Electronic Colloquium on Computational Complexity, Report No. 4 (2020)

The Power of the Combined Basic LP and Affine Relaxation for Promise CSPs*

Joshua Brakensiek† Venkatesan Guruswami‡ Marcin Wrochna§
Stanislav Živný¶

Abstract

In the field of constraint satisfaction problems (CSP), promise CSPs are an exciting new direction of study. In a promise CSP, each constraint comes in two forms: "strict" and "weak," and in the associated decision problem one must distinguish between being able to satisfy all the strict constraints versus not being able to satisfy all the weak constraints. The most commonly cited example of a promise CSP is the approximate graph coloring problem—which has recently seen exciting progress [BKO19, WZ20] benefiting from a systematic algebraic approach to promise CSPs based on "polymorphisms," operations that map tuples in the strict form of each constraint to tuples in the corresponding weak form.

In this work, we present a simple algorithm which in polynomial time solves the decision problem for all promise CSPs that admit infinitely many *symmetric* polymorphisms, that is the coordinates are permutation invariant. This generalizes previous work of the first two authors [BG19]. We also extend this algorithm to a more general class of *block-symmetric* polymorphisms. As a corollary, this single algorithm solves all polynomial-time tractable Boolean CSPs simultaneously. These results give a new perspective on Schaefer's classic dichotomy theorem and shed further light on how symmetries of polymorphisms enable algorithms. Finally, we show that block symmetric polymorphisms are not only sufficient but also necessary for this algorithm to work, thus establishing its precise power.

1 Introduction

A central challenge in the theory of algorithms is to understand the mathematical structure (or lack thereof) that governs the efficient tractability (or intractability) of a computational problem. For the class of constraint satisfaction problems (CSP), a rich algebraic theory culminating in the recent resolution of the Feder-Vardi dichotomy conjecture [FV98] in [Bul17, Zhu17] has established a striking link between problem structure and its tractability. In particular, a CSP is efficiently solvable if and only if its defining relations admit an "interesting" polymorphism. Informally, a polymorphism is a function whose component-wise action preserves membership in

*An extended abstract of part of this work (by the first two authors) appeared in the *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)* [BG20].

†Stanford University, Stanford, CA 94305, USA. Email: `jbrakens@stanford.edu`. Research supported in part by an REU supplement to NSF CCF-1526092 and a NSF Graduate Research Fellowship.

‡Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213. Email: `venkatg@cs.cmu.edu`. Research supported in part by NSF grants CCF-1526092 and CCF-1908125.

§University of Oxford, UK. Email: `marcin.wrochna@cs.ox.ac.uk`. Research supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532).

¶University of Oxford, UK. Email: `standa.zivny@cs.ox.ac.uk`. Research supported by a Royal Society University Research Fellowship and by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532).

ISSN 1433-8092

the relations defining the CSP, and "interesting" means that the function obeys some *non-trivial symmetry*. As an example, for the (efficiently solvable) CSP corresponding to linear equations over a ring R, the 3-ary function $f(x,y,z) = x-y+z$ is a polymorphism (capturing the fact that if v_1, v_2, v_3 are solutions to a linear system, then so is $v_1 - v_2 + v_3$), and it obeys the so-called Mal'tsev symmetry $f(x,y,y) = f(y,y,x) = x$ for all $x, y \in R$. Indeed, generalizing Gaussian elimination, any CSP with such a Mal'tsev polymorphism is efficiently tractable [Bul02, BD06].

Recently, an exciting new direction of study has emerged in the rich backdrop of the complexity dichotomy for CSPs. This concerns a vast generalization of the CSP framework to the class of *promise constraint satisfaction problems* (PCSP). In a promise CSP, each constraint comes in two forms: "strict" and "weak." Given an instance of a PCSP, one must distinguish between being able to satisfy all the strict constraints versus not being able to satisfy all the weak constraints. (This is the *decision* version; in the *search* version, given an instance with a promised assignment satisfying the strong form of the constraints, one seeks an assignment satisfying the weak form of the constraints.) A prime example of a PCSP is the approximate graph coloring problem, where one seeks to color a graph using more colors than its chromatic number.

The formal study of promise CSPs originated in [AGH17] who classified the complexity of a PCSP called $(2+\epsilon)$-SAT. They further defined an extension of polymorphisms to the promise setting and postulated that the structure of those polymorphisms might govern the complexity of a PCSP. (This extension of polymorphisms to the promise setting is quite natural, requiring that the operation map tuples obeying the strict form a constraint to a tuple satisfying its weak form.) Building on the impetus of [AGH17], Brakensiek and Guruswami systematically studied PCSPs under the polymorphic lens and established promising links to the universal-algebraic framework developed for CSPs [BG18, BG19]. It emerged from these works that a rich enough family of polymorphisms leads to efficient algorithms, whereas severely limited polymorphisms are a prescription for hardness. However, unlike for CSPs, there is no sharp transition between these cases — the significant difficulty being that, unlike for CSPs, polymorphisms for PCSPs are *not* closed under composition and lack the rich algebraic structure of a *clone* (c.f., [BKW17]). This nascent algebraic theory for PCSPs was lifted to a more abstract level in [BKO19, BBKO19] and also led to concrete breakthroughs in approximate graph coloring/homomorphisms [BKO19, KO19, WZ20]. In particular, while previous works [BG18, BG19] focused on the actual *form* of the polymorphisms, the results of [BKO19] reveal that it is not the polymorphisms themselves, but rather solely the *symmetries* they possess, that capture the complexity of the associated PCSP, extending a similar phenomenon known earlier for CSPs [BOP18].

This work concerns the theme of designing algorithms for PCSP based on a rich enough family of polymorphisms. Our main result is that the decision version of an arbitrary PCSP admitting an *infinite family of symmetric polymorphisms* — i.e., polymorphisms which are invariant under any permutation of inputs — is tractable (see Theorem 2). Our result also extends to the case of *block-symmetric* polymorphisms (see Theorem 3). That is, the coordinates can be partitioned into "blocks" such that the function is invariant under permutations within each block. Notably, in the block-symmetric case the algorithm is identical–only the analysis changes. Furthermore, the number of blocks is irrelevant, the only assumption we need is that the minimum block size can be made arbitrarily large. Our final result (Theorem 4) shows that block-symmetry is not only sufficient but also necessary for our algorithm to work. In fact, Theorem 4 also establishes that without loss of generality one can assume that there are only two blocks of symmetric coordinates.

Further our algorithm is very simple — it checks if the canonical linear programming (LP) relaxation of the PCSP is feasible, and if so, it further checks if a slight adaptation of a canonical *affine relaxation* is feasible. The algorithm outputs satisfiable if both these relaxations are feasible. The polymorphisms are not used in the algorithm itself and only enter the analysis.

The analysis is short but subtle — if we had symmetric polymorphisms of all arities then it is known that the basic LP relaxation itself correctly decides satisfiability, as one can round the fractional solution to a satisfying assignment using the polymorphism after clearing denominators of the fractional solution [KOT+12, BKW17]. If polymorphisms only exist of certain arities (e.g., all odd majorities), then the LP alone doesn't suffice (e.g., [KOT+12]). We solve a linear system over the integers corresponding to the affine relaxation which lets us adjust the LP solution to match the arity at which a polymorphism exists. As a subtle twist, the affine relaxation is not of the original PCSP, but rather a *refinement* of the CSP which results from throwing out assignments to constraints which were ruled out by the basic LP.

