Hierarchies of Minion Tests for PCSPs through Tensors*†

Lorenzo Ciardo‡ Stanislav Živný§

Abstract

We provide a unified framework to study hierarchies of relaxations for Constraint Satisfaction Problems and their Promise variant. The idea is to split the description of a hierarchy into an algebraic part, depending on a minion capturing the "base level" of the hierarchy, and a geometric part – which we call tensorisation – inspired by multilinear algebra.

We show that the hierarchies of minion tests obtained in this way are general enough to capture the (combinatorial) bounded width and also the Sherali-Adams LP, Sum-of-Squares SDP, and affine IP hierarchies. We exploit the geometry of the tensor spaces arising from our construction to prove general properties of such hierarchies. We identify certain classes of minions, which we call *linear* and *conic*, whose corresponding hierarchies have particularly fine features. Finally, in order to analyse the Sum-of-Squares SDP hierarchy we also characterise the solvability of the standard SDP relaxation through a new minion.

1 Introduction

What are the limits of efficient algorithms and where is the precise borderline of tractability? The *constraint satisfaction problem* (CSP) offers a general framework for studying such fundamental questions for a large class of computational problems [46, 47, 74] but yet for a class that is amenable to identifying the mathematical structure governing tractability. Canonical examples of CSPs are satisfiability or "not-all-equal" satisfiability of 3-CNF formulas (called 3-Sat and 3-Nae-Sat, respectively), linear equations, several variants of (hyper)graph colourings, and the graph clique problem. All CSPs can be seen as homomorphism problems between relational structures [56]: Given two relational structures $\mathbf{X}$ and $\mathbf{A}$, is there a homomorphism from $\mathbf{X}$ to $\mathbf{A}$? Intuitively, the structure $\mathbf{X}$ represents the variables of the CSP instance and their interactions, whereas the structure $\mathbf{A}$ represents the constraint language; i.e., the alphabet and the allowed constraint relations.

The most studied types of CSPs are so-called *non-uniform* CSPs [65, 56, 71, 16], in which the target structure $\mathbf{A}$ is fixed whereas the source structure $\mathbf{X}$ is given on input; this computational problem is denoted by $\mathrm{CSP}(\mathbf{A})$. From the examples above, 3-Sat, 3-Nae-Sat, (hyper)graph colourings with constantly many colours, linear equations of bounded width over finite fields, and linear equations of bounded width over the rationals are all examples of non-uniform CSPs, all on finite domains except the last one [22, 18, 21]. For instance, in the graph c-colouring problem the target structure $\mathbf{A}$ is a c-clique and the structure $\mathbf{X}$ is the input graph. The existence of a homomorphism from a graph to a c-clique is equivalent to the existence of a colouring of the graph with c colours. The graph clique problem is an example of a CSP with a fixed class of source structures [59, 84] but an arbitrary target structure and, thus, it is not a non-uniform CSP.

We will be concerned with polynomial-time tractability of CSPs. Studied research directions include investigating questions such as: Is there a solution [35, 91]? How many solutions are there, exactly [45, 34, 54] or approximately [36, 39]? What is the maximum number of simultaneously satisfied constraints, exactly [44, 64, 88] or approximately [51, 7, 86]? What is the minimum number of simultaneously unsatisfied constraints [67, 50]? Given an almost satisfiable instance, can one find a somewhat satisfying solution [49, 14, 48]? In this paper, we will focus on the following question:

Given a satisfiable instance, can one find a solution that is satisfying in a weaker sense [9, 12, 25]?

*The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein. This work was also supported by UKRI EP/X024431/1.

†The full version of the paper can be accessed at `https://arxiv.org/abs/2207.02277`.

‡Department of Computer Science, University of Oxford, UK.

§Department of Computer Science, University of Oxford, UK.

This was formalised as *promise constraint satisfaction problems* (PCSPs) by Austrin, Guruswami and Håstad [9] and Brakensiek and Guruswami [25]. Let $\mathbf{A}$ and $\mathbf{B}$ be two fixed relational structures[1] such that there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, indicated by $\mathbf{A} \to \mathbf{B}$. Intuitively, the structure $\mathbf{A}$ represents the allowed "strict" constraints and the structure $\mathbf{B}$ represents the corresponding "weak" constraints. An instance of the PCSP over the template $(\mathbf{A}, \mathbf{B})$, denoted by $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, is a relational structure $\mathbf{X}$ such that there is a homomorphism from $\mathbf{X}$ to $\mathbf{A}$. The task is to find a homomorphism from $\mathbf{X}$ to $\mathbf{B}$, which exists by the composition of the two promised homomorphisms. What we described above is the *search* variant of the PCSP. In the *decision* variant, one is given a relational structure $\mathbf{X}$ and the task is to decide whether there is a homomorphism from $\mathbf{X}$ to $\mathbf{A}$ or whether there is not a homomorphism from $\mathbf{X}$ to $\mathbf{B}$. Note that since homomorphisms compose, if $\mathbf{X} \to \mathbf{A}$ then also $\mathbf{X} \to \mathbf{B}$. Thus, the two cases cannot happen simultaneously. It is known that the decision variant of the PCSP reduces to the search variant [12], but it is not known whether there is a reduction in the other direction for all PCSPs. In this paper, we shall use the decision variant.

PCSPs are a vast generalisation of CSPs including problems that cannot be expressed as CSPs. The work of Barto, Bulín, Krokhin, and Opršal [12] lifted and greatly extended the algebraic framework developed for CSPs [65, 33, 17] to the realm of PCSPs. Subsequently, there has been a series of recent works on the computational complexity of PCSPs building on [12], including applicability of local consistency and convex relaxations [24, 28, 37, 41, 5] and complexity of fragments of PCSPs [61, 75, 2, 11, 29, 26, 15, 85]. Strong results on PCSPs have also been established via other techniques than those in [12], mostly analytical methods, e.g., hardness of various (hyper)graph colourings [68, 53, 63, 8] and other PCSPs [19, 31, 27, 20].

An example of a PCSP, identified in [9], is (in the search variant) finding a satisfying assignment to a k-CNF formula given that a g-satisfying assignment exists; i.e., an assignment that satisfies at least g literals in each clause. Austrin et al. established that this problem is NP-hard if $g/k < 1/2$ and solvable via a constant level of the Sherali-Adams linear programming relaxation otherwise [9]. This classification was later extended to problems over arbitrary finite domains by Brandts et al. [29].

A second example of a PCSP, identified in [25], is (in the search variant) finding a "not-all-equal" assignment to a monotone 3-CNF formula given that a "1-in-3" assignment is promised to exist; i.e., given a 3-CNF formula with positive literals only and the promise that an assignment exists that satisfies exactly one literal in each clause, the task is to find an assignment that satisfies one or two literals in each clause. This problem is solvable in polynomial time via a constant level of the Sherali-Adams linear programming relaxation [25] but not via a reduction to finite-domain CSPs [12].

A third example of a PCSP is the well-known *approximate graph colouring* problem: Given a c-colourable graph, find a d-colouring of it, for constants c and d with $c \leq d$. This corresponds to $\mathrm{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$, where $\mathbf{K}_p$ is the clique on p vertices. Despite a long history dating back to 1976 [57], the complexity of this problem is only understood under stronger assumptions [52, 61, 30] and for special cases [66, 68, 60, 63, 23, 12, 75]. It is believed that the problem is NP-hard already in the decision variant [57], i.e., deciding whether a graph is c-colourable or not even d-colourable.

Like all decision problems, PCSPs can be solved by designing *tests*. If a test, applied to a given instance of the problem, is positive then the answer is YES; if it is negative then the answer is NO. The challenge is then to find tests that are able to guarantee a low number – ideally, zero – of false positives and false negatives. Clearly, a test is itself a decision problem. However, its nature may be substantially different, and less complicated, than the nature of the original problem.

