Electronic Colloquium on Computational Complexity, Report No. 59 (2018)

Combining LPs and Ring Equations via Structured Polymorphisms

Joshua Brakensiek* Venkatesan Guruswami†

Abstract

Promise CSPs are a relaxation of constraint satisfaction problems where the goal is to find an assignment satisfying a relaxed version of the constraints. Several well known problems can be cast as promise CSPs including approximate graph and hypergraph coloring, discrepancy minimization, and interesting variants of satisfiability. Similar to CSPs, the tractability of promise CSPs can be tied to the structure of associated operations on the solution space called (weak) polymorphisms. However, compared to CSPs whose polymorphisms are well-structured algebraic objects called clones, weak polymorphisms in the promise world are much less constrained — essentially any infinite family of functions obeying mild conditions can arise as weak polymorphisms. Under the thesis that non-trivial polymorphisms govern tractability, promise CSPs therefore provide a fertile ground for the discovery of novel algorithms.

In previous work, we classified all tractable cases of Boolean promise CSPs when the constraint predicates are symmetric. The algorithms were governed by three kinds of polymorphism families: (i) parity functions, (ii) majority functions, or (iii) a non-symmetric (albeit block-symmetric) family we called alternating threshold. In this work, we provide a vast generalization of these algorithmic results. Specifically, we show that promise CSPs that admit a family of "regional periodic" weak polymorphisms are solvable in polynomial time, assuming that determining which region a point is in can be computed in polynomial time. Such polymorphisms are quite general and are obtained by gluing together several functions that are periodic in the Hamming weights in different blocks of the input. For example, we can have functions that equal parity for relative Hamming weights up to 1/2, and Majority (so identically 1) for weights above 1/2.

Our algorithm is based on a novel combination of linear programming and solving linear systems over rings. We also abstract a framework based on embedding the promise CSP into a CSP over an infinite domain, solving it there (via the said combination of LPs and ring equations), and then rounding the solution to an assignment for the promise CSP instance. The rounding step is intimately tied to the family of weak polymorphisms, and clarifies the connection between polymorphisms and algorithms in this context. Along we way, we use a result due to Adler and Beling that linear programs over $\mathbb{Z}[\sqrt{2}]$ and similar algebraic extensions (instead of $\mathbb{Q}$) can be solved in weakly polynomial time.

*Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: jbrakens@andrew.cmu.edu. Research supported in part by an REU supplement to NSF CCF-1526092.

†Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213. Email: venkatg@cs.cmu.edu. Some of this work was done when the author was visiting the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Research supported in part by NSF grants CCF-1422045 and CCF-1526092.

ISSN 1433-8092

1 Introduction

Constraint satisfaction problems (CSPs) have driven some of the most influential developments in computational complexity, from NP-completeness to the PCP theorem to the Unique Games conjecture to the (recently settled [Bul17, Zhu17]) Feder-Vardi dichotomy conjecture. The dichotomy theorem for CSPs does not just establish that all CSPs are either NP-complete or decidable in polynomial time, it also pinpoints the mathematical structure that allows for efficient algorithms: when the solution space admits certain non-trivial closure operations called *polymorphisms*, the CSP is tractable, and otherwise it is NP-hard. For instance, for linear equations, if v_1, v_2, v_3 are three solutions, then so is $v_1 - v_2 + v_3$, and the underlying polymorphism is $f(x,y,z) = x - y + z$.

Such polymorphisms and resulting CSP algorithms are unfortunately relatively rare. For instance, in the Boolean case, where the dichotomy has been long known [Sch78], there are only three non-trivial tractable cases: Horn SAT (along with its complement dual Horn SAT), 2-CNF satisfiability, and Linear Equations mod 2. The situation for larger domains is similar, with even arity two CSPs like graph k-colorability being NP-hard for $k \geq 3$. One well-studied approach to cope with the prevalent intractability of CSPs is to settle for approximation algorithms that satisfy a guaranteed fraction of constraints (the *Max CSP* problem). This has been a very fruitful avenue of research from both the algorithmic and hardness sides. In this context, a general algorithm based on semidefinite programming is known to deliver approximation guarantees matching the performance of a variant of polymorphisms tailored to optimization (namely "low-influence approximate polymorphisms") [BR15], and the Unique Games conjecture implies this cannot be improved upon [KKMO07, Rag08]. Thus, at least conditionally, we have a link between mathematical structure and the existence of efficient approximation algorithms, although such work does not apply to the approximation of satisfiable CSP instances.

1.1 Promise CSPs and Weak Polymorphisms

The Max CSP framework, however, does not capture problems like approximate graph coloring where one is allowed more colors than the chromatic number of the graph, for example 10-coloring a 3-colorable graph. An extension of CSPs, called promise CSPs, captures such problems. Informally, a promise CSP asks for an assignment to a CSP instance that satisfies a relaxed version of the CSP instance. For instance, given a k-SAT instance promised to have an assignment satisfying 3 literals per clause, we might settle for an assignment satisfying an odd number of literals in each clause. (We will give formal and more general definitions in Section 2, but briefly a promise CSP is defined by pairs of predicates (P_i, Q_i) with $P_i \subseteq Q_i$, and given an instance of CSP with defining predicates $\{P_i\}$, we would like to find an assignment that satisfies the instance when P_i is replaced with Q_i.) A promise CSP called $(2+\varepsilon)$-SAT (and a variant related to 2-coloring low-discrepancy hypergraphs) was studied in [AGH17]. This work also brought to the fore the concept of weak polymorphisms associated with the promise CSP, which are functions that are guaranteed to map tuples in P_i into Q_i for every i, generalizing the concept of polymorphisms from the case when $P_i = Q_i$ (again, see Section 2 for formal definitions). Some new hardness results for graph and hypergraph coloring were then obtained using the weak polymorphism framework in [BG16].

In [BG18], we undertook a systematic investigation of promise CSPs via the lens of weak polymorphisms, building some theory of their structure and interplay with computational complexity. For the latter, there is a Galois correspondence implying that the complexity of a promise CSP is completely dictated by its weak polymorphisms [Pip02]. Thus, from the perspective of classifying the complexity of promise CSPs, one can just focus on weak polymorphisms and forget about the relations defining the CSP.

Our work, however, revealed that the space of weak polymorphisms is very rich. Therefore, the program of classifying the complexity of promise CSPs via weak polymorphisms (along the lines of the successful theory establishing a complexity dichotomy in the case of CSPs) must overcome significant challenges that go well beyond the CSP case. The polymorphisms associated with CSPs are closed under compositions (since the output belongs to the same relation as the inputs), and as a result they belong to a well-structured class of objects in universal algebra called clones. Weak polymorphisms inherently lose this closure under composition (as the output no longer belongs to the same relation as the inputs). They are therefore much less constrained — essentially any family of functions obeying mild conditions (projection-closed and finitizable) can arise as weak polymorphisms [BG18, Pip02]. Further, whereas a single non-trivial polymorphism can suffice for tractability (as it can be composed with itself to give more complex and higher arity functions), in the case of weak polymorphisms we really need an *infinite family* of them in order to develop algorithms for the associated promise CSP. Indeed, the hardness results of [AGH17, BG18] proceed by establishing a junta-like structure for the weak polymorphisms, and thus the lack of a rich infinite family of them.

The vast variety of possible families of weak polymorphisms means that there are still numerous algorithms, and possibly whole new algorithmic paradigms, yet to be discovered in the promise CSP framework. This is the broad agenda driving this work. Our main result in [BG18] classified all tractable cases of Boolean promise CSPs whose defining predicates (P_i, Q_i) are symmetric.[1] The algorithms were governed by (essentially) three nicely structured weak polymorphism families: (i) parity functions, (ii) majority functions, or (iii) a non-symmetric (albeit block-symmetric) family we called alternating threshold (see Theorem 2.2 for the precise statement).

1.2 Our results

This work is motivated by the program of more systematically leveraging families of weak polymorphisms toward the development of new algorithmic approaches to promise CSP. In this vein, we provide a vast generalization of the above-mentioned algorithmic results for symmetric Boolean promise CSPs, by exhibiting algorithms based on rather general (albeit still structured) families of weak polymorphisms.[2] Specifically, we show that promise CSPs that admit a family of "regional periodic" weak polymorphisms are polynomial time solvable. Such polymorphisms are quite general; their precise description is a bit technical but at a high level they are obtained by gluing together, for various ranges of Hamming weights in prescribed blocks of the input, functions that are periodic in the Hamming weights in their respective block.[3]

Below we state a special case of this result when there is only one block, so that the weak polymorphisms are "threshold-periodic" *symmetric* functions (for simplicity, this case is treated first in Section 4, before the more general block-symmetric case in Section 5). Namely, such polymorphisms look at the range of the Hamming weight of its input, based on which it applies a certain periodic function of the Hamming weight. We stress that imposing a symmetry requirement on the weak polymorphisms is very different from imposing a symmetry condition on the predicates. At least for the Boolean domain, the latter was solved[4] in our earlier work [BG18], whereas we are probably still quite far from handling general symmetric weak

[1] A predicate P is symmetric if for all $(x_1,\ldots,x_m) \in P$ and all permutations $\pi : [m] \to [m]$, we have that $(x_{\pi(1)},\ldots,x_{\pi(m)}) \in P$. We say that (P_i, Q_i) is symmetric if both P_i and Q_i are symmetric.

[2] One possible concern is that no promise CSPs (or only "trivial" promise CSPs) admit such a family $\mathscr{F}$ of weak polymorphisms. To see why this is not the case, pick a positive integer L. We can think of $\{0,1\}^{2^L}$ as the set of all function $f : \{0,1\}^L \to \{0,1\}$. Let $P = \{f \mid f(x) = x_i\}$ be a set of L functions. We let $Q = \{g : \{0,1\}^L \to E \mid \text{exists } f \in \mathscr{F} \text{ such that } g \text{ is a projection of } f\}$. (See Section 2 for the definition of a projection.) Then $\mathscr{F} \subset \text{poly}(P,Q)$, and if $\mathscr{F}$ is reasonably structured (like "almost all" of the families considered in this paper), then (P,Q) is a nontrivial problem (e.g., $Q \neq E^{2^L}$). A more detailed discussion of this fact is available in Appendix E of the full version of [BG18].

[3] While we focus on the case that the domain of the P_i's is Boolean (although the Q_i's can be over any finite domain), this is mostly for notational simplicity. Our methods are general enough to be readily adapted to any finite domain; see Section 6.

[4] This classification used an additional assumption that the predicates can be applied to negations of variables.

polymorphism families (though this work is a step in that direction).

Theorem 1.1 (Informal version of Theorem 4.3)**.** *Let E be a finite set, let $0=\tau_0<\tau_1<\cdots<\tau_{k-1}<\tau_k=1$ be a sequence of rationals, let $M=(M_1,M_2,\dots,M_k)$ a sequence of positive integers, and let $\eta_i:\mathbb{Z}/M_i\mathbb{Z}\to E$ be periodic functions for $i=1,\dots,k$. Consider a promise CSP $\{(P_i,Q_i)\}$ with the P_i's and Q_i's defined over the domains $\{0,1\}$ and E, respectively. Further suppose the promise CSP admits a family of weak polymorphisms $f_L:\{0,1\}^L\to E$ for infinitely many L such that*

$$f_L(x)=\begin{cases}\eta_1(0) & \mathrm{Ham}(x)=0\\ \eta_i(\mathrm{Ham}(x) \bmod M_i) & L\tau_{i-1}<\mathrm{Ham}(x)<L\tau_i,\ i=1,2,\dots,k\\ \eta_k(L) & \mathrm{Ham}(x)=L.\end{cases}$$

where $\mathrm{Ham}(x)$ *denotes the Hamming weight of x. Then the promise CSP can be solved in polynomial time.*

As a concrete example, consider $E=\{0,1,2,3\}$ and a promise CSP with a single pair of predicates (P,Q), which are defined to be

$$P=\{x\in\{0,1\}^6:\mathrm{Ham}(x)=3\}$$
$$Q=\{y\in\{0,1,2,3\}^6: y_i\notin\{0,3\}^6\cup\{1,2\}^6 \text{ and } \sum_{i=1}^{6}y_i\equiv 1 \mod 2.\}$$

Note that $P\subseteq Q$, so (P,Q) is a valid pair of predicates for a promise CSP.[5] At first, it is unclear what algebraic structure (P,Q) has, but it turns out for all odd L to have the following weak polymorphism $g_L:\{0,1\}^L\to\{0,1,2,3\}$.

$$g_L(x)=\begin{cases}0 & \mathrm{Ham}(x)<L/2 \text{ and } \mathrm{Ham}(x)\equiv 0 \mod 2\\ 3 & \mathrm{Ham}(x)<L/2 \text{ and } \mathrm{Ham}(x)\equiv 1 \mod 2\\ 2 & \mathrm{Ham}(x)>L/2 \text{ and } \mathrm{Ham}(x)\equiv 0 \mod 2\\ 1 & \mathrm{Ham}(x)>L/2 \text{ and } \mathrm{Ham}(x)\equiv 1 \mod 2.\end{cases}$$

In Theorem 1.1, this corresponds to the choices $k=2,\ \tau_1=1/2,\ M_1=M_2=2,\ \eta_1(0)=0,\ \eta_1(1)=3,$ $\eta_2(0)=2$, and $\eta_2(1)=1$. We leave as an exercise to the reader to check why this family of g_L's are weak polymorphisms of (P,Q). Below, we give an overview of our algorithm for this special case. This serves as an illustration of the crux of our strategy, which involves blending together two broad approaches underlying efficient CSP algorithms, namely linear programming and solving linear systems over rings.

It should be noted that at a high level, CSPs solvable by linear programming relaxations have a connection to "bounded width" constraint satisfaction problems (e.g., [KOT+12]) and CSPs representable as ring equations have Mal'tsev polymorphisms (e.g., [BKW17]). Thus, by "synthesizing" these two techniques, we are understanding promise CSPs (like the (P,Q) just mentioned) which neither method by itself would resolve.

1.3 Overview of ideas for a special case

To give insight into the proof of Theorem 4.3, we give a high-level overview of how to solve promise CSPs using the predicate (P,Q) mentioned in the previous subsection with $P\subset\{0,1\}^6$ and $Q\subset\{0,1,2,3\}^6$. As stated, there is an infinite family of threshold-periodic weak polymorphisms $g_L:\{0,1\}^L\to\{0,1,2,3\}$ (where L is odd).

[5] In Section 2, we allow for a more general mapping $\phi:\{0,1\}\to E$ such that $\phi(P)\subseteq Q$.

Imagine we have an instance of a CSP with constraints from P on Boolean variables $x_1, \dots, x_n$. We seek to find $y_1, \dots, y_n \in \{0,1,2,3\}^m$ which satisfies the corresponding CSP instance with respect to Q. We first construct a Basic LP relaxation.

Basic LP Relaxation. In the Basic LP relaxation, for each $x_i \in \{0,1\}$ we consider a relaxed version $v_i \in [0,1]$. For every constraint $P(x_{i_1}, \dots, x_{i_6})$, we specify $(v_{i_1}, \dots, v_{i_6})$ must live in the convex hull of P. We can find real-valued v_i's which satisfy these conditions in polynomial time.

