fixed Number of Variables. 2020. Accepted for presentation at CTW2020

and publication in AIRO Springer Series. Preprint available at arxiv.org/abs/2003.00963.

[17] Manuel Bodirsky, Barnaby Martin, Michael Pinsker, and András Pongrácz. Constraint satisfaction problems for reducts of homogeneous graphs. *SIAM Journal on Computing*, 48(4):1224–1264, 2019. URL: `https://doi.org/10.1137/16M1082974`.

[18] Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006. URL: `https://doi.org/10.1093/logcom/exi083`.

[19] Manuel Bodirsky and Michał Wrona. Equivalence constraint satisfaction problems. In *Proceedings of the 21th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 16, pages 122–136. Dagstuhl Publishing, 2012. URL: `https://drops.dagstuhl.de/opus/volltexte/2012/3668/pdf/14.pdf`.

[20] V. G. Bodnarčuk, L. A. Kalužnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras, part I and II. *Cybernetics*, 5:243–539, 1969. URL: `https://doi.org/10.1007/BF01070906`.

[21] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL: `https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`.

[22] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 1782–1801, USA, 2018. Society for Industrial and Applied Mathematics. URL: `https://doi.org/10.1137/1.9781611975031.117`.

[23] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010. URL: `https://link.springer.com/book/10.1007%2F978-0-387-70914-7`.

[24] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, Berkeley, CA, USA, 2017. URL: `https://ieeexplore.ieee.org/document/8104069`.

[25] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal*

on Computing, 34:720–742, 2005. URL: https://doi.org/10.1137/S0097539700376676.

[26] Andrei A. Bulatov and Dániel Marx. The complexity of global cardinality constraints. In *Proceedings of the 24th IEEE Annual Symposium on Logic In Computer Science, (LICS*, pages 419–428, Washington, DC, USA, 2009. IEEE Computer Society. URL: https://doi.org/10.1109/LICS.2009.38.

[27] Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st ACM Annual Symposium on Theory of Computing (STOC)*, page 602–613, New York, NY, USA, 2019. Association for Computing Machinery. URL: https://doi.org/10.1145/3313276.3316300.

[28] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the 20th ACM Annual Symposium on Theory of Computing (STOC)*, pages 460–467, New York, NY, USA, 1988. ACM. URL: https://doi.org/10.1145/62212.62257.

[29] Clément Carbonnel, Miguel Romero, and Stanislav Živný. The complexity of general-valued CSPs seen from the other side. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 236–246, 2018. URL: https://ieeexplore.ieee.org/document/8555109.

[30] Clément Carbonnel, Miguel Romero, and Stanislav Živný. Point-width and Max-CSPs. In *Proceedings of the 34th ACM-IEEE Annual Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2019. URL: https://doi.org/10.1109/LICS.2019.8785660.

[31] Tsu-Wu J. Chou and George E. Collins. Algorithms for the solution of systems of linear Diophantine equations. *SIAM Journal on Computing*, 11:687–708, 1982. URL: https://epubs.siam.org/doi/10.1137/0211057.

[32] David Cohen, Martin Cooper, Peter Jeavons, and Andrei Krokhin. Supermodular functions and the complexity of Max-CSP. *Discrete Applied Mathematics*, 149(1):53 – 72, 2005. URL: https://doi.org/10.1016/j.dam.2005.03.003.

[33] David A. Cohen, Martin C. Cooper, Páidí. Creed, Peter G. Jeavons, and Stanislav. Živný. An algebraic theory of complexity for discrete optimization. *SIAM Journal on Computing*, 42(5):1915–1939, 2013. URL: https://epubs.siam.org/doi/abs/10.1137/130906398.

[34] 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. URL: https://doi.org/10.1016/j.artint.2006.04.002.

[35] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: A synopsis. *SIGSAM Bull.*, 10(1):10–12, 1976. URL: https://doi.org/10.1145/1093390.1093393.

[36] Martin C. Cooper. Linear-time algorithms for testing the realisability of line drawings of curved objects. *Artificial Intelligence*, 108(1):31 – 67, 1999. URL: https://doi.org/10.1016/S0004-3702(98)00118-0.

