

Approximate Graph Colouring and Crystals^{*†}

Lorenzo Ciardo[‡]

Stanislav Živný[§]

Abstract

We show that approximate graph colouring is not solved by any level of the affine integer programming (AIP) hierarchy. To establish the result, we translate the problem of exhibiting a graph fooling a level of the AIP hierarchy into the problem of constructing a highly symmetric *crystal* tensor. In order to prove the existence of crystals in arbitrary dimension, we provide a combinatorial characterisation for *realisable* systems of tensors; i.e., sets of low-dimensional tensors that can be realised as the projections of a single high-dimensional tensor.

1 Introduction

The *approximate graph colouring* problem (AGC) asks to find a d -colouring of a given c -colourable graph, where $3 \leq c \leq d$. There is a huge gap in our understanding of the computational complexity of this problem. For an n -vertex graph and $c = 3$, the currently best known polynomial-time algorithm finds a d -colouring of the graph with $d = O(n^{0.19996})$. It has been long conjectured [36] that the problem is NP-hard for any fixed constants $3 \leq c \leq d$ even in the decision variant: Given a graph, output YES if it is c -colourable and output NO if it is not d -colourable.

For $c = d$, the problem becomes the classic c -colouring problem, which appeared on Karp's original list of 21 NP-complete problems [42]. The case $c = 3$, $d = 4$ was only proved to be NP-hard in 2000 [43] (and a simpler proof was given in [38]); more generally, [43] showed hardness of the case $d = c + 2\lfloor c/3 \rfloor - 1$. This was improved to $d = 2c - 2$ in 2016 [12], and recently to $d = 2c - 1$ [6]. In particular, this last result implies hardness of the case $c = 3$, $d = 5$; the complexity of the case $c = 3$, $d = 6$ is still open. Building on [44, 40], NP-hardness was established for $d = \binom{c}{\lfloor c/2 \rfloor} - 1$ for $c \geq 4$ in [57]. NP-hardness of AGC was established for all constants $3 \leq c \leq d$ in [34] under a non-standard variant of the Unique Games Conjecture, in [39] under the d -to-1 conjecture [45] for any fixed d , and (an even stronger statement of distinguishing 3-colourability from not having an independent set of significant size) in [19] under the rich 2-to-1 conjecture [20].

AGC is an example of so called *promise constraint satisfaction problem* (PCSP). For a positive integer k , a k -uniform hypergraph \mathbf{H} consists of a set $\mathcal{V}(\mathbf{H})$ of elements called *vertices* and a set $\mathcal{E}(\mathbf{H}) \subseteq \mathcal{V}(\mathbf{H})^k$ of tuples of k vertices called *hyperedges*.¹ Given two k -uniform hypergraphs \mathbf{G} and \mathbf{H} , a map $f : \mathcal{V}(\mathbf{G}) \rightarrow \mathcal{V}(\mathbf{H})$ is a *homomorphism* from \mathbf{G} to \mathbf{H} if $f(\mathbf{g}) \in \mathcal{E}(\mathbf{H})$ for any $\mathbf{g} \in \mathcal{E}(\mathbf{G})$, where f is applied entrywise to the vertices in \mathbf{g} . We shall denote the existence of a homomorphism from \mathbf{G} to \mathbf{H} by the expression $\mathbf{G} \rightarrow \mathbf{H}$. The PCSP parameterised by two k -uniform hypergraphs \mathbf{H} and $\tilde{\mathbf{H}}$ such that $\mathbf{H} \rightarrow \tilde{\mathbf{H}}$, denoted by $\text{PCSP}(\mathbf{H}, \tilde{\mathbf{H}})$, is the following computational problem: Given a k -uniform hypergraph \mathbf{G} as input, answer YES if $\mathbf{G} \rightarrow \mathbf{H}$ and No if $\mathbf{G} \not\rightarrow \tilde{\mathbf{H}}$. The requirement $\mathbf{H} \rightarrow \tilde{\mathbf{H}}$ implies that the two cases cannot happen simultaneously, as homomorphisms compose; the *promise* is that one of the two cases always happens.² A 2-uniform hypergraph is a *digraph*. Moreover, a p -colouring of a digraph \mathbf{G} is precisely a homomorphism from \mathbf{G} to the *clique* \mathbf{K}_p – i.e., the digraph on vertex set $\{1, \dots, p\}$ such that any pair of distinct vertices is a (directed) edge. Therefore, AGC is $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$.

