Algebraic Approach to Approximation*

Libor Barto
Faculty of Mathematics and Physics
Charles University
Czechia
libor.barto@gmail.com

Silvia Butti
Department of Computer Science
University of Oxford
UK
silvia.butti@cs.ox.ac.uk

Alexandr Kazda
Faculty of Mathematics and Physics
Charles University
Czechia
alex.kazda@gmail.com

Caterina Viola
Department of Mathematics and Computer Science
University of Catania
Italy
caterina.viola@unict.it

Stanislav Živný
Department of Computer Science
University of Oxford
UK
standa.zivny@cs.ox.ac.uk

ABSTRACT

Following the success of the so-called algebraic approach to the study of decision constraint satisfaction problems (CSPs), exact optimization of valued CSPs, and most recently promise CSPs, we propose an algebraic framework for valued promise CSPs.

To every valued promise CSP we associate an algebraic object, its so-called valued minion. Our main result shows that the existence of a homomorphism between the associated valued minions implies a polynomial-time reduction between the original CSPs. We also show that this general reduction theorem includes important inapproximability results, for instance, the inapproximability of almost solvable systems of linear equations beyond the random assignment threshold.

CCS CONCEPTS

• **Theory of computation → Problems, reductions and completeness**; **Constraint and logic programming**.

KEYWORDS

approximation, constraint satisfaction, algebraic approach, polymorphisms

*Libor Barto and Caterina Viola were funded by the European Union (ERC, CoCoSym, 771005). Libor Barto was also funded by (ERC, POCOCOP, 101071674). Caterina Viola was also funded by ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, co-founded by European Union - NextGenerationEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for the m. This work was supported by UKRI EP/X024431/1. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper. The full version of the paper is available at arXiv:2401.15186 [6].

LICS '24, July 8–11, 2024, Tallinn, Estonia

ACM ISBN 979-8-4007-0660-8/24/07
https://doi.org/10.1145/3661814.3662076

ACM Reference Format:
Libor Barto, Silvia Butti, Alexandr Kazda, Caterina Viola, and Stanislav Živný. 2024. Algebraic Approach to Approximation. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3661814.3662076

1 INTRODUCTION

What mathematical structure captures efficient computation? Answering this question is the holy grail of theoretical computer science. Constraint Satisfaction Problems, or CSPs for short, provide an excellent framework to attempt this ambitious research endeavour. On the one hand, CSPs are general enough to include many fundamental problems of interest and allow for general patterns to occur, which rarely happens when studying concrete problems in isolation. On the other hand, CSPs are structured enough so that interesting and nontrivial results can be established. Indeed, while CSPs do not capture[1] all computational problems, both algorithmic and hardness techniques developed in the context of constraint satisfaction are often used beyond the realm of CSPs.

Putting the area of constraint solving aside, there are two main strands of research on the computational complexity of CSPs. The first strand studies decision CSPs on finite [32] and infinite [16] domains, exact solvability of optimization CSPs (known as valued CSPs [29]), and most recently qualitative approximation of decision CSPs (known as promise CSPs, or PCSPs for short [3, 4, 23]). The highlights of this strand include, firstly, complexity classifications of CSPs, e.g., dichotomies for robust solvability of CSPs [8], valued CSPs [43, 45, 51], infinite-domain CSPs [18–20], promise CSPs [23, 33], and in particular a dichotomy for all finite-domain CSPs [26, 54], which gave a positive answer to the long-standing Feder-Vardi conjecture [32]. Secondly, characterizations of the power of various algorithms, e.g., [7, 12, 21, 28, 37, 44, 52].

The second strand studies quantitative approximation of CSPs. The highlights include, e.g., the PCP theorem [1, 2, 31], Håstad's optimal inapproximability results [35], Raghavendra's result that a semidefinite programming relaxation is optimal for all CSPs [48] under Khot's Unique Games Conjecture [40], inapproximability

[1] Up to polynomial-time Turing reductions, CSPs on infinite domains *do* capture all computational problems [17], cf. also [34].

of certain valued CSPs (under the UGC) [42], optimal inapproximability of certain MaxCSPs [27], or the recent line of work on inapproximability of perfectly satisfiable MaxCSPs [13–15].

While the two strands use different mathematical tools (algebraic vs. analytical), there are some common features, e.g., dictatorship testing plays an important role in both PCSPs and approximability. Our paper confirms that this is not a coincidence.

With the general goal to better understand what makes computational problems easy or hard, we aim to provide uniform descriptions of algorithms, tractability boundaries, and reductions. For CSPs studied in the first strand described above, all of these can be described uniformly by means of *polymorphisms*, which can be, informally, thought of as multivariate symmetries of solution spaces of CSPs (although the precise definitions and conditions depend on the type of considered CSPs). Interestingly, it was observed a posteriori in [24] that Raghavendra's result from [48], which falls in the second strand, can be phrased in terms of (a certain type of) polymorphisms, although it remained unclear whether and how polymorphisms determine complexity without the Unique Games Conjecture. The notion of polymorphisms coming from [24] is close to ours, cf. [6].

In the present paper, we introduce and initiate the study of the very general framework of valued PCSPs. It includes, as special cases, (non-valued) PCSPs (and thus also CSPs), valued CSPs, approximation of CSPs (both constant factor and gap variants), Gap Label Cover, and Unique Games. The only previous works on valued PCSPs are the algorithmic results in [5, 53] and the unpublished manuscript of Kazda developing an algebraic theory for constant factor approximation of valued PCSPs [39], cf. [6].

As our main result, we define a notion of polymorphisms for valued PCSPs and show that it leads to polynomial-time reductions. Thus, we take the first step in providing a uniform description of reductions among the very large class of computational problems captured by valued PCSPs.

In order to help the reader and to explain clearly the differences between the previous work on PCSPs and our more general setting of valued PCSPs, we recap in Section 2 the basics of the algebraic theory for non-valued PCSPs. This should be useful in particular as we work in the multi-sorted setting[2] (cf. the discussion at the end of Section 2.3) and with slightly more general notions than is common in the literature. In Section 3 we define valued PCSPs and *plurimorphisms*, the new notion of multivariate symmetry. Then, in Section 4, we prove our first main result, namely that a homomorphism between sets of plurimorphisms of two valued PCSPs implies a polynomial-time reduction between these valued PCSPs. The core of this reduction theorem is that every valued PCSP is polynomial-time equivalent to a valued version of the Minor Condition problem that played a key role in the algebraic approach to non-valued PCSPs [4]. This allows us to circumvent routes via definability as was done in the original algebraic approach to decision non-promise CSPs and valued CSPs [25, 29]. Finally, in Section 5 we give examples of valued homomorphisms, most notably our second main result, which is a valued homomorphism that captures, e.g., Håstad's result on inapproximability of almost-satisfiable systems of linear equations [35].

[2]Different variables can have different domains.

2 PROMISE CSP

In this section we review the basics of the theory of *crisp* (non-valued) Promise CSPs, in a way that mimics our theory-building in the more general valued setting in the next section. The definitions and theorems, whose proofs are provided in [6], essentially follow parts of [4] with some adjustments.[3]

2.1 Preliminaries

For two sets A and Z, the set A^Z is the set of all functions from Z to A. Sometimes it is more natural to regard elements $f \in A^Z$ as tuples of elements of A indexed by elements of Z (we also say a Z-tuple of elements of A). In such a case we use boldface and write, e.g., $\mathbf{a} \in A^Z$. However, there are situations when both viewpoints (as a function or as a tuple) are used within one formula or a proof.

For a finite set Z, a Z-ary *relation on* A is a subset ϕ of A^Z. For $\mathbf{a} \in A^Z$ and a Z-ary relation ϕ, we usually write $\phi(\mathbf{a})$ instead of $\mathbf{a} \in \phi$. In order to succinctly write down a Z-tuple, one can fix a linear order on Z and write a tuple as a sequence of length $|Z|$, e.g., $\mathbf{a} = (\mathbf{a}(z_1), \mathbf{a}(z_2), \dots, \mathbf{a}(z_n))$, where $z_1, \dots, z_n$ is the enumeration of Z in increasing order.

Functions are composed from right to left: If $f : A \to B$ and $g : B \to C$ then the composed function $A \to C$ is denoted by $g \circ f$ or just gf. Note that for a Z-tuple $\mathbf{a} \in A^Z$ and a function $f : A \to B$, the Z-tuple $f \circ \mathbf{a} \in B^Z$ is the tuple obtained by applying f to $\mathbf{a}$ component-wise.

The class of finite sets is denoted by FinSet. We denote by $[n]$ the set $\{1, 2, \dots, n\}$.

2.2 Relational structures

For a set τ, a τ-sorted set A is a collection of sets, one set A_t for each sort $t \in \tau$, and a τ-sorted function between two τ-sorted sets A and B is a collection of functions $A_t \to B_t$, $t \in \tau$. We define these notions formally as follows.

Definition 2.1 (Multi-sorted setting). Let τ be a set (of *sorts symbols*). A τ*-sorted* set is a set A together with a mapping sort : $A \to \tau$. For $t \in \tau$, the t-sort of A is $A_t = \{a \in A \mid \mathrm{sort}(a) = t\}$.

For two τ-sorted sets A, B, a τ*-sorted function* from A to B is a function $f : A \to B$ that preserves sorts, i.e., $\mathrm{sort}(f(a)) = \mathrm{sort}(a)$ for every $a \in A$. The set of τ-sorted mappings from A to B is denoted by B^A, as above. Note that this set is not τ-sorted.

For a τ-sorted set A and a set N, their product is the τ-sorted set $A \times N$ with $\mathrm{sort}(a, n) = a$ for every $a \in A$, $n \in N$. This time, by A^N we denote the τ-sorted set of those mappings $f : N \to A$ such that $f(N) \subseteq A_t$ for some t, with $\mathrm{sort}(f) = t$.

For two τ-sorted sets A and Z, we regard the elements of A^Z also as Z-tuples and subsets of A^Z as Z-ary relations. Similarly as before, Z-tuples can be presented as sequences of length $|Z|$ by fixing a linear order on Z.

Definition 2.2 (Multi-sorted signature). A *multi-sorted signature* Σ is a triple $\Sigma = (\sigma, \tau, \mathrm{ar})$ where σ is a set of *relational symbols*, τ is

[3]The adjustment is mostly in that the arity of a relation or a function can be any finite set N. It is more standard in the CSP literature to only use n-ary relations and functions for a non-negative integer n. We do not see any advantages for the latter (at least in our context) and a lot of disadvantages, such as the need to often choose enumerations, awkward expressions, unnecessary notions, unnecessary abusing notation, void calculations, etc.

a set of *sort symbols*, and ar assigns to each symbol $\phi \in \sigma$ a finite τ-sorted set $\mathrm{ar}(\phi)$, called the *arity* of ϕ.

Such a signature is *finite* if σ is finite.

Symbols Σ, σ, τ, ar are reserved for the objects above and we often keep the notation implicit. A signature is implicitly **multi-sorted and finite**. We will also implicitly assume that τ is finite.

Definition 2.3 (Relational structure). Let Σ be a signature. A *structure in signature* Σ, or Σ*-structure*, $\mathbf{A}$ consists of a τ-sorted set A called the *domain* and an $\mathrm{ar}(\phi)$-ary relation $\phi^{\mathbf{A}}$ on A (i.e., $\phi^{\mathbf{A}} \subseteq A^{\mathrm{ar}(\phi)}$) called the *interpretation* of ϕ in $\mathbf{A}$ for each $\phi \in \sigma$. Such a structure $\mathbf{A}$ is said to be *finite* if A is finite.

We shall use the same letter, but different fonts, to refer to a structure $\mathbf{A}$ (bold) and its domain A (uppercase). A structure is implicitly **finite**.

2.3 Promise CSP

The Promise CSP over a pair of structures $(\mathbf{A}, \mathbf{B})$ can be defined as the problem of deciding whether a conjunctive formula is true in $\mathbf{A}$ or not even true in $\mathbf{B}$.[4] This problem only makes sense if each conjunctive formula true in $\mathbf{A}$ is also true in $\mathbf{B}$. Formal definitions are as follows.

Definition 2.4 (Conjunctive formula). Let Σ be a signature and X a τ-sorted set. A *conjunctive formula* over X in the signature Σ (or *conjunctive* Σ*-formula*) is a formal expression Φ of the form

$$\Phi = \bigwedge_{i \in I} \phi_i(\mathbf{x}_i),$$

where I is a finite nonempty set, and $\phi_i \in \sigma$, $\mathbf{x}_i \in X^{\mathrm{ar}(\phi_i)}$ for all $i \in I$. The conjuncts are called *constraints*.

Given additionally a Σ-structure $\mathbf{A}$, the *interpretation of* Φ *in* $\mathbf{A}$, or *the* X*-ary relation defined in* $\mathbf{A}$ *by* Φ, is the X-ary relation on A defined by

$$\Phi^{\mathbf{A}}(h) \text{ iff } \bigwedge_{i \in I} \phi_i^{\mathbf{A}}(h\mathbf{x}_i).$$

We allow empty formulas ($I = \emptyset$) and interpret them $\Phi^{\mathbf{A}} = A^X$.

Definition 2.5 (PCSP). A pair of relational structures $(\mathbf{A}, \mathbf{B})$ over the same signature Σ is a *promise template* if $\Phi^{\mathbf{A}} \neq \emptyset$ implies $\Phi^{\mathbf{B}} \neq \emptyset$ for every conjunctive formula Φ in the signature Σ.

Given a promise template $(\mathbf{A}, \mathbf{B})$, the Promise Constraint Satisfaction Problem over $(\mathbf{A}, \mathbf{B})$, denoted by $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, is the following problem.

Input a finite τ-sorted set X and conjunctive Σ-formula Φ over X.
Output yes if $\Phi^{\mathbf{A}} \neq \emptyset$; no if $\Phi^{\mathbf{B}} = \emptyset$.[5]

In this context, X is regarded as a set of variables and the τ-sorted functions $h : X \to A$ as assignments of values in A to variables. The fact that $\phi_i^{\mathbf{A}}(h\mathbf{x}_i)$ means that the constraint $\phi_i(\mathbf{x}_i)$ is satisfied in $\mathbf{A}$ by the assignment h. Thus elements of $\Phi^{\mathbf{A}}$ (or $\Phi^{\mathbf{B}}$) can be thought of as solutions of Φ in $\mathbf{A}$ (or $\mathbf{B}$).