Given a PCSP template $(\mathbf{A}, \mathbf{B})$, we may use any (potentially infinite) structure $\mathbf{T}$ to make a test for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$: We simply let the outcome of the test on an instance structure $\mathbf{X}$ be YES if $\mathbf{X} \to \mathbf{T}$, and NO if $\mathbf{X} \not\to \mathbf{T}$. In other words, $\mathrm{CSP}(\mathbf{T})$ is a test for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. Let $\mathbf{X}$ be an instance of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. If $\mathbf{X} \to \mathbf{T}$ whenever $\mathbf{X} \to \mathbf{A}$, the test is guaranteed not to generate false negatives, and we call it *complete*. Since homomorphisms compose, if $\mathbf{A} \to \mathbf{T}$ the test is automatically complete. If $\mathbf{X} \to \mathbf{B}$ whenever $\mathbf{X} \to \mathbf{T}$, the test is guaranteed not to generate false positives, and we call it *sound*. If both of these conditions hold, we say that the test *solves* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. Notice that, in this case, one obtains a reduction of $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ to $\mathrm{CSP}(\mathbf{T})$. The nature of such a test could be substantially different from that of the original problem. The reason of this difference is, ultimately, algebraic. The complexity of both CSPs and PCSPs was shown to be determined by

[1] Unless otherwise stated, we shall use the word "structure" to mean finite-domain structures; if the domain is allowed to be infinite, we shall say it explicitly.

higher-order symmetries of the solution sets of the problems, known as *polymorphisms*, denoted by $\mathrm{Pol}(\mathbf{A})$ for $\mathrm{CSP}(\mathbf{A})$ [33] and by $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ [12]. For CSPs, polymorphisms form *clones*; in particular, they are closed under composition. This means that some symmetries may be obtainable through compositions of other symmetries, so that one can hope to capture properties of entire families of CSPs (e.g., bounded width, tractability, etc.) through the presence of a certain polymorphism and, more generally, to describe their complexity through universal-algebraic tools. A chief example of this approach is the positive resolution of the dichotomy conjecture for CSPs by Bulatov [35] and Zhuk [91], establishing that finite-domain non-uniform CSPs are either in P or are NP-complete. For PCSPs, however, polymorphisms are not closed under composition, and the algebraic structure they are endowed with – known as *minion* – is much less rich and, apparently, harder to understand through the lens of universal algebra.

To make a test $\mathbf{T}$ useful as a polynomial-time algorithm to solve a PCSP, one requires that $\mathrm{CSP}(\mathbf{T})$ should be tractable. It was conjectured in [24] that every tractable (finite-domain) PCSP is solved by a tractable test. In other words, if the conjecture is true, *tests are the sole source of tractability for PCSPs.* For the conjecture to be true, one needs to admit tests on infinite domains: As shown in [12], the PCSP template $(\mathbf{1\text{-}in\text{-}3}, \mathbf{NAE})$ does not admit a finite-domain tractable test; i.e., there is no (finite) structure $\mathbf{T}$ such that $\mathbf{1\text{-}in\text{-}3} \to \mathbf{T} \to \mathbf{NAE}$ and $\mathrm{CSP}(\mathbf{T})$ is tractable.

For a PCSP template $(\mathbf{A}, \mathbf{B})$, one would ideally aim to build tests for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ in a systematic way. One method to do so is by considering tests associated with minions and, in particular, their *free structures.* The free structure $\mathbb{F}_{\mathscr{M}}(\mathbf{A})$ of a minion $\mathscr{M}$ generated by a structure $\mathbf{A}$ [12] is a (potentially infinite) structure obtained, essentially, by simulating the relations in $\mathbf{A}$ on a domain consisting of elements of $\mathscr{M}$. Then, we define $\mathrm{Test}_{\mathscr{M}}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$ if $\mathbf{X} \to \mathbb{F}_{\mathscr{M}}(\mathbf{A})$, and $\textsc{No}$ otherwise. (Note that $\mathbf{X}$ is the input to the problem; the minion $\mathscr{M}$ and the relational structure $\mathbf{A}$, coming from a PCSP template, are (fixed) parameters of the test.)

For certain choices of $\mathscr{M}$, $\mathrm{Test}_{\mathscr{M}}$ is a tractable test; i.e., $\mathrm{CSP}(\mathbb{F}_{\mathscr{M}}(\mathbf{A}))$ is tractable for any $\mathbf{A}$. This is the case for the minions $\mathscr{H} = \mathrm{Pol}(\textsc{Horn-3-Sat})$ (whose elements are nonempty subsets of a given set), $\mathscr{Q}_{\text{conv}}$ (whose elements are stochastic vectors), and $\mathscr{Z}_{\text{aff}}$ (whose elements are affine integers vectors). As it was shown in [12], these three minions correspond to three well-studied algorithmic relaxations: $\mathrm{Test}_{\mathscr{H}}$ is Arc Consistency (AC) [82], $\mathrm{Test}_{\mathscr{Q}_{\text{conv}}}$ is the Basic Linear Programming relaxation (BLP) [76], and $\mathrm{Test}_{\mathscr{Z}_{\text{aff}}}$ is the Affine Integer Programming relaxation ($\mathscr{Z}_{\text{aff}}$) [24]. In [28], the algorithm $\mathrm{BLP} + \mathrm{AIP}$ corresponding to a combination of linear and affine programming was shown to consist in $\mathrm{Test}_{\mathscr{M}_{\mathrm{BA}}}$, where the minion $\mathscr{M}_{\mathrm{BA}}$ is given essentially as the direct sum of $\mathscr{Q}_{\text{conv}}$ and $\mathscr{Z}_{\text{aff}}$. In summary, several widely used algorithms for (P)CSPs are minion tests; in particular, Arc Consistency, which is the simplest example of consistency algorithms, and algorithms based on relaxations.

Convex relaxations have been instrumental in the understanding of the complexity of many variants of CSPs, including constant approximability of Min-CSPs [55, 50] and Max-CSPs [69, 86], robust satisfiability of CSPs [92, 76, 14], and exact solvability of optimisation CSPs [72, 89]. An important line of work focused on making convex relaxations stronger and stronger via the so-called "lift-and-project" method, which includes the Sherali-Adams LP hierarchy [87], the SDP hierarchy of Lovász and Schrijver [81], and the (stronger) SDP hierarchy of Lasserre [77], also known as the Sum-of-Squares hierarchy (see [78] for a comparison of these hierarchies). The study of the power of various hierarchies has led to several breakthroughs, e.g., [1, 38, 73, 58, 90, 79, 79].

In the same spirit as hierarchies of relaxations, the (combinatorial) k-consistency algorithm has been studied [56, 3], where k is an integer bounding the number of variables considered in reasoning about partial solutions; the case $k = 1$ corresponds to Arc Consistency mentioned above. The notion of local consistency, in addition to being one of the central concepts in constraint satisfaction, has also emerged independently in finite model theory [70], graph theory [62], and proof complexity [4]. The power of local consistency for CSPs is now fully understood [32, 13, 10]. Recent work identified a necessary condition on local consistency to solve PCSPs [5].

Contributions The main contribution of this work is the introduction of *a general framework for refining algorithmic relaxations of* (P)CSPs. Given a minion $\mathscr{M}$, we present a technique to systematically turn $\mathrm{Test}_{\mathscr{M}}$ into the corresponding *hierarchy of minion tests*: a sequence of increasingly tighter relaxations $\mathrm{Test}^k_{\mathscr{M}}$ for $k \in \mathbb{N}$. Letting $\mathscr{M}$ be $\mathscr{H}$ (resp., $\mathscr{Q}_{\text{conv}}$, $\mathscr{Z}_{\text{aff}}$), we shall retrieve in this way the bounded width hierarchy [56, 13] (resp., the Sherali-Adams LP hierarchy [87], the affine integer programming hierarchy). Additionally, we describe a new minion $\mathscr{S}$ capturing the power of the basic semidefinite programming relaxation (SDP) [86], and we show that $\mathrm{Test}^k_{\mathscr{S}}$ coincides with the Sum-of-Squares hierarchy [77]. It follows that this framework is able to provide a unified algebraic description of all these four well-known hierarchies of algorithmic relaxations. We point out that, in addition to casting *known* hierarchies of relaxations as hierarchies of minion tests, this approach can be straightforwardly used to design *new* hierarchies by considering different minions, such as $\mathscr{M}_{\mathrm{BA}}$ (capturing $\mathrm{BLP}+\mathrm{AIP}$), see the full version [42].

The technique we adopt to build hierarchies of minion tests is inspired by multilinear algebra. We describe a *tensorisation* construction that turns a given structure $\mathbf{X}$ into a structure $\mathbf{X}^{\textcircled{k}}$ on a different signature, where both the domain and the relations are multidimensional objects living in tensor spaces. Essentially, $\mathrm{Test}^k_{\mathscr{M}}$ works by applying $\mathrm{Test}_{\mathscr{M}}$ to *tensorised* versions of the structures $\mathbf{X}$ and $\mathbf{A}$ rather than to $\mathbf{X}$ and $\mathbf{A}$ themselves. This allows us to describe the functioning of the algorithms in the hierarchy by describing the geometry of a space of tensors – which can be accomplished by using multilinear algebra. As far as we know, this approach has not appeared in the literature on Sherali-Adams, bounded width, Sum-of-Squares, hierarchies of integer programming, and related algorithmic techniques such as the high-dimensional Weisfeiler-Leman algorithm [6, 37].[2]

One interesting feature of our framework is that it is *modular*, in that it allows splitting the description of a hierarchy of minion tests into an *algebraic* part, corresponding to the minion $\mathscr{M}$, and a *geometric* part, entirely dependent on the tensorisation construction and hence common to any hierarchy. By considering certain well-behaved families of minions, which we call *linear* and *conic*, we can then deduce general properties of the corresponding hierarchies by only focussing on the geometry of spaces of tensors.

