Submodular Functions and Valued Constraint Satisfaction Problems over Infinite Domains

Manuel Bodirsky[1]
Institut für Algebra, Technische Universität Dresden
Germany
manuel.bodirsky@tu-dresden.de

Marcello Mamino
Dipartimento di Matematica, Università di Pisa
Italy
marcello.mamino@dm.unipi.it

Caterina Viola[2]
Institut für Algebra, Technische Universität Dresden
Germany
caterina.viola@tu-dresden.de

Abstract

Valued constraint satisfaction problems (VCSPs) are a large class of combinatorial optimisation problems. It is desirable to classify the computational complexity of VCSPs depending on a fixed set of allowed cost functions in the input. Recently, the computational complexity of all VCSPs for finite sets of cost functions over finite domains has been classified in this sense. Many natural optimisation problems, however, cannot be formulated as VCSPs over a finite domain. We initiate the systematic investigation of infinite-domain VCSPs by studying the complexity of VCSPs for piecewise linear homogeneous cost functions. We remark that in this paper the infinite domain will always be the set of rational numbers. We show that such VCSPs can be solved in polynomial time when the cost functions are additionally submodular, and that this is indeed a maximally tractable class: adding any cost function that is not submodular leads to an NP-hard VCSP.

2012 ACM Subject Classification Mathematics of computing → Mathematical optimization, Theory of computation → Complexity theory and logic

Keywords and phrases Valued constraint satisfaction problems, Piecewise linear functions, Submodular functions, Semilinear, Constraint satisfaction, Optimisation, Model Theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.12

1 Introduction

In a *valued constraint satisfaction problem (VCSP)* we are given 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 less than u. By restricting the set of possible cost functions in the input, a great variety of computational optimisation

[1] The first and second author have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681988, CSP-Infinity).
The first author has also received funding from the DFG (Project number 622397)
[2] This author is supported by DFG Graduiertenkolleg 1763 (QuantLA).

problems can be modelled as a valued constraint satisfaction problem. By allowing the cost functions to evaluate to $+\infty$, we can even model 'crisp' constraints, given by relations that have to be satisfied by the variable assignments. Hence the class of (classical) constraint satisfaction problems (CSPs) is a subclass of the class of all VCSPs.

If the domain is finite, the computational complexity of the VCSP has recently been classified for all sets of cost functions, assuming the Feder-Vardi conjecture for classical CSPs [14, 13, 15]. Even more recently, two solutions to the Feder-Vardi conjecture have been announced [18, 6]. These fascinating achievements settle the complexity of the VCSP over finite domains.

Several outstanding combinatorial optimisation problems cannot be formulated as VCSPs over a finite domain, but they can be formulated as VCSPs over the domain $\mathbb{Q}$, the set of rational numbers. One example is the famous linear programming problem, where the task is to optimise a linear function subject to linear inequalities. This can be modelled as a VCSP by allowing unary linear cost functions and cost functions of higher arity to express the crisp linear inequalities. Another example is the minimisation problem for sums of piecewise linear convex cost functions (see, e.g., [5]). Both of these problems can be solved in polynomial time, e.g. by the ellipsoid method (see, e.g., [10]).

Despite the great interest in such concrete VCSPs over the rational numbers in the literature, VCSPs over infinite domains have not yet been studied systematically. In order to obtain general results we need to restrict the class of cost functions that we investigate, because without any restriction it is already hopeless to classify the complexity of infinite-domain CSPs (any language over a finite alphabet is polynomial-time Turing equivalent to an infinite domain CSP [2]). One restriction that captures a variety of optimisation problems of theoretical and practical interest is the class of all *piecewise linear homogeneous* cost functions over $\mathbb{Q}$, defined below. We first illustrate by an example the type of cost functions that we want to capture in our framework.

▶ **Example 1.1.** An internet provider charges the clients depending on the amount of data x downloaded and the amount of data y that is uploaded. The cost function of the provider could be the partial function $f\colon \mathbb{Q}^2 \to \mathbb{Q}$ given by

$$f(x,y) := \begin{cases} 3x & \text{if } 0 \leq y < 2x \\ \frac{3}{2}y & \text{if } 0 \leq 2x \leq y \\ \text{undefined} & \text{otherwise.} \end{cases}$$

A partial function $f\colon \mathbb{Q}^n \to \mathbb{Q}$ is called *piecewise linear homogeneous (PLH)* if it is first-order definable over the structure $\mathfrak{L} := (\mathbb{Q}; <, 1, (c\cdot)_{c\in\mathbb{Q}})$; being undefined at $(x_1,\dots,x_n) \in \mathbb{Q}^n$ is interpreted as $f(x_1,\dots, x_n) = +\infty$. The structure $\mathfrak{L}$ has quantifier elimination (see Section 3.2) and hence there are finitely many regions such that f is a homogeneous linear polynomial in each region; this is the motivation for the name *piecewise linear homogeneous*. The cost function from Example 1.1 is PLH.

The cost function in Example 1.1 satisfies an additional important property: it is *submodular* (defined in Section 3.3). Submodular cost functions naturally appear in several scientific fields such as, for example, economics, game theory, machine learning, and computer vision, and play a key role in operational research and combinatorial optimisation (see, e.g., [9]). Submodularity also plays an important role for the computational complexity of VCSPs over finite domains, and guided the research on VCSPs for some time (see, e.g., [7, 12]), even though this might no longer be visible in the final classification obtained in [14, 13, 15].

In this paper we show that VCSPs for submodular PLH cost functions can be solved in polynomial time (Theorem 5.1 in Section 5). To solve this problem, we first describe how to solve the feasibility problem (does there exist a solution satisfying all crisp constraints) and then how to find the optimal solution. The first step follows from a new, more general polynomial-time tractability result, namely for *max-closed* PLH constraints (Section 4). To then solve the optimisation problem for PLH constraints, we introduce a technique to reduce the task to a problem over a finite domain that can be solved by a fully combinatorial polynomial-time algorithm for submodular set-function optimisation by Iwata and Orlin [11]. Moreover, we show that submodularity defines a *maximal tractable class*: adding any cost function that is submodular leads to an NP-hard VCSP (Section 6). Section 7 closes with some problems and challenges.