[4] This is the decision version. The *search version* is: given a conjunctive formula which is promised to be satisfiable in $\mathbf{A}$, find a satisfying assignment in $\mathbf{B}$. We only consider the decision version but results can be easily adjusted to the search one.

[5] The promise is that we are in one of the two cases, i.e., not in the case that $\Phi^{\mathbf{A}} = \emptyset$ and $\Phi^{\mathbf{B}} \neq \emptyset$.

The standard Constraint Satisfaction Problem over $\mathbf{A}$ [32] is $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$, where typically only single-sorted signatures are considered. Here is a concrete example of a problem that falls into this framework.

Example 2.6 (3LIN2). Given a system of linear equations over the two-element field $\mathbb{Z}_2$ with exactly 3 variables in each equation, the task is to decide whether it has a solution. This problem can be phrased as $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$, where $A = \{0, 1\}$, the signature consists of two $[3]$-ary symbols ϕ_0, ϕ_1, and their interpretation is $\phi_i^{\mathbf{A}}(a_1, a_2, a_3)$ iff $a_1 + a_2 + a_3 = i \pmod 2$.

We denote this PCSP as well as the template by 3LIN2. We will also use this convention for other PCSPs. Templates (and PCSPs) kLIN2 for a positive k are defined similarly.

An example of a "truly" promise problem is the following version of the approximate graph coloring problem.

Example 2.7 (3- versus 5- graph coloring). Given a graph, the task is to accept if it is 3-colorable and reject if it is not 5-colorable. This is $\mathrm{PCSP}(\mathbf{K}_3, \mathbf{K}_5)$, where $\mathbf{K}_k$ denotes a k-clique, that is, a structure with a k-element domain and one binary relational symbol interpreted as the disequality relation on the domain.

The last example is a CSP, but requires two sorts instead of just one. It is a version of the Label Cover problem.

Example 2.8 ($\mathrm{LC}_{D,E}$ – Label Cover). Fix finite disjoint sets D, E. Given a bipartite (multi-)graph with vertex set $U \cup V$ and a constraint $\pi_{uv} : D \to E$ for each edge $\{u, v\}$ in the graph, the task is to decide whether all the constraints can be satisfied, i.e., whether there exist functions $h_D : U \to D$ and $h_E : V \to E$ such that $\pi_{uv}(h_D(u)) = h_E(v)$ for every edge $\{u, v\}$.

This problem is $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$, where the sort symbols are D and E, $A = D \cup E$ (with $\mathrm{sort}(d) = D$ for $d \in D$ and $\mathrm{sort}(e) = E$ for $e \in E$), the signature consists of all functions $\pi : D \to E$ of arity $[2]$ with $\mathrm{sort}(1) = D$, $\mathrm{sort}(2) = E$, interpreted as $\pi^{\mathbf{A}}(d, e)$ iff $\pi(d) = e$. We typically omit the superscript in $\pi^{\mathbf{A}}$, which should not cause a confusion because of the different number of arguments.

The multi-sorted setting is primarily introduced to include problems such as the Label Cover. Note however that single-sorted PCSPs have natural formulations as multi-sorted ones. For instance, 3LIN2 from Example 2.6 can be introduced using a 3-sorted signature, with the 3-sorted domain $\phi_0^{\mathbf{A}} \cup \phi_1^{\mathbf{A}} \cup \{0, 1\}$ (where $\phi_i^{\mathbf{A}}$ is as in the example) and six binary symbols interpreted as the graphs of the projection mappings $\phi_i^{\mathbf{A}} \to \{0, 1\}$. In fact, this transformation from single-sorted to multi-sorted is essentially the reduction from a PCSP to MC discussed at the end of this section.

2.4 Polymorphisms

An N-ary polymorphism of $(\mathbf{A}, \mathbf{B})$ is an N-ary function from A to B that *preserves* every relation, that is, if we apply it component-wise to an N-tuple of tuples from $\phi^{\mathbf{A}}$, then we get a tuple from $\phi^{\mathbf{B}}$ for every ϕ in the common signature of $\mathbf{A}$ and $\mathbf{B}$. We phrase this property in terms of matrices. But first, let us discuss the terminology in the multi-sorted setting. Let A, B, Z be τ-sorted sets and N be a set.

An N*-ary function* f *from* A *to* B is a τ-sorted function $A^N \to B$, i.e., an element of B^{A^N}. It can be regarded as a collection of functions

$f_t : A_t^N \to B_t$, $t \in \tau$. When $|N| = 1$ an N-ary function is called unary.

For $n \in N$, the *N-ary projection to the n-th coordinate* is denoted by proj_n^N, i.e., $\mathrm{proj}_n^N : A^N \to A$ is defined by $\mathrm{proj}_n^N(\mathbf{a}) = \mathbf{a}(n)$. The set A will be clear from the context. For $z \in Z$, the Z-ary projection to the z-th coordinate $\mathrm{proj}_z^Z : A^Z \to A$ is defined by the same formula. Note that its image is contained in $A_{\mathrm{sort}(z)}$.

An element $M \in A^{Z \times N}$ can be regarded as a matrix whose rows are indexed by elements $z \in Z$, columns are indexed by elements $n \in N$, and the (z, n) entry is $M(z, n) \in A_{\mathrm{sort}(z)}$. The N-tuple of columns is denoted by $\mathrm{cols}(M) \in (A^Z)^N$ and, for $n \in N$, the n-th column is denoted by $\mathrm{col}_n(M) \in A^Z$. The Z-tuple of rows is denoted by $\mathrm{rows}(M) \in (A^N)^Z$ and the z-th row by $\mathrm{row}_z(M) \in (A_{\mathrm{sort}(z)})^N \subseteq A^N$.

Definition 2.9 (Polymorphism). Let $(\mathbf{A}, \mathbf{B})$ be a pair of Σ-structures and N a finite set. An *N-ary relation-matrix pair* for $\mathbf{A}$ is a pair (ϕ, M), where $\phi \in \sigma$ and $M \in A^{\mathrm{ar}(\phi) \times N}$ is a matrix whose each column is in $\phi^{\mathbf{A}}$. We denote by $\mathrm{Mat}(\mathbf{A}, N)$ the set of all N-ary relation-matrix pairs for $\mathbf{A}$.

$$\mathrm{Mat}(\mathbf{A}, N) = \{(\phi, M) \mid \phi \in \sigma, M \in A^{\mathrm{ar}(\phi) \times N}, \forall n \in N \; \mathrm{col}_n(M) \in \phi^{\mathbf{A}}\}.$$

An N-ary function f from A to B is a *polymorphism* of $(\mathbf{A}, \mathbf{B})$ if

$$\forall (\phi, M) \in \mathrm{Mat}(\mathbf{A}, N) \;\; \phi^{\mathbf{B}}(f \circ \mathrm{rows}(M)).$$

We denote by $\mathrm{Pol}^{(N)}(\mathbf{A}, \mathbf{B})$ the set of N-ary polymorphisms of $(\mathbf{A}, \mathbf{B})$ and the collection of these sets[6] by

$$\mathrm{Pol}(\mathbf{A}, \mathbf{B}) = (\mathrm{Pol}^{(N)}(\mathbf{A}, \mathbf{B}))_{N \in \mathrm{FinSet}}.$$

An N-ary polymorphism of $(\mathbf{A}, \mathbf{B})$ gives us a way to combine N tuples from a relation $\phi^{\mathbf{A}}$ to get a single tuple from $\phi^{\mathbf{B}}$. This extends to any conjunctive formula: if Φ is a conjunctive formula over X and $M \in A^{X \times N}$ has all the columns in $\Phi^{\mathbf{A}}$, then $\Phi^{\mathbf{B}}(f \mathrm{rows}(M))$. In other words, polymorphisms give us a way to combine $\mathbf{A}$-solutions to get a $\mathbf{B}$-solution.

The collection of polymorphisms of a pair $(\mathbf{A}, \mathbf{B})$ is closed under taking minors in the sense of the following definition. We call such collections function minions.[7]

Definition 2.10 (Function minion). Let A, B be τ-sorted sets and N, N' finite sets. For $f : A^N \to B$ and $\pi : N \to N'$, the *minor of f given by π*, denoted by $f^{(\pi)}$, is the N'-ary function $f^{(\pi)} : A^{N'} \to B$ defined by

$$f^{(\pi)}(\mathbf{a}) = f(\mathbf{a} \circ \pi) \quad \text{ for every } \mathbf{a} \in A^{N'}.$$

A collection $\mathscr{M} = (\mathscr{M}^{(N)})_{N \in \mathrm{FinSet}}$, where $\mathscr{M}^{(N)}$ is a set of N-ary functions from A to B, is a *function minion on* (A, B) if $f^{(\pi)} \in \mathscr{M}^{(N')}$ for every $N, N' \in \mathrm{FinSet}$, $f \in \mathscr{M}^{(N)}$, and $\pi : N \to N'$.

[6]It may seem that $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is a monstrous object: for each finite set N we have a set of N-ary functions from A to B. However, note that N-ary polymorphisms fully determine N'-ary polymorphisms whenever $|N| = |N'|$.

[7]Conversely, "almost" every function minion on finite sets is a minion of polymorphisms. The caveat is that we would need to allow infinite signatures and ignore functions of arity $\emptyset$.

Example 2.11. If $f : A^{[3]} \to A$ and $\pi : [3] \to [2]$ is defined by $\pi(1) = \pi(3) = 2$, $\pi(2) = 1$, then $f^{(\pi)}(a_1, a_2) = f(a_2, a_1, a_2)$. Informally, a minor of f is a function that can be obtained from f by merging and permuting variables (and introducing dummy ones).

Example 2.12. The collection given by $\mathscr{P}^{(N)} = \{\mathrm{proj}_n^N \mid n \in N\}$ is an easy and important example of a function minion on (A, A). Note that $(\mathrm{proj}_n^N)^{(\pi)} = \mathrm{proj}_{\pi(n)}^{N'}$ for every $\pi : N \to N'$.

A fundamental role (though not always explicit) in the CSP theory, as well as for various variants of CSPs, is played by a specific conjunctive formula Φ on the set of variables A^N for some finite set N. Note that assignments from the set of variables to A (to B) are exactly the N-ary functions from A to A (to B). For a fixed pair $(\mathbf{A}, \mathbf{B})$, the formula Φ is created by placing all the possible constraints with the restriction that $\Phi^{\mathbf{A}}(\mathrm{proj}_n^N)$ for every $n \in N$. Then $\Phi^{\mathbf{B}}$ is exactly the set of N-ary polymorphisms of $(\mathbf{A}, \mathbf{B})$.

Proposition 2.13 (Canonical formula). *For every pair $(\mathbf{A}, \mathbf{B})$ of finite Σ-structures and N a finite set, the Σ-formula*[8]

$$\Phi = \bigwedge_{(\phi, M) \in \mathrm{Mat}(\mathbf{A}, N)} \phi(\mathrm{rows}(M))$$

over the set of variables A^N satisfies

- $\Phi^{\mathbf{A}}(\mathrm{proj}_n^N)$ *for every $n \in N$, and*
- $\Phi^{\mathbf{B}} = \mathrm{Pol}^{(N)}(\mathbf{A}, \mathbf{B})$.

Note that, as claimed above, Φ is created by placing all possible constraints so that the first item is satisfied. Indeed, any constraint $\phi(\mathbf{x})$ over A^N is equal to $\phi(\mathrm{rows}(M))$ for some $\phi \in \sigma$ and $M \in A^{\mathrm{ar}(\phi) \times N}$. The fact that $\phi^{\mathbf{A}}(\mathrm{proj}_n^N)$ is exactly saying that $\mathrm{col}_n(M)$ is in $\phi^{\mathbf{A}}$, so satisfying $\phi^{\mathbf{A}}(\mathrm{proj}_n^N)$ for each $n \in N$, is equivalent to $M \in \mathrm{Mat}(\mathbf{A}, N)$.

Applying the canonical formula for a singleton set N gives us a characterization of templates. Item (iii) in the proposition below is in fact often used as a definition of a template.

Proposition 2.14 (Characterization of templates). *Let $(\mathbf{A}, \mathbf{B})$ be a pair of finite Σ-structures. The following are equivalent.*

(i) $(\mathbf{A}, \mathbf{B})$ *is a promise template.*
(ii) *For each conjunctive Σ-formula Φ over the set of variables A, if $\Phi^{\mathbf{A}}(\mathrm{id}_A)$, then $\Phi^{\mathbf{B}} \neq \emptyset$.*
(iii) *There exists a unary polymorphism of $(\mathbf{A}, \mathbf{B})$.*

Starting from the canonical formula, the theory can now go in two directions. The original approach for CSPs from [10, 16, 25, 38] can be formulated, with a slight imprecision, as follows. If $\mathrm{Pol}(\mathbf{A}, \mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{A}', \mathbf{A}')$, then each relation in $\mathbf{A}'$ can be defined by existentially quantifying the canonical formula (for $(\mathbf{A}, \mathbf{A})$) for a suitable N, which then implies $\mathrm{PCSP}(\mathbf{A}', \mathbf{A}') \leq \mathrm{PCSP}(\mathbf{A}, \mathbf{A})$, where $\leq$ denotes the polynomial-time reducibility. This direction can continue by replacing definability with more expressive constructions and thus allowing us to replace the inclusion $\mathrm{Pol}(\mathbf{A}, \mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{A}', \mathbf{A}')$ by weaker requirements, which in turn gives us more reductions. One step in this process replaced the inclusion by the

[8]In the terminology used in e.g. [16], this is the canonical conjunctive query of the N-th power of $\mathbf{A}$. In [38], this construction was called the indicator problem.

existence of so-called minion homomorphisms [10] and this was generalized to PCSPs in [4] based on [23, 47].