Finally, we observe that the scope of this idea is potentially not limited to CSPs: The multilinear pattern that we found at the core of the bounded width, Sherali-Adams, affine integer programming, and Sum-of-Squares hierarchies appears to be transversal to the CSP framework and, instead, inherently connected to the algorithmic techniques themselves, which can be applied to classes of computational problems living beyond the realms of (P)CSPs.

Subsequent work The authors have used the tensorisation methodology introduced in this paper in follow-up work that studied hierarchies of relaxations for PCSPs. In particular, they have shown that the approximate graph colouring problem is not solved by the affine integer programming hierarchy [43] and not even by the (much stronger) lift-and-project hierarchy for the combined basic linear programming and affine integer programming relaxation [40].

2 Background

Notation We denote by $\mathbb{N}$ the set of positive integers. For $k \in \mathbb{N}$, we denote by $[k]$ the set $\{1, \dots, k\}$. We indicate by $\mathbf{e}_i$ the i-th standard unit vector of the appropriate size (which will be clear from the context); i.e., the i-th entry of $\mathbf{e}_i$ is 1, and all other entries are 0. $\mathbf{0}_p$ and $\mathbf{1}_p$ denote the all-zero and all-one vector of size p, respectively, while I_p and $O_{p,q}$ denote the $p \times p$ identity matrix and the $p \times q$ all-zero matrix, respectively. Given a matrix M, we let $\mathrm{tr}(M)$ and $\mathrm{csupp}(M)$ be the trace and the set of indices of nonzero columns of M, respectively. The symbol $\aleph_0$ denotes the cardinality of $\mathbb{N}$.

Promise CSPs A signature σ is a finite set of relation symbols R, each with arity $\mathrm{ar}(R) \in \mathbb{N}$. A σ-structure $\mathbf{A}$ consists of a domain (universe) A and, for each $R \in \sigma$, a relation $R^{\mathbf{A}} \subseteq A^{\mathrm{ar}(R)}$. A σ-structure $\mathbf{A}$ is finite if the size $|A|$ of its domain A is finite. In this case, we often assume that the domain of $\mathbf{A}$ is $A = [n]$.

Let $\mathbf{A}$ and $\mathbf{B}$ be σ-structures. A *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a map $h : A \to B$ such that, for each $R \in \sigma$ with $r = \mathrm{ar}(R)$ and for each $\mathbf{a} = (a_1, \dots, a_r) \in A^r$, if $\mathbf{a} \in R^{\mathbf{A}}$ then $h(\mathbf{a}) = (h(a_1), \dots, h(a_r)) \in R^{\mathbf{B}}$. We denote the existence of a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ by $\mathbf{A} \to \mathbf{B}$. A pair of σ-structures $(\mathbf{A}, \mathbf{B})$ with $\mathbf{A} \to \mathbf{B}$ is called a *promise constraint satisfaction problem* (PCSP) *template*. The PCSP problem parameterised by the template $(\mathbf{A}, \mathbf{B})$, denoted by $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, is the following computational problem: The input is a σ-structure $\mathbf{X}$ and the goal is to answer YES if $\mathbf{X} \to \mathbf{A}$ and NO if $\mathbf{X} \not\to \mathbf{B}$. The promise is that it is not the case that $\mathbf{X} \not\to \mathbf{A}$ and $\mathbf{X} \to \mathbf{B}$. We write $\mathrm{CSP}(\mathbf{A})$ for $\mathrm{PCSP}(\mathbf{A}, \mathbf{A})$, the classic (non-promise) constraint satisfaction problem.

Relaxations and hierarchies The following relaxations of (P)CSPs shall be mentioned in this paper: Arc Consistency (AC) is a propagation algorithm that checks for the existence of assignments satisfying the local constraints of the given (P)CSP instance [82]; the basic linear programming (BLP) relaxation looks for compatible probability distributions on assignments [76]; the affine integer programming (AIP) relaxation turns the constraints into linear equations, that can be solved over the integers using (a variant of) Gaussian elimination [24]; the basic semidefinite programming (SDP) relaxation is essentially a strengthening of BLP,

[2]Butti and Dalmau [37] recently characterised for CSPs when the k-th level of the Sherali-Adams LP programming hierarchy accepts in terms of a construction different from the one introduced in this work. Unlike the tensorisation, the construction considered in [37] yields a relational structure whose domain includes the set of constraints of the original structure.

where probabilities are replaced by vectors satisfying orthogonality requirements [86]; BLP + AIP is a hybrid relaxation combining BLP and AIP [28].

Hierarchies of refinements of some of these relaxations have been used in the literature on (P)CSP. In particular, the bounded width (BW^k) hierarchy (also known as local consistency checking algorithm) refines AC by propagating local solutions over bigger and bigger portions of the instance, while the Sherali-Adams (SA^k), affine integer programming (AIP^k), and Sum-of-Squares hierarchies strengthen the BLP, AIP, and SDP relaxations, respectively, by looking for compatible distributions over bigger and bigger assignments.

The SDP relaxation, as well as the four hierarchies mentioned above, are described in the full version [42]. The other relaxations are not presented in detail in this work. We refer to [12] for AC, BLP, and AIP, and to [28] for BLP + AIP.

Algebraic approach to PCSPs The algebraic theory of PCSPs developed in [12] relies on the notions of polymorphism and minion. Let $\mathbf{A}$ be a σ-structure. For $L \in \mathbb{N}$, the *L-th power* of $\mathbf{A}$ is the σ-structure $\mathbf{A}^L$ with domain A^L whose relations are defined as follows: Given $R \in \sigma$ and an $L \times \mathrm{ar}(R)$ matrix M such that all rows of M are tuples in $R^{\mathbf{A}}$, the columns of M form a tuple in $R^{\mathbf{A}^L}$. An L-ary *polymorphism* of a PCSP template $(\mathbf{A}, \mathbf{B})$ is a homomorphism from $\mathbf{A}^L$ to $\mathbf{B}$. Minions were defined in [12] as sets of functions with certain properties. We shall use here the abstract definition of minions, as first done in [28], cf. also [41]. A *minion* $\mathscr{M}$ consists in the disjoint union of nonempty sets $\mathscr{M}^{(L)}$ for $L \in \mathbb{N}$ equipped with (so-called *minor*) operations $(\cdot)_{/\pi} : \mathscr{M}^{(L)} \to \mathscr{M}^{(L')}$ for all functions $\pi : [L] \to [L']$, which satisfy $M_{/\,\mathrm{id}} = M$ and, for $\pi : [L] \to [L']$ and $\tilde{\pi} : [L'] \to [L'']$, $(M_{/\pi})_{/\tilde{\pi}} = M_{/\tilde{\pi}\circ\pi}$ for all $M \in \mathscr{M}^{(L)}$.

Example 2.1. *The set* $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ *of all polymorphisms of a* PCSP *template* $(\mathbf{A}, \mathbf{B})$ *is a minion with the minor operations defined by* $f_{/\pi}(a_1, \dots, a_{L'}) = f(a_{\pi(1)}, \dots, a_{\pi(L)})$ *for* $f : \mathbf{A}^L \to \mathbf{B}$ *and* $\pi : [L] \to [L']$. *In this minion, the minor operations correspond to identifying coordinates, permuting coordinates, and introducing dummy coordinates (of polymorphisms).*