It should be pointed out that we only solve the *decision* version of the PCSP, and *not* the search version. Unlike CSPs, for promise CSP there is no known reduction from search to decision, even for special cases like approximate graph coloring. Our work might be indicative of the subtle relationship between the search and decision problems for promise CSPs.

We now compare our result here with the previous work [BG19] where an algorithm was given to solve (the search version of) any PCSP that admits an infinite family of *structured symmetric polymorphisms*. Examples of such structured families include threshold and threshold-periodic polymorphisms. The value a threshold polymorphism (for a Boolean PCSP) depends on whether the fraction of 1s in the input belongs in a finite number of intervals. (A basic example consists of Majority functions of odd arities, which are polymorphisms for 2-SAT.) A threshold-periodic polymorphism can have a periodic behavior depending on which interval the Hamming weight belongs to — for example it can be Majority for relative weights in $(1/3, 2/3)$ and parity outside this interval. More generally, one can generalize to the non-Boolean case, as well as for the block-symmetric case, via regional polymorphisms whose value depends on the geometric region in which the vector of frequencies of the inputs to the polymorphisms lies. Due to this geometric interpretation, [BG19] assumes a fixed number of blocks (corresponding to a fixed dimension), whereas our new algorithm and analysis is independent of the number of blocks. The algorithm was a combination of solving the LP relaxation (albeit over a special ring like $\mathbb{Z}[\sqrt{2}]$ rather than the rationals) and the affine relaxation over a large enough finite ring. The analysis relied on the special *structure* of the polymorphisms (beyond their full symmetry). In contrast, our result here is more general, and only requires the polymorphism to be a symmetric function — its exact specifics or structure do not matter. It is encouraging that our methodology is consistent with the algebraic result in [BKO19] that the symmetries possessed by the polymorphisms capture the complexity of the PCSP.

Our result and methods have significance even for normal (non-promise) CSPs. For instance, we get a single unified algorithm to solve all non-trivial tractable cases of Boolean CSPs in Schaefer's classic dichotomy theorem [Sch78], namely 2-SAT, Horn-SAT (or its dual), and Mod-2 Linear Equations. The two main techniques to solve CSPs are local propagation based algorithms (which work for the so-called *bounded-width* CSPs [BK14, KOT+12], etc.) and Gaussian elimination (which is a global algorithm that works for linear equations). The major difficulty in proving the full CSP dichotomy was tackling the complicated ways in which these two very different algorithms might need to be interlaced to solve a general CSP. It is our hope that this work serves as an impetus toward the potential discovery of a more modular CSP algorithm that incorporates together linear programming or its extensions (like Sherali-Adams, or semidefinite programming) and linear equation solving. In this light, it is encouraging that full symmetry of the polymorphisms, which is indeed a strong assumption, is not the limit of our techniques, which also extend to the block-symmetric case.

To put this work in further context, except for [BG19] as mentioned previously, nearly all works in the PCSP literature [AGH17, BG18, FKOS19] focus primarily on the structure of the *relations*. In particular, [BG18, FKOS19] characterized the complexity of all *Boolean symmetric relations* (rather than *symmetric polymorphisms*) which encompass many of the known tractable cases of Boolean PCSP. As classified by [FKOS19], all the relevant tractable polymorphisms are either symmetric functions or one special case of block-symmetric known as alternative threshold (and variants). Thus, in the context of PCSPs, the algorithm in this paper supersedes these previous works. See Section 4 for further discussion.

2 Notation

We let any finite set A or B denote a *domain.* We define a *signature* $\tau := (k_i : i \in I)$ to be an indexed list of integers. A *relation* is a subset $R \subseteq A^k$ for any positive integer k. A *template* with signature τ, often denoted by $\mathbf{A} = \{R_i^A \subseteq A^{k_i} : i \in I\}$, is an indexed set of relations over A such that the i-th relation has arity k_i. A *homomorphism* between templates $\mathbf{A} = \{R_i^A \subseteq A^{k_i}\}$ and $\mathbf{B} = \{R_i^B \subseteq B^{k_i}\}$ with the same signature is a map $\sigma : A \to B$ such that $\sigma(R_i^A) \subseteq R_i^B$ for all $i \in I$ (where σ is applied to a tuple component-wise).

Two templates for which there exists a homomorphism from the first to the second is called a *promise template* and is denoted as $(\mathbf{A}, \mathbf{B})$.

2.1 PCSP: Decision and Search

Consider two templates $\mathbf{A}$ and $\mathbf{B}$ with a homomorphism σ from $\mathbf{A}$ to $\mathbf{B}$. Formally, a *instance* of the *promise CSP* PCSP$(\mathbf{A}, \mathbf{B})$ is a *template* $\mathbf{X}$ with signature τ over domain $X := \{x_1, \ldots, x_n\}$, the *variables*. We say that $\mathbf{X}$ is *satisfiable* if there exists a homomorphism from $\mathbf{X}$ to $\mathbf{A}$ (which by composition implies a homomorphism to $\mathbf{B}$). We say that $\mathbf{X}$ is *unsatisfiable* if there does not exist a homomorphism $\mathbf{X}$ to $\mathbf{B}$.

We let PCSP-Decision(Γ) denote the *decision* problem of distinguishing between satisfiability and unsatisfiability. We let PCSP-Search(Γ) denote the *search* problem of finding an explicit homomorphism from $\mathbf{X}$ to $\mathbf{B}$.

2.2 Polymorphisms

A *polymorphism* of $(\mathbf{A}, \mathbf{B})$ is a map $f : A^L \to B$ such that for all $i \in I$, $R_i^B \supseteq f(R_i^A, \ldots, R_i^A)$ where we define the latter to be $\{(f(x_1^{(1)}, \ldots, x_1^{(L)}), \ldots, f(x_{k_i}^{(1)}, \ldots, x_{k_i}^{(L)})) : x^{(1)}, \ldots, x^{(L)} \in R_i^A\}$. In other words, consider any $A^{L \times \operatorname{ar} R_i^A}$ matrix M, where each *row* is a satisfying assignment to R_i. Let $X \in B^{\operatorname{ar} R_i^B}$ be the result of applying f to each *column* of M. Then, $X \in R_i^B$. We say that L is the *arity* of f. We let Pol$(\mathbf{A}, \mathbf{B})$ denote the set of polymorphisms of $(\mathbf{A}, \mathbf{B})$ (of all arities).

A map $f : A^n \to B$ is said to be *symmetric* if for all $\pi \in S_n$ (the symmetric group on n elements), $f(x_1, \ldots, x_n) = f(x_{\pi(1)}, \ldots, x_{\pi(n)})$.