The second direction that the theory can take, also based on the canonical formula, avoids the definability considerations. Instead, it proves the reduction theorem based on minion homomorphisms in a more direct way. This approach, discovered in [4], is the one we follow in this work.

2.5 Minion homomorphisms and reductions

A minion homomorphism between function minions is a mapping of N-ary functions in the first minion to N-ary functions in the second minion that preserves taking minors. This concept does not depend on concrete functions in the minion, it only depends on the mappings $f \mapsto f^{(\pi)}$. We therefore first introduce an abstraction of function minions that carries exactly this information.

Definition 2.15 (Minion). An (abstract) *minion* $\mathscr{M}$ consists of a collection of sets $(\mathscr{M}^{(N)})_{N\in\mathrm{FinSet}}$, together with a *minor map* $\mathscr{M}^{(\pi)} : \mathscr{M}^{(N)} \to \mathscr{M}^{(N')}$ for every function $\pi : N \to N'$, which satisfies that $\mathscr{M}^{(\mathrm{id}_N)} = \mathrm{id}_{\mathscr{M}^{(N)}}$ for all finite sets N and $\mathscr{M}^{(\pi)} \circ \mathscr{M}^{(\pi')} = \mathscr{M}^{(\pi\circ\pi')}$ whenever such a composition is well-defined. When the minion is clear, we write $f^{(\pi)}$ for $\mathscr{M}^{(\pi)}(f)$.

A minion $\mathscr{M}$ is *nontrivial* if $\mathscr{M}^{(N)}$ is nonempty for every (equivalently some) nonempty N.

The most natural choice of morphisms between minions is minion homomorphisms defined as follows.[9]

Definition 2.16 (Minion homomorphism). Let $\mathscr{M}$ and $\mathscr{M}'$ be minions. A *minion homomorphism* from $\mathscr{M}$ to $\mathscr{M}'$ is a collection of functions $(\xi^{(N)} : \mathscr{M}^{(N)} \to \mathscr{M}'^{(N)})_{N\in\mathrm{FinSet}}$ that preserves taking minors, that is, $\xi^{(N')}(\mathscr{M}^{(\pi)}(f)) = \mathscr{M}'^{(\pi)}(\xi^{(N)}(f))$ for every $N, N' \in \mathrm{FinSet}$, $f \in \mathscr{M}^{(N)}$, and $\pi : N \to N'$.

The reduction theorem discussed above is the following.

Theorem 2.17 (Reductions via minion homomorphism). *Let* $(\mathbf{A},\mathbf{B})$, $(\mathbf{A}',\mathbf{B}')$ *be promise templates. If there is a minion homomorphism from* $\mathrm{Pol}(\mathbf{A},\mathbf{B})$ *to* $\mathrm{Pol}(\mathbf{A}',\mathbf{B}')$*, then we have* $\mathrm{PCSP}(\mathbf{A}',\mathbf{B}') \leq \mathrm{PCSP}(\mathbf{A},\mathbf{B})$.[10]

This reduction theorem explains hardness for CSPs in the following sense: if $\mathrm{PCSP}(\mathbf{A},\mathbf{A})$ cannot be solved in polynomial time by the algorithms in [26, 54], then $\mathrm{Pol}(\mathbf{A},\mathbf{A})$ has a minion homomorphism to the projection minion from Example 2.12 (with $|A| \geq 2$), which has a minion homomorphism to every nontrivial minion.

The modern proof of Theorem 2.17 is via showing that each $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$ is equivalent to a certain computational problem parameterized by the (abstract) minion of polymorphisms, called the *minor condition problem*, and that a minion homomorphism (trivially) gives a reduction between such problems.

Definition 2.18 (Minor Condition Problem). Given a nontrivial minion $\mathscr{M}$ and an integer k, the Minor Condition Problem for $\mathscr{M}$ and k, denoted by $\mathrm{MC}(\mathscr{M},k)$ is the following problem:

Input
1. disjoint sets U and V (the sets of *variables*),
2. a set D_x with $|D_x| \leq k$ for every $x \in U \cup V$ (the *domain* of x),
3. a set of formal expressions of the form $\pi(u) = v$, where $u \in U$, $v \in V$, and $\pi : D_u \to D_v$ (the *minor conditions*).

Output yes: if there exists a function h from $U \cup V$ with $h(x) \in D_x$ (for each $x \in U \cup V$) such that, for each minor condition $\pi(u) = v$, we have $\pi(h(u)) = h(v)$.

no: if there does not exist a function h from $U \cup V$ with $h(x) \in \mathscr{M}^{(D_x)}$ such that, for each minor condition $\pi(u) = v$, we have $\mathscr{M}^{(\pi)}(h(u)) = h(v)$.

The name for the minor condition problem comes from the requirement $\mathscr{M}^{(\pi)}(h(u)) = h(v)$: the element of $\mathscr{M}$ assigned to v must be the minor of the element assigned to u given by π. Note also that $\pi(h(u)) = h(v)$ is equivalent to $\mathscr{P}^{(\pi)}(h(u)) = h(v)$ for the projection minion $\mathscr{P}$ from Example 2.12.

Since $\mathscr{M}$ is nontrivial, an instance cannot simultaneously be a yes and no instance. Indeed, if h witnesses that an instance is a yes instance, then $x \mapsto f^{(\gamma_{h(x)})}$, where $f \in \mathscr{M}^{([1])}$ and $\gamma_{h(x)}$ is the mapping $[1] \to D_x$ with $1 \mapsto h(x)$, witnesses that the instance is not a no instance.

Notice also that an instance of MC is very similar to an instance of LC from Example 2.8. In fact, $\mathrm{MC}(\mathscr{M},k)$ can be phrased as a PCSP over a certain multi-sorted template.

The reduction between two PCSPs in Theorem 2.17 based on a minion homomorphism is a composition of three reductions: from PCSP to MC, from MC (over one minion) to MC (over another one), and from MC to PCSP. Overall, we have the following reductions (depicted as arrows) for templates $(\mathbf{A},\mathbf{B})$, $(\mathbf{A}',\mathbf{B}')$, their polymorphism minions $\mathscr{M}$, $\mathscr{M}'$, and a sufficiently large k:

$$\begin{tikzcd} \mathrm{PCSP}(\mathbf{A}',\mathbf{B}') \arrow[d, leftrightarrow] & \mathrm{PCSP}(\mathbf{A},\mathbf{B}) \arrow[d, leftrightarrow] \\ \mathrm{MC}(\mathscr{M}',k) \arrow[r] & \mathrm{MC}(\mathscr{M},k) \end{tikzcd}$$

3 VALUED PROMISE CSP

The generalization of PCSP to the valued setting is obtained by replacing relations by valued relations, that is, mappings $A^Z \to \mathbb{Q} \cup \{-\infty\}$, and suitably adjusting the concepts. The crisp PCSPs can be modelled as Valued PCSPs with $\{-\infty, 0\}$-valued relations.

This section covers the basics up to a valued and improved version of canonical formulas. A generalization of minion homomorphisms and the main reduction theorem are given in Section 4 and examples of valued homomorphisms are shown in Section 5. The missing proofs are in the full version [6].

3.1 Preliminaries

We denote by $\mathbb{Q}^+$ ($\mathbb{Q}_0^+$) the set of positive (nonnegative) rational numbers and by $\overline{\mathbb{Q}}$ the set of rational numbers together with an additional symbol $-\infty$. We naturally extend the operations and order, leaving $0 \cdot -\infty$ undefined.

We will work with probability distributions on finite sets with rational probabilities, so we can formally regard a probability distribution on N as a function $\mu : N \to \mathbb{Q}_0^+$ such that $\sum_{n\in N} \mu(n) = 1$. We denote by ΔN the set of probability distributions on N. The

[9] In the language of category theory, a minion is simply a functor from the category of finite sets to the category of sets (a minion corresponds to the functor $X \mapsto \mathscr{M}^{(X)}$, $\pi \mapsto \mathscr{M}^{(\pi)}$) and minion homomorphisms are natural transformations. Note that the projection minion from Example 2.12 is naturally equivalent to the inclusion functor.

[10] In fact, Theorem 1 even holds with a log-space reduction, but that will not concern us.

support of a probability distribution $\mu \in \Delta N$ is the set $\mathrm{Supp}(\mu) = \{n \in N \mid \mu(n) > 0\}$.

If $f : N \to N'$ and $\mu \in \Delta N$, we define $f(\mu) \in \Delta N'$ in the natural way $(f(\mu))(n') = \sum_{n; f(n)=n'} \mu(n)$, that is, n' can be sampled according to $f(\mu)$ by sampling n according to μ and computing $n' = f(n)$. We also use the notation $F(\mu)$ when $\mu \in \Delta N$ and F is a probability distribution on a set of mappings $N \to N'$, i.e., to sample $F(\mu)$ we independently sample $n \sim \mu$, $f \sim F$ and compute $f(n)$.

Given $\mu \in \Delta N$ and a function $f : N \to \mathbb{Q}$, we denote by $\mathbb{E}_{n\sim\mu} f(n)$ the expected value of $f(n)$ when n is sampled according to μ, i.e., $\mathbb{E}_{n\sim\mu} f(n) = \sum_{n\in N} \mu(n) f(n)$.

A basic tool for some of the proofs is Farkas' lemma [50]. The following formulation will be convenient for us. In the statement, juxtaposition denotes the standard matrix multiplication, ^{T} is used for the transposition, and $\mathbf{x} \geq 0$ means that all the components are nonnegative.

Theorem 3.1 (Farkas' lemma). *Let I, J be finite sets, $F \in \mathbb{Q}^{I\times J}$, and $\mathbf{q} \in \mathbb{Q}^I$. The following are equivalent.*

(i) $\exists \mathbf{y} \in (\mathbb{Q}_0^+)^J\ F\mathbf{y} \leq \mathbf{q}$.
(ii) $\forall \mathbf{x} \in (\mathbb{Q}_0^+)^I\ (F^T\mathbf{x} \geq 0 \implies \mathbf{q}^T\mathbf{x} \geq 0)$.

3.2 Valued Structures and the Valued PCSP

Definition 3.2 (Valued relational structure). Let τ be a set (of sorts), and let Z and A be τ-sorted sets. A *Z-ary valued relation on A*, or a *Z-ary payoff function on A*, is a function $\phi : A^Z \to \overline{\mathbb{Q}}$. The *feasibility set* of ϕ, denoted by $\mathrm{feas}(\phi)$, is the pre-image of $\mathbb{Q}$ under ϕ.

Let Σ be a signature. A *valued Σ-structure* $\mathbf{A}$ consists of a τ-sorted set A called the *domain* and an $\mathrm{ar}(\phi)$-ary valued relation $\phi^{\mathbf{A}}$ on A, the *interpretation of ϕ in* $\mathbf{A}$, for every $\phi \in \sigma$. Such a structure $\mathbf{A}$ is said to be finite if A is finite. For a rational number c we write $\mathbf{A} \leq c$ if $\phi^{\mathbf{A}}(\mathbf{a}) \leq c$ for every $\phi \in \sigma$ and $\mathbf{a} \in A^{\mathrm{ar}(\phi)}$.

For a valued Σ-structure $\mathbf{A}$, the *feasibility structure*, denoted by $\mathrm{feas}(\mathbf{A})$, is the (non-valued) Σ-structure obtained by replacing each $\phi^{\mathbf{A}}$ by $\mathrm{feas}(\phi^{\mathbf{A}})$.

Definition 3.3 (Payoff formula). Let Σ be a signature and X a finite τ-sorted set. A *payoff formula* over X in the signature Σ, or a *payoff Σ-formula*, is a formal expression of the form

$$\Phi = \sum_{i\in I} w_i \phi_i(\mathbf{x}_i),$$

where I is a finite nonempty set, and $w_i \in \mathbb{Q}_0^+$ (*weights*), $\phi_i \in \sigma$, $\mathbf{x}_i \in X^{\mathrm{ar}(\phi_i)}$ for all $i \in I$. The *weight of* Φ is $w(\Phi) = \sum_{i\in I} w_i$.

Given additionally a valued Σ-structure $\mathbf{A}$, the *interpretation of* Φ *in* $\mathbf{A}$, or *the X-ary valued relation defined in* $\mathbf{A}$ *by* Φ, is the X-ary valued relation on A defined by

$$\Phi^{\mathbf{A}}(h) = \sum_{i\in I} w_i \phi_i^{\mathbf{A}}(h\mathbf{x}_i),$$

where summands $0 \cdot -\infty$ are evaluated as $-\infty$ (but we keep $0 \cdot -\infty$ undefined in different contexts).

We allow empty formulas Φ, and define $w(\Phi) = 0$ and $\Phi^{\mathbf{A}}(h) = 0$.

Note that the convention that $0\cdot-\infty = -\infty$ ensures that $\mathrm{feas}(\Phi^{\mathbf{A}})$ is equal to the interpretation of $\bigwedge_{i\in I} \phi_i(\mathbf{x}_i)$ in $\mathrm{feas}(\mathbf{A})$ and, for $h \in \mathrm{feas}(\Phi^{\mathbf{A}})$, the sum defining $\Phi^{\mathbf{A}}(h)$ does not contain any infinities.

Definition 3.4 (Valued PCSP). A *valued promise template* is a quadruple $(\mathbf{A}, \mathbf{B}, c, s)$ where

- $\mathbf{A}, \mathbf{B}$ are valued relational structures in the same signature Σ, and
- $c, s \in \mathbb{Q}$ are the *completeness* and *soundness* parameters respectively

such that $\exists h\ \Phi^{\mathbf{A}}(h) \geq c\, w(\Phi)$ implies $\exists h\ \Phi^{\mathbf{B}}(h) \geq s\, w(\Phi)$ for every payoff Σ-formula Φ.

Given a valued promise template $(\mathbf{A}, \mathbf{B}, c, s)$, the *Promise Constraint Satisfaction Problem over* $(\mathbf{A}, \mathbf{B}, c, s)$, denoted by $\mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s)$, is the following problem.

Input a finite τ-sorted set X and a payoff Σ-formula Φ over X.
Output yes if $\exists h\ \Phi^{\mathbf{A}}(h) \geq c\, w(\Phi)$; no if $\forall h\ \Phi^{\mathbf{B}}(h) < s\, w(\Phi)$.[11]

Let us start the discussion about this generalization of PCSPs by giving examples of problems included in this framework.