Example 2.2. *Other examples of minions that shall appear frequently in this work are* $\mathscr{Q}_{\mathrm{conv}}$, $\mathscr{Z}_{\mathrm{aff}}$, *and* $\mathscr{H}$, *capturing the power of the algorithms* BLP, AIP, *and* AC, *respectively. The L-ary elements of* $\mathscr{Q}_{\mathrm{conv}}$ *are rational vectors of size L that are stochastic (i.e., whose entries are nonnegative and sum up to* 1*), with the minor operations defined as follows: For* $\mathbf{q} \in \mathscr{Q}_{\mathrm{conv}}{}^{(L)}$ *and* $\pi : [L] \to [L']$, $\mathbf{q}_{/\pi} = P\mathbf{q}$, *where P is the* $L' \times L$ *matrix whose* (i, j)*-th entry is* 1 *if* $\pi(j) = i$, *and* 0 *otherwise.* $\mathscr{Z}_{\mathrm{aff}}$ *is defined similarly to* $\mathscr{Q}_{\mathrm{conv}}$, *the only difference being that its L-ary elements are affine integer vectors (i.e., their entries are integer – possibly negative – numbers and sum up to* 1*).* $\mathscr{H}$ *is the minion of polymorphisms of the* CSP *template* Horn-3-Sat, *i.e., the Boolean structure whose four relations are "$x \wedge y \Rightarrow z$", "$x \wedge y \Rightarrow \neg z$",* $\{0\}$, *and* $\{1\}$. *Equivalently (cf. [12]),* $\mathscr{H}$ *can be described as follows: For any* $L \in \mathbb{N}$, *the L-ary elements of* $\mathscr{H}$ *are Boolean functions of the form* $f_Z(x_1, \dots, x_L) = \bigwedge_{z \in Z} x_z$ *for any* $Z \subseteq [L]$, $Z \neq \emptyset$; *the minor operations are defined as in Example 2.1. We shall also mention the minion* $\mathscr{M}_{\mathrm{BA}}$ *capturing* BLP + AIP. *Its L-ary elements are* $L \times 2$ *matrices whose first column* $\mathbf{u}$ *belongs to* $\mathscr{Q}_{\mathrm{conv}}{}^{(L)}$ *and whose second column* $\mathbf{v}$ *belongs to* $\mathscr{Z}_{\mathrm{aff}}{}^{(L)}$, *and such that if the i-th entry of* $\mathbf{u}$ *is zero then the i-th entry of* $\mathbf{v}$ *is also zero, for each* $i \in [L]$. *The minor operation is defined on each column individually; i.e.,* $[\ \mathbf{u} \quad \mathbf{v}\ ]_{/\pi} = [\ \mathbf{u}_{/\pi} \quad \mathbf{v}_{/\pi}\ ]$.

For two minions $\mathscr{M}$ and $\mathscr{N}$, a *minion homomorphism* $\xi : \mathscr{M} \to \mathscr{N}$ is a map that preserves arities and minors: Given $M \in \mathscr{M}^{(L)}$ and $\pi : [L] \to [L']$, $\xi(M) \in \mathscr{N}^{(L)}$ and $\xi(M_{/\pi}) = \xi(M)_{/\pi}$. We denote the existence of a minion homomorphism from $\mathscr{M}$ to $\mathscr{N}$ by $\mathscr{M} \to \mathscr{N}$.

We will also need the concept of free structure from [12]. Let $\mathscr{M}$ be a minion and let $\mathbf{A}$ be a (finite) σ-structure. The *free structure* of $\mathscr{M}$ generated by $\mathbf{A}$ is a σ-structure $\mathbb{F}_{\mathscr{M}}(\mathbf{A})$ with domain $\mathscr{M}^{(|A|)}$ (potentially infinite). Given a relation symbol $R \in \sigma$ of arity r, a tuple $(M_1, \dots, M_r)$ of elements of $\mathscr{M}^{(|A|)}$ belongs to $R^{\mathbb{F}_{\mathscr{M}}(\mathbf{A})}$ if and only if there is some $Q \in \mathscr{M}^{(|R^{\mathbf{A}}|)}$ such that $M_i = Q_{/\pi_i}$ for each $i \in [r]$, where $\pi_i : R^{\mathbf{A}} \to A$ maps $\mathbf{a} \in R^{\mathbf{A}}$ to its i-th coordinate a_i. The definition of free structure may at this point strike the reader as rather abstract. We shall see that if we consider certain quite general classes of minions then this object unveils an interesting geometric description of linear and multilinear nature.

3 Overview of results and techniques

This section contains statements of the main results. The full version of the paper contains all details and proofs [42].

Let $(\mathbf{A},\mathbf{B})$ be a PCSP template. As discussed in Section 1, any (potentially infinite) structure $\mathbf{T}$ on the same signature as $\mathbf{A}$ and $\mathbf{B}$ can be viewed as a test for the computational problem $\operatorname{PCSP}(\mathbf{A},\mathbf{B})$: Given an instance $\mathbf{X}$, the test returns YES if $\mathbf{X}\to\mathbf{T}$, and NO otherwise. As the next definition illustrates, minions provide a systematic method to build tests for PCSPs.

DEFINITION 3.1. *Let $\mathscr{M}$ be a minion. The* minion test $\operatorname{Test}_{\mathscr{M}}$ *is the computational problem defined as follows: Given two σ-structures $\mathbf{X}$ and $\mathbf{A}$, return* YES *if $\mathbf{X}\to\mathbb{F}_{\mathscr{M}}(\mathbf{A})$, and* NO *otherwise.*

If $\mathbf{X}$ is an instance of $\operatorname{PCSP}(\mathbf{A},\mathbf{B})$ for some template $(\mathbf{A},\mathbf{B})$, we write $\operatorname{Test}_{\mathscr{M}}(\mathbf{X},\mathbf{A})=$ YES if $\operatorname{Test}_{\mathscr{M}}$ applied to $\mathbf{X}$ and $\mathbf{A}$ returns YES (i.e., if $\mathbf{X}\to\mathbb{F}_{\mathscr{M}}(\mathbf{A})$), and we write $\operatorname{Test}_{\mathscr{M}}(\mathbf{X},\mathbf{A})=$ NO otherwise. Note that, in the expression "$\operatorname{Test}_{\mathscr{M}}(\mathbf{X},\mathbf{A})$", $\mathbf{X}$ is the input structure of the PCSP, while $\mathbf{A}$ is the fixed structure from the PCSP template.

Excluding SDP for the moment, it turns out that the algebraic structure lying at the core of all relaxations mentioned in Section 2, of seemingly different nature, is the same, as all of them are minion tests for specific minions.

THEOREM 3.1. ([12, 28, 41]) $\mathrm{AC}=\operatorname{Test}_{\mathscr{H}}$, $\mathrm{BLP}=\operatorname{Test}_{\mathscr{Q}_{\mathrm{conv}}}$, $\mathrm{AIP}=\operatorname{Test}_{\mathscr{Z}_{\mathrm{aff}}}$, $\mathrm{BLP}+\mathrm{AIP}=\operatorname{Test}_{\mathscr{M}_{\mathrm{BA}}}$.

One reason why minion tests are interesting types of tests is that they are always complete.

PROPOSITION 3.1. $\operatorname{Test}_{\mathscr{M}}$ *is complete for any minion $\mathscr{M}$; i.e., for any $\mathbf{X}$ and $\mathbf{A}$ with $\mathbf{X}\to\mathbf{A}$, we have $\mathbf{X}\to\mathbb{F}_{\mathscr{M}}(\mathbf{A})$.*

A second quality of minion tests is that their soundness can be checked algebraically, as stated in the next proposition and shown easily using a compactness argument from [83], cf. [12].

PROPOSITION 3.2. *Let $\mathscr{M}$ be a minion and let $(\mathbf{A},\mathbf{B})$ be a* PCSP *template. Then,* $\operatorname{Test}_{\mathscr{M}}$ *solves* $\operatorname{PCSP}(\mathbf{A},\mathbf{B})$ *if and only if $\mathscr{M}\to\operatorname{Pol}(\mathbf{A},\mathbf{B})$.*

3.1 A minion for SDP The first contribution of this work is to design a minion $\mathscr{S}$ capturing the power of SDP, thus showing that, similarly to AC, BLP, AIP, and BLP + AIP, also SDP is a minion test.

DEFINITION 3.2. *For $L\in\mathbb{N}$, let $\mathscr{S}^{(L)}$ be the set of real $L\times\aleph_0$ matrices M such that*

$$\text{(3.1)}\qquad \textit{(C1)}\ \operatorname{csupp}(M)\ \textit{is finite}\qquad \textit{(C2)}\ MM^T\ \textit{is a diagonal matrix}\qquad \textit{(C3)}\ \operatorname{tr}(MM^T)=1.$$

Given a function $\pi:[L]\to[L']$ and a matrix $M\in\mathscr{S}^{(L)}$, we let $M_{/\pi}=PM$, where P is the $L'\times L$ matrix whose (i,j)-th entry is 1 if $\pi(j)=i$, and 0 otherwise. We set $\mathscr{S}=\bigsqcup_{L\in\mathbb{N}}\mathscr{S}^{(L)}$.

In the full version [42], we prove that the object defined above is indeed a minion and that it captures the power of the SDP relaxation, as stated below.

PROPOSITION 3.3. $\mathrm{SDP}=\operatorname{Test}_{\mathscr{S}}$. *In other words, given two σ-structures $\mathbf{X}$ and $\mathbf{A}$,* $\operatorname{SDP}(\mathbf{X},\mathbf{A})=$ YES *if and only if $\mathbf{X}\to\mathbb{F}_{\mathscr{S}}(\mathbf{A})$.*

Using Proposition 3.2, we obtain a characterisation of the power of the SDP relaxation.