Now consider if we try to round the v_i's right away. Consider a constraint $P(x_{i_1}, \dots, x_{i_6})$, then we know there is a convex combination of elements of P which equals $(v_{i_1}, \dots, v_{i_6})$. A key idea introduced in our previous work [BG18] was that the weights of the convex combination can, in the limit, be approximated by an average of the elements of P using integer weights which sum to an odd number. Imagine this weighted average being arranged as a matrix

$$\begin{array}{c|cccccc|c}
 & x_{i_1}^{(1)} & x_{i_2}^{(1)} & x_{i_3}^{(1)} & x_{i_4}^{(1)} & x_{i_5}^{(1)} & x_{i_6}^{(1)} & \in P \\
 & x_{i_1}^{(2)} & x_{i_2}^{(2)} & x_{i_3}^{(2)} & x_{i_4}^{(2)} & x_{i_5}^{(2)} & x_{i_6}^{(2)} & \in P \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
 & x_{i_1}^{(L)} & x_{i_2}^{(L)} & x_{i_3}^{(L)} & x_{i_4}^{(L)} & x_{i_5}^{(L)} & x_{i_6}^{(L)} & \in P \\
\hline
\text{Average} & \approx v_{i_1} & \approx v_{i_2} & \approx v_{i_3} & \approx v_{i_4} & \approx v_{i_5} & \approx v_{i_6} & \\
g_L & \hat{y}_{i_1} & \hat{y}_{i_2} & \hat{y}_{i_3} & \hat{y}_{i_4} & \hat{y}_{i_5} & \hat{y}_{i_6} &
\end{array}$$

The key observation is that since the L rows have elements of P, we can apply the weak polymorphism g_L to get an element of $(\hat{y}_{i_1}, \dots, \hat{y}_{i_6}) \in Q_i$.

Now think about what happens to x_{i_1}. If $v_{i_1} > 1/2$ then if L is sufficiently large and the integer weights sufficiently accurate, then the Hamming weight of the column $(x_{i_1}^{(1)}, \dots, x_{i_1}^{(L)})$ will be greater than $L/2$, guaranteeing that $\hat{y}_{i_1}$ is 1 or 2. Likewise, if $v_{i_1} < 1/2$, then we can guarantee that $\hat{y}_{i_1}$ is either 0 or 3. We can deftly avoid the case $v_{i_1} = 1/2$ from ever happening, by solving the linear program over a subring of $\mathbb{R}$ that is dense but does not contain $1/2$, such as $\mathbb{Z}[\sqrt{2}]$.[6]

Since the same variable can appear in many predicates in the instance, issues can arise. For the variable x_{i_1} note that the Basic LP made a *global* choice that either $v_{i_1} > 1/2$ or $v_{i_1} < 1/2$. Thus, for every clause that x_{i_1} appears in, the corresponding $\hat{y}_{i_1}$ will always be in $\{0,3\}$ (if $v_{i_1} < 1/2$) or $\{1,2\}$ (if $v_{i_1} > 1/2$). However, this approach on its own cannot globally ensure that $\hat{y}_{i_1}$ is always equal to, say, 0 instead of 3. This due to the current lack of control on the *parity* of how many times each element of P shows up in the matrix above, since this parity is what the weak polymorphism g_L looks at when deciding whether $\hat{y}_{i_1}$ is 0 or 3. Naive attempts to force a certain parity fail, as the same variable needs the same parity assigned across all the constraints it appears in. To repair this, we also need to consider the Affine relaxation.

Affine Relaxation. Here, we let V be the smallest affine subspace (with respect to $\mathbb{F}_2$) which contains P. Then, each constraint $P(x_{i_1}, \dots, x_{i_m})$ is relaxed to $(r_{i_1}, \dots, r_{i_m}) \in V$ where $r_i \in \mathbb{F}_2$. Solving these relaxed constraints can be done in polynomial time using Gaussian Elimination over $\mathbb{F}_2$.

The beauty of utilizing this second relaxation is that whenever we run into the dilemma of $\hat{y}_{i_1} \in \{0,3\}$ or $\hat{y}_{i_1} \in \{1,2\}$, we can break the uncertainty by always setting y_{i_1} to be element with the same parity as r_{i_1}! The reason this works is subtle but powerful. When picking the integer weights of the elements of P, we also require that the Hamming weight of each column modulo 2 is equal to the r_{i_1}'s. When L is really large, changing the parity does not harm the approximation, so the "binning" of $\hat{y_{i_1}} \in \{0,3\}$ or $\hat{y_{i_1}} \in \{1,2\}$ still works via the Basic LP. But now the addition of these r_{i_1}'s via the Affine relaxation further guarantees that across clauses, the $\hat{y_{i_1}}$ chosen always has consistent parity with r_{i_1}. Thus, the $\hat{y_{i_1}}$'s do indeed satisfy all Q

[6] Solving linear programs over such a ring is known to be efficient due to a result of Adler and Beling [AB92]. The authors believe this is the first application of this fact to approximation algorithms. See Section 3.2 for more details.

constraints. This completes the proof that (P,Q) is a tractable promise CSP template.

Note that each of these two relaxations was a "lifting" (or we call *embedding*) of the (P,Q) problem into the Boolean-domain Gaussian elimination problem and the infinite-domain Basic LP relaxation. Section 3 more formally defines how this lifting process works.

1.4 Organization

In Section 2, we describe the notation used for CSPs and promise CSPs, particularly for polymorphisms and weak polymorphisms. In Section 3, we formally define the Basic LP and Affine relaxations (and combined relaxations) of a promise CSP via a notion we call a *promise embedding*. In Section 4, we prove that having an infinite family of threshold-periodic weak polymorphisms implies tractability, proving "warm up" results for threshold polymorphisms and periodic polymorphisms along the way. In Section 5, we show how to extend these reductions to block-symmetric functions known as *regional* and *regional periodic* polymorphisms. In Section 6, we briefly describe how these results can be extended to larger domains. In Section 7, we describe the challenges in further developing the theory of promise CSPs. Appendix A proves that the reductions to finite and infinite domains described in Section 3 are correct and efficient.

On a first reading, we recommend focusing on Section 4 after skimming Sections 2 and 3.

2 Preliminaries

In this section, we include the important definitions and results in the constraint satisfaction literature. In order to accommodate both the theorist and the logician, we give the definitions from multiple perspectives.

2.1 Constraint Satisfaction

In this paper, a constraint satisfaction problem consists of a *domain* D and a set $\Gamma = \{P_i \subseteq D^{\mathrm{ar}_i} : i \in I\}$ of *constraints* or *relations*. Each ar_i is called the *arity* of constraint P_i and the collection $\sigma = \{(i, \mathrm{ar}_i) : i \in I\}$ is called a *signature*. We say that $(x_1, \dots, x_{\mathrm{ar}_i})$ *satisfies* a constraint P_i if $(x_1, \dots, x_{\mathrm{ar}_i}) \in P_i$. This is written as $P_i(x_1, \dots, x_{\mathrm{ar}_i})$. This indexed set of constraints Γ is often referred to as the *template*.[7]

A Γ-*CSP* is a formula written in conjunctive normal form (CNF) with constraints from Γ. That is, for some index set J

$$\Psi(x_1, \dots, x_n) = \bigwedge_{j \in J} P_{i_j}(x_{j_1}, \dots, x_{j_{\mathrm{ar}_{i_j}}}).$$

We say that the formula is satisfiable if there is an assignment of variables which satisfies every clause. The decision problem CSP(Γ) corresponds to the language $\{\Phi : \Phi \text{ is a satisfiable } \Gamma\text{-CSP}\}$. In other words, given Φ, is it satisfiable?

Remark. In the CSP literature, another common way to define a Γ-CSP is consider the domain $X = \{x_1, \dots, x_n\}$ and a template Ψ with signature σ. We say that Ψ is satisfiable, if there is a *homomorphism* (to be defined soon) $f : X \to D$ from Ψ to Γ.

The famous Dichotomy Conjecture of Feder and Vardi [FV98] conjectured that for every finite domain D and template Γ, Γ-CSP is either in P or in NP-complete.

The case $|D| = 2$ was first fully solved by Schaefer [Sch78]. This was later extended to the case $|D| = 3$ by Bulatov [Bul06], and finally general finite D in the recent independent works by Bulatov [Bul17]

[7] The tuple (D, σ, Γ) of the domain, signature and template is known as a *structure*.

and Zhuk [Zhu17]. An extraordinarily important tool in the resolution of the Dichotomy conjecture is *polymorphisms* (e.g., [Che06, BKW17]).

Given a relation $P \subseteq D^{\text{ar}}$ and a function $f : D^L \to E$ (where D and E may be equal), we define $f(P)$ to be[8]

$$\{(f(x_1^{(1)}, \ldots, x_1^{(L)}), \ldots, f(x_{\text{ar}}^{(1)}, \ldots, x_{\text{ar}}^{(L)})) : x^{(1)}, \ldots, x^{(L)} \in P\}.$$

More pictorially (c.f., [BKW17])

$$\begin{array}{lllll}
(x_1^{(1)}, & x_2^{(1)}, & \ldots, & x_{\text{ar}}^{(1)}) & \in P \\
(x_1^{(2)}, & x_2^{(2)}, & \ldots, & x_{\text{ar}}^{(2)}) & \in P \\
\vdots & \vdots & & \vdots & \\
(x_1^{(L)}, & x_2^{(L)}, & \ldots, & x_{\text{ar}}^{(L)}) & \in P \\
\Downarrow f & \Downarrow f & \ldots & \Downarrow f & \\
y_1 & y_2 & \ldots & y_k & \in f(P)
\end{array}$$

Given this notion, we can now define both what a *homomorphism* and what a *polymorphism* are.

Definition 2.1. Let D and E be domains and $\sigma = \{(i, \text{ar}_i) : i \in I\}$ be a signature. Let $\Gamma = \{P_i \subseteq D^{\text{ar}_i} : i \in I\}$ and $\Gamma' = \{P'_i \subseteq E^{\text{ar}_i} : i \in I\}$ be templates with signature σ. A map $f : D \to E$ is a *homomorphism* from Γ to Γ' if $f(P_i) \subseteq P'_i$ for all i.

As an example, consider $D = \{0,1\}$ and $E = \{0,1,2\}$ and $\sigma = \{(1,2)\}$. Consider $\Gamma_{2\text{–col}} = \{P_1 = \{(0,1),(1,0)\} \in D^2\}$ and $\Gamma_{3\text{–col}} = \{Q_1 = \{(0,1),(0,2),(1,0),(1,2),(2,0),(2,1)\} \in E^2\}$, which are the templates for 2-coloring and 3-coloring respectively. Then, the map id_D is a homomorphism from $\Gamma_{2\text{–col}}$ to $\Gamma_{3\text{–col}}$. In other words, any 2-colorable graph is also 3-colorable.

Definition 2.2. Let D be a domain and $\Gamma = \{P_i \subseteq D^{\text{ar}_i} : i \in I\}$ be a template. A *polymorphism* is a function $f : D^L \to D$ for some positive integer L such that $f(P_i) \subseteq P_i$ for all $i \in I$. We let $\text{poly}(\Gamma)$ denote the set of a polymorphisms of Γ.

Intuitively, polymorphisms are algebraic objects which combine solutions of CSPs to produce another solution. We now give a few standard examples.

1. Consider any template Γ. A trivial example of such an f is a *projection* function: for some $i \in [L] := \{1, \ldots, L\}$, for all $x_1, \ldots, x_L \in D$, we let $f(x_1, \ldots, x_L) = x_i$. This function is a polymorphism for every Γ. More generally, we say that a polymorphism is *essentially unary* (or a *dictator*) if f depends on exactly one coordinate (in particular, this does not include constant functions).

2. Consider a polymorphism $f : D^L \to D$ such that $f \in \text{poly}(\Gamma)$. Let $\pi : [L] \to [R]$ be any surjective map, where $R \le L$ is a positive integer. Then, $f^\pi : D^R \to D$ is defined to be

$$f^\pi(x_1, \ldots, x_R) = f(y_1, \ldots, y_L) \text{ where } y_j = x_{\pi(j)} \text{ for all } j \in L.$$

   We have that $f^\pi \in \text{poly}(\Gamma)$ (e.g., [BG18]).

3. Linear Equations. Consider any finite field $\mathbb{F}$. Let

$$\Gamma_{\mathbb{F}\text{–lin}} = \{P_i \subset \mathbb{F}^{\text{ar}_i} : P_i \text{ affine subspace}\}$$

   be a template of linear constraints. Then, the map $f(x,y,z) = x - y + z$ is a polymorphism. In the case $\mathbb{F} = \mathbb{F}_2$, this is called PAR_3.

[8] This corresponds to the $O_f(P)$ notation from [BG18].

4. 2-SAT. The template for 2-SAT can be expressed in a few ways, one is

$$\Gamma_{2\text{-SAT}} = \{P_1 = \{(1,1),(1,0),(0,1)\}, P_2 = \{(1,0),(0,1)\}\}.$$

Then MAJ_3, the majority function on 3 bits, is a polymorphism (e.g., [Che06]).

One reason polymorphisms are so fundamental, is due to an elegant property known as a *Galois correspondence* (or *Galois connection*). From a computational complexity perspective[9], if two finite CSP templates Γ_1 and Γ_2 of the same domain, but not necessarily of the same signature, satisfy $\mathrm{poly}(\Gamma_1) \subseteq \mathrm{poly}(\Gamma_2)$, then there is a polynomial-time reduction from $\mathrm{CSP}(\Gamma_2)$ to $\mathrm{CSP}(\Gamma_1)$. Thus, from a computational complexity perspective, it is sufficient to think about the polymorphisms of a CSP rather than the individual constraints. We can now state Schaefer's theorem rather elegantly.

Theorem 2.1 ([Sch78], as stated in, e.g, [BJK05])**.** *Let $D = \{0,1\}$ and let Γ be a template.* $\mathrm{CSP}(\Gamma) \in \mathsf{P}$ *if and only if*[10] $\mathrm{poly}(\Gamma)$ *has a non-dictator polymorphism. Otherwise,* $\mathrm{CSP}(\Gamma)$ *is* NP*-complete.*

2.2 Promise Constraint Satisfaction

Next, we discuss an approximation variant of CSPs known as *promise CSPs* (or PCSPs), first studied systematically by the authors in [BG18]. Intuitively, a promise CSP is just like a CSP except that the constraints have "slack" to them which allows for an algebraic form of approximation.

Definition 2.3. A *promise domain* is a triple (D,E,ϕ), where ϕ is a map from D to E.

The most commonly used promise domain in this article will be $D = E = \{0,1\}$ and $\phi = \mathrm{id}_D$ is the identity map.

Definition 2.4. Let (D,E,ϕ) be a promise domain and let $\sigma = \{(i,\mathrm{ar}_i) : i \in I\}$ be a signature. A *promise template* $\Gamma = (\Gamma_P,\Gamma_Q)$is a pair of templates $\Gamma_P = \{P_i \in D^{\mathrm{ar}_i}\}$ and $\Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\}$ each with signature σ such that ϕ is a homomorphism from Γ_P to Γ_Q. Each pair (P_i,Q_i) is called a *promise constraint*.

In the simplest case, $D = E$ and $\phi = \mathrm{id}_D$, then the homomorphism condition is equivalent to $P_i \subseteq Q_i$. Note that in general ϕ could be an injection, surjection or neither.