First observe that Valued PCSPs indeed generalize crisp PCSPs: for a crisp template $(\mathbf{A}', \mathbf{B}')$ we can define a valued promise template $(\mathbf{A}, \mathbf{B}, 0, 0)$ by setting $\phi^{\mathbf{A}}(\mathbf{a}) = 0$ if $\mathbf{a} \in \phi^{\mathbf{A}'}$ and $\phi^{\mathbf{A}}(\mathbf{a}) = -\infty$ otherwise for all $\phi \in \sigma$, $\mathbf{a} \in A^{\mathrm{ar}(\phi)}$, and similarly for $\mathbf{B}'$. Clearly, $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}')$ is equivalent to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B}, 0, 0)$.

Another natural valued promise template associated to a crisp template $(\mathbf{A}', \mathbf{B}')$ is $(\mathbf{A}, \mathbf{B}, c, s)$, where $\phi^{\mathbf{A}}(\mathbf{a}) = 1$ if $\mathbf{a} \in \phi^{\mathbf{A}'}$, $\phi^{\mathbf{A}}(\mathbf{a}) = 0$ otherwise, and $c \geq s$ are the completeness and soundness parameters. PCSPs over such templates include e.g. approximation problems for MaxCSPs, such as the following concrete problems.

Example 3.5 ($\mathrm{3LIN2}(c, s)$). Given a weighted system of linear equations over $\mathbb{Z}_2$ with exactly 3 variables in each equation, accept if there exists an assignment that satisfies a c-fraction of the equations (taking weights into account), and reject if there is no assignment that satisfies an s-fraction of the equations.

This problem is $\mathrm{PCSP}(\mathbf{A}, \mathbf{A}, c, s)$ where $A = \{0, 1\}$ and the signature consists (as in Example 2.6) of two $[3]$-ary symbols ϕ_0, ϕ_1 interpreted as $\phi_i^{\mathbf{A}}(a_1, a_2, a_3) = 1$ if $a_1 + a_2 + a_3 = i \pmod 2$ and $\phi_i^{\mathbf{A}}(a_1, a_2, a_3) = 0$ otherwise.

We denote this PCSP as well as the template by $\mathrm{3LIN2}(c, s)$. Note that $\mathrm{3LIN2}(1, 1)$ is another formulation of 3LIN2.

The following maximization version of Example 2.7, first introduced in [46], nicely combines the promise and valued frameworks.

Example 3.6 (Maximum 3- versus 5-coloring of graphs). Given an edge-weighted graph G, the task is to accept if G admits a 3-coloring with a c-fraction of non-monochromatic edges, and reject if G does not admit a 5-coloring with an s-fraction of non-monochromatic edges. But this is just $\mathrm{PCSP}(\mathbf{K}_3, \mathbf{K}_5, c, s)$, where $\mathbf{K}_k$ is the k-clique, interpreted here as having payoff 1 on the edges and 0 on non-edges.

Another example that fits in our framework is a variant of Example 3.6 concerned with a 3- vs 5- coloring of a large induced subgraph of a given graph [36].

A gap version of Example 2.8, the Gap Label Cover problem, is a starting point for many NP-hardness results in approximation.

Example 3.7 ($\mathrm{GLC}_{D,E}(c, s)$: Gap Label Cover). Fix disjoint finite sets D, E and rationals $1 \geq c \geq s > 0$. Given a weighted bipartite

[11] The promise is that we are in one of the two cases, i.e., not in the case that $\forall h\ \Phi^{\mathbf{A}}(h) < c\, w(\Phi)$ and $\exists h\ \Phi^{\mathbf{B}}(h) \geq s\, w(\Phi)$.

graph with vertex set $U \cup V$ and a constraint $\pi_{uv} : D \to E$ for each edge $\{u, v\}$, accept if a c-fraction (taking weights into account) of the constraints can be satisfied, and reject if not even an s-fraction of the constraints can be satisfied.

This problem is $\mathrm{PCSP}(\mathbf{A}, \mathbf{A}, c, s)$, where the sort symbols are D and E, $A = D \cup E$, the signature consists of all functions $\pi : D \to E$ of arity [2] ($\mathrm{sort}(1) = D$, $\mathrm{sort}(2) = E$), interpreted as $\pi^{\mathbf{A}}(d, e) = 1$ if $\pi(d) = e$ and $\pi^{\mathbf{A}}(d, e) = 0$ otherwise.

A consequence of the PCP theorem [1, 31] and the Parallel Repetition theorem [49] is that for every $\epsilon > 0$ there exist D, E such that $\mathrm{GLC}_{D,E}(1, \epsilon)$ is NP-hard.

Problems with $\mathbf{A} \le c$ are said to have *perfect completeness*. By giving up perfect completeness in the Gap Label Cover and restricting the functions $\pi : D \to E$ to be bijections, we obtain the well-known Unique Games problem, a starting point of many conditional NP-hardness results.

Example 3.8 (Unique Games). We fix disjoint sets D and E such that $|D| = |E|$ and $\epsilon > 0$, and define $\mathbf{A}$ as in Example 3.7 but only using bijective $\pi : D \to E$.

The Unique Games Conjecture of Khot [40] states that for every $\epsilon > 0$ there exist D and E such that $\mathrm{PCSP}(\mathbf{A}, \mathbf{A}, 1 - \epsilon, \epsilon)$ is NP-hard.

Nice examples where infinite and nonzero finite payoffs both appear are the vertex cover and independent set problems in graphs. While they are in some sense complementary,[12] it is known that these two problems differ significantly with respect to approximability: vertex cover admits a 2-approximation whereas there is no constant factor approximation for independent set. The following examples show the optimization versions of these problems.

Example 3.9 (Independent Set). An *independent set* in a graph G is a subset S of the vertices of G such that every edge of the graph is incident to *at most* one vertex in S. In the Independent Set problem with parameter $1 \ge c > 0$, the task is, given a vertex-weighted graph G, to accept if G has an independent set of fractional size at least c, and reject otherwise.

Independent Set fits in our framework as $\mathrm{PCSP}(\mathbf{A}, \mathbf{A}, c, c)$ where $1 \ge c > 0$ (the lower bound on the weight of the independent set), $A = \{0, 1\}$, and the signature consists of a unary relation symbol ϕ interpreted as $\phi^{\mathbf{A}}(a) = a$ (enforcing that the fractional size of the independent set is at least c) and a binary relation symbol ψ interpreted as $\psi^{\mathbf{A}}(1, 1) = -\infty$ and $\psi^{\mathbf{A}}(a_1, a_2) = 0$ for all other values of a_1, a_2 (enforcing that if the subset of the vertices that are assigned 1 yields a finite payoff, then it is an independent set).

Example 3.10 (Vertex Cover). A *vertex cover* of a graph G is a subset S of the vertices of G such that every edge of the graph is incident to *at least* one vertex in S. In the Vertex Cover problem with parameter c, the task is to accept if a vertex-weighted graph G has a vertex cover of fractional size at most c, and reject otherwise.

Vertex cover is a minimization problem. However, it can be phrased in our framework as $\mathrm{PCSP}(\mathbf{A}, \mathbf{A}, -c, -c)$, where the domain and signature are as in Example 3.9 but the symbols are interpreted as $\phi^{\mathbf{A}}(a) = -a$, $\psi^{\mathbf{A}}(0, 0) = -\infty$, and $\psi^{\mathbf{A}}(a_1, a_2) = 0$ otherwise.

[12] A set of vertices is independent iff its complement is a vertex cover.

We now discuss several possible variations and modifications of the definition of valued PCSPs, ordered by the significance of the difference they would cause.

First, we have decided for the maximization version of the definition. The corresponding minimization problem can be obtained by multiplying all payoff functions as well as c and s by -1 (cf. Example 3.10), so results for our version can be easily transferred to the minimization version and vice versa.

Second, note that by shifting (and/or scaling) the payoff functions in $\mathbf{A}$ and $\mathbf{B}$ and modifying c and s in the same way, we get an equivalent problem. It would therefore be possible to fix c, s, e.g., to $c = s = 0$ and define a template just as a pair $(\mathbf{A}, \mathbf{B})$. Our choice here was inspired by a more natural formulation of problems such as $3\mathrm{LIN}2(c, s)$.

Third, a natural version of the definition is to require Φ to be *normalized*, that is, $w(\Phi) = 1$. An instance can then be regarded as a probability distribution on constraints; $\Phi^{\mathbf{A}}(h)$ can be interpreted as the expected value of $\phi^{\mathbf{A}}(h\mathbf{x}_i)$ when constraint $\phi(\mathbf{x}_i)$ is selected according to this distribution. Note that an equivalent normalized instance can be obtained by dividing all the weights by $w(\Phi)$, unless $w(\Phi) = 0$, i.e., all the weights are zero. Therefore this alternative formulation differs from our formulation only very slightly.

Fourth, a more substantial change would be to require that all the weights be equal, say to 1. We regard the presented version as slightly more natural. Note however that it is often the case that positive (algorithmic) results work even for the weighted version and negative (hardness) results already for the non-weighted one, by emulating weights via repeated constraints.

Fifth, the most substantial change would be to not fix c, s in advance and rather make them part of the instance. An important intermediate choice is to fix s/c: a template would be a triple $(\mathbf{A}, \mathbf{B}, \kappa)$, an instance would include c (not s), and yes and no would be defined in the same way as in the definition with $s = \kappa c$. Such a framework includes constant factor approximation problems for MaxCSP; for $\kappa = 1$ and $\mathbf{A} = \mathbf{B}$ this framework essentially coincides with general-valued CSPs [30, 43]. In fact, the algebraic framework discovered for $\kappa = 1$ and general $\mathbf{A}, \mathbf{B}$ by Kazda [39] was among the starting points for this work. We give basics of this framework in the full version [6].

3.3 Polymorphisms

A natural generalization of an N-ary operation from A to B to the valued world consists of a probability distribution on N and a probability distribution on a set $\mathcal{F}$ of (normal) N-ary operations from A to B. In our situation $\mathcal{F}$ will be the set of all N-ary polymorphisms of the pair of feasibility structures corresponding to a pair $(\mathbf{A}, \mathbf{B})$ of valued structures. We therefore denote

$$\mathrm{PolFeas}(\mathbf{A}, \mathbf{B}) = \mathrm{Pol}(\mathrm{feas}(\mathbf{A}), \mathrm{feas}(\mathbf{B}))$$

and introduce the following concept.

Definition 3.11 (Weighting). Let $\mathscr{M}$ be a minion and N a finite set. An *N-ary weighting of $\mathscr{M}$* is a pair

$$\Omega = (\Omega^{\mathrm{in}}, \Omega^{\mathrm{out}}) \quad \text{where} \quad \Omega^{\mathrm{in}} \in \Delta N,\ \Omega^{\mathrm{out}} \in \Delta \mathscr{M}^{(N)}.$$

Relation-matrix pairs for valued structures are introduced in an analogous fashion as in the crisp case. Given such a pair and a weighting Ω of $\mathrm{PolFeas}(\mathbf{A},\mathbf{B})$ we have two naturally associated rationals: the expected payoff in $\mathbf{A}$ of the n-th column, when n is selected according to Ω^{in}; and the expected payoff in $\mathbf{B}$ of the tuple obtained by applying f to the rows of the matrix, when f is selected according to Ω^{out}.

Definition 3.12 (Relation-matrix pairs, input and output payoffs). Let $(\mathbf{A},\mathbf{B})$ be a pair of valued Σ-structures, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$, and N a finite set. We define

$$\mathrm{Mat}(\mathbf{A},N) = \{(\phi,M) \mid \phi\in\sigma, M\in A^{\mathrm{ar}(\phi)\times N}, \forall n\in N\ \mathrm{col}_n(M)\in \mathrm{feas}(\phi^{\mathbf{A}})\}.$$

For an N-ary weighting Ω of $\mathscr{M}$ and $(\phi,M)\in\mathrm{Mat}(\mathbf{A},N)$ we define

$$\Omega^{\mathrm{in}}[\phi,M] = \mathop{\mathbb{E}}_{n\sim\Omega^{\mathrm{in}}} \phi^{\mathbf{A}}(\mathrm{col}_n(M)) \text{ and}$$
$$\Omega^{\mathrm{out}}[\phi,M] = \mathop{\mathbb{E}}_{f\sim\Omega^{\mathrm{out}}} \phi^{\mathbf{B}}(f\,\mathrm{rows}(M)).$$

For an N-ary weighting Ω of $\mathscr{M}$ and functions $\alpha : N\to\mathbb{Q}$, $\beta : \mathscr{M}^{(N)}\to\mathbb{Q}$ we define

$$\Omega^{\mathrm{in}}[\alpha] = \mathop{\mathbb{E}}_{n\sim\Omega^{\mathrm{in}}}\alpha(n) \quad\text{and}\quad \Omega^{\mathrm{out}}[\beta] = \mathop{\mathbb{E}}_{f\sim\Omega^{\mathrm{out}}}\beta(f).$$

For a weighting Ω, each relation-matrix pair thus gives us a point $(\Omega^{\mathrm{in}}[\phi,M],\Omega^{\mathrm{out}}[\phi,M])$ in the plane $\mathbb{Q}^2$. We say that Ω is a κ-polymorphism if all these points lie on or above the line with slope κ going through (c,s).

Definition 3.13 (Polymorphisms). Let $(\mathbf{A},\mathbf{B})$ be a pair of valued Σ-structures, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$, and $c,s\in\mathbb{Q}$.