THEOREM 3.2. *Let $(\mathbf{A},\mathbf{B})$ be a* PCSP *template. Then,* SDP *solves* $\operatorname{PCSP}(\mathbf{A},\mathbf{B})$ *if and only if $\mathscr{S}\to\operatorname{Pol}(\mathbf{A},\mathbf{B})$.*

3.2 Tensorisation As discussed earlier, minions give a systematic method for designing tests for (P)CSPs. We now describe a construction, which we call *tensorisation*, that provides a technique to systematically refine minion tests, thus creating hierarchies of progressively stronger algorithms.

Let S be a set and let $k\in\mathbb{N}$. For $\mathbf{n}=(n_1,\dots,n_k)\in\mathbb{N}^k$, $\mathcal{T}^{\mathbf{n}}(S)$ denotes the set of all functions from $[n_1]\times\cdots\times[n_k]$ to S, which we visualise as hypermatrices or tensors. Many of the tensors appearing in this paper are *cubical*, which means that $\mathbf{n}=n\cdot\mathbf{1}_k=(n,\dots,n)$ is a constant tuple.[3]

[3]See the full version [42] for further details on the terminology for tensors.

For $k \in \mathbb{N}$ and a signature σ, $\sigma^{ⓚ}$ is the signature consisting of the same symbols as σ such that each symbol R of arity r in σ has arity r^k in $\sigma^{ⓚ}$.

DEFINITION 3.3. *The k-th* tensor power *of a σ-structure $\mathbf{A}$ is the $\sigma^{ⓚ}$-structure $\mathbf{A}^{ⓚ}$ having domain A^k and relations defined as follows: For each symbol $R \in \sigma$ of arity r in σ, we set $R^{\mathbf{A}^{ⓚ}} = \left\{\mathbf{a}^{ⓚ} : \mathbf{a} \in R^{\mathbf{A}}\right\}$, where, for $\mathbf{a} \in R^{\mathbf{A}}$, $\mathbf{a}^{ⓚ}$ is the tensor in $\mathcal{T}^{r \cdot \mathbf{1}_k}(A^k)$ defined as follows: For any $(i_1, i_2, \dots, i_k) \in [r]^k$, the $(i_1, i_2, \dots, i_k)$-th element of $\mathbf{a}^{ⓚ}$ is $(a_{i_1}, a_{i_2}, \dots, a_{i_k})$.*[4,5]

Notice that $\mathbf{A}^{①} = \mathbf{A}$. Also, the function $R^{\mathbf{A}} \to R^{\mathbf{A}^{ⓚ}}$ given by $\mathbf{a} \mapsto \mathbf{a}^{ⓚ}$ is a bijection, so the cardinality of $R^{\mathbf{A}^{ⓚ}}$ equals the cardinality of $R^{\mathbf{A}}$.

EXAMPLE 3.1. *Let us describe the third tensor power of the 3-clique – i.e., the structure $\mathbf{K}_3^{③}$. The domain of $\mathbf{K}_3^{③}$ is $[3]^3$, i.e., the set of tuples of elements in $[3]$ having length 3. Let R be the symbol corresponding to the binary edge relation in $\mathbf{K}_3$, so that $R^{\mathbf{K}_3} = \{(1,2),(2,1),(2,3),(3,2),(3,1),(1,3)\}$. Then, $R^{\mathbf{K}_3^{③}}$ has arity $2^3 = 8$ and it is a subset of $\mathcal{T}^{2 \cdot \mathbf{1}_3}([3]^3)$. Specifically, $R^{\mathbf{K}_3^{③}} = \{(1,2)^{③},(2,1)^{③},(2,3)^{③},(3,2)^{③},(3,1)^{③},(1,3)^{③}\}$ where, e.g., $(2,3)^{③} = \left[\begin{array}{cc|cc} (2,2,2) & (2,2,3) & (3,2,2) & (3,2,3) \\ (2,3,2) & (2,3,3) & (3,3,2) & (3,3,3) \end{array}\right]$.*[6]

We say that a σ-structure $\mathbf{A}$ is *k-enhanced* if σ contains a k-ary symbol R_k such that $R_k^{\mathbf{A}} = A^k$. Observe that any two σ-structures $\mathbf{A}$ and $\mathbf{B}$ are homomorphic if and only if the structures $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ obtained by adding R_k to their signatures are homomorphic. Hence, $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is equivalent to $\mathrm{PCSP}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$, and considering k-enhanced structures results in no loss of generality. We now give the main definition of this work.

DEFINITION 3.4. *For a minion $\mathscr{M}$ and an integer $k \in \mathbb{N}$, the k-*th level of the minion test $\mathrm{Test}_{\mathscr{M}}$*, denoted by $\mathrm{Test}^k_{\mathscr{M}}$, is the computational problem defined as follows: Given two k-enhanced σ-structures $\mathbf{X}$ and $\mathbf{A}$, return* YES *if $\mathbf{X}^{ⓚ} \to \mathbb{F}_{\mathscr{M}}(\mathbf{A}^{ⓚ})$, and* NO *otherwise.*

Comparing Definition 3.4 with Definition 3.1, we see that $\mathrm{Test}^k_{\mathscr{M}}(\mathbf{X}, \mathbf{A}) = \mathrm{Test}_{\mathscr{M}}(\mathbf{X}^{ⓚ}, \mathbf{A}^{ⓚ})$. In other words, the k-th level of a minion test is just the minion test applied to the tensor power of the structures. We have seen (cf. Proposition 3.1) that a minion test is always complete. It turns out that this property keeps holding for any level of a minion test.

PROPOSITION 3.4. $\mathrm{Test}^k_{\mathscr{M}}$ *is complete for any minion $\mathscr{M}$ and any integer $k \in \mathbb{N}$.*

The proof of Proposition 3.4 relies on the fact that homomorphisms between structures are in some sense invariant under the tensorisation construction.

It is well known that each of the hierarchies of relaxations mentioned in Section 2 has the property that higher levels are at least as powerful as lower levels. As the next result shows, this is in fact a property of all hierarchies of minion tests.

PROPOSITION 3.5. *Let $\mathscr{M}$ be a minion, let $k, p \in \mathbb{N}$ be such that $k > p$, and let $\mathbf{X}, \mathbf{A}$ be two k- and p-enhanced σ-structures. If $\mathrm{Test}^k_{\mathscr{M}}(\mathbf{X}, \mathbf{A}) =$ YES then $\mathrm{Test}^p_{\mathscr{M}}(\mathbf{X}, \mathbf{A}) =$ YES.*

It follows from Proposition 3.5 that, if some level of a minion test is sound for a template $(\mathbf{A}, \mathbf{B})$ (equivalently, if it solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$), then any higher level is sound for $(\mathbf{A}, \mathbf{B})$ (equivalently, solves $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$).

The next theorem is the second main result of this paper. It shows that the framework defined above is general enough to capture four well-studied hierarchies of relaxations for (P)CSPs.

[4]Using the terminology for tensors from the full version [42], $\mathbf{a}^{ⓚ}$ can be more compactly defined as follows: $E_{\mathbf{i}} * \mathbf{a}^{ⓚ} = \mathbf{a}_{\mathbf{i}}$ for any $\mathbf{i} \in [r]^k$.

[5]We can visualise $\mathbf{a}^{ⓚ}$ as the formal Segre outer product of k copies of $\mathbf{a}$ (cf. [80]).

[6]The vertical line separates the two 2×2 layers of the $2 \times 2 \times 2$ tensor.

THEOREM 3.3. (INFORMAL) *If $k \in \mathbb{N}$ is at least the maximum arity of the template,*

- $\mathrm{BW}^k = \mathrm{Test}^k_{\mathscr{H}}$
- $\mathrm{SA}^k = \mathrm{Test}^k_{\mathscr{Q}_{\mathrm{conv}}}$
- $\mathrm{AIP}^k = \mathrm{Test}^k_{\mathscr{Z}_{\mathrm{aff}}}$
- $\mathrm{SoS}^k = \mathrm{Test}^k_{\mathscr{S}}$.

3.3 Linear minions Certain features of the hierarchies of minion tests from Definition 3.4 – in particular, the fact that they are complete (Proposition 3.4) and progressively stronger (Proposition 3.5) – hold true for any minion, as they only depend on basic properties of the tensorisation construction. In order to prove Theorem 3.3, however, it is necessary to dig deeper by investigating how the tensorisation construction interacts with the free structure. In other words, we need to understand the object $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{\textcircled{k}})$. To that end, we isolate a property shared by all minions mentioned in this work: Their objects can be interpreted as matrices, and their minor operations can be expressed as matrix multiplications. We call such minions *linear*.