[37] Nadia. Creignou, Sanjeev. Khanna, and Madhu. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems.* Society for Industrial and Applied Mathematics, 2001. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898718546.

[38] Roman J Dwilewicz. A short history of convexity. *Differential Geometry-Dynamical Systems*, 11:112–129, 2009. URL: emanticscholar.org/paper/A-short-history-of-Convexity-Dwilewicz/c8726356c97e686242cf8e0a8e21004f854cd321.

[39] 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*, 28:57–104, 1999. URL: 10.1137/S0097539794266766.

[40] Jeanne Ferrante and Charles Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975. URL: https://epubs.siam.org/doi/10.1137/0204006.

[41] Satoru Fujishige. *Submodular Functions and Optimization.* Volume 47 of Annals of Discrete Mathematics, 1st edition. North-Holland, Amsterdam, 2005. URL: https://www.elsevier.com/books/submodular-functions-and-optimization/fujishige/978-0-444-52086-9.

[42] Peter Fulla, Hannes Uppman, and Stanislav Živný. The complexity of boolean surjective general-valued CSPs. *ACM Transactions*

on Computation Theory (TOCT), 11(1):4:1–4:31, 2018. URL: `http://doi.acm.org/10.1145/3282429`.

[43] Peter Fulla and Stanislav Živný. A galois connection for weighted (relational) clones of infinite size. *ACM Transactions on Computation Theory (TOCT)*, 8(3), 2016. URL: `https://doi.org/10.1145/2898438`.

[44] Peter Fulla and Stanislav Živný. On planar valued CSPs. *Journal of Computer and System Sciences*, 87:104 – 118, 2017. URL: `https://doi.org/10.1016/j.jcss.2017.03.005`.

[45] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990. URL: `https://dl.acm.org/doi/book/10.5555/574848`.

[46] David Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968. URL: `https://msp.org/pjm/1968/27-1/pjm-v27-n1-p08-p.pdf`.

[47] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988. URL: `http://doi.acm.org/10.1145/48014.61051`.

[48] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Tractable optimization problems through hypergraph-based structural restrictions. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 16–30, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. URL: `https://doi.org/10.1007/978-3-642-02930-1_2`.

[49] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1:1–1:24, 2007. URL: `http://doi.acm.org/10.1145/1206035.1206036`.

[50] Martin Grötschel, Laszlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin Heidelberg, 1993. URL: `https://doi.org/10.1007/978-3-642-97881-4`.

[51] Gregory Gutin, Pavol Hell, Arash Rafiey, and Anders Yeo. A dichotomy for minimum cost graph homomorphisms. *European Journal of Combinatorics*, 29(4):900–911, 2008. URL: `http://dx.doi.org/10.1016/j.ejc.2007.11.012`.

[52] Thomas L. Heath. *The Works of Archimedes: Edited in Modern Notation with Introductory Chapters*. Cambridge Library Collection - Mathematics. Cambridge University Press, 2009. URL: `https://doi.org/10.1017/CBO9780511695124`.

[53] Pavol. Hell and Arash. Rafiey. The dichotomy of minimum cost homomorphism problems for digraphs. *SIAM Journal on Discrete Mathematics*, 26(4):1597–1608, 2012. URL: `https://doi.org/10.1137/100783856`.

[54] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, USA, 1997. URL: `https://dl.acm.org/doi/book/10.5555/262326`.

[55] Rostislav Horcík, Tommaso Moraschini, and Amanda Vidal. An algebraic approach to valued constraint satisfaction. In *Proceedings of the 26th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 82, pages 42:1–42:20, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/7676`.

[56] Satoru Iwata and James B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the 20th ACM-SIAM Annual Symposium on Discrete Algorithms (SODA)*, page 1230–1237, 2009. URL: `https://epubs.siam.org/doi/abs/10.1137/1.9781611973068.133`.