Definition 2.5. Let $\Gamma = \{\Gamma_P,\Gamma_Q\}$ be a promise template over the promise domain (D,E,ϕ). A Γ-*PCSP* is a pair of CNF formulae Ψ_P and Ψ_Q with identical structure. That is, there is an index set J such that

$$\Psi_P(x_1,\ldots,x_n) = \bigwedge_{j\in J} P_{i_j}(x_{j_1},\ldots,x_{j_{\mathrm{ar}_{i_j}}})$$
$$\Psi_Q(y_1,\ldots,y_n) = \bigwedge_{j\in J} Q_{i_j}(y_{j_1},\ldots,y_{j_{\mathrm{ar}_{i_j}}})$$

Remark. Just like a Γ-CSP, a Γ-PCSP can be expressed in the language of homomorphisms. Our domain again $X = \{x_1,\ldots,x_n\}$, and Ψ is a template over the domain X with the same signature as Γ_P and Γ_Q. Satisfying Ψ_P and Ψ_Q corresponds to finding homomorphisms from Ψ to Γ_P and Γ_Q, respectively.

Note that if $x_1,\ldots,x_n$ satisfies Ψ_P, then $\phi(x_1),\ldots,\phi(x_n)$ satisfies Ψ_Q. In particular, satisfying Ψ_Q is "easier" (in a logical, not algorithmic, sense) than satisfying Ψ_P. Thus, we can define a promise problem.

[9] From a logic perspective, there is a *primitive-positive reduction* from Γ_2 to Γ_1.

[10] We assume in this paper that $\mathsf{P} \neq \mathsf{NP}$.

Definition 2.6 (Promise CSP–decision version)**.** Let Γ be a promise CSP. We define PCSP(Γ) to be the following promise decision problem on promise formulae (Ψ_P, Ψ_Q).

- **ACCEPT:** Ψ_P is satisfiable.
- **REJECT:** Ψ_Q is not satisfiable.

This has a corresponding search variant

Definition 2.7 (Promise CSP–search version)**.** Let Γ be a promise CSP. We define Search-PCSP(Γ) to be the following promise search problem on promise formulae (Ψ_P, Ψ_Q).

- Given that Ψ_P is satisfiable, output a satisfying assignment to Ψ_Q.

Unlike classical CSPs, in which the decision and search versions are often[11] polynomial-time equivalent, it is not clear that the decision and search variants of PCSP(Γ) have the same computational complexity, although there no known Γ for which the complexity differs. Even so, there is a reduction from the decision version to the search version: run the algorithm for the search version, and check if it satisfies Ψ_Q.

The following are interesting examples of promise CSPs, (c.f., [BG18]). In the first five examples, the domain is Boolean: $(D = \{0,1\}, D, \mathrm{id}_D)$.

1. **CSPs.** Let $\Gamma = \{P_i \in D^{k_i}\}$ be a CSP over the domain D, then (D, D, id_D) is a promise domain and $\Lambda = (\Gamma, \Gamma)$ is a promise CSP.

2. **(2+ε)-SAT.** Let $\mathrm{NEQ} = \{(0,1),(1,0)\}$. Fix, a positive integer k, and let $P_1 = \{x \in D^{2k+1} : \mathrm{Ham}(x) \geq k\}$ and $Q_1 = \{x \in D^{2k+1} : \mathrm{Ham}(x) \geq 1\}$. Then, $\Gamma = (\{P_1, \mathrm{NEQ}\}, \{Q_1, \mathrm{NEQ}\})$ corresponds to a promise variant of $(2k+1)$-SAT: if every clause in a $(2k+1)$-SAT instance is true for at least k variables, can one find a "normal" satisfying assignment. This problem was shown to be NP-hard by Austrin, Guruswami, and Håstad [AGH17].

3. **Threshold conditions.** Fix α, β such that $1/\alpha + 1/\beta = 1$. Let $\Gamma_P = \{P_i \subset D^{\mathrm{ar}_i} : i \in \{1,2\}\}$ and $\Gamma_Q = \{Q_i \subseteq D^{\mathrm{ar}_i} : \{1,2\}\}$, where the promise constraints are as follows (the choices of $\mathrm{ar}_1, \mathrm{ar}_2, s, t$ are arbitrary).

$$P_1 = \{x \in D^{\mathrm{ar}_1} : \mathrm{Ham}(x) \leq s\} \qquad Q_1 = \{x \in D^{\mathrm{ar}_1} : \mathrm{Ham}(x) \leq \alpha s\}$$
$$P_2 = \{x \in D^{\mathrm{ar}_2} : \mathrm{Ham}(x) \geq \mathrm{ar}_2 - t\} \qquad Q_2 = \{x \in D^{\mathrm{ar}_2} : \mathrm{Ham}(x) \geq \mathrm{ar}_2 - \beta t\}.$$

We have that PCSP(Γ_P, Γ_Q) is tractable (this is also alluded to in [BG18]).

4. **Embedding Linear Equations.** Let $A \subseteq \mathbb{F}_7^{\mathrm{ar}}$ be an affine subspace. Specify a map $h : \mathbb{F}_7 \to \{0,1\}$ such that $h(0) = 0$ and $h(1) = 1$. Let $\Gamma_P = \{A \cap D^{\mathrm{ar}}\}$ and $\Gamma_Q = \{h(A)\}$. Then, (Γ_P, Γ_Q) is tractable by performing Gaussian elimination over $\mathbb{F}_7$, even though the domain is Boolean! This type of promise CSP was briefly alluded to in [BG18], and in this work we systematically study such examples though the theory of *promise embeddings* (see Section 3.1).

5. **Hitting Set.** Let $\Gamma_P := \{P_i \in D^{\mathrm{ar}_i} : i \in I\}$, where $P_i := \{x \in D^{\mathrm{ar}_i} : \mathrm{Ham}(x) = \ell_i\}$ where $\ell_i \in \{1, \ldots, \mathrm{ar}_i - 1\}$. In other words, CSP($\Gamma_P$) corresponds to a generalized hitting set problem: given a collection of hyperedges S_i and targets ℓ_i, find a subset of the vertices S such that $|S \cap S_i| = \ell_i$. Let $\Gamma_Q := \{Q_i = D^{\mathrm{ar}_i} \setminus \{0^{\mathrm{ar}_i}, 1^{\mathrm{ar}_i}\}\}$. Then CSP($\Gamma_Q$) is hypergraph two-coloring (each color appears at least once per hyperedge). Although neither CSP(Γ_P) or CSP(Γ_Q) is tractable in general, PCSP(Γ_P, Γ_Q) is tractable [BG18].

[11] This requires that fixing a variable to a specific value leads to a constraint with the same polymorphisms.

6. **Approximate Graph Coloring.** Let $k \leq \ell$ be positive integer. Let $D = [k]$ and $E = [\ell]$. Then, (D, E, id_D) is a promise domain. Let $\Gamma_{k\text{–col}} = \{P = \{(x,y) \in D^2 : x \neq y\}\}$ and $\Gamma_{\ell\text{–col}} = \{Q = \{(x,y) \in E^2 : x \neq y\}\}$. Then, $\Gamma = (\Gamma_{k\text{–col}}, \Gamma_{\ell\text{–col}})$ is then the promise template for the well-studied *approximate graph coloring problem*: given a graph of chromatic number k, find an ℓ-coloring. This problem has been studied for decades, and it is still unsolved in many cases (e.g., [GK04, KLS00, Hua13, BG16]).

7. **Rainbow Coloring.** Consider $D = [k]$ and $E = \{0,1\}$, with $\phi : D \to E$ being an arbitrary, non-constant map. If $\Gamma_P = \{P = \{x \in D^k : x \text{ is a permutation of } D\}\}$ and $\Gamma_Q = \{Q = E^k \setminus \{0^k, 1^k\}\}$, then $\mathrm{PCSP}(\Gamma_P, \Gamma_Q)$ is the following hypergraph problem: given a k-uniform hypergraph such that there is a k-coloring in which every color appears in every edge, find a hypergraph 2-coloring. A random-walk-based polynomial-time algorithm is reported in [McD93]; semidefinite programming gives another folklore algorithm. A deterministic algorithm based on solving linear programming was found by Alon [personal communication].

Analogous to the utility of polymorphisms for studying CSPs, *weak polymorphisms* are useful for studying promise CSPs. The first formal definition of a weak polymorphism appeared in [AGH17].

Definition 2.8. Let (D, E, ϕ) be a promise domain and $\sigma = \{(i, \mathrm{ar}_i) : i \in I\}$ be a signature. and $\Gamma = (\Gamma_P = \{P_i \subseteq D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \subseteq E^{\mathrm{ar}_i}\})$ be a promise template over this domain. A *weak polymorphism* is a function $f : D^L \to E$ for some positive integer L such that for all $i \in I$, $f(P_i) \subseteq Q_i$. We let $\mathrm{poly}(\Gamma)$ denote the set of weak polymorphisms of Γ.

Like in the case of constraint satisfaction problems, there is a Galois correspondence for promise CSPs ([BG18], following from a result of [Pip02]). Thus, like for CSPs, it suffices to consider the collection of weak polymorphisms and not the particulars of the constraints.

As pointed out in [AGH17], unlike the polymorphisms for "ordinary" CSPs, due to the change in domain of weak polymorphisms, they cannot be composed. Thus, unlike CSPs which deal with families of CSPs closed under compositions and projections (known as *clones*), promise CSPs are determined by families of CSPs closed under only projections[12]. Thus, while the techniques for studying CSPs are (universal) *algebraic*, the necessary techniques for studying promise CSPs are *topological*. In particular, unlike results such as Schaefer's theorem for which the existence of one nontrivial polymorphism (e.g., PAR_3) is enough to imply tractability of promise CSPs, [AGH17] showed that an infinite sequence of weak polymorphisms is necessary to imply tractability. One contribution of this work is that we show that having an infinite sequences of weak polymorphisms which "converge" with respect to a particular topology is sufficient to imply efficient algorithms. We now give a polymorphic reason for why each of the above examples is tractable/non-tractable.

1. **CSPs.** The polymorphisms of Γ are exactly the same as the weak polymorphisms of (Γ, Γ).

2. **(2+ε)-SAT.** [AGH17] showed that MAJ_{2k-1} is a weak polymorphism, but MAJ_{2k+1} (or any function that essentially depends on at least $2k+1$ variables) is not. They exploited this fact to show NP-hardness via a reduction from the PCP theorem.

3. **Threshold conditions.** Consider L such that L/α is not an integer. Then,

$$f_L(x_1, \ldots, x_L) = \begin{cases} 0 & \mathrm{Ham}(x) < L/\alpha \\ 1 & \mathrm{Ham}(x) > L/\alpha \end{cases}$$

[12] There is an additional constraint known as *finitization*, which comes as a technicality when Γ is finite. See the Conclusion for a discussion on how this could be utilized to understand promise CSPs from a topological perspective.

is a weak polymorphism of this problem. This is known as a *threshold polymorphism* and is studied in Section 4.1.

4. **Embedding Linear Equations.** Consider $L \equiv 1 \mod 7$ then
$$f_L(x_1, \ldots, x_L) = h\left(\sum_{i=1}^{L} x_i \mod 7\right)$$
is a family of weak polymorphisms. This is a *periodic polymorphism*, and it is studied in Section 4.2.

5. **Hitting Set.** [BG18] showed that for all odd integers L
$$\mathrm{AT}_L(x_1, \ldots, x_L) = \mathbf{1}[x_1 - x_2 + x_3 - \cdots - x_{L-1} + x_L \geq 1],$$
is a weak polymorphism for this problem. In this paper, we have generalized weak polymorphisms like these to *regional polymorphisms*, which are studied in Section 5.

6. **Approximate Graph Coloring.** As this question is still open, much is not yet understood about the weak polymorphisms. [BG16] showed that when $\ell \leq 2k-2$, then the weak polymorphisms "look" dictatorial when restricted to some subset of the outputs. These polymorphisms are closely connected to the independent sets of *tensor powers of cliques* (e.g., [ADFS04]).

7. **Rainbow Coloring.** This problem has many, many nontrivial weak polymorphisms. For example, for odd L, $f_L : [k]^L \to \{0,1\}$ defined to be
$$f_L(x_1, \ldots, x_L) = \mathbf{1}\left[\sum_{i=1}^{L} x_i \leq \frac{2k+3}{4} \cdot L\right],$$
is a family of weak polymorphisms for this problem. This is an example of a non-Boolean *regional polymorphism*, which is studied in Section 6.

The main result of our previous work [BG18] is as follows.

Theorem 2.2 ([BG18]). *Consider the promise domain $(D = \{0,1\}, D, \mathrm{id}_D)$. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \in D^{\mathrm{ar}_i}\})$ be a CSP with the following technical conditions.*

- *$P_1 = Q_1 = \{(0,1),(1,0)\}$. In other words, variables can be negated.*
- *For all $i \in I$, P_i and Q_i are* symmetric*: if $(x_1, \ldots, x_{ar_i}) \in R_i$ then $(x_{\pi(1)}, \ldots, x_{\pi(ar_i)}) \in R_i$ for all permutations π.*

Then, either $\mathrm{PCSP}(\Gamma)$ *is (promise)* NP*-hard or* $\mathrm{PCSP}(\Gamma)$ *is in* P *and has one of the following* 6 *infinite sequences of weak polymorphisms (coming in* 3 *pairs).*

1. MAJ_L *for all odd $L \geq 3$ or*[13] $\neg\mathrm{MAJ}_L$ *for all odd $L \geq 3$.*
2. PAR_L *for all odd $L \geq 3$ or* $\neg\mathrm{PAR}_L$ *for all odd $L \geq 3$.*
3. AT_L *for all odd $L \geq 3$ or* $\neg\mathrm{AT}_L$ *for all odd $L \geq 3$.*

Although that paper tried to jointly understand both algorithmic and hardness results (with most of the work coming in on the hardness side), the aim of this paper is to develop the algorithmic tools for understanding tractable cases of promise CSPs. In particular, we more deeply explore the power of LP and Affine (i.e., linear equations over a commutative ring) relaxations for solving promise CSPs.

[13] The negation symbol ($\neg$) in front a polymorphism merely means to negate the output.

3 Embeddings and Relaxations

In this section, we build on the theory described in Section 2 to rigorously connect promise CSPs with *relaxations* of these problems (e.g., linear programming relaxations).

3.1 Promise Embedding

Often it is useful to reduce promise CSP to a tractable CSP in another domain, and then map the result back to the original domain. We call this procedure a *promise embedding*.

Definition 3.1. Let (D,E,ϕ) be a promise domain and $\sigma = \{(i,\mathrm{ar}_i) : i \in I\}$ be a signature. Let $\Gamma = (\Gamma_P = \{P_i \subseteq D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \subseteq D^{\mathrm{ar}_i}\})$ be a promise template with this signature. Let F be another (possibly infinite) domain, and let $g : D \to F$, $h : F \to E$ be maps. Let Λ be a set of relations over the domain F. We say that (g,h) is a *promise embedding* of Γ into Λ if there is $\Lambda_\Gamma = \{R_i \in F^{\mathrm{ar}_i}\} \subseteq \Lambda$ with the signature σ such that g is a homomorphism from Γ_P to Λ_Γ and h is a homomorphism from Λ_Γ to Γ_Q.

In practice, D and E will both be finite. Thus, $g : D \to F$ is a finite map (often something canonical, like the identity function) which tells how to express our promise CSP in the new domain. On the other hand, $h : F \to E$, which we call the *rounding function*, is where the “algorithmic magic” takes place. When F is infinite, it is not *a priori* obvious that h has a computationally efficient description, so we often assume that we have *oracle access* to this rounding function. Furthermore, the choice of rounding function h is crucially tied to the weak polymorphisms of Γ. In fact, one can consider h to be the “limit” of a sequence of weak polymorphisms of Γ, or alternatively $\mathrm{poly}(\Gamma)$ is a *discretization* of h. This connection between h and weak polymorphisms is made more clear in Section 4.