2 Valued Constraint Satisfaction Problems

A *valued constraint language* Γ *(over D)* (or simply *language*) consists of

- a signature τ consisting of function symbols f, each equipped with an arity $\mathrm{ar}(f)$,
- a 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 Γ be a valued constraint language with a finite signature τ. The *valued constraint satisfaction problem for* Γ, denoted by $\mathrm{VCSP}(\Gamma)$, is the following computational problem.

▶ **Definition 2.1.** 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 $u_I \in \mathbb{Q} \cup \{+\infty\}$.

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

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

is finite, and if so, whether there is one whose cost is smaller or equal to u_I.

A *solution* of an instance of $\mathrm{VCSP}(\Gamma)$ is a tuple $x \in D^{|V_I|}$ such that $x \in \mathrm{dom}(f)$ for all valued constraint f in the instance.
Note that since the signature τ of Γ is finite, it is inessential for the computational complexity of $\mathrm{VCSP}(\Gamma)$ how the function symbols in ϕ_I are represented. The function described by the expression ϕ_I is also called the *objective function*. When $u_I = +\infty$ then this problem is called the *feasibility* problem, which can also be modelled as a (classical) constraint satisfaction problem. The choice of defining the VCSP as a decision problem and not as an optimisation problem is motivated by two major issues that do not occur in the finite domain case: in the infinite domain setting one needs to decide whether the infimum is attained, and to model the case in which the infimum is $-\infty$.

Many well-known optimisation problems can only be formulated when we allow infinite domains D.

▶ **Example 2.2.** Let Γ be the valued constraint language with signature $\tau = \{g_1, g_2, g_3\}$ and the cost functions

- $g_1^\Gamma \colon \mathbb{Q} \to \mathbb{Q}$ defined by $g_1(x) = -x$,
- $g_2^\Gamma \colon \mathbb{Q}^2 \to \mathbb{Q}$ defined by $g_2(x,y) := \min(x,-y)$, and
- $g_3^\Gamma \colon \mathbb{Q}^3 \to \mathbb{Q}$ defined by $g_3(x,y,z) := \max(x,y,z)$.

Two examples of instances of VCSP(Γ) are

$$\begin{aligned} & g_1(x)+g_1(y)+g_1(z)+g_2(x,y) \\ & +g_3(x,y,z)+g_3(x,x,x)+g_3(x,x,x) \qquad (1) \\ \text{and} \quad & g_1(x)+g_1(y)+g_1(z) \\ & +g_3(x,y,z)+g_3(x,x,y)+g_3(y,z,z) \qquad (2) \end{aligned}$$

We can make the cost function described by the expression in (1) arbitrarily small by fixing x to 0 and choosing y and z sufficiently large. On the other hand, the minimum for the cost function in (2) is 0, obtained by setting x, y, z to 0. Note that g_1 and g_3 are convex functions, but g_2 is not, nevertheless, as we will see later, VCSP(Γ) can be solved in polynomial time.

3 Cost functions over the rationals

In this section we describe 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.

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

▶ **Example 3.1.** Let $\mathfrak{S}$ be the structure with domain $\mathbb{Q}$ and the signature $\{+, 1, \leq\}$ 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.

▶ **Example 3.2.** Let $\mathfrak{L}$ be the structure with the (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).

3.1 Quantifier Elimination

Let τ be a signature. We adopt the usual definition of first-order logic.

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 3.3** ([8])**.** *The structure $\mathfrak{S}$ from Example 3.1 has quantifier elimination.*

▶ **Theorem 3.4.** *The structure $\mathfrak{L}$ from Example 3.2 has quantifier elimination.*

Proof. See the appendix. ◀

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

- if it has no variables, then it is equivalent to $\top$ or $\bot$,
- if it 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 it is equivalent to a formula without variables, and otherwise it 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 it has two variables, say x and y, then it 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 is equivalent to a formula with at most one variable, and otherwise it is equivalent to $x \, \sigma \, \frac{d}{c}{\cdot}y$ or to $\frac{d}{c}{\cdot}y \, \sigma \, x$.

3.2 Piecewise Linear Homogeneous 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 with domain A. 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 will be functions from $\mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$. It is sometimes convenient to view a cost function as a partial function over $\mathbb{Q}$. If $t \in \mathbb{Q}^{\operatorname{ar}(f)} \setminus \operatorname{dom}(f)$ we interpret this as $f(t) = +\infty$.

▶ **Definition 3.5.** 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 constraint language Γ is called *piecewise linear* (*piecewise linear homogeneous*) if every cost function in Γ is PL (or PLH, respectively).

Every piecewise linear homogeneous cost function is also piecewise linear, since all functions of the structure $\mathfrak{L}$ are clearly first-order definable in $\mathfrak{S}$. The cost functions in the valued constraint language from Example 2.2 are PLH.

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 cost functions over a finite domain D. Indeed, every VCSP for a valued constraint language over a finite domain is also a VCSP for a language that is PLH. To see this, suppose that $f\colon D^d \to \mathbb{Q} \cup \{+\infty\}$ is such a cost function, identifying D with a subset of $\mathbb{Q}$ in an arbitrary way. Then the function $f'\colon \mathbb{Q}^d \to \mathbb{Q} \cup \{+\infty\}$ defined by $f'(x_1, \ldots, x_n) := f(x_1, \ldots, x_n)$ if $x_1, \ldots, x_n \in D$, and $f'(x_1, \ldots, x_n) = +\infty$ otherwise, is PLH.

3.3 Submodularity

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

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

▶ **Definition 3.6.** 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, $y \in D^n$

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

Note that in particular if $x, y \in \mathrm{dom}(f)$, then $\min(x,y) \in \mathrm{dom}(f)$ and $\max(x,y) \in \mathrm{dom}(f)$. All cost functions in Example 2.2 are submodular.
Other examples of submodular PLH functions are those that can be written as the maximum of two increasing linear homogeneous functions or as the minimum of two linear homogeneous functions with different monotonicity.