- Let $\kappa\in\mathbb{Q}_0^+$. An N-ary weighting Ω of $\mathscr{M}$ is a *κ-polymorphism* of $(\mathbf{A},\mathbf{B},c,s)$ if
$$\forall(\phi,M)\in\mathrm{Mat}(\mathbf{A},N)\ \ \Omega^{\mathrm{out}}[\phi,M]-s\ge\kappa(\Omega^{\mathrm{in}}[\phi,M]-c).$$
- An N-ary weighting Ω of $\mathscr{M}$ is a *polymorphism* of $(\mathbf{A},\mathbf{B},c,s)$ if it is a κ-polymorphism for some $\kappa\in\mathbb{Q}_0^+$.
- A finite family $(\Omega_j)_{j\in J}$ of weightings of $\mathscr{M}$ of arities $\mathcal{N} = (N_j)_{j\in J}$ is an $\mathcal{N}$-ary *plurimorphism* of $(\mathbf{A},\mathbf{B},c,s)$ if there exists $\kappa\in\mathbb{Q}_0^+$ such that every Ω_j is a κ-polymorphism.

We will denote by $\kappa\text{-}\mathrm{Pol}^{(N)}(\mathbf{A},\mathbf{B},c,s)$, $\mathrm{Pol}^{(N)}(\mathbf{A},\mathbf{B},c,s)$, and $\mathrm{Plu}^{(\mathcal{N})}(\mathbf{A},\mathbf{B},c,s)$ the sets of all N-ary κ-polymorphism, N-ary polymorphisms, and $\mathcal{N}$-ary plurimorphisms, respectively, and by $\kappa\text{-}\mathrm{Pol}(\mathbf{A},\mathbf{B},c,s)$, $\mathrm{Pol}(\mathbf{A},\mathbf{B},c,s)$, and $\mathrm{Plu}(\mathbf{A},\mathbf{B},c,s)$ the collections of the corresponding morphisms indexed by their arities.

For a polymorphism Ω, all points in $\mathbb{Q}^2$ determined by relation-matrix pairs lie above or on a line going through (c,s) with a nonnegative slope κ. In particular, these points avoid the region $R=\{(x,y)\mid x\ge c, y<s\}$ and so does any convex combination of these points (since half-planes are convex). It is easy to see that, conversely, if the convex hull of these points avoids R, then Ω is a polymorphism. This is phrased more generally for plurimorphisms in item (iii) of the following proposition, in the language of probabilities. It is also geometrically clear that it is enough to require that the convex hulls of two points avoid R, leading to item (ii).

Proposition 3.14 (Alt. definitions of plurimorphisms). *Let $(\mathbf{A},\mathbf{B})$ be a pair of valued Σ-structures. Further, let $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$, $c,s\in\mathbb{Q}$, and let $(\Omega_j)_{j\in J}$ be a finite family of weightings of $\mathscr{M}$ of arities $(N_j)_{j\in J}$. The following are equivalent.*

(i) *$(\Omega_j)_{j\in J}$ is a plurimorphism of $(\mathbf{A},\mathbf{B},c,s)$.*
(ii) *Each pair $(\Omega_j,\Omega_{j'})$ with $j,j'\in J$ is a plurimorphism of $(\mathbf{A},\mathbf{B},c,s)$.*
(iii) *For every probability distribution*
$$\mu\in\Delta\{(j,\phi,M)\mid j\in J,(\phi,M)\in\mathrm{Mat}(\mathbf{A},N_j)\}$$
we have that
$$\mathop{\mathbb{E}}_{(j,\phi,M)\sim\mu}\Omega_j^{\mathrm{in}}[\phi,M]\ge c \implies \mathop{\mathbb{E}}_{(j,\phi,M)\sim\mu}\Omega_j^{\mathrm{out}}[\phi,M]\ge s.$$

In the crisp case, we observed that polymorphisms give us a way to combine solutions in $\mathbf{A}$ to obtain solutions in $\mathbf{B}$. Item (iii) with $|J|=1$ can be used to show a valued version of this fact: if Ω is a polymorphism of $(\mathbf{A},\mathbf{B},c,s)$, Φ is a normalized payoff formula over X, and $M\in A^{X\times N}$ is such that the expected payoff in $\mathbf{A}$ of the n-th column when $n\sim\Omega^{\mathrm{in}}$ is at least c, then the expected payoff in $\mathbf{B}$ of $f\mathrm{rows}(M)$ when $f\sim\Omega^{\mathrm{out}}$ is at least s. Details are provided in [6].

We also remark that the notion of (c,s)-approximate polymorphism of Brown-Cohen and Raghavendra from [24] (implied by Definitions 1.6 and 1.9 in their paper) is essentially introduced as in item (iii) of Proposition 3.14 (for $|J|=1$ and $\mathbf{A}=\mathbf{B}$ and uniform distribution Ω^{in}).

Unlike in the crisp case, we do not introduce a concept of valued function minion. The reason is that we currently do not know for sure what the right choice of closure properties would be, so that valued function minions would be exactly collections of plurimorphisms of templates. The obvious properties come from item (ii) and the fact that κ-polymorphisms are closed under convex combinations and taking minors (defined naturally, see [6]).

3.4 Canonical payoff formulas

A natural valued refinement of canonical formulas in Proposition 2.13 is the following fact. It is useful for characterizing templates and definability (which is not discussed in this work), but the main theorem requires a more complex version of canonical formulas, presented in Proposition 3.17.

Proposition 3.15 (Canonical payoff formula). *Let $(\mathbf{A},\mathbf{B})$ be a pair of valued Σ-structures, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$, $c,s\in\mathbb{Q}$, N a finite set, $\alpha : N\to\mathbb{Q}$, and $\beta : \mathscr{M}^{(N)}\to\mathbb{Q}$. Suppose further that if $\mathbf{A}\le c$, then $\alpha\le c$ (i.e., $\alpha(n)\le c$ for all $n\in N$). Then the following are equivalent.*

(i) *For each $\kappa\in\mathbb{Q}_0^+$ and each $\Omega\in\kappa\text{-}\mathrm{Pol}^{(N)}(\mathbf{A},\mathbf{B},c,s)$, $\Omega^{\mathrm{out}}[\beta]-s\ge\kappa(\Omega^{\mathrm{in}}[\alpha]-c)$.*
(ii) *There exists a payoff formula Φ over the set of variables A^N such that*
$$\forall n\in N \qquad \Phi^{\mathbf{A}}(\mathrm{proj}_n^N)-c\,w(\Phi)\ge\alpha(n)-c$$
$$\forall f\in\mathscr{M}^{(N)} \qquad \Phi^{\mathbf{B}}(f)-s\,w(\Phi)\le\beta(f)-s$$
$$\mathrm{feas}(\Phi^{\mathbf{B}})=\mathscr{M}^{(N)}.$$

Proof. We first observe that item (ii) is equivalent to the following condition.

(iii) There exist $w_{\phi,M}\in\mathbb{Q}_0^+$, where $(\phi,M)\in\mathrm{Mat}(\mathbf{A},N)$, such that the payoff formula
$$\Phi=\sum_{(\phi,M)\in\mathrm{Mat}(\mathbf{A},N)} w_{\phi,M}\,\phi(\mathrm{rows}(M))$$

satisfies all the inequalities (so we skip the requirement on $\mathrm{feas}(\Phi^{\mathbf{B}})$).

Indeed, if (iii), then $\mathrm{feas}(\Phi^{\mathbf{B}})$ is, as we noted after defining interpretations of payoff formulas, the interpretation of $\bigwedge_{(\phi,M)} \phi(\mathrm{rows}(M))$ in $\mathrm{feas}(\mathbf{B})$, which is $\mathscr{M}^{(N)}$ by Proposition 2.13.

On the other hand, every constraint over the set of variables A^N is of the form $\phi(\mathrm{rows}(M))$ for some matrix M. If (ii), then the first type of inequalities ensures that only $\phi(\mathrm{rows}(M))$ with $M \in \mathrm{Mat}(\mathbf{A}, N)$ show up in Φ. By summing up weights and giving weight zero to constraints that do not show up, we obtain Φ as in (iii).

Condition (iii) is equivalent, by definitions, to the following system of linear inequalities with unknowns $w_{\phi,M} \in \mathbb{Q}_0^+$.

$$\forall n \in N \sum_{(\phi,M)} -(\phi^{\mathbf{A}}(\mathrm{col}_n(M)) - c) w_{\phi,M} \le -(\alpha(n) - c)$$
$$\forall f \in \mathscr{M}^{(N)} \sum_{(\phi,M)} (\phi^{\mathbf{B}}(f\mathrm{rows}(M)) - s) w_{\phi,M} \le \beta(f) - s.$$

Note that, since $\mathrm{Mat}(\mathbf{A}, N)$ only contains matrices whose columns are in $\mathrm{feas}(\phi^{\mathbf{A}})$, all the coefficients in the above system of inequalities are finite.

Let F be the coefficient matrix of the system and $\mathbf{q}$ the right-hand side vector. Note that F can be naturally regarded as a rational matrix of type $(N \cup \mathscr{M}^{(N)}) \times \mathrm{Mat}(\mathbf{A}, N)$ (where the union should formally be disjoint). Schematically, the system $F\mathbf{y} \le \mathbf{q}$ is

$$\begin{array}{cc} & (\phi, M) \\ \begin{array}{c} n \\ \\ f \end{array} & \left(\begin{array}{c} \vdots \\ \cdots\cdots -(\phi^{\mathbf{A}}(\mathrm{col}_n(M)) - c) \cdots\cdots \\ \vdots \\ \hline \vdots \\ \cdots\cdots \phi^{\mathbf{B}}(f\mathrm{rows}(M)) - s \cdots\cdots \\ \vdots \end{array}\right) \mathbf{y} \le \left(\begin{array}{c} \vdots \\ -(\alpha(n) - c) \\ \vdots \\ \hline \vdots \\ \beta(f) - s \\ \vdots \end{array}\right) \end{array}$$

By the Farkas' lemma (Theorem 3.1), the above system of inequalities (and then item (ii)) is equivalent to

$$\forall \mathbf{x} \in (\mathbb{Q}_0^+)^{N \cup \mathscr{M}^{(N)}} \ (F^T \mathbf{x} \ge 0 \implies \mathbf{q}^T \mathbf{x} \ge 0). \tag{FE}$$

Vectors $\mathbf{x}$ correspond to pairs of vectors $(\mathbf{x}^{\mathrm{in}} \in (\mathbb{Q}_0^+)^N, \mathbf{x}^{\mathrm{out}} \in (\mathbb{Q}_0^+)^{\mathscr{M}^{(N)}})$ and these can be written as $(\theta^{\mathrm{in}}\Omega^{\mathrm{in}}, \theta^{\mathrm{out}}\Omega^{\mathrm{out}})$ where $\theta^{\mathrm{in}}, \theta^{\mathrm{out}} \in \mathbb{Q}_0^+$, $\Omega^{\mathrm{in}} \in \Delta N$, and $\Omega^{\mathrm{out}} \in \Delta\mathscr{M}^{(N)}$. Condition (FE) is thus

$$\begin{aligned} &\forall \theta^{\mathrm{in}},\ \theta^{\mathrm{out}} \in \mathbb{Q}_0^+ \ \forall \Omega \ N\text{-ary weighting of } \mathscr{M} \\ &(\forall (\phi, M) \in \mathrm{Mat}(\mathbf{A}, N) \ \theta^{\mathrm{out}}(\Omega^{\mathrm{out}}[\phi, M] - s) \\ &\quad \ge \theta^{\mathrm{in}}(\Omega^{\mathrm{in}}[\phi, M] - c)) \\ &\quad \implies \theta^{\mathrm{out}}(\Omega^{\mathrm{out}}[\beta] - s) \ge \theta^{\mathrm{in}}(\Omega^{\mathrm{in}}[\alpha] - c). \end{aligned} \tag{1}$$

For $\theta^{\mathrm{out}} > 0$, the condition is exactly saying that for each $\kappa \in \mathbb{Q}_0^+$ and $\Omega \in \kappa\text{-}\mathrm{Pol}^{(N)}(\mathbf{A}, \mathbf{B}, c, s)$, we have $\Omega^{\mathrm{out}}[\beta] - s \ge \kappa(\Omega^{\mathrm{in}}[\alpha] - c)$, where $\kappa = \theta^{\mathrm{in}}/\theta^{\mathrm{out}}$, which is exactly (i). It remains to observe that (1) is void for $\theta^{\mathrm{out}} = 0$ and $\theta^{\mathrm{in}} > 0$. Indeed, the left-hand side of the implication in (1) holds only if $\mathbf{A} \le c$ (by considering matrices with all the columns equal). In that case we have $\alpha \le c$ by assumptions of the proposition, therefore the right-hand side of the implication holds as well. □

Proposition 3.16 (Characterization of templates). *Let $(\mathbf{A}, \mathbf{B})$ be a pair of valued Σ-structures and $c, s \in \mathbb{Q}$. The following are equivalent.*

(i) *$(\mathbf{A}, \mathbf{B}, c, s)$ is a valued promise template.*
(ii) *For each payoff formula Φ over the set of variables A*
$$\Phi^{\mathbf{A}}(\mathrm{id}_A) \ge cw(\Phi) \implies \exists h \in B^A \ \Phi^{\mathbf{B}}(h) \ge sw(\Phi).$$
(iii) *There exists a unary polymorphism of $(\mathbf{A}, \mathbf{B}, c, s)$.*

Now we state the mentioned more complex version of canonical formula required for the main theorem. The difference is that instead of having one instance Φ as in item (ii) of Proposition 3.15, we simultaneously create multiple instances $(\Phi_j)_{j\in J}$ and allow suitable scaling and shifts. Moreover, in order to slightly simplify our formulation of the valued version of the minor condition problem, we also shift the α by c and β by s.

Proposition 3.17 (Improved canonical payoff formulas). *Let*

- *$(\mathbf{A}, \mathbf{B})$ be a pair of Σ-structures, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A}, \mathbf{B})$, $c, s \in \mathbb{Q}$,*
- *$(N_j)_{j\in J}$ a family of finite sets (arities) with J finite,*
- *$(\alpha_j)_{j\in J}$ a family of functions $\alpha_j : N_j \to \mathbb{Q}$, and*
- *$(\beta_j)_{j\in J}$ a family of functions $\beta_j : \mathscr{M}^{(N_j)} \to \mathbb{Q}$.*

The following are equivalent.