To exemplify this definition, we give a few examples of promise embeddings.

1. Let Γ be any CSP over D, then the promise CSP (Γ,Γ) embeds into Γ via $(g,h) = (\mathrm{id}_D, \mathrm{id}_D)$
2. Recall from Example 6 of Section 2.2 that $\Gamma = (\Gamma_{k\text{–col}}, \Gamma_{\ell\text{–col}})$ is the promise template for the k vs. ℓ approximate graph coloring problem. For any $m \in \{k,\dots,\ell\}$, we have that $(\mathrm{id}_{[k]}, \mathrm{id}_{[m]})$ is promise embedding of Γ into $\Gamma_{m\text{–col}}$. In other words, any algorithm which solves the m-coloring problem can also solve the k vs. ℓ approximate graph coloring problem.
3. Consider $\Gamma = (\Gamma_P, \Gamma_Q)$ from Example 4 of Section 2.2 with affine subspace $A \le \mathbb{F}_7^{\mathrm{ar}}$ and the map $h : \mathbb{F}_7 \to \{0,1\}$. Then, $(\mathrm{id}_{\{0,1\}}, h)$ is a promise embedding from Γ to $\Lambda = \{A\}$.

To be the best of the authors' knowledge, all known polynomial time algorithms for solving promise CSPs, involve embedding the given promise template Γ into a judiciously chosen Λ for which CSP(Λ) is polynomial-time tractable. In fact, the authors conjecture that any tractable promise CSP must embed into some (possibly infinite) tractable CSP.

However, even though Γ has a finite domain, Λ often necessarily has infinite domain, even when Γ is Boolean. In fact, the Basic LP and Affine relaxations, explained in the following sections, are instances of embedding into an infinite Λ. This is another reason why classifying the computational complexity of promise CSPs is so much more difficult than for ordinary CSPs, and perhaps partially explains the difficulty of the computational complexity community's struggle to resolve the approximate graph coloring problem.

3.2 Basic LP Relaxation

The Basic LP relaxation is a widespread tool in approximation algorithms, often giving optimal results. For example, the resolution of the Finite-Valued CSP (VCSP) dichotomy due to Thapper and Živný [TZ16] showed that all tractable instances can be solved with a Basic LP relaxation. In [BG18], the Basic LP was one of the classes of algorithms exhibited in tractable promise CSPs (used for the MAJ and AT families). In this work, we vastly generalize the usage of such an algorithm.

Fix a template $\Gamma = \{P_i \subseteq D^{\mathrm{ar}_i}\}$ and consider an instance of CSP(Γ)

$$\Psi(x_1, \ldots, x_n) := \bigwedge_{j \in J} P_{i_j}(x_{j_1}, \ldots, x_{j_{\mathrm{ar}_{i_j}}}).$$

Fix a subring $A \subset \mathbb{R}$ which is to be the domain of our Basic LP. (Typically, $A = \mathbb{Q}$, but for reasons we are soon to see, other commutative rings are useful.)

Fix a positive integer $k \geq 1$ and a map $g : D \to A^k$. Then, for each $P_i \in \Gamma$, $g(P_i)$ is a cloud of points in $(A^k)^{\mathrm{ar}_i} \subseteq R^{k\,\mathrm{ar}_i}$. Recall the notion of a *convex hull* of a set of points $S \in \mathbb{R}^n$

$$\mathrm{Conv}(S) = \left\{ \sum_{i=1}^{\ell} \alpha_i z_i : \alpha_i \in [0,1], z_i \in S, \sum_{i=1}^{\ell} \alpha_i = 1 \right\}.$$

We let $\mathrm{Conv}_A(S) = \mathrm{Conv}(S) \cap A^n$. If we assume that each P_i has constant size, then $\mathrm{Conv}(g(P_i))$ can be specified by a constant number of linear inequalities.

The following is the Basic LP relaxation.

- Input: $\Psi(x_1, \ldots, x_n) := \bigwedge_{j \in J} P_{i_j}(x_{j_1}, \ldots, x_{j_{\mathrm{ar}_{i_j}}})$, an instance of CSP($\Gamma$).
- Variables: each x_i is replaced by $v_i \in A^k$.
- Constraints:
  - For each x_i, specify that $v_i \in \mathrm{Conv}(g(D))$.
  - For each constraint $P_{i_j}(x_{j_1}, \ldots, x_{j_{\mathrm{ar}_{i_j}}})$ specify that

$$(v_{j_1}, \ldots, v_{j_{\mathrm{ar}_{i_j}}}) \in \mathrm{Conv}(g(P_{i_j})).$$

Relaxation 3.1. The Basic LP relaxation of CSP(Γ).

Remark. Since our primary goal is feasibility, our LP relaxations do not have objective functions.

If we assume that Γ is finite, each P_i has constant size, the size of this LP relaxation is linear in the size of the input Ψ, with constant factors depending on the specific Γ. Note that if $A = \mathbb{Q}$, we can test feasibility and output a solution in polynomial time. Like most uses of linear programming in approximation algorithms, an LP solution, once found, is *rounded* to solve the problem at hand. Due to technical restrictions of the rounding algorithms in this paper, there are often edge cases, for which rounding will not work. For example, the procedure "round to the nearest integer" does not work for $v_{i,j} = 1/2$. In [BG18], the authors used an ad-hoc approach for avoiding these $1/2$ situations, but it turns out these can be solved in a more principled manner by solving the LP over a different ring other than $\mathbb{Q}$. Of course, linear programs over certain rings such as $A = \mathbb{Z}$, are not solvable in polynomial time unless $\mathsf{P} = \mathsf{NP}$, so we need to look at so-called *LP-solvable* rings.

Definition 3.2. A countable subring $A \subset \mathbb{R}^k$ for some positive integer k is *LP-solvable* if linear programs over A can be exactly solved in (weakly) polynomial time.

Thus, $A = \mathbb{Q}$ is LP-solvable, but $A = \mathbb{Z}$ is not. For technical reasons, we assume that $\mathbb{R}$ is *not* LP-solvable because, in general, elements of $\mathbb{R}$ do not have a finite description.

Even with these restrictions, there is still a diversity of A which are LP-solvable. Results of Adler and Beling [AB92, Bel01], show that algebraic extensions of $\mathbb{Z}$, such as $\mathbb{Z}[\sqrt{q}]$ for q non-square are LP-solvable. The usefulness of this fact is that edge cases like rounding $1/2$ can be avoided by solving the LP over, say, the ring $\mathbb{Z}[\sqrt{2}]$. The authors are unaware of a previous application of this fact to approximation algorithms of CSPs.

This procedure of rewriting a CSP as a Basic LP and then rounding can be expressed in the language of a promise embedding. Let $A \subset \mathbb{R}^k$ be an LP-solvable ring and let Γ be a promise relation over the promise domain (D, E, ϕ). As first introduced in Section 3.1, let $g : D \to A$ be any map, and let $h : A \to E$ be our rounding function. Let $\Lambda_A = \{R : R = \mathrm{Conv}_A(S), S \subset A^\ell \text{ finite}, \ell \geq 1\}$ be the family of convex subsets of A.

Theorem 3.1. *Let $A \subset \mathbb{R}^k$ be an LP-solvable ring. Let (D, E, ϕ) be a finite promise domain. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i} : i \in I\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\})$ be a finite promise CSP. Let $(g : D \to A, h : A \to E)$ be a promise embedding of Γ into Λ_A. Then $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$, in which P^h is the family promise languages which can be computed in polynomial time given oracle access to h.*

This result is proven in Appendix A. Intuitively, Theorem 3.1 abstracts away the fine details of working with the Basic LP and reduces the task to showing the existence of a promise embedding.

3.3 Affine Relaxation

Another broad class of algorithms studied in the CSP literature correspond to solving system of linear equations over some commutative ring. Such algorithms are captured in the CSP literature under the broader class of CSPs with a *Mal'tsev polymorphism*: a function on three variables such that $\varphi(x, y, y) = \varphi(y, y, x) = x$ always. Such CSPs are known to be tractable (e.g., [BKW17]). For commutative rings, the canonical Mal'tsev polymorphism is $\varphi(x, y, z) = x - y + z$, when the domain is a finite ring. Such algorithms are not restricted to finite domains: linear equations over $\mathbb{Q}$ can be solved in polynomial time using Gaussian elimination, and linear equations over $\mathbb{Z}$ can be solved in polynomial time by computing the Hermite Normal Form [KB79]. This leads to the natural notion of *LE-solvable* rings.

Definition 3.3. Define a commutative ring B to be *LE-solvable*, if systems of linear equations over R can be efficiently solved in (weakly) polynomial time.

By the discussion above, all finite commutative rings are LE-solvable, as well as the infinite rings $\mathbb{Z}^k$ for any natural number k. Furthermore, every LP-solvable ring is LE-solvable as LPs are more expressive than LEs. Just as LPs relax sets to the their convex hulls, linear equations relax sets to their *affine hulls*. Given a subset $S \subseteq R^k$, define the *affine hull* to be

$$\mathrm{Aff}(S) = \left\{ r_1 s^1 + \cdots + r_k s^k : \forall j, s^j \in S \text{ and } \sum_{j=1}^{k} r_j = 1 \right\}.$$

Note that by design $S \subseteq \mathrm{Aff}(S)$. Furthermore, if S is finite, checking whether $x \in \mathrm{Aff}(S)$ can constrained by two linear conditions over R.

We can now define the *Affine relaxation* of any CSP. Fix a finite domain D and an LE-solvable ring R. Also pick any embedding map $g : D \to R$. Let $\Gamma = \{P_i \subseteq D^{\mathrm{ar}_i}\}$ be any template over D. Note that $g(P_i)$ is some finite subset of R^{ar_i}. This leads to our description of an affine relaxation.

- Input: $\Psi(x_1,\ldots,x_n) := \bigwedge_{j\in J} P_{i_j}(x_{j_1},\ldots,x_{j_{\mathrm{ar}_{i_j}}})$, instance of CSP($\Gamma$),
- Variables: each x_i is replaced by $w_i \in R$.
- Constraints: For each constraint $P_{i_j}(x_{j_1},\ldots,x_{j_{\mathrm{ar}_{i_j}}})$ of Ψ, specify that

$$(w_{j_1},\ldots,w_{j_{\mathrm{ar}_{i_j}}}) \in \mathrm{Aff}(g(P_{i_j})).$$

Relaxation 3.2. The Affine relaxation of CSP(Γ).

By definition, $\mathrm{Aff}(g(P_i))$ has a constant-sized description, since each P_i is of constant size, and there are finitely many possible values for $g(P_i)$, a lookup table of the linear constraints can be formed. Thus, the system can be generated in linear time, and so it can be solved in polynomial time whenever R is LE-solvable. Let

$$\Theta_R = \{\mathrm{Aff}(S) : \exists k, S \subseteq R^k \text{ finite}\}.$$

This leads to an analogue of Theorem 3.1 for the infinite template over R

Theorem 3.2. *Let R be an LE-solvable ring. Let (D,E,ϕ) be a finite promise domain, and let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\})$ be a finite promise CSP over this promise domain. Let (g,h) be a promise embedding of Γ into Θ_R. Then,* PCSP(Γ) $\in \mathsf{P}^h$.

This result is proven in Appendix A.

3.4 Combined Relaxation

The true power of this promise embedding perspective is revealed when these relaxations are combined using *direct products*.

Consider CSP templates Λ_1 and Λ_2 over domains F_1 and F_2 (not necessarily finite), respectively. We define the *direct product* $\Lambda_1 \times \Lambda_2$ to be the CSP template over the domain $F_1 \times F_2$ such that

$$\Lambda_1 \times \Lambda_2 = \{R_1 \times R_2 \subseteq (F_1 \times F_2)^{\mathrm{ar}} : R_1 \in \Lambda_1, R_2 \in \Lambda_2 \text{ same arity ar}\},$$

where $(R_1 \times R_2)((x_1,y_1),\ldots,(x_{\mathrm{ar}},y_{\mathrm{ar}})) = R_1(x_1,\ldots,x_{\mathrm{ar}}) \wedge R_2(y_1,\ldots,y_{\mathrm{ar}})$.

Note that up to relabeling coordinates, the direct product is commutative and associative, allowing the seamless combination of two or more CSP templates.

Fix a sequence of LP-solvable rings $\mathscr{A} := (A_1,\ldots,A_\ell)$ and a sequence of LE-solvable rings $\mathscr{R} := (R_1,\ldots,R_m)$. Now define the template

$$\Xi_{\mathscr{A},\mathscr{R}} := \Lambda_{A_1} \times \cdots \times \Lambda_{A_\ell} \times \Theta_{R_1} \times \cdots \times \Theta_{R_m}.$$

It turns out promise homomorphisms to this template correspond to algorithms which *combine* linear programming and affine equation solving.

Theorem 3.3. *Let $\mathscr{A} := (A_1,\ldots,A_\ell)$ be a sequence of LP-solvable rings, and let $\mathscr{R} := (R_1,\ldots,R_m)$ be a sequence of LE-solvable rings. Let (D,E,ϕ) be a finite promise domain, and let $\Gamma = (\Gamma_P,\Gamma_Q)$ be a finite promise CSP over this domain. Let (g,h) be a promise embedding of Γ into $\Xi_{\mathscr{A},\mathscr{R}}$. Then,* PCSP($\Gamma$) $\in \mathsf{P}^h$.

This result is proven in Appendix A. Theorems 3.1, 3.2, and 3.3 are useful in that if we can show for a particular promise template Γ that a promise homomorphism (g,h) exists to a suitable Λ_A, Θ_R or $\Xi_{\mathscr{A},\mathscr{R}}$ *and h* is proven to be polynomial-time computable, then we can show that $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$. The subsequent sections establish this result for a variety of Γ. Due to the complexity of such Γ, we refer to them via the structure of their weak polymorphisms $\mathrm{poly}(\Gamma)$.

4 Threshold-Periodic Weak Polymorphisms

In this section and the subsequent one, we assume that our promise domain (D,E,ϕ) satisfies $D = \{0,1\}$ and E is any finite domain with any inclusion map $\phi : D \to E$. Restricting D to be Boolean allows for a simplified presentation, the results of Sections 4 and 5 can be extended to larger domains, as described in Section 6.

4.1 Threshold Polymorphisms

Many polymorphisms which are considered in classical CSP theory, such as the OR, AND, and MAJ functions, can be thought of as *threshold functions*. That is, the value of each of these polymorphisms only depends on whether the Hamming weight of the input is above a certain threshold. In this subsection, we consider a generalization of such functions to multiple thresholds.

Definition 4.1. A *threshold sequence* is a finite sequence of rational[14] numbers $\tau_0 = 0 < \tau_1 < \cdots < \tau_k = 1$.

For $x \in \{0,1\}^L$, we let $\mathrm{Ham}(x)$ be the Hamming weight of x, i.e., the number of bits of x set to 1.

Definition 4.2. Let $T = \{\tau_0, \tau_1, \ldots, \tau_k\}$ be a threshold sequence and $\eta : \{0,1,\ldots,k+1\} \to E$ be any map. Let L be a positive integer such that $L\tau_i$ is not an integer for any $i \in \{1,\ldots,k-1\}$. Then, define $\mathrm{THR}_{T,\eta,L} : \{0,1\}^L \to \{0,1\}$ to be the following polymorphism.

$$\mathrm{THR}_{T,\eta,L}(x) = \begin{cases} \eta(0) & \mathrm{Ham}(x) = 0 \\ \eta(i) & L\tau_{i-1} < \mathrm{Ham}(x) < L\tau_i, 1 \le i \le k \\ \eta(k+1) & \mathrm{Ham}(x) = L. \end{cases}$$

The function η is closely connected to the rounding function h from the definition of a promise embedding (Section 3.1). In essence, η is finite description or discretization of h.