4 Tractability of Max-Closed PLH Constraints

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

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

We can associate to Γ the following relational structure Feas(Γ): for every cost function f of arity n from Γ the signature of Feas(Γ) contains a relation symbol R_f of arity n such that $R_f^{\mathrm{Feas}(\Gamma)} = \mathrm{dom}(f)$.

Every polynomial-time algorithm for VCSP(Γ), in particular, has to solve CSP(Feas(Γ)). In fact, an instance ϕ of VCSP(Γ) can be translated into an instance ψ of CSP(Feas(Γ)) by replacing subexpressions of the form $f(x_1,\dots,x_n)$ in ϕ by $R_f(x_1,\dots,x_n)$ and by replacing $+$ by $\wedge$. 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 4.1.** Let $\mathfrak{A}$ be a structure with relational signature τ and domain A. Then a function $g\colon A^k \to A$ is called a *polymorphism* of $\mathfrak{A}$ if for all $R \in \tau$ we have that $R^{\mathfrak{A}}$ is *preserved* by g, namely $g(x^1,\dots,x^k) \in R^{\mathfrak{A}}$ for all $x^1,\dots,x^k \in R^{\mathfrak{A}}$ (where g is applied component-wise).

▶ **Definition 4.2.** A relation $R \subseteq \mathbb{Q}^n$ is called *piecewise linear homogeneous (PLH)* if it is first-order definable over $\mathfrak{L}$ (see Example 3.2).

In general, a valued constraint language can have infinitely many cost functions. If we consider Γ to be a finite submodular PLH valued constraint language, then Feas(Γ) is a relational structure all of whose relations are

- PLH, and
- preserved by the polymorphisms max and min.

We observed that for the polynomial-time tractability of VCSP(Γ) we need, in particular, that CSP(Feas(Γ)) be tractable. In this section we prove a more general result:

▶ **Theorem 4.3.** *Let $\mathfrak{A}$ be a structure having domain $\mathbb{Q}$ and finite relational signature τ. Assume that for all $R \in \tau$, 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 [4]. In particular, there, the weaker bound NP $\cap$ co-NP has been shown for a larger class, and polynomial tractability only for a smaller class (which does not contain many max-closed PLH relations, for instance $x \geq \max(y, z)$).

We use a technique introduced in [3] which relies on the following concept.

▶ **Definition 4.4.** Let $\mathfrak{A}$ be a structure with a finite relational signature τ. A *sampling algorithm* for $\mathfrak{A}$ takes as input a positive integer d and computes a finite τ-structure $\mathfrak{B}$ such that every finite conjunction of atomic τ-formulas having at most d distinct free variables is satisfiable in $\mathfrak{A}$ if, and only if, it is satisfiable in $\mathfrak{B}$. A sampling algorithm is called *efficient* if its running time is bounded by a polynomial in d.

The definition above is a slight re-formulation of Definition 2.2 in [3], 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.

▶ **Definition 4.5.** A k-ary function $g\colon D^k \to D$ is called *totally symmetric* if $g(x_1, \ldots, x_k) = g(y_1, \ldots, y_k)$ for all $x_1, \ldots, x_k, y_1, \ldots, y_k \in D$ such that $\{x_1 \ldots, x_k\} = \{y_1, \ldots, y_k\}$.

▶ **Theorem 4.6** (Bodirsky-Macpherson-Thapper, [3], 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.*

In this section, we study $\mathrm{CSP}(\mathfrak{A})$, where $\mathfrak{A}$ is a τ-structure satisfying the hypothesis of Theorem 4.3. We give a formal definition of the *numerical data* in $\mathfrak{A}$, we will need it later on. By quantifier elimination (Theorem 3.4), we can write each of the finitely many relations $R^{\mathfrak{A}}$ for $R \in \tau$ as a quantifier-free τ_0-formula ϕ_R. We can assume (as in the proof of Theorem 3.4) that all formulas ϕ_R are positive (namely contain no negations). From now on, we will fix one such representation. Let $\mathrm{At}(\phi_R)$ denote the set of atomic subformulas of ϕ_R. Each atomic τ_0-formula is of the form $t_1 \stackrel{<}{=} t_2$, where t_1 and t_2 are terms. We call the atomic formula non-trivial if it is not equivalent to $\perp$ or $\top$, from now on we make the following assumptions on the atomic formulas (cf. again the proof of Theorem 3.4)

- that atomic formulas except $\top, \perp$ are non-trivial
- that the functions $k\cdot$ are never composed, because $k \cdot h \cdot x$ can be replaced by $(kh) \cdot x$
- that, in any atomic formula $k \cdot x_i \stackrel{<}{=} h \cdot x_j$, the constants k and h are not both negative.

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

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

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

We describe now the numeric domain $\mathbb{Q}^\star$ in which our algorithm operates.

▶ **Definition 4.7.** We call $\mathbb{Q}^\star$ the ordered $\mathbb{Q}$-vector space

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

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 component-wise

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

The order is induced by $\mathbb{Q}$ extended with $0<\boldsymbol{\epsilon}\ll 1$, namely the lexicographical order of the components x and y

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

$\mathbb{Q}$ is clearly embedded in $\mathbb{Q}^\star$ (the embedding is given by the map $k\mapsto k+0\boldsymbol{\epsilon}$).
Any τ_0-formula has an obvious interpretation in any ordered $\mathbb{Q}$-vector space Q extending $\mathbb{Q}$, and, in particular, in $\mathbb{Q}^\star$.

▶ **Proposition 4.8.** *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. It follows from [17, Chapter 1, Remark 7.9] that the first-order theory of ordered $\mathbb{Q}$-vector spaces in the signature $\tau_0\cup\{+,-\}$ is complete. As a consequence the formula $\forall x_1\dots x_d\phi(x_1\dots x_d)\leftrightarrow\psi(x_1\dots x_d)$ holds in $\mathbb{Q}$ if and only if it does in Q. ◀

The proposition gives us a natural extension $\mathfrak{A}^\star$ of $\mathfrak{A}$ to the domain $\mathbb{Q}^\star$. Namely 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). Similarly, we will see that, as long as satisfiability is concerned, there is no difference between $\mathfrak{A}$ and $\mathfrak{A}^\star$.