2.3 Basic LP and Affine Relaxation

As is well-studied in the CSP literature (e.g., [RS09, TZ17]), we consider the canonical linear programming relaxation of a CSP instance, often refer to as the "Basic LP" or "BLP." For our CSP instance $\Psi_{\mathbf{A}}$, we represent the assignment to x_i by a (rational) probability distribution of weights $\{w_i(A)\}_{a \in A}$ summing to 1. We also have a probability distribution over the satisfying assignments to each constraint, which we denote as $p_j(y)$, where $j \in [m]$ is the index of the constraint and $y \in A_j$ is the potential assignment. Explicitly, the linear constraints are as follows.

$$w_i(a) \geq 0 \qquad \text{for all } i \in [n] \text{ and } a \in A \tag{1}$$

$$p_j(y) \geq 0 \qquad \text{for all } j \in [m] \text{ and } y \in A_i \tag{2}$$

$$\sum_{a \in A} w_i(a) = 1 \qquad \text{for all } i \in [n] \tag{3}$$

$$\sum_{y \in A_j} p_j(y) = 1 \qquad \text{for all } j \in [m] \tag{4}$$

$$\sum_{\substack{y \in A_j \\ y_i = a}} p_j(y) = w_i(a) \qquad \begin{matrix}\text{for all } i \in [n], a \in A, j \in [m] \\ \text{with } x_i \text{ in } A_j\end{matrix}. \tag{5}$$

We let $\mathrm{LP}_{\mathbb{Q}}(\Psi_{\mathbf{A}})$ denote the rational polytope of solutions. By a theorem of [GLS93] (c.f., [BG19]), we can efficiently find a *relative interior point* in this polytope. In particular, at such a point, each coordinate is positive if and only if it is positive at some point in the polytope.[1]

In addition to the Basic LP, we also consider the affine relaxation of a Promise CSP. In essence we solve the same linear system, but instead of enforcing each variable to be a nonnegative rational, we enforce that it is an integer (possibly negative). This can be solved in polynomial time via [KB79] (see also [BG19] for a more detailed discussion of this approach). We let $r_i(a) \in \mathbb{Z}$ replace $w_i(a)$ for all $a \in A$ and $q_i(y) \in \mathbb{Z}$ replace $p_i(y) \in \mathbb{Q}$. Explicitly,

$$\sum_{a \in A} r_i(a) = 1 \qquad \text{for all } i \in [n] \tag{6}$$

$$\sum_{y \in A_j} q_j(y) = 1 \qquad \text{for all } j \in [m] \tag{7}$$

$$\sum_{\substack{y \in A_j \\ y_i = a}} q_j(y) = r_i(a) \qquad \begin{matrix}\text{for all } i \in [n], a \in A, j \in [m] \\ \text{with } x_i \text{ in } A_j\end{matrix}. \tag{8}$$

We let $\mathrm{Aff}_{\mathbb{Z}}(\Psi_{\mathbf{A}})$ denote the integral lattice of solutions.

3 BLP+Affine Algorithm and analysis for symmetric polymorphisms

In the BLP+Affine algorithm, we seek to throw out any assignment to constraint for which the LP determines to have weight 0. In particular, for each constraint A_i, we let $A'_i \subseteq A_i$ be the set of assignments which have nonzero weight according to the LP solution. We let $\Psi'_{\mathbf{A}}$ be the *refined* CSP built from A'_i's instead of A_i's.

The algorithm is presented in Figure 1.

Definition 1. We say the BLP+Affine algorithm *solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ if for every instance $\mathbf{X}$, the existence of solutions $(w_i \colon A \to \mathbb{Q}_{\geq 0})_{i \in X}$ and $(r_i \colon A \to \mathbb{Z})_{i \in X}$ to the BLP and Affine relaxations of $\mathbf{X} \to \mathbf{A}$ such that $\mathrm{supp}(r) \subseteq \mathrm{supp}(w)$ implies the existence of a homomorphism $\mathbf{X} \to \mathbf{B}$.

As stated in the introduction, both the algorithm and the proof are structured similarly to those of [KOT+12] and [BG19]. Like in those works, the weights of the LP solution and affine relaxation are used to construct a list of assignments which are plugged into the relevant polymorphism. The novel contribution here is that a single argument can cover any infinite symmetric family of polymorphisms.

[1] For our specialized LP, we do not need such a hammer. We can instead solve the LP repeatedly, each time maximizing a different variable as the objective function. Averaging the results would then yield a solution such that each variable is positive if and only if it is positive in some LP solution.

1. Find a relative interior point in $\mathrm{LP}_{\mathbb{Q}}(\Psi_{\mathbf{A}})$. If no solution exists, **Reject**.
2. Refine $\Psi_{\mathbf{A}}$ to $\Psi'_{\mathbf{A}}$ by throwing out assignments to constraints which have weight 0 according to the relative interior point.
3. If $\mathrm{Aff}_{\mathbb{Z}}(\Psi'_{\mathbf{A}})$ is empty, **Reject**. Else, **Accept**.

Figure 1: BLP+Affine algorithm

Theorem 2. *Let* $(\mathbf{A}, \mathbf{B})$ *be a promise template over any finite domain such that* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *has symmetric polymorphisms of arbitrarily large arities. Then, BLP+Affine solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$.

Proof. If the Basic LP fails to have a solution, then $\Psi_{\mathbf{A}}$ is not satisfiable. Note that the refinement $\Psi'_{\mathbf{A}}$ includes every possible assignment which is in the support of some LP solution, including integral solutions. Thus, any solution to $\Psi_{\mathbf{A}}$ is also a solution to $\Psi'_{\mathbf{A}}$. Thus, if the affine relaxation of $\Psi'_{\mathbf{A}}$ fails to have a solution then $\Psi'_{\mathbf{A}}$ (and thus $\Psi_{\mathbf{A}}$) is unsatisfiable.

We use the notation of Section 2.3. Among all the coordinates in the LP solution–the w's and P's–let ℓ be the least common denominator of these rational numbers. Let M be the maximum absolute value of any integer which appears in the affine solution (both the variable weights and the constraint weights). Let $f : A^L \to B$ be a symmetric polymorphism of arity $L > (M+1)\ell^2$. Now write $L = u\ell + v$ where $u \in \mathbb{Z}^+$ and $v \in \{0, \ldots, \ell - 1\}$. Note that $u \geq (M+1)\ell$.

We seek to show there exists an assignment to $\Psi_{\mathbf{B}}$. For each $i \in [n]$ and $a \in A$, let

$$W_i(a) := u\ell w_i(a) + v r_i(a).$$

For a fixed $i \in [n]$, note that by Eq. (3) and (6)

$$\sum_{a \in A} W_i(a) = \sum_{a \in A} u\ell w_i(a) + v r_i(a) = u\ell + v = L.$$

Also, for fixed $i \in [n]$ and $a \in A$, either $w_i(a) = 0$, which implies that $r_i(a) = 0$ by the refinement, so $W_i(a) = 0$. Otherwise, $w_i(a) \geq 1/\ell$, so

$$W_i(a) \geq u\ell(1/\ell) + v(-M) \geq (M+1)\ell - \ell M > 0.$$

We claim that the assignment

$$X_i := f(\ldots, \underbrace{a, \ldots, a}_{W_i(a) \text{ times } \forall a \in A}, \ldots)$$

satisfies $\Psi_{\mathbf{B}}$–since f is symmetric, only the quantity of each $a \in A$ in the input matters. To verify this, fix a constraint B_j (with A'_j the corresponding constraint in $\Psi'_{\mathbf{A}}$), for $j \in [m]$ and assume WLOG it is on variables $x_1, \ldots, x_k$.