[57] Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bullettin of the European Association for Theoretical Computer Science (EATCS)*, 113:21–55, 2014. Errata can be found at http://community.dur.ac.uk/andrei.krokhin/papers/BEATCScolumn errata.txt. URL: `http://eatcs.org/beatcs/index.php/beatcs/article/view/266/249`.

[58] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906. URL: `https://doi.org/10.1007/BF02418571`.

[59] Peter Jonsson and Christer Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102(1):143–155, 1998. URL: `https://doi.org/10.1016/S0004-3702(98)00031-9`.

[60] Peter Jonsson, Fredrik Kuivinen, and Johan Thapper. Min CSP on four elements: Moving beyond submodularity. In *Proceedings of the*

17th International Conference on Principles and Practice of Constraint Programming (CP), pages 438–453, 2011. URL: 10.1007/978-3-642-23786-7_34.

[61] Peter Jonsson and Johan Thapper. Constraint satisfaction and semi-linear expansions of addition over the rationals and the reals. *Journal of Computer and System Sciences*, 82(5):912–928, 2016. URL: https://doi.org/10.1016/j.jcss.2016.03.002.

[62] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979. URL: https://doi.org/10.1137/0208040.

[63] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. URL: http://dx.doi.org/10.1007/BF02579150.

[64] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103, Boston, MA, 1972. Springer US. URL: https://doi.org/10.1007/978-1-4684-2001-2_9.

[65] Leonid Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1097, 1979. URL: https://doi.org/10.1016/0041-5553(80)90061-0.

[66] Leonid Khachiyan and Lorant Porkolab. Integer optimization on convex semialgebraic sets. *Discrete & Computational Geometry*, 23:207–224, 2000. URL: https://doi.org/10.1007/PL00009496.

[67] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001. URL: http://dx.doi.org/10.1137/S0097539799349948.

[68] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolinek. The complexity of general-valued CSPs. In *Proceedings of the 56th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1246–1258, 2015. URL: 10.1109/FOCS.2015.80.

[69] 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. URL: `https://doi.org/10.1137/130945648`.

[70] Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *Journal of ACM*, 60(2):10:1–10:38, 2013. URL: `http://doi.acm.org/10.1145/2450142.2450146`.

[71] Michael Kompatscher and Trung Van Pham. A complexity dichotomy for poset constraint satisfaction. In *Proceedings of the 34th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 66, pages 47:1–47:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/6985`.

[72] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983. Special Issue of the 9th International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982. URL: `http://www.sciencedirect.com/science/article/pii/0304397582901256`.

[73] 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), Part I*, pages 846–858, 2015. URL: `https://doi.org/10.1007/978-3-662-47672-7_69`.

[74] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4), 2011. URL: `https://doi.org/10.1145/1989734.1989736`.

[75] Andrei Krokhin and Stanislav Živnỳ. The complexity of valued CSPs. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7, pages 233–266. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/6966`.

[76] Gábor Kun, Ryan M. O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference ((ITCS)*, page 484–495, New York, NY, USA, 2012. ACM. URL: `https://doi.org/10.1145/2090236.2090274`.

[77] Gábor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. In *Proceedings of the 41st ACM Annual Symposium on Theory of Computing (STOC)*, pages 725–734, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org/10.1145/1536414.1536512.

[78] Richard E. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22(1):155–171, 1975. URL: https://doi.org/10.1145/321864.321877.

[79] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983. URL: http://www.jstor.org/stable/3689168.

[80] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. URL: https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[81] Dániel Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *Journal of ACM*, 60(6):42:1–42:51, 2013. URL: http://doi.acm.org/10.1145/2535926.

[82] Gregor Matl and Stanislav Živný. Beyond boolean surjective VCSPs. In Rolf Niedermeier and Christophe Paul, editors, *Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126, pages 52:1–52:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10291.

[83] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31(1):114–127, 1984. URL: http://doi.acm.org/10.1145/2422.322418.