▶ **Corollary 4.9.** *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. From Proposition 4.8 observing that ϕ (resp. $\phi^\star$) is unsatisfiable if and only if it is equivalent to $\perp$. ◀

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

▶ **Theorem 4.10.** *There is an efficient sampling algorithm for $\mathfrak{A}^\star$.*

Assuming, for a moment, Theorem 4.10, it is easy to prove Theorem 4.3.

Proof of Theorem 4.3. By Proposition 4.7, for all $k\geq 1$ the function

$$(x_1,\dots,x_k)\mapsto\max(x_1,\dots,x_k)$$

is a k-ary totally symmetric polymorphism of $\mathrm{CSP}(\mathfrak{A}^\star)$. Therefore, $\mathrm{CSP}(\mathfrak{A}^\star)$ is in P by Theorem 4.10 and Theorem 4.6. Finally, by Corollary 4.9, $\mathrm{CSP}(\mathfrak{A}^\star)$ and $\mathrm{CSP}(\mathfrak{A})$ are equivalent. ◀

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 $t_1=t_2$.

▶ **Lemma 4.11.** *Let Φ be a finite set of atomic τ_0-formulas having free variables in $\{x_1 \dots x_d\}$. Assume that $\bar{\Phi} := \bigcup_{\phi \in \Phi} \bar{\phi}$ has a simultaneous solution $(x_1 \dots x_d) \in \mathbb{Q}^{>0}$ in positive numbers. Then $\bar{\Phi}$ has a solution taking values in the set $C_{\Phi,d} \subset \mathbb{Q}$ defined as follows*

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

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

Proof. See the appendix. ◀

▶ **Lemma 4.12.** *Let Φ be a finite set of atomic τ_0-formulas having free variables in $\{x_1 \dots x_d\}$. Assume that the formulas in Φ are simultaneously satisfiable in $\mathbb{Q}$. Then they are simultaneously satisfiable in $D_{\Phi,d} := -C^{\star}_{\Phi,d} \cup \{0\} \cup C^{\star}_{\Phi,d}$ where*

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

$C_{\Phi,d}$ is defined as in Lemma 4.11, and $-C^{\star}_{\Phi,d}$ denotes the set $\{-x \mid x \in C^{\star}_{\Phi,d}\}$.

Proof. See the appendix. ◀

Proof of Theorem 4.10. 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 $\mathrm{At}(\tau) := \bigcup_{R \in \tau} \mathrm{At}(\phi_R)$, namely 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 has size polynomial 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 4.12. In fact, consider a set Ψ of atomic τ-formulas having free variables $x_1 \dots x_d$. Assume that Ψ is satisfied in $\mathfrak{A}^{\star}$ by one assignment $x_i = a_i$ for $i \in \{1 \dots d\}$. For each $\phi_R \in \Psi$ let $\Phi_R \subset \mathrm{At}(\phi_R)$ be the set of atomic subformulas of ϕ_R which are satisfied by our assignment a_i. Clearly the atomic τ_0-formulas $\Phi := \bigcup_{\phi_R \in \Psi} \Phi_R$ are simultaneously satisfiable. Remembering that the formulas ϕ_R have no negations by construction, it is obvious that any simultaneous solution of Φ must also satisfy Ψ. By Lemma 4.12, 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. ◀

5 Tractability of Submodular PLH Valued Constraints

Here we extend the method developed in Section 4 to the treatment of VCSPs. To better highlight the parallel with Section 4, so that the reader already familiar with it may quickly get an intuition of the arguments here, we will use identical notations to represent corresponding objects. This choice has the drawback that some symbols, notably $\mathbb{Q}^{\star}$, need to be re-defined (the new $\mathbb{Q}^{\star}$, for instance, will contain the old one). In this section, we will sometimes skip details that can be borrowed unchanged from Section 4.

Our goal is to prove the following result

▶ **Theorem 5.1.** *Let Γ be a PLH valued finite constraint language. Assume that all cost functions in Γ are submodular. Then* $\mathrm{VCSP}(\Gamma)$ *is polynomial-time solvable.*

Let us begin with the new definition of $\mathbb{Q}^{\star}$.

▶ **Definition 5.2.** We let $\mathbb{Q}^\star$ denote the ring $\mathbb{Q}((\boldsymbol{\epsilon}))$ of formal Laurent power series in the indeterminate $\boldsymbol{\epsilon}$. Namely $\mathbb{Q}^\star$ 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 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 $0 < \boldsymbol{\epsilon} \ll 1$, namely

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

It is well known that $\mathbb{Q}^\star$ is an ordered field, namely 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 $m \leq n$

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

▶ **Definition 5.3.** We define a new structure $\mathfrak{L}^\star$, that is both an extension and an expansion of $\mathfrak{L}$ (see Example 3.2), namely it has $\mathbb{Q}^\star$ as domain and $\tau_1 := \tau_0 \cup \{k\}_{k \in \mathbb{Q}^\star_{-1,1}}$ as signature, where 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 (zero-ary functions).

Notice that, for technical reasons, we allow only constants in $\mathbb{Q}^\star_{-1,1}$. During the rest of this section, τ_1-formulas will be interpreted in the structure $\mathfrak{L}^\star$. We make on τ_1-formulas the same assumptions of Section 4 (that atomic subformulas are non-trivial and not negated), also $H(\Phi)$ and $K(\Phi)$ where Φ is a set of atomic τ_1-formulas are defined similarly to Section 4. Observe that the reduct of $\mathfrak{L}^\star$ obtained by restricting the language to τ_0 is *elementarily equivalent* to $\mathfrak{L}$, namely it satisfies the same first-order sentences.

The following lemmas 5.4, 5.5, and 5.6 are analogues of Lemma 4.11 and Lemma 4.12.