For all assignments $y \in A'_j$ define

$$P_j(y) := u\ell p_j(y) + v q_j(y).$$

By Eqs. 4 and 7,

$$\sum_{y \in A'_j} P_j(y) = u\ell \sum_{y \in A'_j} p_j(y) + v \sum_{y \in A'_j} q_j(y) = L.$$

Since $p_j(y) > 0$ for all $y \in A'_j$ by definition, we have by similar logic as for $W_i(a)$ that

$$P_j(y) \geq u\ell(1/\ell) + v(-M) \geq (M+1)\ell - \ell M > 0.$$

Further note that by Eqs. 5 and 8

$$\begin{aligned} W_i(a) &= u\ell \sum_{\substack{y\in A'_j \\ y_i = a}} w_i(a) + v \sum_{\substack{y\in A'_j \\ y_i = a}} r_i(a) \\ &= \sum_{\substack{y\in A'_j \\ y_i = a}} P_j(y) \end{aligned} \tag{9}$$

For each $j \in [m]$ consider a matrix $M(j) \in A^{L\times k}$, where exactly $P_j(y)$ of the rows are equal to y. When f is applied to the columns of $M(j)$, the result will satisfy $B_i(x_1, \ldots, x_k)$ in $\Psi_{\mathbf{B}}$. For all $i \in [k]$ and $a \in A$, the number of times that a appears in column i is precisely $W_i(a)$ by Eq. (9). Thus, f applied to the columns is precisely $(X_1, \ldots, X_k)$. In other words, the assignment of X_i for $i \in [n]$ satisfies $\Psi_{\mathbf{B}}$, so the algorithm is correct. □

Remark. Another algorithm which works is to solve the Basic LP of $\Psi_{\mathbf{A}}$, but to find the solution in $\mathbb{Z}[\sqrt{2}]$ instead of $\mathbb{Q}$, using the algorithm from [BG19]. In this case, Steps 2 and 3 can be omitted. This works because the algorithm for finding such a solution needs to solve the rational linear program and solve the subsequent linear system. Further details are omitted.

4 Extension of Analysis to Block Symmetric Polymorphisms

We say that a map $f : A^L \to B$. is *block-symmetric* if there exists a partition of the coordinates of f into blocks $B_1 \cup \cdots \cup B_k = [L]$ such that f is permutation-invariant within each coordinate block B_i. We define the *width* of f to be the minimum size of any block.[2] A natural example of a block symmetric polymorphism with nontrivial width is *alternating threshold* first studied in [BG18]

$$AT(x_1, \ldots, x_L) = 1[x_1 - x_2 + x_3 - \cdots \pm x_L \geq 1].$$

In this case, the blocks are the odd and even coordinates. This polymorphism arises in the context of $\mathbf{A}$ corresponding to 1-in-3 SAT and $\mathbf{B}$ corresponding to NAE-SAT. Recent work shows that this PCSP, although simple to state, is not reducible from any finite-domain CSP [BBKO19].

We now show an analogue of Theorem 2 for block-symmetric polymorphisms. Remarkably, the algorithm is identical to the one for ordinary symmetric polymorphisms and is independent of the number of blocks. In particular, it could be that the Promise CSP has finitely many polymorphisms for any particular number of blocks, yet has infinitely many block-symmetric polymorphisms of increasing width.

As discussed in [BG19, FKOS19], nearly all known tractable Boolean PCSPs have polymorphisms which are either symmetric (such as threshold functions) or block-symmetric (such as alternating threshold). Thus, except for those PCSPs which are "homomorphic relaxations" of a larger finite domain (P)CSP (c.f., [BG19, BBKO19]), the algorithm presented here supersedes those works in the context of decision PCSP.

Theorem 3. *Let* $(\mathbf{A}, \mathbf{B})$ *be a promise template over any finite domain such that* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *has block-symmetric polymorphisms of arbitrarily large width. Then,* PCSP-Decision(Γ) *has a polynomial-time algorithm.*

[2] Note that a function f might have different partitions into symmetric blocks, we define the width to be the maximum width over all such partitions. In particular, every $f : A^L \to B$ is block-symmetric with width 1.

Proof. The proof proceeds much like that of Theorem 2. As before, we know that if the algorithm rejects, when $\Psi_{\mathbf{A}}$ is unsatisfiable. We seek to show that if the algorithm accepts, then $\Psi_{\mathbf{B}}$ is satisfiable.

Again, let ℓ be the least common denominator of all coordinates in the LP solution. Let M be the maximum absolute value of any integer which appears in the affine solution. Let $f : A^{B_1 \cup \cdots \cup B_\kappa} \to B$ be a block-symmetric polymorphism such that each block B_b, with $b \in [\kappa]$, has size greater than $(M+1)\ell^2$. Let $L_b = |B_b|$. Similar to before, for all $b \in [\kappa]$, write $L_b = u_b\ell + v_b$ where $u_b \in \mathbb{Z}^+$ and $v \in \{0, \ldots, \ell - 1\}$. Note that $u_b \geq (M+1)\ell$.

We seek to show there exists an assignment to $\Psi_{\mathbf{B}}$. For each $b \in [\kappa]$, $i \in [n]$ and $a \in A$, let

$$W_{b,i}(a) := u_b \ell w_i(da + v_b r_i(a)).$$

For a fixed $b \in [\kappa]$ and $i \in [n]$, by similar logic to the proof of Theorem 2, we have that $W_{b,i}(a) \geq 0$ for all $a \in A$ and

$$\sum_{a \in A} W_{b,i}(a) = \sum_{a \in A} u_b \ell w_i(a) + v_b r_i(a) = u_b \ell + v_b = L_b.$$

We now claim that the assignment

$$X_i := f(\ldots, \underbrace{\underbrace{a, \ldots, a}_{W_{1,i}(a) \text{ times}}, \ldots, }_{L_1 \text{ total}} \ldots, \underbrace{\ldots, \underbrace{a, \ldots, a}_{W_{\kappa,i}(a) \text{ times}}, \ldots}_{L_\kappa \text{ total}})$$

satisfies $\Psi_{\mathbf{B}}$. To verify this, fix a constraint B_j (with A'_j the corresponding constraint in $\Psi'_{\mathbf{A}}$), for $j \in [m]$ and assume WLOG it is on variables $x_1, \ldots, x_k$.