(i) *For each $\kappa \in \mathbb{Q}_0^+$ and for each family $(\Omega_j)_{j\in J}$ of κ-polymorphisms of $(\mathbf{A}, \mathbf{B}, c, s)$,*
$$\kappa \sum_{j\in J} \Omega_j^{\mathrm{in}}[\alpha_j] \ge 0 \implies \sum_{j\in J} \Omega_j^{\mathrm{out}}[\beta_j] \ge 0.$$
(ii) *There exist a family of payoff formulas $(\Phi_j)_{j\in J}$ with Φ_j over the set of variables A^{N_j}, a number $\gamma \in \mathbb{Q}_0^+$ (scaling factor), and families of rational numbers $(\delta_j^{\mathrm{in}}, \delta_j^{\mathrm{out}})_{j\in J}$ (input and output shifts) such that*

$$\begin{aligned} &\forall j \in J \ \forall n \in N_j && \Phi_j^{\mathbf{A}}(\mathrm{proj}_n^{N_j}) - c\, w(\Phi_j) \ge \gamma\alpha_j(n) + \delta_j^{\mathrm{in}} \\ &\forall j \in J \ \forall f \in \mathscr{M}^{(N_j)} && \Phi_j^{\mathbf{B}}(f) - s\, w(\Phi_j) \le \beta_j(f) - \delta_j^{\mathrm{out}} \\ & && \sum_{j\in J} \delta_j^{\mathrm{in}} \ge 0 \\ & && \sum_{j\in J} \delta_j^{\mathrm{out}} \ge 0 \\ &\forall j \in J && \mathrm{feas}(\Phi_j^{\mathbf{B}}) = \mathscr{M}^{(N_j)}. \end{aligned}$$

Moreover, for a fixed $(\mathbf{A}, \mathbf{B})$, if there is an upper bound on the sizes of the N_j then the payoff formulas $(\Phi_j)_{j\in J}$, the scaling factor, and the shifts can be computed from α_j, β_j, N_j (or decided that such formulas do not exist) in polynomial time in the size of the input.

4 VALUED MINION HOMOMORPHISMS AND REDUCTIONS

For the reason explained in the last paragraph of Section 3.3 it is not clear yet what a valued minion should be. For now we choose the most liberal definition, which should be regarded as temporary.

On the other hand, the concept of valued minion homomorphism is quite natural.

Definition 4.1 (Valued minion). Let $\mathscr{M}$ be a minion. A *valued minion over $\mathscr{M}$* is a collection $\mathbb{M} = (\mathbb{M}^{(\mathcal{N})})$ indexed by finite families of finite sets $\mathcal{N} = (N_j)_{j \in J}$ such that elements of $\mathbb{M}^{(\mathcal{N})}$ are families $(\Omega_j)_{j \in J}$ where each Ω_j is an N_j-ary weighting of $\mathscr{M}$.

Note that the collection of plurimorphisms of a valued promise template is a valued minion over the minion of polymorphisms of its feasibility template.

Definition 4.2 (Valued minion homomorphisms). Let $\mathbb{M}, \mathbb{M}'$ be valued minions over minions $\mathscr{M}$ and $\mathscr{M}'$, respectively. A *valued minion homomorphism* $\mathbb{M} \to \mathbb{M}'$ is a probability distribution Ξ on the set of minion homomorphisms $\mathscr{M} \to \mathscr{M}'$ such that, for every finite set J, every family of finite sets $\mathcal{N} = (N_j)_{j \in J}$, and every $(\Omega_j)_{j \in J} \in \mathbb{M}^{(\mathcal{N})}$, we have $(\Xi(\Omega_j))_{j \in J} \in \mathbb{M}'^{(\mathcal{N})}$, where $\Xi(\Omega_j) = (\Omega_j^{\text{in}}, \Xi(\Omega_j^{\text{out}}))$.[13]

We are ready to state the main theorem of this paper.

Theorem 4.3 (Reductions via valued minion homomorphism). *Let $(\mathbf{A}, \mathbf{B}, c, s)$ and $(\mathbf{A}', \mathbf{B}', c', s')$ be valued promise templates such that the former one has a no instance. If there is a valued minion homomorphism from $\mathrm{Plu}(\mathbf{A}, \mathbf{B}, c, s)$ to $\mathrm{Plu}(\mathbf{A}', \mathbf{B}', c', s')$, then $\mathrm{PCSP}(\mathbf{A}', \mathbf{B}', c', s') \leq \mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s)$.*

The proof uses the same strategy as in the crisp case. We introduce a valued version of the minor condition problem (VMC) and prove that each PCSP is equivalent to a VMC. Moreover, valued minion homomorphisms give us reductions between VMCs, cf. the following figure, in which $(\mathbf{A}, \mathbf{B}, c, s)$, $(\mathbf{A}', \mathbf{B}', c', s')$ are valued promise templates, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A}, \mathbf{B})$, $\mathscr{M}' = \mathrm{PolFeas}(\mathbf{A}', \mathbf{B}')$, $\mathbb{M} = \mathrm{Plu}(\mathbf{A}, \mathbf{B}, c, s)$, $\mathbb{M}' = \mathrm{Plu}(\mathbf{A}', \mathbf{B}', c', s')$, and k is sufficiently large.

$$\begin{tikzcd} \mathrm{PCSP}(\mathbf{A}', \mathbf{B}', c', s') \arrow[d, leftrightarrow] & \mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s) \arrow[d, leftrightarrow] \\ \mathrm{VMC}(\mathbb{M}', \mathscr{M}', k) \arrow[r] & \mathrm{VMC}(\mathbb{M}, \mathscr{M}, k) \end{tikzcd}$$

4.1 Valued Minor Conditions

Definition 4.4 (Valued Minor Condition Problem). Given a minion $\mathscr{M}$, a valued minion $\mathbb{M}$ over $\mathscr{M}$, and an integer k, the Valued Minor Condition Problem for $\mathscr{M}$, $\mathbb{M}$, and k, denoted by $\mathrm{VMC}(\mathscr{M}, \mathbb{M}, k)$, is the following problem.

Input
1. disjoint sets U and V (the sets of *variables*),
2. a set D_x with $|D_x| \leq k$ for every $x \in U \cup V$ (the *domain* of x),
3. a set of formal expressions of the form $\pi(u) = v$, where $u \in U, v \in V$, and $\pi : D_u \to D_v$ (the *minor conditions*),
4. for each $u \in U$, a pair of functions $\alpha_u : D_u \to \mathbb{Q}$, $\beta_u : \mathscr{M}^{(D_u)} \to \mathbb{Q}$ (the *input and output payoff functions*) which satisfy the following condition.

$$\forall (\Omega_u)_{u \in U} \in \mathbb{M}^{(D_u)_{u \in U}}$$
$$\sum_{u \in U} \Omega_u^{\text{in}}[\alpha_u] \geq 0 \implies \sum_{u \in U} \Omega_u^{\text{out}}[\beta_u] \geq 0. \quad (\star)$$

Output yes if there exists a function h from $U \cup V$ with $h(x) \in D_x$ (for each $x \in U \cup V$) such that, for each minor condition $\pi(u) = v$, we have $\pi(h(u)) = h(v)$, and $\sum_{u \in U} \alpha_u(h(u)) \geq 0$.

no if there does not exist a function h from $U \cup V$ with $h(x) \in \mathscr{M}^{(D_x)}$ such that, for each minor condition $\pi(u) = v$, we have $\mathscr{M}^{(\pi)}(h(u)) = h(v)$, and $\sum_{u \in U} \beta_u(h(u)) \geq 0$.

Note that unlike in the crisp case, the VMC is not a PCSP over a valued promise template, at least not in an obvious way. We also remark that, because of our temporary, too liberal definition of valued minions, the VMC does not need to make sense – the sets of yes and no instances can intersect. We show in the full version [6] that the VMC makes sense for plurimorphism minions of valued promise templates.

The proof of Theorem 4.3 is based on three reductions: from PCSP to VMC, from VMC to PCSP, and between VMCs. The last one is the simplest.

Proposition 4.5 (Between VMCs). *Let $\mathbb{M}, \mathbb{M}'$ be valued minions over minions $\mathscr{M}$ and $\mathscr{M}'$, respectively, such that there exists a valued minion homomorphism $\mathbb{M} \to \mathbb{M}'$. Then $\mathrm{VMC}(\mathscr{M}', \mathbb{M}', k) \leq \mathrm{VMC}(\mathscr{M}, \mathbb{M}, k)$ for any positive integer k.*

Proof sketch. For an instance of $\mathrm{VMC}(\mathscr{M}', \mathbb{M}', k)$ we produce an instance of $\mathrm{VMC}(\mathscr{M}, \mathbb{M}, k)$ that is unchanged except for applying Ξ to the output payoff functions, that is, for each $u \in U$ we define $\beta_u = \mathbb{E}_{\xi \sim \Xi} \beta'_u \circ \xi^{(D_u)}$, where β'_u is the output payoff function for u in the original instance. The correctness of this reduction is verified in the full version [6]. ◻

4.2 From PCSP to VMC

The following object is useful for the reduction from a PCSP to VMC (in the crisp setting as well).

Definition 4.6 (Canonical matrix). Let $\psi \subseteq A^Z$ be a relation. The *canonical matrix* for ψ is the matrix $\mathrm{CM}[\psi] \in A^{Z \times \psi}$ defined by

$$\mathrm{CM}[\psi](z, \mathbf{a}) = \mathbf{a}(z) \quad \text{for every } z \in Z,\ \mathbf{a} \in \psi.$$

Proposition 4.7 (From PCSP to VMC). *Let $(\mathbf{A}, \mathbf{B}, c, s)$ be a valued promise template, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A}, \mathbf{B})$, and $\mathbb{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B}, c, s)$. If k is a sufficiently large integer, then we have $\mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s) \leq \mathrm{VMC}(\mathscr{M}, \mathbb{M}, k)$.*

Proof sketch. From a payoff formula $\Phi = \sum_{i \in I} w_i \phi_i(\mathbf{x}_i)$ over X (where $w_i \in \mathbb{Q}_0^+$), we create an instance of $\mathrm{VMC}(\mathscr{M}, \mathbb{M}, k)$ as follows. The sets U, V, D_x and minor conditions are created by applying the reduction detailed in the full version [6] to the crisp template $(\mathrm{feas}(\mathbf{A}), \mathrm{feas}(\mathbf{B}))$ and conjunctive formula $\bigwedge_{i \in I} \phi_i(\mathbf{x}_i)$. That is, for a large enough k we define these objects as follows.

1. $U = I$, $V = X$.
2. $D_i = \mathrm{feas}(\phi_i^{\mathbf{A}})$, $D_x = A_{\mathrm{sort}(x)}$ for each $i \in I, x \in X$.

[13] Recall here that $\Xi(\Omega_j^{\text{out}})$ is the probability distribution that it sampled by sampling $\xi \sim \Xi$, sampling $f \sim \Omega_j^{\text{out}}$, and computing $\xi(f)$.

3. For each $i \in I$ and $z \in \mathrm{ar}(\phi_i)$, introduce the constraint $\pi_{i,z}(i) = \mathbf{x}_i(z)$, where $\pi_{i,z}$ is the domain-codomain restriction of $\mathrm{proj}_z^{\mathrm{ar}(\phi_i)}$ to $\mathrm{feas}(\phi_i^{\mathbf{A}})$ and $A_{\mathrm{sort}(z)}$.

The input and output payoff functions are defined as follows.

4. $\alpha_i(\mathbf{a}) = w_i(\phi_i^{\mathbf{A}}(\mathbf{a}) - c)$,
$\beta_i(f) = w_i(\phi_i^{\mathbf{B}}(f\ \mathrm{rows}(\mathrm{CM}[\mathrm{feas}(\phi_i^{\mathbf{A}})])) - s)$
for each $i \in I$, $\mathbf{a} \in \mathrm{feas}(\phi_i^{\mathbf{A}})$, and $f \in \mathscr{M}^{(D_i)}$.

For this to be a valid instance, we require $k \geq |\mathrm{feas}(\phi^{\mathbf{A}})|$ and $k \geq |A_{\mathrm{sort}(x)}|$ for every $\phi \in \sigma$ and $x \in X$.

We need to verify condition ($\star$). In fact, a stronger condition holds:

$$\forall \kappa \in \mathbb{Q}_0^+\ \forall i \in I\ \forall \Omega \in \kappa\text{-}\mathrm{Pol}^{(D_i)}(\mathbf{A}, \mathbf{B})\ \Omega^{\mathrm{out}}[\beta_i] \geq \kappa \Omega^{\mathrm{in}}[\alpha_i] \quad (\star\star)$$

Notice that condition ($\star\star$) is indeed stronger than ($\star$): assuming ($\star\star$), $(\Omega_i)_{i \in I} \in \mathbb{M}^{(D_i)_{i\in I}}$, and $\sum_{i \in I} \Omega_i^{\mathrm{in}}[\alpha_i] \geq 0$, we obtain $\sum_{i \in I} \Omega_i^{\mathrm{out}}[\beta_i] \geq \sum_{i \in I} \kappa \Omega_i^{\mathrm{in}}[\alpha_i] \geq 0$, as required.

Condition ($\star\star$) and the soundness and completeness of this reduction are verified in the full version [6]. ◻

It would be possible to make a version of VMC based on ($\star\star$), κ-polymorphisms for various κ, and appropriately defined valued minions and homomorphism (and this is in fact what is done for the constant factor approximation setting in [6] for a fixed κ). The chosen version, albeit more complicated because of the concept of plurimorphisms, requires less information about κ-polymorphisms and gives a stronger reduction result.

4.3 From VMC TO PCSP

The idea of the reduction from VMC to PCSP is similar to the proof in the crisp case (cf. [6]) but we use improved canonical payoff formulas (Proposition 3.17) instead of canonical formulas (Proposition 2.13). A technical issue is that condition ($\star$) only guarantees condition (i) in Proposition 3.17 for $\kappa > 0$. This causes a slight complication.

Proposition 4.8 (From VMC to PCSP). *Let $(\mathbf{A}, \mathbf{B}, c, s)$ be a valued promise template such that $\mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s)$ has a no instance, $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A}, \mathbf{B})$, and $\mathbb{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B}, c, s)$. For any positive integer k, $\mathrm{VMC}(\mathscr{M}, \mathbb{M}, k) \leq \mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s)$.*

Proof sketch. Consider a VMC instance as described in Definition 4.4.