To get intuition, here are examples of common polymorphisms and their corresponding parameters as threshold functions.

	MAJ_L	OR_L	AND_L
T	$\{0,1/2,1\}$	$\{0,1\}$	$\{0,1\}$
η	$(0,0,1,1)$	$(0,1,1)$	$(0,0,1)$

This now leads to our first main result.

Theorem 4.1. *Let $T = \{\tau_0,\ldots,\tau_k\}$ be a threshold sequence with a corresponding map $\eta : \{0,\ldots,k+1\} \to E$. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i} \in I\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i} : i \in I\})$ be a promise template such that* $\mathrm{THR}_{T,\eta,L} \in \mathrm{poly}(\Gamma)$ *for infinitely many L. Then,* $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$.

[14] These could also be real numbers under suitable computational assumptions, but for simplicity of exposition we assume all thresholds are rational

The proof is essentially a direct generalization of the arguments in Section 3.2 of [BG18].

Proof. Let $A = \mathbb{Z}[\sqrt{2}]$, which as previously stated is LP-solvable by a theorem of Adler and Beling [AB94]. We claim that there is a promise embedding from Γ into Λ_A via the following maps[15] $g : D \to A$ and $h : A \to E$:

$$g(d) = d$$
$$h(v) = \begin{cases} \eta(0) & v \le 0 \\ \eta(i) & \tau_{i-1} < v < \tau_i, 1 \le i \le k \\ \eta(k+1) & v \ge 1 \end{cases}$$

Define $\Lambda_\Gamma := \{R_i := \mathrm{Conv}_A(g(P_i)) : P_i \in \Gamma_P\}$. Since $g(P_i) \subset \mathrm{Conv}_A(g(P_i))$, we have that g is a homomorphism from Γ_P to Λ_Γ. We claim that h is a homomorphism from Λ_Γ to Γ_Q. In other words, we seek to show that $h(\mathrm{Conv}_A(g(P_i))) \subseteq Q_i$. For any $V \in \mathrm{Conv}_A(g(P_i))$, since $g(P_i)$ is finite, there exist elements $X^1, \dots, X^m \in P_i$ and weights[16] $\alpha_1, \dots, \alpha_m \in (0, 1]$ summing to 1 such that

$$V = \alpha_1 g(X^1) + \cdots + \alpha_m g(X^m).$$

Fix L to be sufficiently large (to be specified later). We can pick nonnegative integers $w_1, \dots, w_m$ such that $w_1 + \cdots + w_m = L$ and $|w_i - \alpha_i L| \le 1$ (start with $w_i = \lfloor \alpha_i L \rfloor$ for all i and then increase weights one-by-one until the sum is L). Now compute

$$Y := \mathrm{THR}_{T,\eta,L}(\underbrace{X^1, \dots, X^1}_{w_1 \text{ copies}}, \dots, \underbrace{X^m, \dots, X^m}_{w_m \text{ copies}}) \in Q_i.$$

Define for each coordinate $j \in \{1, \dots, \mathrm{ar}_i\}$

$$s_j := \frac{1}{L} \mathrm{Ham}(\underbrace{X^1_j, \dots, X^1_j}_{w_1 \text{ copies}}, \dots, \underbrace{X^m_j, \dots, X^m_j}_{w_m \text{ copies}}).$$

Then, by design, $Y_j = h(s_j)$. We also know that

$$\frac{1}{L}|s_j - V_j| = \sum_{\substack{a=1 \\ X^a_j = 1}}^{m} \left| \frac{w_a}{L} - \alpha_a \right| \le \frac{m}{L}$$

Since $V \in A^{\mathrm{ar}_i}$, we have three cases

1. If $V_j \le 0$, then $X^a_j = 0$ for all j. Then, $V_j = s_j = 0$ so $Y_j = h(V_j)$.
2. If $V_j \ge 1$, then $X^a_j = 1$ for all j. Then, $V_j = s_j = 1$ so $Y_j = h(V_j)$.
3. If $V_j \in (0, 1)$, then $\tau_{i-1} < V_j < \tau_i$ for some $i \in \{1, \dots, k\}$. Thus, as $L \to \infty$, s_j will get sufficiently close to V_j that $\tau_{i-1} < s_j < \tau_i$. Thus, $Y_j = h(s_j) = h(V_j)$.

Thus, since $Y \in Q_i$, we have that $h(V) \in Q_i$, establishing the promise embedding is valid.

By Theorem 3.1, we have that $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. Note that h is polynomial-time computable, since computing h involves checking a constant number of inequalities in A. Thus, $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$, as desired. ☐

[15]For type-theoretic reasons, we define h on the full domain of A rather than $[0, 1] \cap A$.

[16]Note that the weights might not be in A.

4.2 Periodic Polymorphisms

Instead of having our threshold functions be piece-wise constant, we can consider periodic polymorphisms.

Definition 4.3. Let M be a positive integer, and let $\eta : \mathbb{Z}/M\mathbb{Z} \to E$ be any map. Let L be a positive integer. Define $\mathrm{PER}_{M,\eta,L}$ to be the following function

$$\mathrm{PER}_{M,\eta,L}(x) = \eta(k) \text{ if } \mathrm{Ham}(x) \equiv k \mod M.$$

As stated earlier, Example 4 from Section 2.2 is a periodic polymorphism.

Theorem 4.2. *Let M be a positive integer, and let $\eta : \mathbb{Z}/M\mathbb{Z} \to E$ be any function. Let $\Gamma = (\Gamma_P = \{P_i \subseteq D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \subseteq E^{\mathrm{ar}_i}\})$ be a promise template on the Boolean domain such that $\mathrm{PER}_{M,\eta,L} \in \mathrm{poly}(\Gamma)$ for infinitely many L. Then, $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$.*

Proof. For these infinitely many L, consider the remainders when they are divided by M. Since there are only finitely many remainders, there exists $r \in \{0,\dots,M-1\}$ such that $L \equiv r \mod M$ infinitely often.

Consider the ring $R = \mathbb{Z}/M\mathbb{Z}$. We seek to show that there is a promise embedding of Γ into Θ_R via the maps $g : \{0,1\} \to R$ and $h : R \to E$ where

$$g(x) = \begin{cases} 0 & x = 0 \\ r & x = 1 \end{cases}$$
$$h = \eta.$$

Note that η is a "discretization" of h, but since R is a finite domain, η can be used for h.

Consider $\Theta_\Gamma = \{R_i := \mathrm{Aff}(g(P_i)) : P_i \in \Gamma_P\}$. Since $g(P_i) \subseteq \mathrm{Aff}(g(P_i))$, we have that g is a homomorphism from Γ_P to Θ_Γ. We claim that h is a homomorphism from Θ_Γ to Γ_Q. In other words, for all $(P_i, Q_i) \in \Gamma$, we seek to show that $h(\mathrm{Aff}(g(P_i))) \subseteq Q_i$. For any $V \in \mathrm{Aff}(g(P_i))$, we have that there exist $X^1,\dots,X^k \in P_i$ as well as ring elements $r_1,\dots,r_k \in R$ such that $r_1 + \cdots + r_k = 1$ and

$$V = r_1 g(X^1) + \ldots + r_k g(X^k).$$

For some sufficiently large $L \equiv r \mod M$ for which $\mathrm{PER}_{M,\eta,L} \in \mathrm{poly}(\Gamma)$, pick nonnegative integers $w_1,\dots,w_k$ such that $w_i \equiv r_i r \mod M$ and $w_1 + \cdots + w_k = L$. By starting with the w_i's as small as possible and then increment by M, this is possible as long as $L \geq Mk$. Now, since $\mathrm{PER}_{M,\eta,L} \in \mathrm{poly}(\Gamma)$, we have that

$$Y := \mathrm{PER}_{M,\eta,L}(\underbrace{X^1,\dots,X^1}_{w_1 \text{ copies}},\dots,\underbrace{X^k,\dots,X^k}_{w_k \text{ copies}}) \in Q_i.$$

For each coordinate $i \in \{1,\dots,\mathrm{ar}_i\}$, we have that by definition of PER,

$$Y_i = \eta\left(\sum_{j=1}^k w_j X_i^j \mod M\right) = \eta\left(\sum_{j=1}^k r_j (r X_i^j) \mod M\right) = h\left(\sum_{j=1}^k r_j g(X^j)\right) = h(V_i).$$

Since $h(V) = Y \in Q_i$, we know that h is a homomorphism from Θ_Γ to Γ_Q, as desired.

Since R is a finite commutative ring, we have that R is LE-solvable. Thus, by Theorem 3.2, we have that $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. Since $h = \eta$ is a constant-sized function, $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$, as desired. $\square$

4.3 Threshold-periodic Polymorphisms

It turns out that threshold polymorphisms and periodic polymorphisms can be combined in nontrivial ways

Definition 4.4. Let $T = \{\tau_0 = 0, \tau_1, \dots, \tau_k = 1\}$ be a threshold sequence, $M = (M_0, \dots, M_k)$ be a sequence of positive integers, and $H = (\eta_1, \dots, \eta_k)$ be a sequence of maps $\eta_i : \mathbb{Z}/M_i\mathbb{Z} \to E$. Let L be a positive integer such that $L\tau_i$ is not an integer for any $i \in \{1, \dots, k-1\}$. Then, define $\text{THR-PER}_{T,M,H,L} : \{0,1\}^L \to E$ to be the following polymorphism.

$$\text{THR-PER}_{T,M,H,L}(x) = \begin{cases} \eta_1(0) & \text{Ham}(x) = 0 \\ \eta_i(\text{Ham}(x) \mod M_i) & L\tau_{i-1} < \text{Ham}(x) < L\tau_i, 1 \le i \le k \\ \eta_k(L) & \text{Ham}(x) = L. \end{cases}$$

For technical reasons, we have to have that values at Hamming weights 0 and L be consistent with the periodic patterns in the intervals $(0, \tau_1)$ and $(\tau_{k-1}, 1)$, respectively.

Theorem 4.3. *Let T, M, H be defined as above. Let Γ be a promise template on the Boolean domain such that* $\text{THR-PER}_{T,M,H,L} \in \text{poly}(\Gamma)$ *for infinitely many L. Then,* $\text{PCSP}(\Gamma) \in \mathsf{P}$.

Proof. Let $M_{\text{lcm}} = \text{lcm}(M_0, \dots, M_k)$. Like in the periodic case, there must be some $r \in \mathbb{Z}/M_{\text{lcm}}\mathbb{Z}$ such that $L \equiv r \mod M_{\text{lcm}}$ for infinitely many L for which $\text{THR-PER}_{T,M,H,L} \in \text{poly}(\Gamma)$.

Pick an LP-solvable ring A such that $\tau_i \notin A$ for all $i \in \{1, \dots, k-1\}$. Let $R = \mathbb{Z}/M_{\text{lcm}}\mathbb{Z}$. We claim that there is a promise embedding of Γ into $\Xi_{A,R}$ via (g,h) where

$$\begin{aligned} g(0) &= (0,0) \in A \times R \\ g(1) &= (1,r) \in A \times R \\ h(x,y) &= \begin{cases} \eta_1(y \mod M_0) & x \le 0 \\ \eta_i(y \mod M_i) & \tau_{i-1} < x < \tau_i, 1 \le i \le k \\ \eta_k(y \mod M_k) & x \ge 1 \end{cases} \end{aligned}$$

The justification of the embedding is a merging of the methods of Theorem 4.1 and Theorem 4.2. Since we desire ”access” to each coordinate of g, we let g_A be the first coordinate and g_R be the second coordinate.

Consider $\Xi_\Gamma := \{R_i := (\text{Conv}_A(g_A(P_i))_1), \text{Aff}_R(g_R(P_i))) : P_i \in \Gamma_P\}$ note that $\Xi_\Gamma \subset \Xi_{A,R}$. By design, g is a homomorphism from Γ_P to Ξ_Γ. Thus, it suffices to show that for any $(V,W) \in R_i$, we have that $h(V,W) \in Q_i$.

By definition, if $(V,W) \in R_i$, there exists $X^1, \dots, X^m \in P_i$ as well as $\alpha_1, \dots, \alpha_m \in [0,1]$ summing to 1 and $r_1, \dots, r_m \in R$ summing to 1 such that[17]

$$\begin{aligned} V &= \alpha_1 g_A(X^1) + \cdots + \alpha_m g_A(X^m) \\ W &= r_1 g_R(X^1) + \cdots + r_m g_R(X^m). \end{aligned}$$

Pick L sufficiently larger (to be specified) such that $\text{THR-PER}_{T,M,H,L} \in \text{poly}(\Gamma)$ with $L \equiv r \mod M_{\text{lcm}}$. We now need to delicately find integer weights $w_1, \dots, w_m$ such that the following properties hold

$$\begin{aligned} \sum_{i=1}^m w_i &= L \\ w_i &\equiv r_i r \mod M \text{ for all } i \\ |w_i - \alpha_i L| &\le M \text{ for all } i. \end{aligned}$$

[17] The reason these can be simultaneously true is that we can set some α_i's and r_i's to 0.

Note that the first two conditions are consistent because $\sum_{i=1}^m r_i r \equiv r \equiv L \mod M$. Such w_i's can be constructed by first setting each w_i to be the greatest integer at most $\alpha_i L$ which is equivalent to $r_i r \mod M$. Then, one can increase the w_i's by M one-by-one until they sum to L.

With these in hand, consider

$$Y := \text{THR-PER}_{T,M,H,L}(\underbrace{X^1,\ldots,X^1}_{w_1\text{ copies}},\ldots,\underbrace{X^m,\ldots,X^m}_{w_m\text{ copies}}) \in Q_i.$$

Define for each coordinate $j \in \{1,\ldots,\mathrm{ar}_i\}$

$$s_j^A := \frac{1}{L}\,\mathrm{Ham}(\underbrace{X_j^1,\ldots,X_j^1}_{w_1\text{ copies}},\ldots,\underbrace{X_j^m,\ldots,X_j^m}_{w_m\text{ copies}})$$

$$s_j^R := \sum_{a=1}^m w_a X_j^a \mod M$$

Then, by design, $Y_j = h(s_j^A, s_j^R)$. We also know that

$$\frac{1}{L}|s_j^A - V_j| = \sum_{\substack{a=1\\X_j^a=1}}^k \left|\frac{w_a}{L} - \alpha_a\right| \le \frac{Mm}{L}$$

as well as

$$s_j^R = \sum_{a=1}^m r_j(rX_i^j) \mod M = \sum_{a=1}^m r_j g_R(X_i^j) = W_j.$$

Since $V \in A^{\mathrm{ar}_i}$, we have that $\tau_{i-1} < V_j < \tau_i$ for $i \in \{2,\ldots,k-1\}$ or $\tau_0 \le V_j < \tau_1$ or $\tau_{k-1} < V_j \le \tau_k$. In any case, as $L \to \infty$, s_j^A will get sufficiently close to V_j so that it falls into the same interval as V_j. Thus, $Y_j = h(s_j^A, s_j^R) = h(V_j, W_j)$. Therefore, $h(V,W) = Y \in Q_i$, establishing the promise embedding is valid.