For all $b \in [\kappa]$ and assignments $y \in A'_j$ define

$$P_{b,j}(y) := u_b \ell p_j(y) + v_b q_j(y).$$

By Eqs. 4 and 7,

$$\sum_{y \in A'_j} P_{b,j}(y) = u_b \ell \sum_{y \in A'_j} p_j(y) + v_b \sum_{y \in A'_j} q_j(y) = L_b.$$

By similar logic in previous arguments,

$$P_{b,j}(y) \geq u_b \ell (1/\ell) + v_b(-M) \geq (M+1)\ell - \ell M > 0.$$

Further note that by Eqs. 5 and 8

$$\begin{aligned} W_{b,i}(a) &= u_b \ell \sum_{\substack{y \in A'_j \\ y_i = a}} w_i(d) + v_b \sum_{\substack{y \in A'_j \\ y_i = a}} r_i(a) \\ &= \sum_{\substack{y \in A'_j \\ y_i = a}} P_{b,j}(y) \end{aligned} \tag{10}$$

For each $j \in [m]$ consider a matrix $M(j) \in A^{L \times k}$, where exactly $P_{b,j}(y)$ of the rows are equal to y in the rows indexed by block B_b. When f is applied to the columns of $M(j)$, the result will satisfy $B_i(x_1, \ldots, x_k)$ in $\Psi_{\mathbf{B}}$. For all $i \in [k]$ and $a \in A$, the number of times that a appears in column i and row-block B_b is precisely $W_{b,i}(a)$ by Eq. (10). Thus, f applied to the columns is precisely $(X_1, \ldots, X_k)$. In other words, the assignment of X_i for $i \in [n]$ satisfies $\Psi_{\mathbf{B}}$, so the algorithm is correct. $\square$

5 Characterizing the algorithm's power

In this section, we characterize the power of the BLP+Affine algorithm from Figure 1 exactly. Recall, we denote the domains of templates $\mathbf{A}, \mathbf{B}, \mathbf{X}$ as A, B, X.

Theorem 4. *Let* $\mathbf{A}, \mathbf{B}$ *be (finite) templates. The following are equivalent:*

- *BLP+Affine algorithm solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$*.*
- $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *has block-symmetric polymorphisms of arbitrarily high width.*
- *For every* $L \in \mathbb{N}$*,* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *has a block-symmetric polymorphism of arity* $2L+1$ *with two symmetric blocks of variables of width* L *and* $L+1$*, respectively.*

We need a few definitions and fundamental facts from [BKO19, BBKO19]. For a function $f\colon A^n \to B$ and a function $\pi\colon [n] \to [m]$, the *minor* of f obtained from π is the function $g\colon A^m \to B$ defined as

$$g(x_1, \ldots, x_m) := f(x_{\pi(1)}, \ldots, x_{\pi(n)}).$$

We write $g = f_{/\pi}$. Minors give the set of polymorphisms $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ the following structure.

Definition 5. A *minion* $\mathcal{M}$ consists of sets $\mathcal{M}^{(n)}$ for $n \in \mathbb{N}$ and functions $(\cdot)_{/\pi}\colon \mathcal{M}^{(n)} \to \mathcal{M}^{(m)}$ for all functions $\pi\colon [n] \to [m]$, such that compositions agree: $(f_{/\pi})_{/\tau} = f_{/\tau\circ\pi}$ for $\pi\colon [n] \to [m]$, $\tau\colon [m] \to [k]$, and $f_{/\mathrm{id}} = f$. We write $\mathcal{M}$ for the disjoint union of $\mathcal{M}^{(n)}$, $n \in \mathbb{N}$, and $\mathrm{ar}(f) = n$ for $f \in \mathcal{M}^{(n)}$.

A minion homomorphism $\xi\colon \mathcal{M} \to \mathcal{N}$ is a function which preserves arity and minors: $\mathrm{ar}(\xi(f)) = \mathrm{ar}(f)$ and $\xi(f_{/\pi}) = \xi(f)_{/\pi}$ for all functions $\pi\colon [n] \to [m]$.

Note that the objects in a minion do not have to be functions and the set $\mathcal{M}^{(L)}$ does not have to be finite, though this is the case for $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ with finite $\mathbf{A}, \mathbf{B}$. As an important example, consider the minion $\mathcal{Q}_{conv}$ of convex functions, i.e. functions of the form $w_1x_1 + \cdots + w_Lx_L$ for $\sum_1^L w_i = 1$, $w_i \in \mathbb{Q}_{\geq 0}$. We can describe the same minion more concisely by identifying a convex L-ary function with its L-tuple of coefficients $(w_1, \ldots, w_L)$. This minion characterizes the power of BLP in the sense that BLP solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ (as in Definition 1) if and only if $\mathcal{Q}_{conv}$ admits a minion homomorphism to $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$, as proved by Barto et al. [BBKO19, Theorem 7.9]. As our proof straightforwardly extends this part of the argument, we refer the reader to Appendix A for an exposition of it.

We now define the minion that plays the role of $\mathcal{Q}_{conv}$ for the BLP+Affine relaxation. It assigns two coefficients to every coordinate $i \in [L]$.

Definition 6. The minion $\mathcal{M}_{\mathrm{BLP+Aff}}$ is defined as follows: for $L \in \mathbb{N}$, the "L-ary objects" of the minion are

$$\begin{aligned} \mathcal{M}^{(L)}_{\mathrm{BLP+Aff}} := \{(w, r) \mid w\colon [L] \to \mathbb{Q}_{\geq 0}, &\qquad \textstyle\sum_{i\in[L]} w(i) = 1 \\ r\colon [L] \to \mathbb{Z}, &\qquad \textstyle\sum_{i\in[L]} r(i) = 1 \\ &\forall_{i\in[L]} \quad w(i) = 0 \implies r(i) = 0\ \}. \end{aligned}$$

Observe the domain of the minion is $\{(a, b) \in \mathbb{Q}_{\geq 0} \times \mathbb{Z} : a = 0 \implies b = 0\}$.

For $\pi\colon [L] \to [L']$ and $(w, r) \in \mathcal{M}^{(L)}_{\mathrm{BLP+Aff}}$, we define the minor $(w, r)_{/\pi}$ as (w', r'), where

$$\begin{aligned} w'(i) &:= w(\pi^{-1}(i)) = \textstyle\sum_{j\in\pi^{-1}(i)} w(j) \\ r'(i) &:= r(\pi^{-1}(i)), \qquad\qquad \text{for } i \in [L']. \end{aligned}$$

It is easy to check this indeed defines a minion (the $w(i) = 0 \implies r(i) = 0$ condition is preserved when taking a minor and composition of minors works as expected). One could also think of a pair $(w, r) \in \mathcal{M}^{(L)}_{\text{BLP+Aff}}$ as an L-ary function on $\mathbb{Q}^2$, $f(\binom{x_1}{y_1}, \dots, \binom{x_n}{y_n}) = \binom{\sum w(i)x_i}{\sum r(i)x_i}$.

As we explain in Appendix A, the minion $\mathcal{M}_{\text{BLP+Aff}}$ characterizes the BLP+Affine relaxation as follows.