We try to find a collection of payoff formulas $(\Phi_u)_{u \in U}$ and rationals $\gamma \geq 0$, δ_u^{in}, δ_u^{out} such that all the properties in item (ii) of Proposition 3.17 are satisfied (where $J = U$ and $N_u = D_u$). By that proposition, one can find such a collection or decide that it does not exist, in polynomial time.

If we found such formulas, then the reduction is done very much like in the crisp case, as follows.

In the first step we also define a payoff formula Φ_v for each $v \in V$ as $\Phi_v = \sum_{(\phi, M) \in \mathrm{Mat}(\mathbf{A}, D_v)} 0 \cdot \phi(\mathrm{rows}(M))$ over the set of variables A^{D_v}, i.e., all the constraints are given zero weight. Now we have a payoff formula Φ_x over the set of variables A^{D_x} for every $x \in U \cup V$. We make the variable sets disjoint and define Φ as the sum of all the Φ_x, so Φ is a payoff formula over a set of variables Y which is a disjoint union of A^{D_x}. Note that assignments $f : Y \to A$ correspond exactly to collections $(f_x : A^{D_x} \to A)_{x \in X}$, and similarly for B.

In the second step, we create from Φ the resulting instance Ψ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B}, c, s)$ by identifying, for each minor condition $\pi(u) = v$ and each $\mathbf{a} \in A^{D_v}$, the variables $(\mathbf{a}\pi, u)$ and $(\mathbf{a}, v)$. Now assignments for Ψ from the new set of variables to A correspond to those that satisfy $f_u^{(\pi)} = f_v$ for each minor condition $\pi(u) = v$, and an analogous observation holds for assignments to B.

Properties in (ii) guarantee the completeness and soundness of this reduction; details are in the full version [6], where we also deal with the case that formulas Φ_u do not exist (which is the reason for the slightly unpleasant assumption that no instances exist). ◻

5 EXAMPLES OF HOMOMORPHISMS

Examples of valued minion homomorphisms of course include minion homomorphisms for crisp templates. More precisely, if $(\mathbf{A}, \mathbf{B}, 0, 0)$ and $(\mathbf{A}', \mathbf{B}', 0, 0)$ are templates such that all symbols in all four structures are interpreted as $(-\infty, 0)$-valued relations and ξ is a minion homomorphism from $\mathrm{PolFeas}(\mathbf{A}, \mathbf{B})$ to $\mathrm{PolFeas}(\mathbf{A}', \mathbf{B}')$, then the probability distribution Ξ that assigns probability one to ξ is a valued minion homomorphism from $\mathrm{Plu}(\mathbf{A}, \mathbf{B}, 0, 0)$ to $\mathrm{Plu}(\mathbf{A}', \mathbf{B}', 0, 0)$.

Recall that the reduction theorem fully explains hardness for crisp CSPs (but not for crisp PCSPs [4]). In this section we discuss two types of situations in which a reduction is (or is not) explained by the reduction theorem in the valued setting.

5.1 Gadget reductions

We start with a very simple example of a gadget reduction.

Example 5.1 (3LIN2$(c, s) \leq$ 5LIN2(c, s), $c \geq s$). Rename relational symbols for 3LIN2(c, s) (see Example 2.6) to ϕ_0', ϕ_1' to distinguish them from relational symbols for 5LIN2(c, s). The reduction is: replace every constraint $\phi_i'(x_1, x_2, x_3)$ in the input payoff formula by $\phi_i(x_1, x_2, x_3, x_3, x_3)$.

In the example, the gadget for $\phi_i'(x_1, x_2, x_3)$ is the payoff formula $\phi_i(x_1, x_2, x_3, x_3, x_3)$. More generally, one can replace each constraint by an arbitrary payoff formula, possibly introducing additional variables. The following observation formulates (simplified) natural conditions under which such a gadget replacement is a reduction; moreover, the reduction is explained by the reduction theorem, via particularly strong homomorphisms.

Proposition 5.2 (Gadgets and homomorphisms). *Let $(\mathbf{A}, \mathbf{B}, c, s)$ and $(\mathbf{A}', \mathbf{B}', c', s')$ be valued promise templates in signatures Σ, Σ' such that $\mathrm{PolFeas}(\mathbf{A}, \mathbf{B}) = \mathrm{PolFeas}(\mathbf{A}', \mathbf{B}')$. Suppose that for every $\phi \in \Sigma'$ of arity $Z = \mathrm{ar}(\phi)$ there exists a payoff Σ-formula Ψ over the set of variables $Z \cup Y$, where Y is a finite set disjoint from Z, such that the following conditions hold.*

- $\forall \mathbf{a}_Z \in A^Z\ \exists \mathbf{a}_Y \in A^Y\ \Psi^{\mathbf{A}}(\mathbf{a}_Z, \mathbf{a}_Y) - c\, w(\Psi) \geq \phi^{\mathbf{A}'}(\mathbf{a}_Z) - c'$
- $\forall \mathbf{b}_Z \in B^Z\ \forall \mathbf{b}_Y \in B^Y\ \Psi^{\mathbf{B}}(\mathbf{b}_Z, \mathbf{b}_Y) - s\, w(\Psi) \leq \phi^{\mathbf{B}'}(\mathbf{b}_Z) - s'$

Then, for every $\kappa \in \mathbb{Q}_0^+$, every κ-polymorphism of $(\mathbf{A}, \mathbf{B}, c, s)$ is also a κ-polymorphism of $(\mathbf{A}', \mathbf{B}', c', s')$. In particular, Ξ with probability one on the identity is a valued minion homomorphism from $\mathrm{Plu}(\mathbf{A}, \mathbf{B}, c, s)$ to $\mathrm{Plu}(\mathbf{A}', \mathbf{B}', c', s')$.

The question when payoff Σ-formulas Ψ in Proposition 5.2 exist (and, more generally, understanding of more complex gadget reductions) is closely related to questions about definability — a direction we leave for future work.

Sometimes a reduction can be obtained by replacing constraints by gadgets as above and merging some of the additional variables, cf. [11]. The following is an example that can be explained by the reduction theorem (see [6]).

*Example 5.3 (*3LIN2$(c,s) \leq$ 4LIN2(c,s), $c \geq s$*).* We replace every constraint $\phi_i(x_1,x_2,x_3)$ by $\phi_i(x_1,x_2,x_3,z)$, where z is a fresh variable common to all the constraints. The reduction works since if an assignment for the new instance assigns 0 to z, then forgetting z gives an assignment for the original instance with the same payoff; and if z is assigned 1, then we additionally flip the values $0 \leftrightarrow 1$.

We do not have a satisfactory general explanation of the previous example. In fact, we do not think it is possible to provide it using our version of the reduction theorem. This restraint stems from the following example.

*Example 5.4 (*3LIN2 *and* 3LIN2$(1,1)$*).* The 3LIN2 problem can be seen as $(-\infty, 0)$-valued. Then, 3LIN2 and 3LIN2$(1,1)$ are equivalent problems. Nevertheless, there is even no minion homomorphism from PolFeas(3LIN2$(1,1)$) (which consists of all Boolean functions) to PolFeas(3LIN2) (which consists of parity functions depending on odd number of coordinates), since e.g. every minion homomorphism maps commutative binary operations (which the first minion has) to commutative binary operations (which the second minion does not have).

5.2 Homomorphisms to Gap Label Cover

Recall from Example 3.7 that for every $1 \geq \epsilon > 0$, there exist D, E such that $\mathrm{GLC}_{D,E}(1,\epsilon)$ is NP-hard. This result is a starting point for many inapproximability results, including those in an influential paper by Håstad [35].

The following proposition gives a sufficient condition for a reduction from the Gap Label Cover, in particular, it isolates the core of Håstad's results. The reduction and its correctness is simple (similar to the reduction from MC to PCSP in the crisp case, cf. [6]) and not needed in this paper, so we leave it to the reader.

The statement uses instances over the set of variables $A^D \cup A^E$. Note that assignments $A^D \cup A^E \to A'$ exactly correspond to pairs of functions $(f_D : A^D \to A', f_E : A^E \to A')$ and we write such assignments in this way.

Proposition 5.5 (Reductions from GLC). *Let* $(\mathbf{A},\mathbf{B},c,s)$ *be a valued promise template,* $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$*, D and E finite disjoint sets, and $\epsilon \in \mathbb{R}$. Suppose that there exist*

- *mappings* $\Lambda_D : \mathscr{M}^{(D)} \to \Delta D$ *and* $\Lambda_E : \mathscr{M}^{(E)} \to \Delta E$,
- *for every* $\pi : D \to E$ *a normalized payoff formula* Φ_π *over the set of variables* $A^D \cup A^E$*, and a linear nondecreasing function* $\gamma_\pi : \mathbb{R} \to \mathbb{R}$ *with* $\gamma_\pi(s) \geq \epsilon$

such that for every $\pi : D \to E$

1. $\Phi^{\mathbf{A}}_\pi(\mathrm{proj}^D_d, \mathrm{proj}^E_{\pi(d)}) \geq c$ *for every* $d \in D$*, and*
2. $\gamma_\pi(\Phi^{\mathbf{B}}_\pi(f_D,f_E)) \leq \mathbb{E}_{\substack{d\sim\Lambda_D(f_D)\\ e\sim\Lambda_E(f_E)}} \pi(d,e)$ *for every* $f_D \in \mathscr{M}^{(D)}$, $f_E \in \mathscr{M}^{(E)}$.

Then $\mathrm{GLC}_{D,E}(1,\epsilon) \leq \mathrm{PCSP}(\mathbf{A},\mathbf{B},c,s)$.

*Example 5.6 (*3LIN2$(1-\delta, 1/2+\delta)$*).* The first inapproximability result in [35] follows from the fact that for each $1/4 \geq \delta > 0$, there exists ϵ (namely $16\delta^3$) such that the template 3LIN2$(1-\delta,1/2+\delta)$ satisfies the conditions of Proposition 5.5 for every D and E.

The mapping Λ_D (and similarly Λ_E) is a composition of two mappings. The first one is "folding" from $\mathscr{M}^{(D)}$ to the set $\mathscr{F}$ of *folded* functions, i.e., those satisfying $f(\mathbf{a}) = 1 - f(1-\mathbf{a})$ (where $(1-\mathbf{a})(z) = 1 - \mathbf{a}(z)$ at each coordinate z). The second one assigns to $f \in \mathscr{F}$ a probability distribution on D based on the size of the Fourier coefficients of f. The definition of Φ_π is according to the "long code test" in [35]. Details are worked out in the full version [6].

We will show in Theorem 5.8 that the reduction in Proposition 5.5 is explained by a valued minion homomorphism. But first we spell out and somewhat simplify the condition for homomorphisms into the plurimorphisms of $\mathrm{GLC}_{D,E}(1,\epsilon)$. The simplification is that we only need to consider polymorphisms, not plurimorphisms; it follows from the proof that this is always the case when considering homomorphisms to PCSPs with perfect completeness.

Note that N-ary functions in $\mathrm{PolFeas}(\mathrm{GLC}_{D,E}(1,\epsilon))$ correspond exactly to pairs $(p_D : D^N \to D, p_E : E^N \to E)$, and every pair correspond to such a function since all tuples are feasible.

Proposition 5.7 (Simplified homomorphisms to GLC). *Let* $(\mathbf{A},\mathbf{B},c,s)$ *be a valued promise template,* $\mathscr{M} = \mathrm{PolFeas}(\mathbf{A},\mathbf{B})$*, D, E finite sets, $\epsilon \in \mathbb{R}$, and Ξ a probability distribution on the set of minion homomorphisms* $\mathscr{M} \to \mathrm{PolFeas}(\mathrm{GLC}_{D,E}(1,\epsilon))$*. The following are equivalent.*

(i) Ξ *is a valued minion homomorphism from* $\mathrm{Plu}(\mathbf{A},\mathbf{B},c,s)$ *to* $\mathrm{Plu}(\mathrm{GLC}_{D,E}(1,\epsilon))$.
(ii) *For every* $N \in \mathrm{FinSet}$, $\Omega \in \mathrm{Pol}^{(N)}(\mathbf{A},\mathbf{B},c,s)$, $\pi : D \to E$, $\mathbf{d} \in D^N$*, and* $\mathbf{e} \in E^N$

$$\forall n \in \mathrm{Supp}(\Omega^{\mathrm{in}})\ \pi(\mathbf{d}(n)) = \mathbf{e}(n) \quad \Rightarrow \mathop{\mathbb{E}}_{(p_D,p_E)\sim\Xi(\Omega^{\mathrm{out}})} \pi(p_D(\mathbf{d}), p_E(\mathbf{e})) \geq \epsilon.$$

Theorem 5.8 (Reductions from GLC via homomorphisms). *Under the assumptions of Proposition 5.5, there is a valued minion homomorphism from* $\mathrm{Plu}(\mathbf{A},\mathbf{B},c,s)$ *to* $\mathrm{Plu}(\mathrm{GLC}_{D,E}(1,\epsilon))$.

Proof sketch. Let $\Lambda_D, \Lambda_E, \Phi_\pi, \gamma_\pi$ be as in Proposition 5.5. We start by defining a probability distribution Ξ on the set of minion homomorphisms $\mathscr{M} \to \mathrm{PolFeas}(\mathrm{GLC}_{D,E}(1,\epsilon))$. A minion homomorphism ξ is sampled from Ξ as follows.

- Pick $\lambda_D : \mathscr{M}^{(D)} \to D$ by sampling $\lambda_D(f) \in D$ according to $\Lambda_D(f)$, independently for each $f \in \mathscr{M}^{(D)}$.
- Pick $\lambda_E : \mathscr{M}^{(E)} \to E$ similarly using $\Lambda_E(f)$.
- Define ξ by $\xi^{(N)}(f) = (\xi^{(N)}_D(f), \xi^{(N)}_E(f))$ for every $N \in$ FinSet and $f \in \mathscr{M}^{(N)}$, where for $\mathbf{d} \in D^N$, $\mathbf{e} \in E^N$ we define

$$\xi^{(N)}_D(f)(\mathbf{d}) = \lambda_D(f^{(\mathbf{d})}), \quad \xi^{(N)}_E(f)(\mathbf{e}) = \lambda_E(f^{(\mathbf{e})}).$$

The verification that ξ preserves minors and that Ξ is a valued minion homomorphism is in the full version [6].