DEFINITION 3.5. *A minion $\mathscr{M}$ is* linear *if there exists a semiring $\mathcal{S}$ with additive identity $0_{\mathcal{S}}$ and multiplicative identity $1_{\mathcal{S}}$ and a number $d \in \mathbb{N} \cup \{\aleph_0\}$ (called* depth*) such that*

1. *the elements of $\mathscr{M}^{(L)}$ are $L \times d$ matrices whose entries belong to $\mathcal{S}$, for each $L \in \mathbb{N}$;*
2. *given $L, L' \in \mathbb{N}$, $\pi : [L] \to [L']$, and $M \in \mathscr{M}^{(L)}$, $M_{/\pi} = PM$, where P is the $L' \times L$ matrix such that, for $i \in [L']$ and $j \in [L]$, the (i, j)-th entry of P is $1_{\mathcal{S}}$ if $\pi(j) = i$, and $0_{\mathcal{S}}$ otherwise.*

As illustrated in the next proposition, the family of linear minions is rich enough to include the minions associated with all minion tests studied in the literature of PCSPs, including SDP.

PROPOSITION 3.6. *The following minions are linear:*[7]

- $\mathscr{Q}_{\mathrm{conv}}$, *with $\mathcal{S} = \mathbb{Q}$ and $d = 1$*
- $\mathscr{H}$, *with $\mathcal{S} = (\{0, 1\}, \vee, \wedge)$ and $d = 1$*
- $\mathscr{S}$, *with $\mathcal{S} = \mathbb{R}$ and $d = \aleph_0$.*
- $\mathscr{Z}_{\mathrm{aff}}$, *with $\mathcal{S} = \mathbb{Z}$ and $d = 1$*
- $\mathscr{M}_{\mathrm{BA}}$, *with $\mathcal{S} = \mathbb{Q}$ and $d = 2$*

Recall that, as per Definition 3.1, the minion test associated with a minion $\mathscr{M}$ works by checking whether a given instance is homomorphic to the free structure of $\mathscr{M}$; in other words, $\mathrm{Test}_{\mathscr{M}}$ for a template $(\mathbf{A}, \mathbf{B})$ is essentially $\mathrm{CSP}(\mathbb{F}_{\mathscr{M}}(\mathbf{A}))$. It is then worth checking what the latter object looks like in the case that $\mathscr{M}$ is linear. The next remark shows that, in this case, $\mathbb{F}_{\mathscr{M}}(\mathbf{A})$ has a simple matrix-theoretic description.

REMARK 3.1. *Given a linear minion $\mathscr{M}$ with semiring $\mathcal{S}$ and depth d, and a σ-structure $\mathbf{A}$, the free structure $\mathbb{F}_{\mathscr{M}}(\mathbf{A})$ of $\mathscr{M}$ generated by $\mathbf{A}$ has the following description:*

- *The elements of its domain $\mathscr{M}^{(|A|)}$ are $|A| \times d$ matrices having entries in $\mathcal{S}$.*
- *For $R \in \sigma$ of arity r, the elements of $R^{\mathbb{F}_{\mathscr{M}}(\mathbf{A})}$ are tuples of the form $(P_1 Q, \ldots, P_r Q)$, where $Q \in \mathscr{M}^{(|R^{\mathbf{A}}|)}$ is a $|R^{\mathbf{A}}| \times d$ matrix having entries in $\mathcal{S}$ and, for $i \in [r]$, P_i is the $|A| \times |R^{\mathbf{A}}|$ matrix whose $(a, \mathbf{a})$-th entry is $1_{\mathcal{S}}$ if $a_i = a$, and $0_{\mathcal{S}}$ otherwise.*[8]

3.4 Multilinear tests We say that a test is *multilinear* if it can be expressed as $\mathrm{Test}^k_{\mathscr{M}}$ for some linear minion $\mathscr{M}$ and some integer k. In the same way as, for a template $(\mathbf{A}, \mathbf{B})$, $\mathrm{Test}_{\mathscr{M}}$ is essentially $\mathrm{CSP}(\mathbb{F}_{\mathscr{M}}(\mathbf{A}))$, it follows from Definition 3.4 that $\mathrm{Test}^k_{\mathscr{M}}$ is essentially $\mathrm{CSP}(\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{\textcircled{k}}))$, as it checks for the existence of a homomorphism between the tensor power of the instance and the free structure of $\mathscr{M}$ generated by the tensor power of $\mathbf{A}$.

In the full version [42], we show that, if $\mathscr{M}$ is linear, $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{\textcircled{k}})$ is a space of tensors endowed with relations that can be described through a tensor operation called *contraction*. Hence, the matrix-theoretic description in Remark 3.1 is naturally extended to a tensor-theoretic description. To give a first glance of this object, we illustrate below the structure of $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{\textcircled{k}})$ in the case that $\mathscr{M} = \mathscr{Q}_{\mathrm{conv}}$, $k = 3$, and $\mathbf{A} = \mathbf{K}_3$.

[7] It is not hard to verify that also the minion $\mathscr{C}$ capturing the power of the CLAP algorithm from [41] is linear, with $\mathcal{S} = \mathbb{Q}$ and $d = \aleph_0$.

[8] We shall often write 0 and 1 for $0_{\mathcal{S}}$ and $1_{\mathcal{S}}$ to avoid cumbersome notation. The relevant semiring $\mathcal{S}$ will always be clear from the context.

EXAMPLE 3.2. *Let us denote $\mathbb{F}_{\mathscr{Q}_{\text{conv}}}(\mathbf{K}_3^{③})$ by $\boldsymbol{F}$. The domain of $\boldsymbol{F}$ is the set of nonnegative tensors in $\mathcal{T}^{3\cdot\mathbf{1}_3}(\mathbb{Q})$ whose entries sum up to 1. The relation $R^{\boldsymbol{F}}$ is the set of those tensors $M \in \mathcal{T}^{2\cdot\mathbf{1}_3}(\mathcal{T}^{3\cdot\mathbf{1}_3}(\mathbb{Q})) = \mathcal{T}^{6\cdot\mathbf{1}_3}(\mathbb{Q})$ such that there exists a stochastic vector $\mathbf{q} = (q_1, \ldots, q_6) \in \mathscr{Q}_{\text{conv}}{}^{(6)}$ (which should be interpreted as a probability distribution over the elements of $R^{\mathbf{K}_3}$, i.e., over the directed edges in $\mathbf{K}_3$) for which the $\mathbf{i}$-th block $M_{\mathbf{i}}$ of M satisfies $M_{\mathbf{i}} = \mathbf{q}_{/\pi_{\mathbf{i}}}$ for each $\mathbf{i} \in [2]^3$. It will follow from the results in the full version [42] that, for example,*

$$M_{(1,1,1)} = \left[\begin{array}{ccc|ccc|ccc} q_1+q_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_2+q_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_4+q_5 \end{array}\right],$$

$$M_{(2,1,2)} = \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & q_1 & 0 & 0 & 0 & q_6 \\ q_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_3 \\ q_5 & 0 & 0 & 0 & q_4 & 0 & 0 & 0 & 0 \end{array}\right].$$

Figure 1 illustrates the tensor $M \in R^{\boldsymbol{F}}$ corresponding to the uniform distribution $\mathbf{q} = \frac{1}{6} \cdot \mathbf{1}_6$.

In the full version [42], we investigate the geometry of $\mathbb{F}_{\mathscr{M}}(\mathbf{A})$ for a linear minion $\mathscr{M}$. As hinted by Example 3.2, we shall see that this object is a space of sparse tensors, whose nonzero entries form regular patterns (cf. the full version [42]). This feature becomes more evident for higher values of the level k. In turn, the geometry of $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{ⓚ})$ is reflected in the properties of the homomorphisms ξ from $\mathbf{X}^{ⓚ}$ to it – which, by virtue of Definition 3.4, are precisely the solutions sought by $\text{Test}^k_{\mathscr{M}}$. For instance, the full version [42] distils the "consistency requirements" of the BW^k, SA^k, AIP^k, and SoS^k hierarchies – that enforce compatibility between partial assignments[9] from $\mathbf{X}$ to $\mathbf{A}$ – into the single tensor equation $\xi(\mathbf{x_i}) = \Pi_{\mathbf{i}} \overset{k}{*} \xi(\mathbf{x})$. For $k = 1$, the equation is vacuous, since in this case $\Pi_{\mathbf{i}}$ is the identity matrix and $\mathbf{x_i} = \mathbf{x}$ (cf. the full version [42] for the notation). As k increases, it produces a progressively richer system of symmetries that must be satisfied by ξ, which corresponds to a progressively stronger relaxation. Concretely, we shall use results on the geometry of $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{ⓚ})$ to prove Theorem 3.3.