Lemma 7. *Let* $\mathbf{A}, \mathbf{B}$ *be finite templates such that* $\mathbf{A} \to \mathbf{B}$*. The following are equivalent:*

- *BLP+Affine solves* $\operatorname{PCSP}(\mathbf{A}, \mathbf{B})$
- $\mathcal{M}_{\text{BLP+Aff}}$ *admits a minion homomorphism to* $\operatorname{Pol}(\mathbf{A}, \mathbf{B})$*.*

We now reinterpret this last condition in terms of concrete polymorphisms. One direction is simple:

Lemma 8. *Suppose* $\mathcal{M}_{\text{BLP+Aff}}$ *admits a minion homomorphism to some minion* $\mathcal{N}$*. Then for every* $L \in \mathbb{N}$*,* $\mathcal{N}$ *contains a block-symmetric polymorphism of arity* $2L + 1$ *with two blocks of width* L *and* $L + 1$*.*

Proof. Given $L \in \mathbb{N}$, consider the following object $(w, r) \in \mathcal{M}^{(2L+1)}_{\text{BLP+Aff}}$: take $w(i) := \frac{1}{2L+1}$ and $r(i) := (-1)^{i+1}$ for $i = 1, \dots, 2L + 1$. For every permutation $\pi\colon [L] \to [L]$ which maps odd coordinates to odd coordinates (and even to even), $(w, r)_{/\pi} = (w, r)$. Thus the image of (w, r) in $\mathcal{N}$ has the same property, i.e. it has arity $2L + 1$ and it is symmetric on odd coordinates as well as on even coordinates. $\square$

The idea for the other direction is essentially the same as in the proof of Theorem 2 and 3. We apply it to construct a minion homomorphism from every finite subset of $\mathcal{M}_{\text{BLP+Aff}}$ and use a compactness argument.

Lemma 9. *Suppose the minion* $\mathcal{N} = \operatorname{Pol}(\mathbf{A}, \mathbf{B})$ *(for* $\mathbf{A}, \mathbf{B}$ *finite) contains block-symmetric polymorphisms of arbitrarily high width. Then* $\mathcal{M}_{\text{BLP+Aff}}$ *admits a minion homomorphism to* $\mathcal{N}$*.*

Proof. To avoid cumbersome notation we present the proof only for the case of one block, i.e. we assume that $\mathcal{N}$ contains symmetric polymorphisms of arbitrarily high arity. This extends to more blocks just as Theorem 3 extends Theorem 2.

We define finite subsets of $\mathcal{M}_{\text{BLP+Aff}}$ as follows. For $L, \ell, M \in \mathbb{N}$, let $\mathcal{M}^{(L)}_{\ell,M}$ be the subset of those $(w, r) \in \mathcal{M}^{(L)}_{\text{BLP+Aff}}$ such that $\ell w(i) \in \mathbb{Z}$ for $i \in [L]$ and $\sum_i |r(i)| \leq M$. Observe that $\mathcal{M}^{(L)}_{\ell,M}$ is a finite set (since the numbers $\ell w(i)$ are L non-negative integers summing to ℓ and the numbers $r(i)$ are L integers between $-M$ and M). Denote $\mathcal{M}_{\ell,M} := \bigcup_{L\in\mathbb{N}} \mathcal{M}^{(L)}_{\ell,M}$.

For fixed ℓ, M, we define a minion homomorphism from $\mathcal{M}_{\ell,M}$ to $\mathcal{N}$ as follows. Let $f \in \mathcal{N}$ be a function of some arity $L^* \geq M\ell^2$. Let $u, v \in \mathbb{N}$ be numbers such that $L^* = u\ell + v$, $v \in \{0, \dots, \ell - 1\}$. Then $u \geq M\ell$.

Take $L \in \mathbb{N}$ and $(w, r) \in \mathcal{M}^{(L)}_{\ell,M}$. For $i \in [L]$, the number $n_i := u\ell w(i) + vr(i)$ is a non-negative integer. Since $\sum_i n_i = u\ell + v = L^*$, we can map (w, r) to the L-ary minor $g := f(x_1, x_1, x_1, \dots, x_L, x_L)$ of the L^*-ary function f where x_i is repeated n_i times, for $i \in [L]$. We claim that this map is a minion homomorphism from $\mathcal{M}_{\ell,M}$ to $\mathcal{N}$ (in fact to the subminion of minors of f). Indeed, for $\pi\colon [L] \to [L']$, consider the minor $g_{/\pi}$ of g identifying x_j for $j \in \pi^{-1}(i)$ into a single variable y_i (for $i \in [L']$). We have that $g_{/\pi}$ is also a minor of f where y_i is repeated $\sum_{j\in\pi^{-1}(i)} n_j$ times. That is, y_i is repeated $u\ell w(\pi^{-1}(i)) + vr(\pi^{-1}(i))$ times. By symmetry of f the ordering does not matter, thus $g_{/\pi}$ (the minor of the image of f) is the same as the image of the minor $f_{/\pi}$.

We conclude with a compactness argument similar to that of Remark 7.13 in [BBKO19]. For $k \in \mathbb{N}$, let $\mathcal{M}_k := \bigcup_{L \le k} \mathcal{M}^{(L)}_{k!,k}$. Then $\mathcal{M}_k$ is finite, $\mathcal{M}_k \subseteq \mathcal{M}_{k+1}$ and $\bigcup_{k \in \mathbb{N}} \mathcal{M}_k = \mathcal{M}_{\mathrm{BLP+Aff}}$. Consider the possible minion homomorphisms from $\mathcal{M}_k$ to $\mathcal{N}$, or more precisely, restrictions of homomorphisms obtained above to $\mathcal{M}_k$ (since $\mathcal{M}_k$ itself is technically not a minion). There are only finitely many possible such restrictions $\mathcal{M}_k \to \mathcal{N}$, because $\mathcal{M}_k$ is finite, the arities of images in $\mathcal{N}$ are bounded, and hence the number of possible images in $\mathcal{N}$ is also finite. Consider an infinite tree with restrictions from any $\mathcal{M}_k$ to $\mathcal{N}$ as nodes, the trivial map from $\mathcal{M}_0 = \emptyset$ being the root, and the parent of a function $\mathcal{M}_{k+1} \to \mathcal{N}$ being its restriction to $\mathcal{M}_k$. This is an infinite tree (because for each k we have some minion homomorphism from a superset of $\mathcal{M}_k$ to $\mathcal{N}$) that is connected (because everyone is connected through its ancestors to the root) and finitely branching (because there are only finitely many restrictions $\mathcal{M}_k \to \mathcal{N}$, for any fixed k). Therefore, by König's lemma, the tree contains an infinite path $\zeta_k \colon \mathcal{M}_k \to \mathcal{N}$ of homomorphisms that are restrictions of each other. Their union is then a homomorphism from $\bigcup_{k \in \mathbb{N}} \mathcal{M}_k = \mathcal{M}_{\mathrm{BLP+Aff}}$ to $\mathcal{N}$. $\square$

This concludes the proof of Theorem 4.

6 Concluding Thoughts

We conclude with a few natural directions of future inquiry raised by this work.