^{*}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/2210.08293>.

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

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

¹Unless otherwise stated, all hypergraphs appearing in this paper are finite, meaning that their vertex set is finite.

²It is customary to study PCSPs on more general objects known as *relational structures*, which consist of a collection of relations of arbitrary arities on a vertex set.

By letting $\mathbf{H} = \tilde{\mathbf{H}}$ in the definition of a PCSP, one obtains the standard (non-promise) *constraint satisfaction problem* (CSP). PCSPs were introduced in [4, 14] as a general framework for studying approximability of perfectly satisfiable CSPs and have emerged as a new direction in constraint satisfaction that requires different techniques than CSPs. Recent works on PCSPs include those using analytical methods [10, 20, 15, 11] and those building on algebraic methods [13, 16, 57, 39, 2, 5, 17, 25, 7, 29, 51] developed in [6]. However, most basic questions are still left open, including applicability of different types of algorithms. Remarkably, most algorithmic techniques in constraint satisfaction can be broadly classified into two distinct classes: Algorithms enforcing some type of local consistency, and algorithms related to (generalisations of) linear equations.

The first class of algorithms is based on the following idea: Rather than directly checking for the existence of a global map between \mathbf{G} and \mathbf{H} satisfying constraints (i.e., a homomorphism), which may not be doable in polynomial time, one tries to draw an *atlas of charts* covering each region of the instance \mathbf{G} . The charts are partial homomorphisms, i.e., homomorphisms from a substructure of \mathbf{G} to \mathbf{H} ; the atlas must have the property that the maps are consistent, i.e., whenever two regions overlap, there exist charts of the regions that agree on the intersection. The *bounded width* (or *local-consistency checking*) algorithm outputs YES if and only if such an atlas exists – which can be checked in polynomial time provided that the size of each chart is bounded [35]. More powerful versions of this technique require that the charts of each region should be sampled according to some probability distribution. In this case, the consistency requirement of the atlas is stronger, as it asks that, whenever two regions overlap, the probability distribution over the charts of the intersection should be exactly the *marginal* of the distributions over the charts of the two regions. Concretely, checking for the existence of such a “random atlas” amounts to solving a linear program, and results in the so-called Sherali-Adams LP hierarchy [54], which is provably more powerful than bounded width [3]. Treating probabilities as vectors satisfying certain orthogonality requirements gives an even stronger algorithm based on semidefinite programming, known as the sum-of-squares or Lasserre SDP hierarchy [55, 52, 49]. In general, the existence of a (random) atlas is not sufficient to deduce that a *planisphere* (i.e., a global map satisfying all constraints) exists. In fact, if $P \neq NP$, we do not expect polynomial-time algorithms to solve NP-hard problems. Thus, a well-established line of work has sought to prove lower bounds on the efficacy of these consistency algorithms; see [1, 18, 27, 48, 37] for lower bounds on LPs arising from lift-and-project hierarchies such as that of Sherali-Adams, and [56, 50, 26] for lower bounds on SDPs.

Any PCSP can be formulated as a system of linear equations over $\{0, 1\}$. The second class of algorithms essentially consists in solving the equations using (some variant or a generalisation of) Gaussian elimination. This requires relaxing the problem by admitting a larger range for the variables in the equations (as, in general, the system cannot be efficiently solved over $\{0, 1\}$). In particular, it is possible to solve the system in polynomial time over \mathbb{Z} ([41], cf. also [13]) – which results in the *affine integer programming* (AIP) relaxation, also known as *linear Diophantine equations*,³ that we consider in this work. The “base level” of this algorithm was studied in [13, 16] in the context of PCSPs, and its power was characterised algebraically in [6]. This algorithmic technique is substantially different from the first class of algorithms: Instead of looking for an atlas of charts faithfully describing regions of the world – i.e., a system of *local assignments satisfying the constraints* – the algorithms of the second class aim to draw a possibly imprecise planisphere – i.e., a *global assignment satisfying a relaxed version of the constraints*. In the context of CSPs, the elusive interaction between consistency-checking and methods based on (generalisations of) Gaussian-elimination was the major obstacle to the proof of the Feder-Vardi dichotomy conjecture [35], that was finally settled in [24] and, independently, in [59].