[84] Theodore Samuel Motzkin. *Beiträge zur Theorie der Linearen Ungleichungen.* PhD thesis, University of Basel, Azriel, Jerusalem, 1936. English Translation: *Contributions to the theory of linear inequalities*, RAND Corporation Translation 22, The RAND Corporation, Santa Monica, California, 1952. Reprinted in: *Theodore S. Motzkin: Selected Papers* (D. Cantor, B. Gordon, B. Rothschild, eds.), Birkhäuser, Boston, Massachussetts, 1983, pp. 1-80. URL: https://www.springer.com/gb/book/9780817630874.

[85] Constantin Niculescu and Lars-Erik Persson. *Convex functions and their applications*. CMS Books in Mathematics. Springer, 2018.

[86] Cédric Pralet and Gérard Verfaillie. Time-dependent simple temporal networks: Properties and algorithms. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, page 536–539. AAAI Press, 2014. URL: https://www.aaai.org/ocs/index.php/ICAPS/ICAPS14/paper/view/8667.

[87] Liqun Qi. Directed submodularity, ditroids and directed submodular flows. *Mathematical Programming*, 42(1-3):579–599, 1988. URL: https://doi.org/10.1007/BF01589420.

[88] R. Tyrrell Rockafellar and Roger J. B. Wets. *Variational Analysis*, volume 317. Springer-Verlag, Berlin, 1998. URL: https://sites.math.washington.edu/~rtr/papers/rtr169-VarAnalysis-RockWets.pdf.

[89] Kees Roos. Linear optimization: Theorems of the alternative. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*, pages 1878–1881. Springer US, Boston, MA, 2009. URL: https://doi.org/10.1007/978-0-387-74759-0_334.

[90] Hemut H. Schaefer and Manfred P. Wolf. *Topological Vector Spaces*, volume 248 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, NY, second edition edition, 1999. URL: https://doi.org/10.1007/978-1-4612-1468-7.

[91] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., USA, 1986. URL: https://pdfs.semanticscholar.org/3ce2/d233cee585ecff73729836918ba87195c18f.pdf.

[92] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346 – 355, 2000. URL: https://doi.org/10.1006/jctb.2000.1989.

[93] Przemyslaw Slusarczyk and Remigiusz Baran. Piecewise-linear subband coding scheme for fast image decomposition. *Multimedia Tools and Applications*, 75(17):10649–10666, 2016. URL: https://doi.org/10.1007/s11042-014-2173-1.

[94] Charles Swartz. A general Farkas lemma. *Journal of Optimization Theory and Applications*, 46(2):237–244, 1985. URL: https://doi.org/10.1007/BF00938427.

[95] 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)*, volume 5, pages 657–668, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2010/2493`.

[96] Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 669–678, Washington, DC, USA, 2012. IEEE Computer Society. URL: `https://doi.org/10.1109/FOCS.2012.25`.

[97] Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. *Journal of the ACM*, 63(4):37:1–37:33, 2016. URL: `http://doi.acm.org/10.1145/2974019`.

[98] Johan Thapper and Stanislav Živný. The power of Sherali–Adams relaxations for general-valued CSPs. *SIAM Journal on Computing*, 46(4):1241–1279, 2017. URL: `https://doi.org/10.1137/16M1079245`.

[99] Lou Van Den Dries. *Tame topology and o-minimal structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998. URL: `http://dx.doi.org/10.1017/CBO9780511525919`.

[100] Margaret H. Wright. The interior-point revolution in optimization: history, recent developments, and lasting consequences. *Bullettin of the American Mathematical Society*, 42:39–56, 2005. URL: `https://doi.org/10.1090/S0273-0979-04-01040-7`.

[101] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, 2017. URL: `http://ieee-focs.org/FOCS-2017-Papers/3464a331.pdf`.

[102] Stanislav Živnỳ. *The complexity of valued constraint satisfaction problems*. Springer Publishing Company, Incorporated, 2012. URL: `https://dl.acm.org/doi/book/10.5555/2412078`.

List of Figures

List of Algorithms

I herewith declare that I have produced this thesis without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This thesis has not previously been presented in identical or similar form to any other German or foreign examination board.

Caterina Viola
Dresden, 18 February 2020