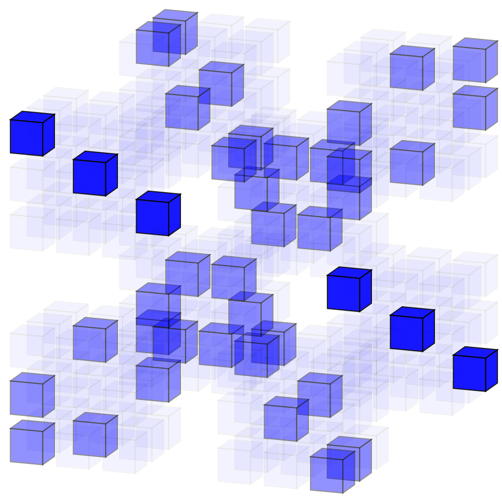
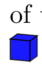


Figure 1: A tensor $M \in R^{\mathbf{F}}$ from Example 3.2, corresponding to the uniform distribution on the set of edges of $\mathbf{K}_3$. The opacity of a cell is proportional to the value of the corresponding entry:  $= \frac{1}{3}$,  $= \frac{1}{6}$,  $= 0$.

3.5 Conic minions A primary message of this work is that the tensorisation construction gives a correspondence between the algebraic properties of a minion and the algorithmic properties of the hierarchy of tests built on the minion. For example, we have seen that if the minion is linear some general properties of the solutions of the hierarchy can be deduced by studying the geometry of $\mathbb{F}_{\mathscr{M}}(\mathbf{A}^{ⓚ})$. Now, the bounded width hierarchy has the property that it only seeks assignments that are partial homomorphisms; similarly, the Sherali-Adams and Sum-of-Squares hierarchies only assign a positive weight to solutions satisfying local constraints. The next definition identifies the minion property guaranteeing this algorithmic feature.

DEFINITION 3.6. *A linear minion $\mathscr{M}$ of depth d is* conic *if, for any $L \in \mathbb{N}$ and for any $M \in \mathscr{M}^{(L)}$, (i) $M \neq O_{L,d}$, and (ii) for any $V \subseteq [L]$, the following implication is true:*[10]

$$\textstyle\sum_{i \in V} M^T \mathbf{e}_i = \mathbf{0}_d \quad \Rightarrow \quad M^T \mathbf{e}_i = \mathbf{0}_d \;\; \forall i \in V.$$

Paraphrasing Definition 3.6, a linear minion $\mathscr{M}$ is conic if any matrix in $\mathscr{M}$ is nonzero and has the property that, whenever some of its rows sum up to the zero vector, each of those rows is the zero vector. All minions appearing in Proposition 3.6 are conic, with the exception of $\mathscr{Z}_{\text{aff}}$.

[9]Cf. the "closure under restriction" property of BW^k and the requirements ♣2 and ♠3 in the full version [42].

[10]As usual, the sum, product, 0, and 1 operations appearing in this definition are to be meant in the semiring $\mathcal{S}$ associated with the linear minion $\mathscr{M}$.

PROPOSITION 3.7. *$\mathscr{H}$, $\mathscr{Q}_{\text{conv}}$, $\mathscr{M}_{\text{BA}}$, and $\mathscr{S}$ are conic minions*[11]*, while $\mathscr{Z}_{\text{aff}}$ is not.*

It turns out that this simple property guarantees that the hierarchies of tests built on conic minions only look at assignments yielding partial homomorphisms. It also follows that conic hierarchies are not fooled by small instances: The full version [42] establishes that the k-th level of such hierarchies is able to correctly classify instances on k (or fewer) elements[12] – as it is well known for the bounded width, Sherali-Adams, and Sum-of-Squares hierarchies.

References

[1] S. Arora, S. Rao, and U. V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2), 2009.

[2] K. Asimi and L. Barto. Finitely tractable promise constraint satisfaction problems. In *Proc. 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21)*, volume 202 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

[3] A. Atserias, A. A. Bulatov, and V. Dalmau. On the Power of k-Consistency. In *Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP'07)*, volume 4596 of *Lecture Notes in Computer Science*, pages 279–290. Springer, 2007.

[4] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[5] A. Atserias and V. Dalmau. Promise Constraint Satisfaction and Width. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1129–1153, 2022, arXiv:2107.05886.

[6] A. Atserias and E. N. Maneva. Sherali-Adams Relaxations and Indistinguishability in Counting Logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

[7] P. Austrin. Towards Sharp Inapproximability for Any 2-CSP. *SIAM J. Comput.*, 39(6):2430–2463, 2010.

[8] P. Austrin, A. Bhangale, and A. Potukuchi. Improved inapproximability of rainbow coloring. In *Proc. 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1479–1495, 2020, arXiv:1810.02784.

[9] P. Austrin, V. Guruswami, and J. Håstad. $(2+\epsilon)$-Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017, eccc:2013/159.

[10] L. Barto. The collapse of the bounded width hierarchy. *J. Log. Comput.*, 26(3):923–943, 2016.

[11] L. Barto, D. Battistelli, and K. M. Berg. Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case. In *Proc. 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21)*, volume 187 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, arXiv:2010.04623.

[12] L. Barto, J. Bulín, A. A. Krokhin, and J. Opršal. Algebraic approach to promise constraint satisfaction. *J. ACM*, 68(4):28:1–28:66, 2021, arXiv:1811.00970.

[13] L. Barto and M. Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM*, 61(1), 2014. Article No. 3.

[14] L. Barto and M. Kozik. Robustly solvable constraint satisfaction problems. *SIAM J. Comput.*, 45(4):1646–1669, 2016, arXiv:1512.01157.

[15] L. Barto and M. Kozik. Combinatorial Gap Theorem and Reductions between Promise CSPs. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1204–1220, 2022, arXiv:2107.09423.

[16] L. Barto, A. Krokhin, and R. Willard. Polymorphisms, and how to use them. In A. Krokhin and S. Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017.

[17] L. Barto, J. Opršal, and M. Pinsker. The wonderland of reflections. *Isr. J. Math*, 223(1):363–398, Feb 2018, arXiv:1510.04521.

[18] L. Barto and M. Pinsker. Topology is irrelevant (in a dichotomy conjecture for infinite domain constraint satisfaction problems). *SIAM J. Comput.*, 49(2):365–393, 2020.

[19] A. Bhangale and S. Khot. Optimal Inapproximability of Satisfiable k-LIN over Non-Abelian Groups. In *Proc. 53rd Annual ACM Symposium on Theory of Computing (STOC'21)*, pages 1615–1628. ACM, 2021, arXiv:2009.02815.

[20] A. Bhangale, S. Khot, and D. Minzer. On Inapproximability of Satisfiable k-CSPs: I. In *Proc. 54th Annual ACM Symposium on Theory of Computing (STOC'22)*, pages 976–988. ACM, 2022.

[21] M. Bodirsky, F. R. Madelaine, and A. Mottet. A proof of the algebraic tractability conjecture for monotone monadic SNP. *SIAM J. Comput.*, 50(4):1359–1409, 2021.

[11] The minion $\mathscr{C}$ associated with the CLAP algorithm from [41] can be easily shown to be conic as well.

[12] We may informally express this fact by saying that conic hierarchies are "sound in the limit".

[22] M. Bodirsky, B. Martin, M. Pinsker, and A. Pongrácz. Constraint satisfaction problems for reducts of homogeneous graphs. *SIAM J. Comput.*, 48(4):1224–1264, 2019.

[23] J. Brakensiek and V. Guruswami. New hardness results for graph and hypergraph colorings. In *Proc. 31st Conference on Computational Complexity (CCC'16)*, volume 50 of *LIPIcs*, pages 14:1–14:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[24] J. Brakensiek and V. Guruswami. An algorithmic blend of LPs and ring equations for promise CSPs. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 436–455, 2019, arXiv:1807.05194.

[25] J. Brakensiek and V. Guruswami. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.*, 50(6):1663–1700, 2021, arXiv:1704.01937.

[26] J. Brakensiek and V. Guruswami. The quest for strong inapproximability results with perfect completeness. *ACM Trans. Algorithms*, 17(3):27:1–27:35, 2021.

[27] J. Brakensiek, V. Guruswami, and S. Sandeep. Conditional Dichotomy of Boolean Ordered Promise CSPs. In *Proc. 48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*, volume 198 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, arXiv:2102.11854.

[28] J. Brakensiek, V. Guruswami, M. Wrochna, and S. Živný. The power of the combined basic LP and affine relaxation for promise CSPs. *SIAM J. Comput.*, 49:1232–1248, 2020, arXiv:1907.04383.

[29] A. Brandts, M. Wrochna, and S. Živný. The complexity of promise SAT on non-Boolean domains. *ACM Trans. Comput. Theory*, 13(4):26:1–26:20, 2021, arXiv:1911.09065.

[30] M. Braverman, S. Khot, N. Lifshitz, and D. Minzer. An Invariance Principle for the Multi-slice, with Applications. In *Proc. 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'21)*, pages 228–236. IEEE, 2021.