Since A is LP-solvable and R is LE-solvable, by Theorem 3.3, we have that $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. Since h only needs to check thresholds and then use a finite lookup table, h can be computed in polynomial time in the description of the input. Thus, $\mathrm{PCSP}(\Gamma) \in \mathsf{P}$, as desired. □

5 Regional Boolean polymorphisms

So far, all of the families of weak polymorphisms we have studied are Boolean, symmetric; that is, they only depend on the Hamming weight of the input vector. This section describes how these results can be extended to special kinds of *block symmetric* functions. Like in the previous section, our promise domain is always $(D = \{0,1\}, E, \phi)$.

Definition 5.1. Let b and L be positive integers. A function $f : D^L \to E$ is b-block symmetric, if there is a partition $[L] = B_1 \cup B_2 \cup \cdots \cup B_b$ such that for all $(x_1,\ldots,x_L) \in D^L$ and any permutation $\pi : [L] \to [L]$ such that $\pi(B_i) = B_i$ for all i.

$$f(x_1,\ldots,x_L) = f(x_{\pi(1)},\ldots,x_{\pi(L)}).$$

In other words, f is b-block symmetric with the corresponding partition $B_1 \cup \cdots \cup B_b$, then, $f(x)$ depends only on $(\mathrm{Ham}_{B_1}(x),\ldots,\mathrm{Ham}_{B_b}(x))$, where $\mathrm{Ham}_{B_i}(x)$ is the sum of the coordinates with indices in B_i. Analogous to how a symmetric function can be thought of as a function on the real interval $[0,1]$, a b-block symmetric function can be thought of as a function on $[0,1]^b$.

5.1 Regional Weak Polymorphisms

Even going from $[0,1]$ to $[0,1]^2$, the ways of splitting up space can become rather complex. Thus, instead of giving an explicit description like for threshold polymorphisms, we discuss a generalization which we call *open partition weak polymorphisms*. First, we need to define what an *open partition* is.

Definition 5.2. Let $A_1,\dots,A_b \subset \mathbb{R}$ be dense commutative rings. Let $\mathfrak{A} := (A_1 \times A_2 \times \cdots \times A_b) \cap [0,1]^b$. Let E be a set. A function $\mathrm{Part} : \mathfrak{A} \to E$ is an *open partition* if for all $x \in \mathfrak{A}$, there exists $\varepsilon > 0$, such that for all $y \in \mathfrak{A}$ with $|x-y| < \varepsilon$, we have $\mathrm{Part}(y) = \mathrm{Part}(x)$. In other words, for all $e \in E$, $\mathrm{Part}^{-1}(e)$is open in the Euclidean topology induced by $\mathfrak{A}$.

We also have a slightly more general notion called an *integer open partition* which allows for arbitrary values to be set at the corners of the hypercube $[0,1]^b$.

Definition 5.3. Let $A_1,\dots,A_b \subset \mathbb{R}$ be dense commutative rings. Let $\mathfrak{A} := (A_1 \times A_2 \times \cdots \times A_b) \cap [0,1]^b$. Let E be a set. A function $\mathrm{Part} : \mathfrak{A} \to E$ is an *integer open partition* if for all $x \in \mathfrak{A} \setminus \{0,1\}^b$, there exists $\varepsilon > 0$, such that for all $y \in \mathfrak{A}$ with $|x-y| < \varepsilon$, we have $\mathrm{Part}(y) = \mathrm{Part}(x)$. In other words,for all $e \in E$, $\mathrm{Part}^{-1}(e) \setminus \{0,1\}^b$ is open in the Euclidean topology induced by $\mathfrak{A}$.

Going back to the 1-dimensional case, consider $A_1 = \mathbb{Z}[\sqrt{2}]$ so that $\mathfrak{A} = A_1 \cap [0,1]$. Also, let our range be $E = \{0,1\}$. The partition corresponding to the MAJ polymorphism is then

$$\mathrm{Part}_{\mathrm{MAJ}}(x) = \begin{cases} 0 & x < 1/2 \\ 1 & x > 1/2. \end{cases}$$

This function is an open partition because the apparent boundary element $1/2$ does not exist in $\mathbb{Z}[\sqrt{2}]$. On the other hand, the partition corresponding to the AND polymorphism.

$$\mathrm{Part}_{\mathrm{AND}}(x) = \begin{cases} 0 & x < 1 \\ 1 & x = 1 \end{cases}$$

is not an open partition because $\mathrm{Part}^{-1}(1)$ has boundary. Yet, it is an integer open partition because the only boundary term has integer coordinates.

A more complex example in two dimensions is as follows. Let $\mathfrak{A} = (\mathbb{Z}[\sqrt{2}] \times \mathbb{Z}[\sqrt{3}]) \cap [0,1]^2$ and let

$$\mathrm{Part}_{\mathrm{AT}}(x,y) = \begin{cases} 0 & x < y \\ 1 & x > y \\ 0 & x = y = 0 \\ 1 & x = y = 1 \end{cases}$$

See Figure 1. Note that since the two coordinates are in the rings $\mathbb{Z}[\sqrt{2}]$ and $\mathbb{Z}[\sqrt{3}]$, $x = y$ if and only if x and y are both integers. Thus, the only boundary terms have integer coordinates, so $\mathrm{Part}_{\mathrm{AT}}$ is an integer open partition. As hinted by the name, $\mathrm{Part}_{\mathrm{AT}}$ is connected to the family of weak polymorphisms AT_L. This connection is made more explicit soon.

For a more nontrivial example, consider $E = \{0,1,2,3,4\}$ and $\mathfrak{A} = (\mathbb{Z}[\sqrt{2})^2 \cap [0,1]^2$. Let

$$\mathrm{Part}_{\mathrm{circle}}(x,y) = \begin{cases} 0 & x < 1/2 \text{ and } y < 1/2 \text{ and } (x-1/2)^2 + (y-1/2)^2 > 1/13 \\ 1 & x < 1/2 \text{ and } y > 1/2 \text{ and } (x-1/2)^2 + (y-1/2)^2 > 1/13 \\ 2 & x > 1/2 \text{ and } y < 1/2 \text{ and } (x-1/2)^2 + (y-1/2)^2 > 1/13 \\ 3 & x > 1/2 \text{ and } y > 1/2 \text{ and } (x-1/2)^2 + (y-1/2)^2 > 1/13 \\ 4 & (x-1/2)^2 + (y-1/2)^2 < 1/13 \end{cases}$$

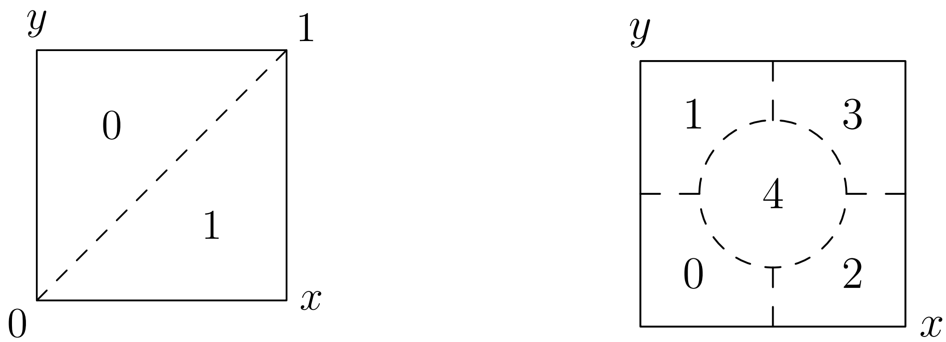

Figure 1: Plots of $\mathrm{Part}_{\mathrm{AT}}(x,y)$ and $\mathrm{Part}_{\mathrm{circle}}(x,y)$. The dashed lines represent the boundary between the regions. The 0 and 1 in the corners of the square for $\mathrm{Part}_{\mathrm{AT}}(x,y)$ represents the value chosen at those corners.

In this case, $\mathrm{Part}_{\mathrm{circle}}$ is an open partition, since the equations $x = 1/2$, $y = 1/2$ and $(x-1/2)^2 + (y-1/2)^2 = 1/13$ have no solutions in $(\mathbb{Z}[\sqrt{2}])^2$.

Although an integer open partition Part is only defined in $\mathfrak{A}$, we can extend it to a substantial portion of $[0,1]^b$. This is useful when we desire to discretize Part by wanting know its value at particular rational coordinates (which may not be in $\mathfrak{A}$).

Definition 5.4. Let $\mathrm{Part} : \mathfrak{A} \to E$ be an integer open partition. Define $\overline{\mathrm{Part}} : [0,1]^b \to E \cup \{\perp\}$ to be the partial function for which

$$\overline{\mathrm{Part}}(x) = \begin{cases} \mathrm{Part}(x) & x \in \{0,1\}^n \\ e \in E & \exists \varepsilon > 0, \forall y \in \mathfrak{A}, |x-y| < \varepsilon \text{ implies } \mathrm{Part}(x) = e \\ \perp & \text{otherwise.} \end{cases}$$

Note that since Part is an integer open partition, $\mathrm{Part}(x) = \overline{\mathrm{Part}}(x)$ for all $x \in \mathfrak{A}$. Since $\mathfrak{A}$ is dense in $[0,1]^b$, and $\mathfrak{A} \subset (\overline{\mathrm{Part}})^{-1}(E)$ is open, we have that $(\overline{\mathrm{Part}})^{-1}(\perp) = [0,1]^b \setminus (\overline{\mathrm{Part}})^{-1}(E)$ is nowhere dense, although it may have positive Lebesgue measure.

As alluded to earlier, these partitions Part, which will end up being our rounding functions, are discretized to form a collection of polymorphisms. Recall our domain $D = \{0,1\}$ is Boolean for this section.

Definition 5.5. Let $A_1, \ldots, A_b \subset \mathbb{R}$ be dense commutative rings. Let $\mathfrak{A} = A_1 \times \cdots \times A_b \cap [0,1]^b$. Let $\mathrm{Part} : \mathfrak{A} \to E$ be an integer open partition. Let $L_1, \ldots, L_b$ be positive integers such that for all $k_i \in \{0, 1, \ldots, L_i\}$ for all $k_i \in [b]$, we have that $\overline{\mathrm{Part}}\left(\frac{k_1}{L_1}, \ldots, \frac{k_b}{L_b}\right) \neq \perp$. Let $L = \sum_{i=1}^{b} L_i$ and let $\mathscr{B} = (B_1, \ldots, B_b)$ be a partition of $[L]$ such that $|B_i| = L_i$ for all $i \in \{1, \ldots, b\}$. Define the *regional weak polymorphism* $\mathrm{REG}_{\mathrm{Part},\mathscr{B}} : D^{B_1} \times \cdots \times D^{B_b} \to E$ to be

$$\mathrm{REG}_{\mathrm{Part},\mathscr{B}}(x) = \overline{\mathrm{Part}}\left(\frac{\mathrm{Ham}_{B_1}(x)}{L_1}, \cdots \frac{\mathrm{Ham}_{B_b}(x)}{L_b}\right).$$

For example, $\mathrm{REG}_{\mathrm{Part}_{\mathrm{AT}},(\{1,3,\ldots,2k+1\},\{2,4,\ldots,2k\})}$ is the same as AT_{2k+1} up to a permutation of the coordinates.

Now, we can prove that having an infinite collection of regional weak polymorphisms implies tractability as long as Part is efficiently computable.

Theorem 5.1. *Let $A_1,\ldots,A_b \subset \mathbb{R}$ be LP-solvable subrings. Let $\mathfrak{A} = A_1 \times \cdots \times A_b \cap [0,1]^b$. Let $\mathrm{Part} : \mathfrak{A} \to E$ be an integer open partition. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i} : i \in I\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\})$ be a promise template. Assume that for all positive integers ℓ, there exists $\mathrm{REG}_{\mathrm{Part},\mathscr{B}} \in \mathrm{poly}(\Gamma)$ such that $|B_i| \ge \ell$ for all $B_i \in \mathscr{B}$. Then,* $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^{\mathrm{Part}}$.

Proof. Let $\mathscr{A} = (A_1,\ldots,A_b)$. By Theorem 3.3, it suffices to show that there is a promise embedding of Γ into $\Xi_{\mathscr{A}}$. The embedding map $g : D \to \mathfrak{A}$ is just

$$g(d) = \begin{cases} (0,\ldots,0) & d = 0 \\ (1,\ldots,1) & d = 1. \end{cases}$$

As suggested by the theorem statement, the rounding map is precisely[18] the integer open partition $h = \mathrm{Part}$.

Now, let $\Xi_\Gamma = \{R_i := \mathrm{Conv}_{A_1}(g_1(P_i)) \times \cdots \times \mathrm{Conv}_{A_b}(g_b(P_i)) \in \mathfrak{A}^{\mathrm{ar}_i} : i \in I\}$. By design, g is a homomorphism from Γ_P to Ξ_Γ. The heart of the argument is to show that h is a weak polymorphism from Ξ_Γ to Γ_Q. In other words, we need to show for all $i \in I$, that $h(R_i) \subseteq Q_i$. Fix $V \in R_i$ and view $V = (V_1,\ldots,V_b) \in A_1^{\mathrm{ar}_i} \times \cdots \times A_b^{\mathrm{ar}_i}$. With $V_a = (V_{a,1},\ldots,V_{a,\mathrm{ar}_i}) \in A_a^{\mathrm{ar}_i}$.

List the elements $X^1,\ldots,X^m \in P_i$. For all $a \in \{1,2,\ldots,b\}$, because $V_a \in \mathrm{Conv}_{A_a}(g_1(P_i))$, we have that there exists weights $\alpha_{a,j} \in [0,1]$ with $j \in \{1,\ldots,m\}$ such that

$$V_a = \sum_{j=1}^m \alpha_{a,j} X^j.$$

(Note that g can be omitted, since it is the identity map on each coordinate.) Pick ℓ sufficiently large (to be determine later), such that $\mathrm{REG}_{h,\mathscr{B}} \in \mathrm{poly}(\Gamma)$ and $|B_a| \ge \ell$ for all $B_a \in \mathscr{B}$. Then, using a nearly identical argument as the one in Theorem 4.1, we can find integer weights $w_{a,j}$ such that $\sum_{j=1}^m w_{a,j} = |B_a|$ for all $a \in \{1,2,\ldots,b\}$ and

$$\left|\frac{w_{a,j}}{|B_a|} - \alpha_{a,j}\right| \le \frac{1}{|B_a|} \le \frac{1}{\ell}.$$

Furthermore, we can ensure that $w_{a,j} = 0$ whenever $\alpha_{a,j} = 0$ Fix $k \in \{1,\ldots,\mathrm{ar}_i\}$. There are essentially two cases to consider

- If $W_k := (V_{1,k},\ldots,V_{b,k}) \in [0,1]^b \setminus \{0,1\}^b$, consider $\varepsilon > 0$ such that $\overline{\mathrm{Part}}(x) = \overline{\mathrm{Part}}(W_k)$ for all $|x - W_k| < \varepsilon$. Then, if ℓ is chosen such that $\frac{b}{\ell} < \varepsilon$, then for each $k \in \{1,\ldots,\mathrm{ar}_i\}$

$$\begin{aligned}
&\mathrm{REG}_{\mathrm{Part},\mathscr{B}}(\underbrace{X_k^1,\ldots,X_k^1}_{w_{1,1}\text{ copies}},\ldots,\underbrace{X_k^m,\ldots,X_k^m}_{w_{1,m}\text{ copies}},\underbrace{X^1,\ldots,X^1}_{w_{2,1}\text{ copies}},\ldots,\underbrace{X_k^m,\ldots,X_k^m}_{w_{2,m}\text{ copies}},\ldots\underbrace{X_k^1,\ldots,X_k^1}_{w_{b,1}\text{ copies}},\ldots,\underbrace{X_k^m,\ldots,X_k^m}_{w_{b,m}\text{ copies}})\\
&\qquad = \overline{\mathrm{Part}}\left(\frac{\sum_{j=1}^m w_{1,j}X_k^j}{|B_1|},\ldots,\frac{\sum_{j=1}^m w_{b,j}X_k^j}{|B_b|}\right)\\
&\qquad = \overline{\mathrm{Part}}\left(\sum_{j=1}^m \alpha_{1,j}X_k^j,\ldots,\sum_{j=1}^m \alpha_{b,j}X_k^j\right) \text{ (within } \varepsilon)\\
&\qquad = \overline{\mathrm{Part}}(W_k)\\
&\qquad = \mathrm{Part}(W_k).
\end{aligned}$$

[18]Technically, the domain of h is $A_1 \times \cdots \times A_b$, whereas the domain of Part is $[0,1]^b$. This can be "fixed" by having h return a default value (e.g., 0) when the input is outside $[0,1]^b$.