▶ **Lemma 5.4.** *Let Φ be a finite set of atomic τ_1-formulas having free variables in $\{x_1 \dots x_d\}$. Call $\bar{\Phi}$ the set $\bar{\phi} \mid \phi \in \Phi$. Suppose that there is $0 < r \in \mathbb{Q}^\star$ such that all satisfying assignments of $\bar{\Phi}$ in the domain $\mathbb{Q}^\star$ also satisfy $0 < x_i \leq r$ for all i. Let u, $\alpha_1 \dots \alpha_d$ be elements of $\mathbb{Q}^\star$. Assume that the formulas in Φ are simultaneously satisfiable by a point $(x_1 \dots x_d) \in \mathbb{Q}^\star$ such that $\sum_i \alpha_i x_i < u$. Let us define the set*

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

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

Proof. See the appendix. ◀

▶ **Lemma 5.5.** *Let Φ be a finite set of atomic τ_1-formulas having free variables in $\{x_1 \dots x_d\}$. Suppose that there are $0 < l < r \in \mathbb{Q}^\star$ such that all satisfying assignments of Φ in the domain $\mathbb{Q}^\star$ also satisfy $l < x_i < r$ for all i. Let $\alpha_1 \dots \alpha_d$ be rational numbers and $u \in \mathbb{Q}^\star_{-1,1}$. Assume that the formulas in Φ are simultaneously satisfiable by a point $(x_1 \dots x_d) \in \mathbb{Q}^\star$ such that $\sum_i \alpha_i x_i \le u$. Then the same formulas are simultaneously satisfiable by a point $(x'_1 \dots x'_d) \in (C^\star_{\Phi,d})^d \subseteq (\mathbb{Q}^\star)^d$ such that $\sum_i \alpha_i x'_i \le u$ where*

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

Proof. See the appendix. ◀

▶ **Lemma 5.6.** *Let Φ be a finite set of atomic τ_0-formulas having free variables in $\{x_1 \dots x_d\}$. Let u, $\alpha_1 \dots \alpha_d$ be rational numbers. Then the following are equivalent*

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

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

Proof. The implication *2* → *1* is immediate observing that the conditions Φ and $\sum_i \alpha_i x_i \le u$ are first-order definable in $\mathfrak{S}$. In fact, any assignment with values in $D_{\Phi,d}$ satisfying the conditions is, in particular, an assignment in $\mathbb{Q}^\star$, and, by 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 $i \in \{1 \dots d\}$. We pre-process the formulas in Φ producing a new set of atomic formulas Φ' as follows. We replace all variables 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 $-y_i$ according 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 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 positive rational numbers with $\sum_i \alpha'_i y_i \le u$. Observing that a positive rational x always satisfies $\boldsymbol{\epsilon} < x < \boldsymbol{\epsilon}^{-1}$, we see that Φ' satisfies the hypothesis of Lemma 5.5 with $l = \boldsymbol{\epsilon}$ and $r = \boldsymbol{\epsilon}^{-1}$. Hence the statement. ◀

Two roads diverge now. Clearly the formulas Φ in Lemma 5.6 are going to define a piece of the domain of a piecewise linear homogeneous function, while the coefficients α_i define the function on that piece. We could decide to interpret our PLH functions in the domain $\mathbb{Q}^\star$ or we could decide to substitute a suitably small rational value of $\boldsymbol{\epsilon}$ in the formal expression of $D_{\Phi,d}$ and 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 $\boldsymbol{\epsilon}$). It is not clear which road is the less traveled by. For reasons that will be discussed in Subsection 5.1 we take the one of transferring.

It is obvious that one can extend Definition 2.1 considering VCSPs whose cost functions take values in any totally ordered ring containing $\mathbb{Q}$, and in particular in $\mathbb{Q}^\star$. We will need to establish the basics of such extended VCSPs. More precisely, we will need to prove Corollary 5.12 hereafter, that builds on a *fully combinatorial* algorithm (Theorem 5.11) due to Iwata and Orlin [11].

▶ **Definition 5.7.** Let R be a totally ordered commutative ring with unit. 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 (see [1], Chapters 3-4) solving it by performing only additions and comparisons of elements in R as fundamental operations. We recall that a uniform machine on a totally ordered commutative ring with unity operates on strings of symbols that represent elements of an ordered commutative ring, rather than bits as in classical Turing machines. (Notice that in such a machine there are no machine-constants except 1.)

▶ **Definition 5.8.** A set function is a function ψ defined on the set 2^V, of subsets of a given set V.

▶ **Definition 5.9.** A set function $\psi\colon 2^V \to Q$ with values in a totally ordered Abelian group Q is submodular if for all $U, W \in 2^V$

$$\psi(U) + \psi(W) \geq \psi(U \cap W) + \psi(U \cup W).$$

▶ **Definition 5.10.** 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 [16, Section 6]. 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 of Section 6 in [16] 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 5.11** (Iwata-Orlin [11] + Schrijver [16]). *There exists a fully combinatorial polynomial-time algorithm over $\mathbb{Q}$ that*

- *taking as input a finite set $Q = \{1, \dots, n\}$ and a ring family, $\mathcal{C} \subseteq 2^Q$, represented as in [16, Section 6] (namely as above),*
- *having access to an oracle computing 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 5.12.** *Let R be a totally ordered commutative ring with unit (for instance $\mathbb{Q}^\star$), there exists a fully combinatorial polynomial-time algorithm over R that*

- *taking as input a finite set $Q = \{1, \dots, n\}$ and a ring family, $\mathcal{C} \subseteq 2^Q$, represented as in Theorem 5.11,*
- *having access to an oracle computing 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 5.11 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 5.11.
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 $E_\psi := \{g_1, \ldots, g_m\}$ be a set of free generators of R_ψ. For any tuple $r = (r_1, \ldots, r_m) \in \mathbb{Q}^m$, we define a group homomorphism $h_r : R_\psi \to \mathbb{Q}$, 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 of the form $\pm x_1 \pm x_2 \ldots \pm x_k$, with $k \leq N$, $x_1, x_2, \ldots, x_k \in E_\psi$.