Inspecting the proofs of Theorems 2 and 3, in order to yield a search algorithm (and not just a decision algorithm), it would suffice to compute:

$$X_i := f(\dots, \underbrace{a, \dots, a}_{W_i(a) \text{ times}}, \dots).$$

In previous work [BG19], Brakensiek and Guruswami circumvented this problem by assuming that f has special structure (such as being a threshold function, etc.). Even then, we often only assumed that you had oracle access to the structure of f. Thus, except for some simple cases studied in the paper, truly polynomial-time search algorithms remain elusive. Perhaps one could hope for a search algorithm like the decision algorithm presented in this paper which is oblivious to the underlying polymorphisms (as long as they are symmetric/block-symmetric).

Question. Is there an "oblivious" polynomial-time algorithm for the search version of Promise CSPs with infinitely many symmetric polymorphisms?

We note that an oblivious polynomial-time algorithm is also not known for the search version of Promise CSPs with symmetric polymorphisms of all arities (which capture the power of BLP [BBKO19, Theorem 7.9]) and for the search version of Promise CSPs with alternating polymorphisms of all odd arities (which capture the power of the affine relaxation [BBKO19, Theorem 7.19]).

Otherwise, one could hope to prove a "structure theorem" that every Promise CSP with infinitely many symmetric polymorphisms also has an infinite threshold-periodic family. As [BG19] shows, such polymorphisms can get exceedingly complicated, suggesting that such a characterization may only be possible in the Boolean case.

Question. Does every Boolean PCSP with infinitely many symmetric polymorphisms have an infinite threshold-periodic family?

Even without a structure theorem, one could perhaps hope to compute the pertinent values of f "on the fly," but this seems difficult in our current formulation as the arity of f could be exponentially large in the input size!

While Theorem 4 characterizes the power of the BLP+Affine algorithm, it is still worthwhile to ask how this compares to other classes of templates, in particular those studied for non-promise CSPs. The following example of a simple template not solved by the BLP+Affine relaxation was communicated to us by Jakub Opršal.

Example 10. Let $\mathbf{A}$ be the disjoint union of a directed 2-cycle $\{0, 1\}$ and a directed 3-cycle $\{0', 1', 2'\}$. Then $\mathbf{A}$ is tractable template (i.e. $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$ is solvable in polynomial time, in fact $\mathrm{Pol}(\mathbf{A}, \mathbf{A})$ has cyclic polymorphisms of every prime arity $p > 3$) but has no non-trivial block-symmetric polymorphisms.

Proof. To see it admits no block-symmetric polymorphisms f of width greater than one, observe that every such width can be represented as $2n + 3n'$ for some $n, n' \in \mathbb{N}$, hence every block can be filled with n copies of values $0, 1$ and n' copies of $0', 1', 2'$, giving some input $\bar{v}$ to f. But f should give the same output on the input $\bar{v}^{\oplus 1}$ consisting of n copies of $1, 0$ and n' copies of $1', 2', 0'$. Since $(v_i, v_i^{\oplus 1})$ is an arc of $\mathbf{A}$ for every i and since f is a polymorphism, $(f(\bar{v}), f(\bar{v}^{\oplus 1}))$ would be a loop in $\mathbf{A}$, a contradiction.

We now observe that $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$ has a straightforward polynomial time algorithm. For each connected component of constraints, the variables must map to either $\{0, 1\}$ or $\{0', 1', 2'\}$. The first case is equivalent to testing if the graph of constraints is bipartite. The latter can be done by a breath-first search which checks that all directed cycles have length a multiple of 3. □

We also know that since $\mathrm{Pol}(\mathbf{A}, \mathbf{A})$ has a majority polymorphism (simply let $f(x, y, z)$ output x if $x = y$ and z otherwise), $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$ can be solved in polynomial time via the $(2, 3)$-consistency algorithm, 3-rounds of Sherali-Adams, or the canonical SDP relaxation (see also [BK14, TZ17, BKW17]). Informally, these relaxations ensure that there are locally consistent assignments to every (constant-sized) subset of variables. This consistency is quite powerful. For instance, 2-SAT can be solved by the BLP+Affine relaxation or 3 rounds of Sherali-Adams, but not the BLP by itself. This suggests the tantalising possibility that an analogous hierarchy could provide a uniform algorithm for all tractable non-promise CSPs.

Question. Which (decision) promise CSPs can be solved via constantly many rounds of the Sherali-Adams hierarchy for the BLP+Affine relaxation? Does this capture all tractable non-promise CSPs?

Acknowledgments

We thank Libor Barto, Andrei Krokhin, and Jakub Opršal for useful comments and encouragement. We also thank anonymous reviewers for many helpful comments.

A From relaxations to minion homomorphisms

In this section, we formally define the minion $\mathcal{Q}_{conv}$ and explain how free structures relate minions to BLP and Affine relaxations. We carry over the notation from Section 5.

Definition 11. The minion $\mathcal{Q}_{conv}$ is defined as follows: for $L \in \mathbb{N}$, the "L-ary object" of the minion are

$$\mathcal{Q}_{conv}^{(L)} := \{\ w \colon [L] \to \mathbb{Q}_{\geq 0} \ \mid\ \textstyle\sum_{i \in [L]} w(i) = 1\ \};$$

for $\pi\colon [L] \to [L']$ and $w \in \mathcal{Q}_{conv}^{(L)}$, we define the minor $w_{/\pi}$ of w as

$$w_{/\pi}(i) = w(\pi^{-1}(i)) = \sum_{j \in \pi^{-1}(i)} w(j).$$

Let us describe how $\mathcal{Q}_{conv}$ characterizes the power of the basic linear programming relaxation; the case of BLP+Affine will be entirely analogous. Recall that for an instance $\mathbf{X}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, a solution to the BLP relaxation assigns to each variable $i \in X$ a distribution $w_i\colon A \to \mathbb{Q}_{\geq 0}$ with $\sum_{a \in A} w_i(a) = 1$. It also assigns to each constraint j of $\mathbf{X}$ a distribution over satisfying assignments $p_j\colon R^A \to \mathbb{Q}_{\geq 0}$ with sum 1. Finally, the relaxation requires that for a variable i in a constraint j of $\mathbf{X}$, the assignment of $a \in A$ to i has value $w_i(a) = \sum_y p_j(y)$, where the sum runs over all satisfying assignments $y \in R^A$ of the constraint where the variable i takes value a.

In other words, $w_i(a) = p_j(\pi^{-1}(a))$, where $\pi = \pi_{j \to i}\colon R^A \to A$ maps a satisfying assignment y to the value of variable i in constraint j. That is, w_i, as an object of $\mathcal{Q}_{conv}^{|A|}$, is required to be the minor of $p_j \in \mathcal{Q}_{conv}^{|R^A|}$ obtained from π. Thus the BLP relaxation of $\mathbf{X}$ is satisfiable if and only if one can assign some $w_i \in \mathcal{Q}_{conv}^{|A|}$ to each variable $i \in X$ so that the following holds for every constraint j of $\mathbf{X}$: there is a $p_j \in \mathcal{Q}_{conv}^{|R^A|}$ such that for all variables i in j, $w_i = p_j/\pi_{j \to i}$. This can be phrased as the existence of a homomorphism from $\mathbf{X}$ to the following structure.