- Otherwise, if $W_k \in \{0,1\}^b$, whenever $\alpha_{a,j} \neq 0$, we must have that $V_{a,k} = X_k^j$. Since $\alpha_{a,j} = 0$ implies $w_{a,j} = 0$, we have that

$$\begin{aligned}
&\mathrm{REG}_{\mathrm{Part},\mathscr{B}}(\text{same as above}) \\
&\qquad = \overline{\mathrm{Part}}\left(\frac{\sum_{j=1}^m w_{1,j}X_k^j}{|B_1|}, \dots, \frac{\sum_{j=1}^m w_{1,j}}{|B_b|}\right) \\
&\qquad = \overline{\mathrm{Part}}\left(\frac{\sum_{j=1}^m w_{1,j}V_k^j}{|B_1|}, \dots, \frac{\sum_{j=1}^m w_{1,j}}{|B_b|}\right) \\
&\qquad = \overline{\mathrm{Part}}(W_k) \\
&\qquad = \mathrm{Part}(W_k) \text{ (because } W_k \in \{0,1\}^b.
\end{aligned}$$

In either case, we have that $\mathrm{Part}(V)$ is the output of $\mathrm{REG}_{\mathrm{Part},\mathscr{B}}(P_i) \subseteq Q_i$. Thus, we have the aforementioned promise embedding. ☐

5.2 Regional Periodic Weak Polymorphisms

Just as threshold polymorphisms can be generalized to threshold-periodic polymorphisms, we have that regional polymorphisms can be generalized to *regional periodic* weak polymorphisms.

Recall that if $\mathfrak{A} := A_1 \times \cdots \times A_b \cap [0,1]$, where the A_i's are dense commutative rings, then $\mathrm{Part} : \mathfrak{A} \to E$ is an open partition if for all $e \in E$, $\mathrm{Part}^{-1}(e)$ is relatively open with respect to the Euclidean topology induced by $\mathfrak{A}$. In other words, for all $e \in E$, there exists $\Omega_e \subset \mathbb{R}^b$ open such that $\mathrm{Part}^{-1}(e) = \Omega_e \cap \mathfrak{A}$. We call Ω_e a *region* of Part. Note that for all $x \in \Omega_e \cap [0,1]^b$, $\overline{\mathrm{Part}}(x) = e$. Given this, we can now define *regional periodic weak polymorphisms*.

Definition 5.6. Let $A_1, \dots, A_b \subset \mathbb{R}$ be dense commutative rings. Let $\mathfrak{A} = A_1 \times \cdots \times A_b$ be a product of subrings of $\mathbb{R}$. Let S be a finite set, and let $\mathrm{Part} : \mathfrak{A} \to S$ be an open partition. Let $L_1, \dots, L_b$ be positive integers such that for all $k_i \in \{0,1,\dots,L_i\}$ for all $i \in [b]$, we have that $\overline{\mathrm{Part}}\left(\frac{k_1}{L_1}, \dots, \frac{k_b}{L_b}\right) \neq \perp$. Let $L = \sum_{i=1}^b L_i$ and let $\mathscr{B} = (B_1, \dots, B_b)$ be a partition of $[L]$ such that $|B_i| = L_i$ for all $i \in [b]$. For each $k \in S$, let J_k be an ideal of $\mathbb{Z}^b$ such that $\mathbb{Z}^b/J_k$ is finite. Let $\mathscr{M} = \{M_k : \mathbb{Z}^b/J_k \to E \mid k \in S\}$ be a collection of maps. Define the *regional periodic polymorphism* $\text{REG-PER}_{\mathrm{Part},\mathscr{B},\mathscr{M}} : D^{B_1} \times \cdots \times D^{B_b} \to E$ to be

$$\text{REG-PER}_{\mathrm{Part},\mathscr{B},\mathscr{M}}(x) = M_k((\mathrm{Ham}_{B_1}(x), \dots, \mathrm{Ham}_{B_b}(x)) \bmod J_k) \text{ where } k = \overline{\mathrm{Part}}\left(\frac{\mathrm{Ham}_{B_1}(x)}{L_1}, \cdots \frac{\mathrm{Ham}_{B_b}(x)}{L_b}\right).$$

Figure 2 shows an example of a partition with periodic functions added. Now, we can prove a more general result.

Theorem 5.2. *Let $A_1, \dots, A_b \subset \mathbb{R}$ be LP-solvable rings. Let $\mathfrak{A} = A_1 \times \cdots \times A_b \cap [0,1]^b$. Let $\mathrm{Part} : \mathfrak{A} \to S$ be an open partition. Let $\mathscr{M} = \{M_k : \mathbb{Z}^b/J_k \to E \mid k \in S\}$ be a collection of maps. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i} : i \in I\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\})$ be a promise template. Assume that for all positive integers ℓ, there exists $\mathrm{REG}_{\mathrm{Part},\mathscr{B},\mathscr{M}} \in \mathrm{poly}(\Gamma)$ such that $|B_i| \geq \ell$ for all $B_i \in \mathscr{B}$. Then, $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^{\mathrm{Part}}$.*

The proof has similar structure to the proof of the threshold-periodic case, Theorem 4.3.

Proof. Given a sequence of blocks $\mathscr{B}$, we can define its *residue* with respect to $\mathscr{M}$ to be the sequence

$$\mathrm{Res}_{\mathscr{M}}(\mathscr{B}) = (\mathscr{B} \mod J_k : k \in S).$$

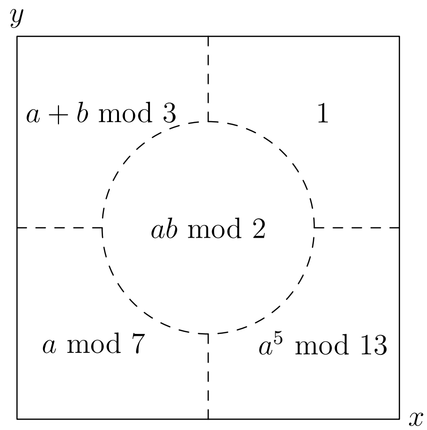

Figure 2: Plot of $\text{Part}_{\text{circle}}(x,y)$ in the same style as Figure 1. Within each region is a function $(a,b) \mapsto M_i(a,b)$ whose domain is some quotient of $\mathbb{Z}^2$, where a and b represent the Hamming weights of each block in the corresponding regional periodic polymorphism.

Note that since $\mathbb{Z}^b/J_k$ is a finite quotient for all $k \in S$, the set of all possible residues is finite. Thus, there exists a residue $\hat{r} := (\hat{r}_k : k \in S)$ such that $\hat{r} = \text{Res}_{\mathscr{M}}(\mathscr{B})$ for infinitely many $\mathscr{B}$ such that $\text{REG}_{\text{Part},\mathscr{B},\mathscr{M}} \in \text{poly}(\Gamma)$ and $\min\{|B_i|\}$ is arbitrarily large.

We now apply the ring-theoretic Chinese Remainder Theorem to our quotient rings. Let $J = \bigcap_{k \in S} J_k$ be the intersection of the ideals of $\mathbb{Z}^b$. Note that J is also an ideal of $\mathbb{Z}^b$. Furthermore, $\mathbb{Z}^b/J$ is finite, as any two elements of $x, y \in \mathbb{Z}^b$ with the same residue satisfy $x - y \in J_k$ for all $k \in S$, so $x - y \in J$, implying a finite number of cosets. This also implies that we can identify $\hat{r}$ with an element of $\mathbb{Z}^b/J$.

Now, let $\mathscr{A} = (A_1, \ldots, A_b)$ and $\mathscr{R} = (\mathbb{Z}^b/J)$. Recall that $\Xi_{\mathscr{A},\mathscr{R}}$ is the direct product

$$\Xi_{\mathscr{A},\mathscr{R}} = \left(\bigtimes_{j=1}^{b} \Lambda_{A_j} \right) \times \Theta_{\mathbb{Z}^b/J}.$$

We claim that there is a promise embedding of Γ into $\Xi_{\mathscr{A},\mathscr{R}}$ via the maps (g,h) where

$$\begin{aligned} g(0) &= (\underbrace{0, \ldots, 0}_{b \text{ terms}}, 0) \\ g(1) &= (\underbrace{1, \ldots, 1}_{b \text{ terms}}, \hat{r}) \\ h(x_1, \ldots, x_b, r) &= M_k(r) \text{ where } k = \text{Part}(x_1, \ldots, x_b). \end{aligned} \tag{1}$$

Analogous to the proof of Theorem 4.3, define

$$\Xi_\Gamma := \{R_i := (\text{Conv}_{A_1}(g_1(P_i)), \ldots, \text{Conv}_{A_b}(g_b(P_i)), \text{Aff}_{\mathbb{Z}^b/J}(g_{b+1}(P_i))) : P_i \in \Gamma_P\}.$$

As before, by definition, g is a homomorphism from Γ_P to Ξ_Γ. Thus, it suffices to show

$$\text{for all } (V_1, \ldots, V_b, W) \in R_i, \text{ we have that } h(V_1, \ldots, V_b, W) \in Q_i.$$

Let $X^1, \ldots, X^m$ be the elements of P_i. For all $j \in [b]$, since $V_j \in \text{Conv}_{A_j}(g_j(P_i))$, we have that there exists $\alpha_{j,1}, \ldots, \alpha_{j,m} \in [0,1]$ summing to 1 such that

$$V_j = \alpha_{j,1} g_a(X^1) + \cdots + \alpha_{j,m} g_a(X^m).$$

Likewise, since $W \in \mathrm{Aff}_{\mathbb{Z}^b/J}(g_{b+1}(P_i))$, we have that there exist $r_1, \ldots, r_m \in \mathbb{Z}^b/J_m$ which sum to the identity $(1, \ldots, 1) \in \mathbb{Z}^b/J_a$ such that

$$W = r_1 g_{b+1}(X^1) + \cdots + r_m g_{b+1}(X^m).$$

Fix ℓ sufficiently large (to be specified later) and $\mathscr{B} = (B_1, \ldots, B_b)$ with $|B_j| \geq \ell$ for all $j \in [b]$ such that $\mathrm{Res}_{\mathscr{M}}(\mathscr{B}) = \hat{r} \in \mathbb{Z}^b/J$. For any such $j \in [b]$ find[19] weights $w_{j,1}, \ldots, w_{j,m}$ satisfying the following conditions:

$$\sum_{k=1}^{m} w_{j,k} = |B_j| \text{ for all } j \in [b] \qquad \text{(cardinality condition)}$$

$$(w_{1,k}, \ldots, w_{b,k}) \in r_k \hat{r} + J \text{ for all } k \in [m] \qquad \text{(coset condition)}$$

$$\left| w_{j,k} - \alpha_{j,k} |B_{j,k}| \right| \leq 2|\mathbb{Z}^b/J| bm \text{ for all } j \in [b], k \in [m]. \qquad \text{(approximation condition)}$$

Now consider

$$\begin{aligned} Y := \text{REG-PER}_{\text{Part},\mathscr{B},\mathscr{M}}(&\underbrace{X^1, \ldots, X^1}_{w_{1,1} \text{ copies}}, \ldots, \underbrace{X^m, \ldots, X^m}_{w_{1,m} \text{ copies}}, \\ &\underbrace{X^1, \ldots, X^1}_{w_{2,1} \text{ copies}}, \ldots, \underbrace{X^m, \ldots, X^m}_{w_{2,m} \text{ copies}}, \ldots \\ &\underbrace{X^1, \ldots, X^1}_{w_{b,1} \text{ copies}}, \ldots, \underbrace{X^m, \ldots, X^m}_{w_{b,m} \text{ copies}}) \in Q_i. \end{aligned}$$

We seek to prove that $h(V_1, \ldots, V_b, W) = Y$, showing that $h(V_1, \ldots, V_b, W) \in Q_i$.

For each $j \in [b]$ and each coordinate $k \in \{1, \ldots, \mathrm{ar}_i\}$ (recall ar_i is the arity of P_i, Q_i and R_i) we can define

$$\begin{aligned} s_k^{A_j} &:= \frac{1}{|B_j|} \mathrm{Ham}(\underbrace{X_k^1, \ldots, X_k^1}_{w_{j,1} \text{ copies}}, \ldots, \underbrace{X_k^m, \ldots, X_k^m}_{w_{j,m} \text{ copies}}). \\ &= \frac{1}{|B_j|} \sum_{\beta=1}^{m} w_{j,\beta} X_k^{\beta} \\ &\approx \sum_{\beta=1}^{m} \alpha_{j,\beta} X_k^{\beta} = V_{j,k}, \end{aligned}$$

where $\approx$ means $O(\frac{1}{\ell})$ error. Thus, if ℓ is sufficiently large, for all $k \in [\mathrm{ar}_i]$, $(s_k^{A_1}, \ldots, s_k^{A_b})$ will be in the same region of Part as $(V_{1,k}, \ldots, V_{b,k})$ because the regions are relatively open.

[19] This can be done by first estimating $\hat{w}_{j,k} = \lfloor \alpha_{j,k} |B_j| \rfloor$ and then adjusting each as little as possible (at most $|\mathbb{Z}^b/J|$) so that $(\hat{w}_{1,k}, \ldots, \hat{w}_{b,k})$ are in the appropriate cosets. Then, it is not hard to check by the compatibility of the conditions that

$$T := (|B_1|, \ldots, |B_b|) - \left(\sum_{k=1}^{m} w_{1,k}, \ldots, \sum_{k=1}^{m} w_{b,k} \right) \in J.$$

If we add T to $(\hat{w}_{1,1}, \ldots, \hat{w}_{b,1})$, then the cardinality and coset constraints are satisfied. Note that the entries of T are bounded by $|\mathbb{Z}^b/J|m$, so the approximation condition is also still satisfied.