[31] M. Braverman, S. Khot, and D. Minzer. On rich 2-to-1 games. In *Proc. 12th Innovations in Theoretical Computer Science Conference (ITCS'21)*, volume 185 of *LIPIcs*, pages 27:1–27:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[32] A. Bulatov. Bounded relational width. Unpublished manuscript, 2009.

[33] A. Bulatov, P. Jeavons, and A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

[34] A. A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013.

[35] A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330, 2017, arXiv:1703.03021.

[36] A. A. Bulatov, M. E. Dyer, L. A. Goldberg, M. Jerrum, and C. McQuillan. The expressibility of functions on the Boolean domain, with applications to Counting CSPs. *J. ACM*, 60(5):32, 2013.

[37] S. Butti and V. Dalmau. Fractional Homomorphism, Weisfeiler-Leman Invariance, and the Sherali-Adams Hierarchy for the Constraint Satisfaction Problem. In *Proc. 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21)*, volume 202 of *LIPIcs*, pages 27:1–27:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, arXiv:2107.02956.

[38] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer. Approximate Constraint Satisfaction Requires Large LP Relaxations. *J. ACM*, 63(4):34:1–34:22, 2016.

[39] X. Chen, M. E. Dyer, L. A. Goldberg, M. Jerrum, P. Lu, C. McQuillan, and D. Richerby. The complexity of approximating conservative counting CSPs. *J. Comput. Syst. Sci.*, 81(1):311–329, 2015.

[40] L. Ciardo and S. Živný. Approximate graph colouring and the hollow shadow. Technical report, 2022, arXiv:2211.03168.

[41] L. Ciardo and S. Živný. CLAP: A New Algorithm for Promise CSPs. *SIAM Journal on Computing*, 2022, arXiv:2107.05018.

[42] L. Ciardo and S. Živný. Hierarchies of minion tests for PCSPs through tensors. Technical report, 2022, arXiv:2207.02277.

[43] L. Ciardo and S. Živný. Approximate graph colouring and crystals. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*, 2023, arXiv:2210.08293. To appear.

[44] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *J. Comput. Syst. Sci.*, 51(3):511–522, 1995.

[45] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.

[46] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

[47] N. Creignou, P. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints - An Overview of Current Research Themes*, volume 5250 of *Lecture Notes in Computer Science*. Springer, 2008.

[48] V. Dalmau, M. Kozik, A. A. Krokhin, K. Makarychev, Y. Makarychev, and J. Opršal. Robust algorithms with polynomial loss for near-unanimity CSPs. *SIAM J. Comput.*, 48(6):1763–1795, 2019, arXiv:1607.04787.

[49] V. Dalmau and A. A. Krokhin. Robust Satisfiability for CSPs: Hardness and Algorithmic Results. *ACM Trans.*

Comput. Theory, 5(4):15:1–15:25, 2013.
[50] V. Dalmau, A. A. Krokhin, and R. Manokaran. Towards a characterization of constant-factor approximable finite-valued CSPs. *J. Comput. Syst. Sci.*, 97:14–27, 2018.
[51] V. G. Deineko, P. Jonsson, M. Klasson, and A. A. Krokhin. The approximability of MAX CSP with fixed-value constraints. *J. ACM*, 55(4):16:1–16:37, 2008.
[52] I. Dinur, E. Mossel, and O. Regev. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.
[53] I. Dinur, O. Regev, and C. Smyth. The hardness of 3-uniform hypergraph coloring. *Comb.*, 25(5):519–535, Sept. 2005.
[54] M. E. Dyer and D. Richerby. An Effective Dichotomy for the Counting Constraint Satisfaction Problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013.
[55] A. Ene, J. Vondrák, and Y. Wu. Local Distribution and the Symmetry Gap: Approximability of Multiway Partitioning Problems. In *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*, pages 306–325. SIAM, 2013.
[56] T. Feder and M. 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.
[57] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976.
[58] M. Ghosh and M. Tulsiani. From Weak to Strong Linear Programming Gaps for All Constraint Satisfaction Problems. *Theory Comput.*, 14(1):1–33, 2018.
[59] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM*, 54(1):1–24, 2007.
[60] V. Guruswami and S. Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM Journal on Discrete Mathematics*, 18(1):30–40, 2004.
[61] V. Guruswami and S. Sandeep. d-To-1 Hardness of Coloring 3-Colorable Graphs with O(1) Colors. In *Proc. 47th International Colloquium on Automata, Languages, and Programming (ICALP'20)*, volume 168 of *LIPIcs*, pages 62:1–62:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.
[62] P. Hell, J. Nešetřil, and X. Zhu. Duality and polynomial testing of tree homomorphisms. *Transactions of the American Mathematical Society*, 348(4):1281–1297, 1996.
[63] S. Huang. Improved hardness of approximating chromatic number. In *Proc. 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques and the 17th International Workshop on Randomization and Computation (APPROX-RANDOM'13)*, pages 233–243. Springer, 2013, arXiv:1301.5216.
[64] A. Huber, A. A. Krokhin, and R. Powell. Skew Bisubmodularity and Valued CSPs. *SIAM J. Comput.*, 43(3):1064–1084, 2014.
[65] P. G. Jeavons, D. A. Cohen, and M. Gyssens. Closure Properties of Constraints. *J. ACM*, 44(4):527–548, 1997.
[66] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. *Comb.*, 20(3):393–415, 2000.
[67] S. Khanna, M. Sudan, L. Trevisan, and D. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.
[68] S. Khot. Improved Inaproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*, pages 600–609. IEEE Computer Society, 2001.
[69] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.
[70] P. G. Kolaitis and M. Y. Vardi. On the expressive power of datalog: Tools and a case study. *J. Comput. Syst. Sci.*, 51(1):110–134, 1995.
[71] P. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61(2):302–332, 2000.
[72] V. Kolmogorov, J. Thapper, and S. Živný. The power of linear programming for general-valued CSPs. *SIAM J. Comput.*, 44(1):1–36, 2015, arXiv:1311.4219.
[73] P. K. Kothari, R. Meka, and P. Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. In *Proc. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17)*, pages 590–603. ACM, 2017.
[74] A. Krokhin and S. Živný, editors. *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
[75] A. A. Krokhin, J. Opršal, M. Wrochna, and S. Živný. Topology and adjunction in promise constraint satisfaction. *SIAM Journal on Computing*, 2022, arXiv:2003.11351.
[76] G. Kun, R. O'Donnell, S. Tamaki, Y. Yoshida, and Y. Zhou. Linear programming, width-1 CSPs, and robust

satisfaction. In *Proc. 3rd Innovations in Theoretical Computer Science (ITCS'12)*, pages 484–495. ACM, 2012.

[77] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769, 2002.

[78] M. Laurent. A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003.

[79] J. R. Lee, P. Raghavendra, and D. Steurer. Lower Bounds on the Size of Semidefinite Programming Relaxations. In *Proc. 47th Annual ACM on Symposium on Theory of Computing (STOC'15)*, pages 567–576. ACM, 2015.

[80] L.-H. Lim. Tensors and hypermatrices. In L. Hogben, editor, *Handbook of Linear Algebra*, pages 231–260. CRC Press Boca Raton, 2nd edition, 2014.

[81] L. Lovász and A. Schrijver. Cones of Matrices and Set-Functions and 0-1 Optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[82] A. K. Mackworth. Consistency in networks of relations. *Artif. Intell.*, 8(1):99–118, 1977.

[83] A. Malcev. Untersuchungen aus dem gebiete der mathematischen logik. *Journal of Symbolic Logic*, 2(2), 1937.

[84] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM*, 60(6), 2013. Article No. 42.

[85] T. Nakajima and S. Živný. Linearly ordered colourings of hypergraphs. In *Proc. 49th International Colloquium on Automata, Languages, and Programming (ICALP'22)*, volume 229 of *LIPIcs*, pages 128:1–128:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, arXiv:2204.05628.

[86] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254, 2008.

[87] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discret. Math.*, 3(3):411–430, 1990.

[88] J. Thapper and S. Živný. The complexity of finite-valued CSPs. *J. ACM*, 63(4):37:1–37:33, 2016, arXiv:1210.2987.

[89] J. Thapper and S. Živný. The power of Sherali-Adams relaxations for general-valued CSPs. *SIAM J. Comput.*, 46(4):1241–1279, 2017, arXiv:1606.02577.

[90] M. Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pages 303–312. ACM, 2009.

[91] D. Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020, arXiv:1704.01914.

[92] U. Zwick. Finding almost-satisfying assignments. In *Proc. 13th Annual ACM Symposium on the Theory of Computing (STOC'98)*, pages 551–560. ACM, 1998.