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. To see this, 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 $r_1, \ldots, r_m$. 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 $i \in \{1, \ldots, m\}$, 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 [7, 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 5.13.** *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 $f : Q^d \to R$,*
- *given the representation of $\mathcal{C}_f$ as in Theorem 5.11,*

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 [7]. ◀

Proof of Theorem 5.1. Similarly to the proof of Theorem 4.3, we will use a sampling technique. Namely, given an instance I of VCSP(Γ), we will employ Lemma 5.6 to fix a finite structure Γ_I, of size (and also representation size) polynomial in $|V_I|$, having a subset $\mathbb{Q}^\star_I$ of $\mathbb{Q}^\star_{-1,4}$ as domain, such that the variables V_I of I have an assignment in $\mathbb{Q}$ having cost $\leq u_I$ if and only if they have one in $\mathbb{Q}^\star_I$. Once we have Γ_I, we will conclude by Lemma 5.13.

The 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 4.8 the choice of ϕ_f is immaterial. Remains to define the domain $\mathbb{Q}^\star_I \subset \mathbb{Q}^\star$.

By quantifier-elimination (Theorem 3.4), any piecewise linear homogeneous cost function $f\colon \mathbb{Q}^n \to \mathbb{Q} \cup \{+\infty\}$ can be written as

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

where $t_{f,1},\dots,t_{f,m_f}$ are τ_0-terms, $\chi_{f,1},\dots,\chi_{f,m_f}$ are conjunctions of atomic τ_0-formulas with variables from $\{x_1,\dots,x_n$, and $\chi_{f,1},\dots,\chi_{f,m_f}$ 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 $1 \le i \le m_f$, into the set Φ. Clearly Φ is finite and depends only on the fixed language Γ. Finally, $\mathbb{Q}^\star_I := D_{\Phi,|V_I|}$ as defined in Lemma 5.6.

The size of $\mathbb{Q}^\star_I$ is clearly polynomial by simple inspection of the definition. Its representation has also polynomial size 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 $\le u_I$ we have, a fortiori, an assignment $V_I \to \mathbb{Q}^\star$ of value $\le u_I$, hence, by the usual completeness of the first-order theory of ordered $\mathbb{Q}$-vector spaces, there is an assignment $V_I \to \mathbb{Q}$ with the same property.

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

$$\phi_I = \sum_{i=1}^{m} f_i(x^i_1,\dots,x^i_{\mathrm{ar}(f_i)})$$

(cf. Definition 2.1). 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 5.6, 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 5.13 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 without obstacle). The oracle computing f is straightforward to implement since sums and comparisons in $\mathbb{Q}^\star$ merely reduce to the corresponding component-wise 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 component-wise minimal feasible assignment $\alpha_x\colon V_I \to \mathbb{Q}^\star_I$ that gives to x a value $\ge q$ (which exists observing that the set of feasible assignments is min-closed). This algorithm is easy to construct observing that the feasibility problem is a min-closed CSP. 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 4.3, adding a constraint $x \geq k$ for increasing values of $k \in \mathbb{Q}_I^\star$. ◀

5.1 Why $\mathbb{Q}^\star$?

It might appear that in more than one occasion we chose to work in mathematically overcomplicated structures. For example, the algorithm for Theorem 5.1 merely manipulates points in $\mathbb{Q}^\star_{-1,4}$, which is just $\mathbb{Q}^6$ with the lexicographic order, yet we went to the trouble of introducing the field of formal Laurent power series. More radically, one might observe that assigning a rational value to the formal variable $\boldsymbol{\epsilon}$ small enough, we could have mapped the entire algorithm to $\mathbb{Q}$, thus dispensing with non-Archimedean extensions entirely. As we believe to owe to our reader an explanation for this, we better give three.

First, the idea of limiting our horizon to $\mathbb{Q}^\star_{-1,4} \simeq \mathbb{Q}^6$ might seem a simplification, but, in practice, it makes things more complicated. For example, in several places we used the fact that $\mathbb{Q}^\star$ has a field structure to make proofs more direct and intuitive. Second, going for the most elementary exposition, namely choosing an $\boldsymbol{\epsilon}$ small enough explicitly, would have completely obfuscated any idea in the arguments, which would have been converted in some unsightly bureaucracy of inequalities. Even computationally, mapping everything to $\mathbb{Q}$ is tantamount as converting arrays of small integers into bignums by concatenation, hardly an improvement. Finally, the existence of an efficiently computable rational value of $\boldsymbol{\epsilon}$ that works is not necessary for our method, even though, in this case, a posteriori, such an $\boldsymbol{\epsilon}$ exists.

Our third, and most important, justification, is that we desire to present the approach used in this paper, which is quite generic, as much as the results. To this aim, it is convenient to express the underlying ideas in their natural language. For example, Corollary 5.12 is a completely black-boxed way to transfer combinatorial algorithms between domains that share some algebraic structure. We do not claim great originality in that observation, yet we believe that the method is interesting, and worthy of being presented in the cleanest form that we could devise.

6 Maximal Tractability

A *sublanguage* of a valued constraint language Γ is a valued constraint language that can be obtained from Γ by dropping some of the cost functions.

▶ **Definition 6.1.** Let $\mathcal{V}$ be a class of valued constraint languages over a fixed domain D and let Γ be a language of $\mathcal{V}$. We say that Γ is *maximally tractable within* $\mathcal{V}$ if

- VCSP(Γ') is polynomial time solvable for every finite sublanguage Γ' of Γ; and
- for every valued constraint language Δ in $\mathcal{V}$ properly containing Γ, there exists a finite sublanguage Δ' of Δ such that VCSP(Δ') is NP-hard.

Using [7, Theorem 6.7], it is easy to show the following. (See the appendix for details.)