Definition 12. For a template $\mathbf{A}$, the *free structure* $\mathbb{F}_{conv}(\mathbf{A})$ is a template with domain $\mathcal{Q}_{conv}^{|A|}$. For each relation R^A of arity k in $\mathbf{A}$, there is a relation $R^{\mathbb{F}}$ of the same arity in $\mathbb{F}_{\mathrm{BLP+Aff}}(\mathbf{A})$ defined as follows: $w_1, \ldots, w_k \in \mathcal{Q}_{conv}^{(|A|)}$ are in the relation $R^{\mathbb{F}}$ if there is some $p \in \mathcal{Q}_{conv}^{(|R^A|)}$ such that for each $i \in [k]$, $w_i = p/\pi_i$. Here $\pi_i\colon R^A \to A$ maps $y \in R^A \subseteq A^k$ to its i-th coordinate $y[i] \in A$.

In essence, the free structure encodes the BLP+Affine relaxation of A as a CSP template.

Let us write $\mathbf{X} \to \mathbf{A}$ if there exists a homomorphism from $\mathbf{X}$ to $\mathbf{A}$. We can now say formally: an instance $\mathbf{X}$ of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ has a satisfiable BLP relaxation if and only if $\mathbf{X} \to \mathbb{F}_{conv}(\mathbf{A})$. Thus the BLP relaxation *solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ (as in Definition 1) if and only if for every instance $\mathbf{X}$, $\mathbf{X} \to \mathbb{F}_{conv}(\mathbf{A})$ implies $\mathbf{X} \to \mathbf{B}$. A standard compactness argument shows that this, in turn, is equivalent to $\mathbb{F}_{conv}(\mathbf{A}) \to \mathbf{B}$.

A fundamental property of free structures is that the condition $\mathbb{F}_{conv}(\mathbf{A}) \to \mathbf{B}$ is equivalent to the existence of a minion homomorphism from $\mathcal{Q}_{conv}$ to $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ (Lemma 4.4 in [BBKO19]). Finally, this last condition can be proved equivalent to the existence of symmetric polymorphisms of all arities $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$, concluding the characterization of the power of BLP for PCSPs:

Theorem 13 (Theorem 7.9 in [BBKO19])**.** *Let* $\mathbf{A}, \mathbf{B}$ *be finite templates such that* $\mathbf{A} \to \mathbf{B}$. *The following are equivalent:*

- *BLP solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ *(as in Definition 1),*
- $\forall_{\mathbf{X}} \quad \mathbf{X} \to \mathbb{F}_{conv}(\mathbf{A}) \implies \mathbf{X} \to \mathbf{B}$,
- $\mathbb{F}_{conv}(\mathbf{A}) \to \mathbf{B}$,
- $\mathcal{Q}_{conv}$ *admits a minion homomorphism to* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$,
- $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *contains symmetric polymorphisms of every arity.*

All this extends straightforwardly to the BLP+Affine relaxation, except for the last step. The definition of the free structure (Definition 12) applies to any minion $\mathcal{M}$ in place of $\mathcal{Q}_{conv}$, giving a template $\mathbb{F}_{\mathcal{M}}(\mathbf{A})$. We define an appropriate minion $\mathcal{M}_{\text{BLP+Aff}}$ in Section 5 and we denote $\mathbb{F}_{\text{BLP+Aff}}(\mathbf{A})$ as a shorthand for $\mathbb{F}_{\mathcal{M}_{\text{BLP+Aff}}}(\mathbf{A})$. Exactly as before, a simple compactness argument and Lemma 4.4 from [BBKO19] imply the following:

Lemma 14. *Let* $\mathbf{A}, \mathbf{B}$ *be finite templates such that* $\mathbf{A} \to \mathbf{B}$. *The following are equivalent:*

- *BLP+Affine solves* $\text{PCSP}(\mathbf{A}, \mathbf{B})$*,*
- $\forall_{\mathbf{X}}\ \ \mathbf{X} \to \mathbb{F}_{\text{BLP+Aff}}(\mathbf{A}) \implies \mathbf{X} \to \mathbf{B}$*,*
- $\mathbb{F}_{\text{BLP+Aff}}(\mathbf{A}) \to \mathbf{B}$*,*
- $\mathcal{M}_{\text{BLP+Aff}}$ *admits a minion homomorphism to* $\text{Pol}(\mathbf{A}, \mathbf{B})$*.*

This lemma is used in Section 5.

Let us note that in [BBKO19], the Affine relaxation alone was similarly characterized by the minion $\mathcal{Z}_{\text{aff}}$, defined analogously to $\mathcal{Q}_{conv}$, except with integer (not necessarily non-negative) coefficients: the L-ary objects are $r\colon [L] \to \mathbb{Z}$ such that $\sum_{i\in[L]} r(i) = 1$.

References

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

[BBKO19] Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *arXiv:1811.00970 [cs, math]*, 2019.

[BD06] Andrei Bulatov and Víctor Dalmau. A Simple Algorithm for Mal'tsev Constraints. *SIAM J. Comput.*, 36(1):16–27, July 2006.

[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.

[BG19] Joshua Brakensiek and Venkatesan Guruswami. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 436–455, Philadelphia, PA, USA, 2019. Society for Industrial and Applied Mathematics.

[BG20] Joshua Brakensiek and Venkatesan Guruswami. Symmetric polymorphisms and efficient decidability of PCSPs. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 297–304. SIAM, 2020. arXiv:1907.04383.

[BK14] Libor Barto and Marcin Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1):3:1–3:19, January 2014.

[BKO19] Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic Approach to Promise Constraint Satisfaction. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 602–613, New York, NY, USA, 2019. ACM.

[BKW17] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and How to Use Them. *Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany*, 2017.

[BOP18] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Israel Journal of Mathematics*, 223(1):363–398, February 2018.

[Bul02] Andrei A. Bulatov. Mal'tsev constraints are tractable. *Electronic Colloquium on Computational Complexity (ECCC)*, (034), 2002.

[Bul17] Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330. IEEE Computer Society, 2017.

[FKOS19] Miron Ficak, Marcin Kozik, Miroslav Olšák, and Szymon Stankiewicz. Dichotomy for Symmetric Boolean PCSPs. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:12, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[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.

[GLS93] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer Science & Business Media, 1993.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix. *SIAM Journal on Computing*, 8(4):499–507, November 1979.

[KO19] Andrei Krokhin and Jakub Opršal. The complexity of 3-colouring H-colourable graphs. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS'19)*. IEEE, 2019. arXiv:1904.03214.

[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.

[RS09] P. Raghavendra and D. Steurer. How to Round Any CSP. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 586–594, October 2009.

[Sch78] Thomas J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226. ACM, 1978.

[TZ17] Johan Thapper and Stanislav Živný. The Power of Sherali–Adams Relaxations for General-Valued CSPs. *SIAM Journal on Computing*, 46(4):1241–1279, 2017.

[WZ20] Marcin Wrochna and Stanislav Živný. Improved hardness for H-colourings of G-colourable graphs. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1426–1435. SIAM, 2020. arXiv:1907.00872.

[Zhu17] Dmitriy Zhuk. A Proof of CSP Dichotomy Conjecture. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342. IEEE Computer Society, 2017.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il