We remark that the construction of ξ from $\lambda = (\lambda_D, \lambda_E)$ is not ad hoc: every minion homomorphism ξ can be constructed in this way. We refer to [4, Lemma 4.4] for details in the one-sorted case. ◻

6 CONCLUSION

Our main result, Theorem 4.3, shows that computational complexity is determined by symmetries in the vast framework of valued PCSPs. We see this result as a step towards the general goal of providing uniform descriptions of algorithms, tractability boundaries, and reductions.

Crisp non-promise CSPs already include many important combinatorial problems. Valued PCSPs generalize this framework in two directions: towards approximation (promises) and optimization (values). A further vast enlargement in the combinatorial direction would be provided by incorporating interesting classes of infinite structures. In fact, [22, 53] already contribute to this project in the constant factor setting.

There are still many basic questions and theory-building tasks left open already for finite-domain valued PCSPs: to incorporate the trivial reduction as in e.g. Example 5.4; to characterize gadget reductions (or versions of definability) in terms of polymorphisms or plurimorphisms; to characterize plurimorphism valued minions of templates; to clarify whether plurimorphisms are necessary to determine computational complexity or enough information is provided already by polymorphisms; to develop methods for proving nonexistence of homomorphisms; to revisit the valued CSP dichotomy without fixed threshold [43] and Raghavendra's result on unique games hardness of approximation for all MaxCSPs [48]; among others. An interesting special case for a full complexity classification is the valued non-promise CSPs with fixed threshold.

The most exciting (and likely challenging) research goal to us is to improve the reduction theorem so that it explains the PCP theorem [31] (hardness of Gap Label Cover) or even the Unique Games Conjecture [40] (hardness of Unique Games), or some special cases such as the 2-to-2 Conjecture (now theorem [41]). Crucially, both Gap Label Cover and Unique Games are within our framework, and we can now thus at least specify the aim: to weaken the concept of valued minion homomorphism so that, e.g., plurimorphisms of $\mathrm{GLC}_{D,E}(1,\epsilon)$ homomorphically map to the projection minion. A reason for cautious optimism is the recent "Baby PCP" paper [9] that contributes to this effort in the crisp setting.

REFERENCES

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. 1998. Proof Verification and the Hardness of Approximation Problems. *J. ACM* 45, 3 (1998), 501–555. https://doi.org/10.1145/278298.278306

[2] Sanjeev Arora and Shmuel Safra. 1998. Probabilistic Checking of Proofs: A New Characterization of NP. *J. ACM* 45, 1 (1998), 70–122. https://doi.org/10.1145/273865.273901

[3] Per Austrin, Venkatesan Guruswami, and Johan Håstad. 2017. $(2+\epsilon)$-Sat Is NP-hard. *SIAM J. Comput.* 46, 5 (2017), 1554–1573. https://doi.org/10.1137/15M1006507

[4] Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. 2021. Algebraic Approach to Promise Constraint Satisfaction. *J. ACM* 68, 4 (2021), 28:1–28:66. https://doi.org/10.1145/3457606

[5] Libor Barto and Silvia Butti. 2022. Weisfeiler-Leman Invariant Promise Valued CSPs. In *Proc. 28th International Conference on Principles and Practice of Constraint Programming (CP'22) (LIPIcs, Vol. 235)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 4:1–4:17. https://doi.org/10.4230/LIPIcs.CP.2022.4

[6] Libor Barto, Silvia Butti, Alexandr Kazda, Caterina Viola, and Stanislav Živný. 2024. *Algebraic Approach to Approximation.* Technical Report. arXiv:2401.15186

[7] Libor Barto and Marcin Kozik. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM* 61, 1 (2014). https://doi.org/10.1145/2556646 Article No. 3.

[8] Libor Barto and Marcin Kozik. 2016. Robustly Solvable Constraint Satisfaction Problems. *SIAM J. Comput.* 45, 4 (2016), 1646–1669. https://doi.org/10.1137/130915479 arXiv:1512.01157

[9] Libor Barto and Marcin Kozik. 2022. Combinatorial Gap Theorem and Reductions between Promise CSPs. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. SIAM, 1204–1220. https://doi.org/10.1137/1.9781611977073.50 arXiv:2107.09423

[10] Libor Barto, Jakub Opršal, and Michael Pinsker. 2018. The wonderland of reflections. *Isr. J. Math* 223, 1 (Feb 2018), 363–398. https://doi.org/10.1007/s11856-017-1621-9 arXiv:1510.04521

[11] Mihir Bellare, Oded Goldreich, and Madhu Sudan. 1998. Free Bits, PCPs, and Nonapproximability—Towards Tight Results. *SIAM J. Comput.* 27, 3 (1998), 804–915. https://doi.org/10.1137/S0097539796302531

[12] Joel Berman, Pawel Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2010. Varieties with few subalgebras of powers. *Trans. Am. Math. Soc.* 362, 3 (2010), 1445–1473.

[13] Amey Bhangale, Subhash Khot, and Dor Minzer. 2022. On Approximability of Satisfiable k-CSPs: I.. In *Proc. 54th Annual ACM Symposium on Theory of Computing (STOC'22)*. ACM, 976–988. https://doi.org/10.1145/3519935.3520028

[14] Amey Bhangale, Subhash Khot, and Dor Minzer. 2023. On Approximability of Satisfiable k-CSPs: II. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 632–642. https://doi.org/10.1145/3564246.3585120

[15] Amey Bhangale, Subhash Khot, and Dor Minzer. 2023. On Approximability of Satisfiable k-CSPs: III. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 643–655. https://doi.org/10.1145/3564246.3585121

[16] Manuel Bodirsky. 2021. *Complexity of infinite-domain constraint satisfaction.* Vol. 52. Cambridge University Press.

[17] Manuel Bodirsky and Martin Grohe. 2008. Non-dichotomies in Constraint Satisfaction Complexity. In *Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP'08) (Lecture Notes in Computer Science, Vol. 5126)*. Springer, 184–196. https://doi.org/10.1007/978-3-540-70583-3_16

[18] Manuel Bodirsky and Jan Kára. 2010. The complexity of temporal constraint satisfaction problems. *J. ACM* 57, 2 (2010), 9:1–9:41. https://doi.org/10.1145/1667053.1667058

[19] Manuel Bodirsky, Barnaby Martin, and Antoine Mottet. 2018. Discrete Temporal Constraint Satisfaction Problems. *J. ACM* 65, 2 (2018), 9:1–9:41. https://doi.org/10.1145/3154832 arXiv:1503.08572

[20] Manuel Bodirsky and Michael Pinsker. 2015. Schaefer's Theorem for Graphs. *J. ACM* 62, 3 (2015), 19:1–19:52. https://doi.org/10.1145/2764899

[21] Manuel Bodirsky and Jakub Rydval. 2023. On the Descriptive Complexity of Temporal Constraint Satisfaction Problems. *J. ACM* 70, 1 (2023), 2:1–2:58. https://doi.org/10.1145/3566051

[22] Manuel Bodirsky, Žaneta Semanišinová, and Carsten Lutz. 2023. The Complexity of Resilience Problems via Valued Constraint Satisfaction Problems. abs/2309.15654 (2023). https://doi.org/10.48550/ARXIV.2309.15654 arXiv:2309.15654

[23] Joshua Brakensiek and Venkatesan Guruswami. 2021. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.* 50, 6 (2021), 1663–1700. https://doi.org/10.1137/19M128212X arXiv:1704.01937

[24] Jonah Brown-Cohen and Prasad Raghavendra. 2015. Combinatorial Optimization Algorithms via Polymorphisms. *CoRR* abs/1501.01598 (2015). arXiv:1501.01598 http://arxiv.org/abs/1501.01598

[25] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM J. Comput.* 34, 3 (2005), 720–742. https://doi.org/10.1137/S0097539700376676

[26] Andrei A. Bulatov. 2017. A Dichotomy Theorem for Nonuniform CSPs. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*. 319–330. https://doi.org/10.1109/FOCS.2017.37 arXiv:1703.03021

[27] Siu On Chan. 2016. Approximation Resistance from Pairwise-Independent Subgroups. *J. ACM* 63, 3 (2016), 27:1–27:32. https://doi.org/10.1145/2873054

[28] Lorenzo Ciardo and Stanislav Živný. 2023. CLAP: A New Algorithm for Promise CSPs. *SIAM J. Comput.* 52, 1 (2023), 1–37. https://doi.org/10.1137/22M1476435 arXiv:2107.05018

[29] David A. Cohen, Martin C. Cooper, Páidí Creed, Peter G. Jeavons, and Stanislav Živný. 2013. An Algebraic Theory of Complexity for Discrete Optimization. *SIAM J. Comput.* 42, 5 (2013), 1915–1939. https://doi.org/10.1137/130906398

[30] David A. Cohen, Martin C. Cooper, Peter Jeavons, and Andrei A. Krokhin. 2006. The complexity of soft constraint satisfaction. *Artif. Intell.* 170, 11 (2006), 983–1016. https://doi.org/10.1016/J.ARTINT.2006.04.002

[31] Irit Dinur. 2007. The PCP theorem by gap amplification. *J. ACM* 54, 3 (2007), 12. https://doi.org/10.1145/1236457.1236459

[32] Tomás Feder and Moshe Y. Vardi. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.* 28, 1 (1998), 57–104. https://doi.org/10.1137/S0097539794266766

[33] Miron Ficak, Marcin Kozik, Miroslav Olšák, and Szymon Stankiewicz. 2019. Dichotomy for Symmetric Boolean PCSPs. In *Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)*, Vol. 132. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 57:1–57:12. https://doi.org/10.4230/LIPIcs.ICALP.2019.57 arXiv:1904.12424

[34] Pierre Gillibert, Julius Jonusas, Michael Kompatscher, Antoine Mottet, and Michael Pinsker. 2022. When Symmetries Are Not Enough: A Hierarchy of Hard Constraint Satisfaction Problems. *SIAM J. Comput.* 51, 2 (2022), 175–213. https://doi.org/10.1137/20M1383471

[35] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (2001), 798–859. https://doi.org/10.1145/502090.502098

[36] Yahli Hecht, Dor Minzer, and Muli Safra. 2023. NP-Hardness of Almost Coloring Almost 3-Colorable Graphs. In *Proc. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'23) (LIPIcs, Vol. 275)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 51:1–51:12. https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2023.51

[37] Pawel M. Idziak, Petar Markovic, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2010. Tractability and Learnability Arising from Algebras with Few Subpowers. *SIAM J. Comput.* 39, 7 (2010), 3023–3037. https://doi.org/10.1137/090775646

[38] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. 1997. Closure Properties of Constraints. *J. ACM* 44, 4 (1997), 527–548. https://doi.org/10.1145/263867.263489

[39] Alexander Kazda. 2021. Minion homomorphisms give reductions between promise valued CSPs. (2021). Unpublished manuscript.

[40] Subhash Khot. 2002. On the power of unique 2-prover 1-round games. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*. ACM, 767–775. https://doi.org/10.1145/509907.510017

[41] Subhash Khot, Dor Minzer, and Muli Safra. 2023. Pseudorandom sets in Grassmann graph have near-perfect expansion. *Ann. Math.* 198, 1 (2023), 1 – 92. https://doi.org/10.4007/annals.2023.198.1.1

[42] Subhash Khot and Oded Regev. 2008. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.* 74, 3 (2008), 335–349. https://doi.org/10.1016/J.JCSS.2007.06.019

[43] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. 2017. The Complexity of General-Valued CSPs. *SIAM J. Comput.* 46, 3 (2017), 1087–1110. https://doi.org/10.1137/16M1091836 arXiv:1502.07327

[44] Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. 2015. The power of linear programming for general-valued CSPs. *SIAM J. Comput.* 44, 1 (2015), 1–36. https://doi.org/10.1137/130945648 arXiv:1311.4219

[45] Marcin Kozik and Joanna Ochremiak. 2015. Algebraic Properties of Valued Constraint Satisfaction Problem. In *Proc. 42nd International Colloquium on Automata, Languages, and Programming (ICALP'15) (Lecture Notes in Computer Science, Vol. 9134)*. Springer, 846–858. https://doi.org/10.1007/978-3-662-47672-7_69

[46] Tamio-Vesa Nakajima and Stanislav Živný. 2023. *Maximum k- vs. l-colourings of graphs*. Technical Report. arXiv:2311.00440

[47] Nicholas Pippenger. 2002. Galois theory for minors of finite functions. *Discrete Mathematics* 254, 1-3 (2002), 405–419.

[48] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP?. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC'08)*. ACM, 245–254. https://doi.org/10.1145/1374376.1374414

[49] Ran Raz. 1998. A Parallel Repetition Theorem. *SIAM J. Comput.* 27, 3 (1998), 763–803. https://doi.org/10.1137/S0097539795280895

[50] Alexander Schrijver. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., USA.

[51] Johan Thapper and Stanislav Živný. 2016. The Complexity of Finite-Valued CSPs. *J. ACM* 63, 4 (2016), 37:1–37:33. https://doi.org/10.1145/2974019 arXiv:1210.2987

[52] Johan Thapper and Stanislav Živný. 2017. The power of Sherali–Adams relaxations for general-valued CSPs. *SIAM J. Comput.* 46, 4 (2017), 1241–1279. https://doi.org/10.1137/16M1079245 arXiv:1606.02577

[53] Caterina Viola and Stanislav Živný. 2021. The Combined Basic LP and Affine IP Relaxation for Promise VCSPs on Infinite Domains. *ACM Trans. Algorithms* 17, 3 (2021), 21:1–21:23. https://doi.org/10.1145/3458041

[54] Dmitriy Zhuk. 2020. A Proof of the CSP Dichotomy Conjecture. *J. ACM* 67, 5 (2020), 30:1–30:78. https://doi.org/10.1145/3402029 arXiv:1704.01914