▶ **Theorem 6.2.** *The valued constraint language consisting of all submodular PLH cost functions is maximally tractable within the class of PLH valued constraint languages.*

7 Conclusion and Outlook

We have presented a polynomial-time algorithm for submodular PLH cost functions over the rationals. In fact, our algorithm not only decides the feasibility problem and whether there exists a solution of cost at most u_I, but can also be adapted to efficiently compute the infimum of the cost of all solutions (which might be $-\infty$), and decides whether the infimum is attained. The modification is straightforward observing that the sample computed does not depend on the threshold u_I.

We also showed that submodular PLH cost functions are *maximally tractable* within the class of PLH cost functions. 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.

Another challenge is to extend our tractability result to the class of all submodular *piecewise linear* VCSPs. We believe that submodular piecewise linear VCSPs are in P, too. But note that already the 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 a polymorphism) does *not* admit an efficient sampling algorithm (it is easy to see that for every $d \in \mathbb{N}$ every d-sample must have exponentially many vertices in d), so a different approach than the approach in this paper is needed.

References

1 Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation*. Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.

2 Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, pages 184–196. Springer Verlag, July 2008.

3 Manuel Bodirsky, Dugald Macpherson, and Johan Thapper. Constraint satisfaction tractability from semi-lattice operations on infinite sets. *Transaction of Computational Logic (ACM-TOCL)*, 14(4):1–30, 2013.

4 Manuel Bodirsky and Marcello Mamino. Tropically convex constraint satisfaction. *Theory of Computing Systems*, pages 1–29, 2017. An extended abstract of the paper appeared under the title “Max-Closed Semilinear Constraints" in the proceedings of CSR'16; preprint available under ArXiv:1506.04184.

5 Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

6 Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330, 2017.

7 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.

8 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.

9 Satoru Fujishige. *Submodular Functions and Optimization*. volume 58 of Annals of Discrete Mathematics. North-Holland, Amsterdam, 2005. 2nd edition.

10 M. Grötschel, L. Lovász, and L. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Heidelberg, 1994. 2nd edition.

11 Satoru Iwata and James B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1230–1237, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=1496770.1496903`.

12 Peter Jonsson, Fredrik Kuivinen, and Johan Thapper. Min CSP on four elements: Moving beyond submodularity. In *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, pages 438–453, 2011.

13 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolinek. The complexity of general-valued csps. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1246–1258, 2015.

14 Vladimir Kolmogorov, Johan Thapper, and Stanislav Zivny. The power of linear programming for general-valued csps. *SIAM J. Comput.*, 44(1):1–36, 2015.

15 Marcin Kozik and Joanna Ochremiak. Algebraic properties of valued constraint satisfaction problem. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 846–858, 2015.

16 Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

17 Lou van den Dries. *Tame topology and o-minimal structures*, volume 248 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998. `doi:10.1017/CBO9780511525919`.

18 Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342, 2017.

A Appendix

A.1 Quantifier Elimination: Proof of Theorem 3.4

To prove Theorem 3.4 it suffices to prove the following lemma.

▶ **Lemma A.1.** *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 seven 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, $\varphi = \bigwedge_i \chi_i \wedge \bigwedge_i \chi'_i \wedge \bigwedge_i \chi''_l$, 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. Therefore the procedure is correct. ◀

Proof (of Theorem 3.4). 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 A.1, 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 A.1, $\exists x.\neg\varphi'$ 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 $Q_1x_1Q_2x_2\cdots Q_nx_n.\varphi'$ 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, $Q_2x_2\cdots Q_nx_n.\varphi'$ is equivalent to a quantifier -free τ_0-formula ψ. Therefore, φ is equivalent to $Q_1x_1.\psi$, that is, a τ_0-formula with one quantifier that is equivalent to a quantifier-free τ_0-formula, again by inductive hypothesis. ◀

A.2 Proof of Lemma 4.11 and Lemma 4.12

Proof of Lemma 4.11. 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 that 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 of $\Psi_1 \cup \Psi_2$ 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 x_i such that there is a constraint of the form $h \cdot x_i = k \cdot 1$, or all constraints are of the form $h \cdot x_i = h' \cdot x_j$. In the first case assign $x_i = \frac{k}{h}$, in the second assign one of the variables x_i arbitrarily to 1, then, in any case, since the diameter of the connected component is $< d$, all variables in it are forced to take values in $C_{\Phi,d}$ by simple propagation of x_i. ◀

Proof of Lemma 4.12. First we fix a solution $x_i = a_i$ for $i = 1 \dots d$ of Φ. In general, some of the values a_i will be positive, some 0, and some 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 x_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 4.11, 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 $-d \leq n_1 \dots n_d \leq d$ be integers such that for all i, j

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

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 $y_i = c_i + n_i c_i \boldsymbol{\epsilon} \in \mathbb{Q}^\star$ satisfies all formulas of Φ^+. To check this, we consider the different cases for atomic formulas

- $k \cdot y_i < h \cdot y_j$: if $kc_i < hc_j$ this 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 $kb_i = hb_j$ and $kc_i = hc_j$, therefore $\frac{b_i}{c_i} = \frac{b_j}{c_j}$, and, as a consequence, also $n_i = n_j$ from which the statement.
- $k \cdot 1 < h \cdot y_j$: similarly to the first case, if $k < hc_j$ this is immediate. Otherwise $k = hc_j$, so k and h are positive, the constraint

  $$k \cdot 1 < hc_j + hn_jc_j\boldsymbol{\epsilon}$$

  is equivalent to $0 < n_j$, in other words $1 < \frac{b_j}{c_j}$, which follows 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 $\frac{b_j}{c_j} = 1$, so $n_j = 0$ and the case follows.
- $k \cdot y_i = h \cdot 1$: as the case above. ◀

A.3 Proof of Lemma 5.4 and Lemma 5.5

Proof of Lemma 5.4. As in the proof of Lemma 4.11 (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