Furthermore, for all $k \in [\mathrm{ar}_i]$ define

$$\begin{aligned} s_k^{\mathscr{R}} &= \sum_{\beta=1}^{m} (w_{1,\beta}, \ldots, w_{b,\beta}) X_k^\beta \\ &\in J + \sum_{\beta=1}^{m} r_j (\hat{r} X_k^\beta) \\ &= J + \sum_{\beta=1}^{m} r_j g_{b+1}(X_k^\beta) = W_k. \end{aligned}$$

Thus, for all $k \in [\mathrm{ar}_i]$,

$$\begin{aligned} Y_k &= M_{\overline{\mathrm{Part}}(s_k^{A_1}, \ldots, s_k^{A_b})}(s_k^{\mathscr{R}}) \\ &= M_{\mathrm{Part}(V_{1,k}, \ldots, V_{b,k})}(W_k) \text{ (by above discussion)} \\ &= h_k(V_1, \ldots, V_b, W) \text{ (by (1))}. \end{aligned}$$

Thus, $Y = h(V_1, \ldots, V_b, W)$, so we have established the promise embedding. Since each A_i is LP-solvable, and $\mathbb{Z}^b/J$ is a finite commutative ring (and so is LE-solvable), by Theorem 3.3, we have that $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^{\mathrm{Part}}$. □

6 Extending to Larger Domains

Given the established framework, the extension from Boolean to non-Boolean domains is not much more difficult. The main change is that instead of embedding the domain D in the interval $[0,1]$, we embed in the *standard D-simplex*.

For a finite domain D, we denote $\mathbb{R}^D$ as the set of possible $|D|$-tuples of real numbers, indexed by elements of D. The standard D-simplex is defined to be $\Delta^D := \{x \in \mathbb{R}^D : x_d \geq 0 \text{ for all } d \in D, \sum_{d \in D} x_d = 1\}$.

Thus, for the Basic LP, working with symmetric or block-symmetric functions on non-Boolean domains is similar to working with block-symmetric Boolean polymorphisms, except that the domains are simplices or Cartesian products of simplices instead of the hypercube.

The Affine relaxations are also nearly identical, as we can embed our CSP over the domain D into some quotient of $\mathbb{Z}^D$, since all finite commutative rings are LE-solvable.

As a result, regional and regional periodic weak polymorphisms can be extended to non-Boolean domains with only minor changes in their definitions. Likewise, their corresponding theorems can be proved with nearly identical proofs. As a result, we reserve giving the details of these extensions for a future version of the paper.

7 Conclusion

Our algorithms show how rich and diverse algorithms can be for promise CSPs as in comparison to classical CSP theory. In particular, finite promise CSPs can often demand algorithms which require infinite domains! There are many challenges for extending these algorithmic results to wider classes of weak polymorphisms. These challenges range from more topological inquiries to fundamental questions about infinite-domain CSPs.

One aspect of promise CSPs that was not utilized in this paper is that when the template Γ is finite, $\mathrm{poly}(\Gamma)$ is "finitizable" (c.f., [BG18]), which means that there exists a constant $R_\Gamma > 0$, such that $f \in \mathrm{poly}(\Gamma)$

if and only if all of its projections of arity R_Γ are in $\text{poly}(\Gamma)$. Such a property may give a topological foothold (e.g., compactness) which could allow for more general classification. For instance, it is certainly possible that if a Γ is finite, and $\text{poly}(\Gamma)$ contains (block) symmetric polynomials for arbitrarily large arities, then $\text{poly}(\Gamma)$ contains an infinite family of Regional or Regional Periodic (or some slight variant) polymorphisms with consistent parameters. To prove such a result, a topological theory of weak polymorphisms needs to be developed.

Another important question is whether generalizations of the Basic LP, such as the Sherali-Adams or Sum-of-Squares hierarchies correspond to classes of infinite CSPs into which we can embed finite Promise-CSPs. Semidefinite programming may be especially useful for non-Boolean domains, as there is an algorithm known for Example 7 of Section 2.2, using SDPs [folklore].

From a polymorphic standpoint, one might wonder if block-symmetric functions are the only tractable families? We conjecture that the tractable families may be captured by *block transitive* weak polymorphisms. that is $f : D^{B_1} \times \cdots \times D^{B_d} \to E$ which have the property that for all $i, j \in B_k$ for some k there is a permutation π of the coordinates such that $\pi(i) = j$.

Note that there is still much work that needs to be done on the hardness side of the dichotomy. As shown by the struggles of the hardness of approximation community to solve the approximate graph coloring problem, stronger versions of the PCP theorem are desired. The recent breakthrough on the 2–to–2 conjecture [DKK$^+$16, KMS17, DKK$^+$17, KMS18] is an encouraging step in this direction, although its impact on promise CSPs such as the approximate graph coloring problem is limited due to the fact that the current version lacks perfect completeness.

Another exciting direction for future exploration is understanding, for both CSPs and promise CSPs, what insight these polymorphisms shed on the existence of "fast" exponential-time algorithms for NP-hard instances. Such questions have been investigated for CSPs, most notably the work of [JLNZ13], which showed that the fundamental universal algebraic object in *partial polymorphisms*, maps $f : D^L \to D \cup \{\perp\}$ for which the tuples mapping to $\perp$ are ignored. They also identified the "easiest" NP-hard templates, but indicated that an exhaustive classification is currently out of reach. The perspective given in this work of considering threshold-periodic and regional-periodic polymorphisms can be easily extended to partial polymorphisms by adding $\perp$ as an extra element of the domain. The study of these families of partial polymorphisms and their utility in designing algorithms beating brute force will be the subject of future work.

Acknowledgments

The authors thank Anupam Gupta and Ryan O'Donnell for helpful discussions about linear programming.

A Proofs of the Promise Homomorphism Theorems

Theorem 3.1. *Let $A \subset \mathbb{R}^k$ be an LP-solvable ring. Let (D, E, ϕ) be a finite promise domain. Let $\Gamma = (\Gamma_P = \{P_i \in D^{\text{ar}_i} : i \in I\}, \Gamma_Q = \{Q_i \in E^{\text{ar}_i}\})$ be a finite promise CSP. Let $(g : D \to A, h : A \to E)$ be a promise embedding of Γ into Λ_A. Then $\text{PCSP}(\Gamma) \in \mathsf{P}^h$, in which P^h is the family promise languages which can be computed in polynomial time given oracle access to h.*

Proof of Theorem 3.1. We give an algorithm for both the decision and search version.

- Write the Basic LP relaxation of $\Psi_P(x_1,\ldots,x_n)$.
- Solve the Basic LP over the ring A to get a solution $(v_i \in A)_{i\in[n]}$. **Reject** if no solution.
- For all $i \in [n]$, set $y_i := h(v_i)$. **Accept** and output $(y_1,\ldots,y_n)$.

Algorithm A.1. Solving and rounding a Basic LP.

First we explain why this algorithm is correct. Assume Ψ_P has a satisfying assignment, then the Basic LP must also have a satisfying assignment. Let $\Lambda_\Gamma = \{R_i := \mathrm{Conv}_A(S_i) : i \in I, S_i \subset A^{\mathrm{ar}_i}\}$. Since g is a homomorphism from Γ_P to Λ_Γ, we have that $g(P_i) \subset R_i$ for all $i \in I$. In particular, this implies that $\mathrm{Conv}_A(g(P_i)) \subset R_i$. Thus, any solution to the Basic LP is a satisfying assignment of

$$\Psi_R(x_1,\ldots,x_n) := \bigwedge_{j\in J} R_{i_j}(x_{j_1},\ldots,x_{j_{\mathrm{ar}_{i_j}}}).$$

Now, since h is a homomorphism is a from Λ_Γ to Γ_Q, any satisfying assignment to Ψ_R (and thus to the Basic LP) maps via h to a satisfying assignment to Ψ_Q. Thus, the algorithm correctly solves the search problem, and thus it also solves the decision problem.

Finally, we explain why this algorithm lies in P^h. Note that that Basic LP can be computed in linear time in the size of Ψ_P, and thus the instance can be solved in polynomial time since A is LP-solvable. The "rounding" step uses an oracle to h, so $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. □

Theorem 3.2. *Let R be an LE-solvable ring. Let (D,E,ϕ) be a finite promise domain, and let $\Gamma = (\Gamma_P = \{P_i \in D^{\mathrm{ar}_i}\}, \Gamma_Q = \{Q_i \in E^{\mathrm{ar}_i}\})$ be a finite promise CSP over this promise domain. Let (g,h) be a promise embedding of Γ into Θ_R. Then,* $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$.

Proof of Theorem 3.2. Consider the following algorithm.

- Write the affine relaxation of $\Psi_P(x_1,\ldots,x_n)$.
- Solve the affine relaxation over the ring R to get a solution $r_1,\ldots,r_n \in R$. **Reject** if no solution.
- For all $i \in [n]$, set $y_i := h(r_i)$. **Accept** and output $(y_1,\ldots,y_n)$.

Algorithm A.2. Solving and rounding an affine relaxation.

First we explain why this algorithm is correct. Assume Ψ_P has a satisfying assignment, then the Affine relaxation must also have a satisfying assignment. Let $\Theta_\Gamma = \{R_i := \mathrm{Aff}_R(S_i) : i \in I, S_i \subset R^{\mathrm{ar}_i}\}$. Since g is a homomorphism from Γ_P to Λ_Γ, we have that $g(P_i) \subset R_i$ for all $i \in I$. In particular, this implies that $\mathrm{Aff}_R(g(P_i)) \subset R_i$. Thus, any solution to the Basic LP is a satisfying assignment of

$$\Psi_R(x_1,\ldots,x_n) := \bigwedge_{j\in J} R_{i_j}(x_{j_1},\ldots,x_{j_{\mathrm{ar}_{i_j}}}).$$

Now, since h is a homomorphism is a from Λ_Γ to Γ_Q, any satisfying assignment to Ψ_R (and thus to the Affine relaxation) maps via h to a satisfying assignment to Ψ_Q. Thus, the algorithm correctly solves the search problem, and thus it also solves the decision problem.

Like in the previous proof, the relaxation has size linear in the input. Since R is LE-solvable, the relaxation can be solved in polynomial time. The last step uses an oracle to h, so $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. $\square$

Theorem 3.3. *Let $\mathscr{A} := (A_1, \ldots, A_\ell)$ be a sequence of LP-solvable rings, and let $\mathscr{R} := (R_1, \ldots, R_m)$ be a sequence of LE-solvable rings. Let (D, E, ϕ) be a finite promise domain, and let $\Gamma = (\Gamma_P, \Gamma_Q)$ be a finite promise CSP over this domain. Let (g, h) be a promise embedding of Γ into $\Xi_{\mathscr{A},\mathscr{R}}$. Then,* $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$.

Proof of Theorem 3.3. We use an algorithm which is a combination of the techniques in Theorem 3.1 and Theorem 3.2.

- For each $A_j \in \mathscr{A}$
  - Write the Basic LP relaxation of $\Psi_P(x_1, \ldots, x_n)$.
  - Solve the Basic LP over the ring A to get a solution $(v_{j,i} \in A)_{i \in [n]}$. **Reject** if no solution.
- For each $R_j \in \mathscr{R}$
  - Write the affine relaxation of $\Psi_P(x_1, \ldots, x_n)$.
  - Solve the affine relaxation over the ring R to get a solution $r_{j,1}, \ldots, r_{j,n} \in R$. **Reject** if no solution.
- For all $i \in [n]$, set $y_i := h(v_{1,i}, \ldots, v_{\ell,i}, r_{1,i}, \ldots, r_{m,i})$. **Accept** and output $(y_1, \ldots, y_n)$.

Algorithm A.3. Solving multiple Basic LPs and affine relaxations with simultaneous rounding.

Let $\Xi_\Gamma = \{R_i : i \in I\}$ be the particular CSP with signature the same signature as Γ such that g is a homomorphism from Γ_P to Ξ_Γ and h is a homomorphism from Ξ_Γ to Γ_Q. Assume that Ψ_P has a satisfying assignment, then each Basic LP and Affine relaxation is satisfiable. Then, by the same logic as the previous two proofs, the solutions to all the linear programs and linear systems put together satisfies the corresponding instance of Ξ_Γ:

$$\Psi_R(x_1, \ldots, x_n) := \bigwedge_{j \in J} R_{i_j}(x_{j_1}, \ldots, x_{j_{\mathrm{ar}_{i_j}}}).$$

Finally, since h is a homomorphism from Ξ_Γ to Γ_Q, any satisfying assignment to Ψ_R maps to a satisfying assignment to Ψ_Q, so the algorithm is correct for the search version and thus also for the decision version.

Like in the previous proof, the relaxation has size linear in the input. Since each A_i is LP-solvable and each R_i is LE-solvable, the relaxation can be solved in polynomial time. The last step uses an oracle to h, so $\mathrm{PCSP}(\Gamma) \in \mathsf{P}^h$. $\square$

References

[AB92] Ilan Adler and Peter A. Beling. Polynomial algorithms for LP over a subring of the algebraic integers with applications to LP with circulant matrices. *Math. Program.*, 57:121–143, 1992.

[AB94] Ilan Adler and Peter A. Beling. Polynomial algorithms for linear programming over the algebraic numbers. *Algorithmica*, 12(6):436–457, 1994.

[ADFS04] N. Alon, I. Dinur, E. Friedgut, and B. Sudakov. Graph Products, Fourier Analysis and Spectral Techniques. *Geometric & Functional Analysis GAFA*, 14(5):913–940, 2004.

[AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad. (2+ε)-sat is np-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017.

[Bel01] Peter A. Beling. Exact algorithms for linear programming over algebraic extensions. *Algorithmica*, 31(4):459–478, 2001.

[BG16] Joshua Brakensiek and Venkatesan Guruswami. New hardness results for graph and hypergraph colorings. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 14:1–14:27. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[BG18] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1782–1801. SIAM, 2018. Full version available as ECCC TR16-183.

[BJK05] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the Complexity of Constraints Using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, January 2005.

[BKW17] Libor Barto, Andrei A. Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei A. Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[BR15] Jonah Brown-Cohen and Prasad Raghavendra. Combinatorial optimization algorithms via polymorphisms. *CoRR*, abs/1501.01598, 2015.

[Bul06] Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.

[Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In Umans [Uma17], pages 319–330.

[Che06] Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 37(4):85–114, 2006.

[DKK+16] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? *Electronic Colloquium on Computational Complexity (ECCC)*, 23:198, 2016.

[DKK+17] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. On non-optimally expanding sets in grassmann graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:94, 2017.

[FV98] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[GK04] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM J. Discrete Math.*, 18(1):30–40, 2004.

[Hua13] Sangxia Huang. Improved hardness of approximating chromatic number. In Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, volume 8096 of *Lecture Notes in Computer Science*, pages 233–243. Springer, 2013.

[JLNZ13] Peter Jonsson, Victor Lagerkvist, Gustav Nordh, and Bruno Zanuttini. Complexity of SAT problems, clone theory and the exponential time hypothesis. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1264–1277. SIAM, 2013.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[KLS00] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.

[KMS17] Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and grassmann graphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 576–589. ACM, 2017.

[KMS18] Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:6, 2018.

[KOT+12] Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 csps, and robust satisfaction. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 484–495. ACM, 2012.

[McD93] Colin McDiarmid. A Random Recolouring Method for Graphs and Hypergraphs. *Combinatorics, Probability and Computing*, 2(3):363–365, September 1993.

[Pip02] Nicholas Pippenger. Galois theory for minors of finite functions. *Discrete Mathematics*, 254(1-3):405–419, 2002.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254. ACM, 2008.

[Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In Richard J. Lipton, Walter A. Burkhard, Walter J. Savitch, Emily P. Friedman, and Alfred V. Aho, editors, *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 216–226. ACM, 1978.

[TZ16] Johan Thapper and Stanislav Zivny. The complexity of finite-valued csps. *J. ACM*, 63(4):37:1–37:33, 2016.

[Uma17] Chris Umans, editor. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. IEEE Computer Society, 2017.

[Zhu17] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In Umans [Uma17], pages 331–342.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il