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

and $\Psi_1 \cup \Psi_2 \cup \Psi_3$ is satisfiable by an assignment with $\sum_i \alpha_i x_i < u$. As in the proof of Lemma 4.11 the set of solutions of $\Psi_1 \cup \Psi_2$ satisfying $\sum_i \alpha_i x_i < u$ 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 $\sum_i \alpha_i x_i < u$

taking values in $C_{\Phi,d}$. The proof of Lemma 4.11 shows that there is a solution of $\Psi_1 \cup \Psi_2$ taking values in $C_{\Phi,d}$ without necessarily meeting the requirement that $\sum_i \alpha_i x_i < u$. We will 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 $\sum_i \alpha_i a_i < u$ and $\sum_i \alpha_i b_i \geq u$. We know that the first exists, and we assume the second towards a contradiction. The two assignments must differ, so, without loss of generality $a_1 \neq b_1$. 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 $t \geq 0$

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

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

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

is either $\geq 2r$ or < 0 depending on the sign of $(a-b)$. 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 $0 < x_i \leq r$. ◀

Proof of Lemma 5.5. We consider two cases: either all satisfying assignments satisfy the inequality $\sum_i \alpha_i x_i \geq u$ or there is a satisfying assignment $(x_1 \ldots x_d)$ for Φ such that $\sum_i \alpha_i x_i < u$.

In the first case, we claim that all satisfying assignments, in fact, satisfy $\sum_i \alpha_i x_i = u$. In fact, assume that $x_i = a_i, b_i$ are two satisfying assignments such that $\sum_i \alpha_i a_i = u$ and $v := \sum_i \alpha_i b_i > u$. As in the proof of Lemma 5.4, consider assignments of the form $x_i(t) = (1+t)a_i - tb_i$ for $t \in \mathbb{Q}^\star$. Clearly $\sum_i \alpha_i x_i(t) = u - t(v-u) < u$ for all $t > 0$. As in Lemma 5.4, the new assignment must satisfy all equality constraints in Φ. Each inequality constraint implies a strict inequality on t (remember that Φ only has strict inequalities). Since all of these must be satisfied by $t = 0$, there is an open interval of acceptable values of t around 0, and, in particular, an acceptable $t > 0$. 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 4.12, replacing the formal symbol $\boldsymbol{\epsilon}$ in that proof by $\boldsymbol{\epsilon}^3$. Namely take a satisfying assignment $x_i = b_i$ for Φ, and, by Lemma 5.4, one 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 5.4. 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 4.12. Observe that we have to replace $\boldsymbol{\epsilon}$ in Lemma 4.12 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 5.4 there is an assignment $x_i = c_i \in C_{\Phi,d}$ such that $\sum_i \alpha_i c_i < u$ 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 $\sum_i \alpha_i y_i < u$ we write

$$\sum_i \alpha_i y_i = \sum_i \alpha_i c_i + \boldsymbol{\epsilon}^3 \sum_i \alpha_i n_i c_i < u$$

because the first summand is in $\mathbb{Q}^{\star}_{-1,1}$ and $< u$, therefore the second summand is neglected in the lexicographical order. ◀

A.4 Proof of the maximal tractability

In this appendix we prove Theorem 6.2. We will make use of the following result.

▶ **Theorem A.2** (Cohen-Cooper-Jeavons-Krokhin, [7], Theorem 6.7). *Let D be a finite totally ordered set. Then the valued constraint language consisting of all submodular cost functions over D is maximally tractable within the class of all valued constraint languages over D.*

We show that the class of submodular piecewise linear homogeneous languages is maximally tractable within the class of PLH valued constraint languages.

▶ **Definition A.3.** Given a finite set $D \subset \mathbb{Q}$, we define the partial function $\chi_D \colon \mathbb{Q} \to \mathbb{Q}$ by

$$\chi_D(x) = \begin{cases} 0 & x \in D \\ +\infty & x \in \mathbb{Q} \setminus D. \end{cases}$$

For every finite set $D \subset \mathbb{Q}$, the cost function χ_D is submodular and PLH.

▶ **Definition A.4.** 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 $\hat{f} \colon \mathbb{Q}^n \to \mathbb{Q}$, by

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

Note that the canonical extension of a submodular function over a finite domain is submodular and PLH.

Proof of Theorem 6.2. Polynomial-time tractability of the VCSP for finite sets of submodular PLH cost functions has been shown in Theorem 5.1.

Now suppose that f is a cost function over $\mathbb{Q}$ that is not submodular, i.e., there exists a couple of points, $a := (a_1, \dots, a_n)$, $b := (b_1, \dots, b_n) \in \mathbb{Q}^n$ such that

$$f(a) + f(b) < f(\min(a, b)) + f(\max(a, b)).$$

Let Γ_D be the language of all submodular functions on

$$D := \{a_1, \dots, a_n, b_1, \dots, b_n\} \subset \mathbb{Q}\,.$$

Notice that $f|_D$ is not submodular, for our choice of D. Therefore, by Theorem A.2, there exists a finite language $\Gamma'_D \subset \Gamma_D$ such that $\mathrm{VCSP}(\Gamma'_D \cup \{f|_D\})$ is NP-hard.

We define the finite submodular PHL language Γ' by replacing every cost function g in Γ'_D by its canonical extension $\hat{g}$. Then $\Gamma' \cup \{f, \chi_D\}$, where χ_D is defined as in Definition A.3, has an NP-hard VCSP. Indeed, for every instance I of $\mathrm{VCSP}(\Gamma'_D \cup \{f|_D\})$, we define an instance J of $\mathrm{VCSP}(\Gamma' \cup \{f, \chi_D\})$ in the following way:

- replace every function symbol g in ϕ_I by the symbol for its canonical extension,
- replace the function symbol for $f|_D$ in ϕ_I by f, and
- add to J_ϕ the summand $\chi_D(v)$ for every variable $v \in V_I$.

Because of the terms involving χ_D, the infimum of ϕ_J is smaller than $+\infty$ if, and only if, it is attained in a point having coordinates in D. Therefore, the infimum of ϕ_J coincides with the infimum of ϕ_I. Since J is computable in polynomial-time from I, the NP-hardness of VCSP$(\Gamma' \cup \{f, \chi_D\})$ follows from the NP-hardness of VCSP$(\Gamma' \cup \{f|_D\})$. ◀