If, as conjectured, AGC is an NP-hard problem and $P \neq NP$, neither of the two algorithmic techniques should be able to solve it. In a striking sequence of works [46, 32, 33, 47], the 2-to-2 conjecture of Khot [45] (with imperfect completeness) was resolved. As detailed in [47], this implies (together with [39]) that polynomially many levels of the sum-of-squares hierarchy do not solve AGC, which implies the same result for polynomially many levels of the weaker Sherali-Adams and bounded width algorithms. Recent work [3] established that even sublinear levels of bounded width do not solve AGC.

Contributions In this paper, we focus on the second class of algorithms and show that no level of the affine integer programming hierarchy solves AGC. Recently, [30] described a linear-algebraic characterisation of the algorithm in terms of a geometric construction called *tensorisation*. Using this characterisation as a black box, we translate the problem of finding an instance of AGC fooling the algorithm into the problem of finding a tensor

³A hierarchy based on AIP (with additional local-consistency conditions) was considered in [9], where a lower bound on its power was shown for the graph isomorphism problem.

with many symmetries, which we call a *crystal*. Our main technical contribution is the construction of crystals. More precisely, we prove the following result: Given a collection of low-dimensional tensors (“pictures”) satisfying a compatibility requirement, it is possible to build a high-dimensional tensor such that by projecting it onto low-dimensional hyperplanes one recovers the pictures. Variants of this problem have appeared in the literature in combinatorial matrix theory. In particular, the problem of constructing a matrix (i.e., a 2-dimensional tensor) having prescribed row-sum and column-sum vectors (i.e., 1-dimensional projections) has been studied for different classes of matrices, such as nonnegative integer matrices [23], $(0, 1)$ matrices [31, 53], alternating-sign matrices [58], and sign-restricted matrices [22], see also the survey [8]. For example, the Gale-Ryser theorem [53] provides a characterisation, based on the notion of majorisation, of the pairs of vectors \mathbf{r}, \mathbf{c} for which there exists a $(0, 1)$ matrix whose row-sum and column-sum vectors are \mathbf{r} and \mathbf{c} , respectively. In a similar fashion, we not only show that a tensor having prescribed low-dimensional projections exists, but we also prove that a natural necessary combinatorial condition is in fact also sufficient for a system of low-dimensional “picture” tensors in order to be the set of projections of a high-dimensional tensor.

We point out that our proof is constructive, as it allows to explicitly find a tensor with the desired characteristics. As far as we know, the problem of reconstructing a tensor from low-dimensional projections has hitherto only been studied for matrices (but cf. [21], where a related problem is investigated in three dimensions in the restricted setting of alternating-sign 3-dimensional tensors). However, in order to rule out affine integer programming as an algorithm to solve AGC for all numbers of colours, we need to build crystals of arbitrarily high dimension and hence approach the reconstruction problem for arbitrarily high-dimensional tensors. In addition to its direct application to the non-solvability of AGC, we believe that our result might be of independent interest to the linear algebra and tensor theory communities.

In Section 2, we present our main results. The details of all results and proofs can be found in the full version of this paper [28].

2 Overview

Let $k \geq 2$ be an integer. Given a set V , we define $\binom{V}{\leq k} = \{S \subseteq V : 1 \leq |S| \leq k\}$. Let \mathbf{G} and \mathbf{H} be two digraphs. We introduce a variable $\lambda_S(f)$ for every $S \in \binom{\mathcal{V}(\mathbf{G})}{\leq k}$ and every function $f : S \rightarrow \mathcal{V}(\mathbf{H})$, and a variable $\lambda_{\mathbf{g}}(f)$ for every $\mathbf{g} = (g_1, g_2) \in \mathcal{E}(\mathbf{G})$ and every $f : \{g_1, g_2\} \rightarrow \mathcal{V}(\mathbf{H})$. The k -th level of the AIP hierarchy is given by the following constraints:

$$\begin{aligned}
 \text{(AIP1)} \quad & \sum_{f: S \rightarrow \mathcal{V}(\mathbf{H})} \lambda_S(f) = 1 & S \in \binom{\mathcal{V}(\mathbf{G})}{\leq k} \\
 \text{(AIP2)} \quad & \lambda_R(f) = \sum_{\tilde{f}: S \rightarrow \mathcal{V}(\mathbf{H}), \tilde{f}|_R = f} \lambda_S(\tilde{f}) & \emptyset \neq R \subseteq S \in \binom{\mathcal{V}(\mathbf{G})}{\leq k}, f : R \rightarrow \mathcal{V}(\mathbf{H}) \\
 \text{(AIP3)} \quad & \lambda_R(f) = \sum_{\tilde{f}: \{g_1, g_2\} \rightarrow \mathcal{V}(\mathbf{H}), \tilde{f}|_R = f} \lambda_{\mathbf{g}}(\tilde{f}) & \mathbf{g} \in \mathcal{E}(\mathbf{G}), \emptyset \neq R \subseteq \{g_1, g_2\}, f : R \rightarrow \mathcal{V}(\mathbf{H}) \\
 \text{(AIP4)} \quad & \lambda_{\mathbf{g}}(f) = 0 & \mathbf{g} \in \mathcal{E}(\mathbf{G}), f : \{g_1, g_2\} \rightarrow \mathcal{V}(\mathbf{H}) \text{ with } f(\mathbf{g}) \notin \mathcal{E}(\mathbf{H}).
 \end{aligned}$$

We say that $\text{AIP}^k(\mathbf{G}, \mathbf{H}) = \text{YES}$ if the system above admits a solution such that all variables take integer values. For a fixed k , this can be checked in polynomial time in the number of vertices of the input digraph \mathbf{G} by solving a polynomial-sized system of linear equations over the integers [41]. (For the “base level” of the hierarchy $k = 1$, cf. [28, Appendix A].)

Let $\tilde{\mathbf{H}}$ be a digraph such that $\mathbf{H} \rightarrow \tilde{\mathbf{H}}$. One easily checks that $\text{AIP}^k(\mathbf{G}, \mathbf{H}) = \text{YES}$ if $\mathbf{G} \rightarrow \mathbf{H}$; we say that the k -th level of AIP *solves* $\text{PCSP}(\mathbf{H}, \tilde{\mathbf{H}})$ if $\mathbf{G} \rightarrow \tilde{\mathbf{H}}$ whenever $\text{AIP}^k(\mathbf{G}, \mathbf{H}) = \text{YES}$. Clearly, if $\text{AIP}^k(\mathbf{G}, \mathbf{H}) = \text{YES}$ then $\text{AIP}^{k'}(\mathbf{G}, \mathbf{H}) = \text{YES}$ for any level k' lower than k . It follows that if some level of the hierarchy solves $\text{PCSP}(\mathbf{H}, \tilde{\mathbf{H}})$ then any *higher* level of the hierarchy also solves it. It is worth noticing that the AIP hierarchy does not enforce consistency, in the sense that it is possible that a partial assignment is given nonzero weight without being a partial homomorphism. This is in sharp contrast to the “consistency-enforcing” algorithms mentioned in the Introduction, such as the bounded-width, Sherali-Adams LP, and Lasserre SDP hierarchies. We now state the first main result of this work.

THEOREM 2.1. *No level of the AIP hierarchy solves approximate graph colouring; i.e., for any fixed $3 \leq c \leq d$, there is no k such that the k -th level of AIP solves $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$.*

2.1 Affine integer programming and tensors In order to prove Theorem 2.1, we need to find instances of AGC that fool the AIP hierarchy. Rather than working with the hierarchy itself, we shall lift the analysis to a tensor-theoretic framework. Next, we define some terminology on tensors that will be used throughout the paper.

Given n in the set \mathbb{N} of positive integer numbers, we let $[n] = \{1, \dots, n\}$. We also let $[0] = \emptyset$. Given a tuple $\mathbf{n} = (n_1, \dots, n_q) \in \mathbb{N}^q$ for some $q \in \mathbb{N}$, we let $[\mathbf{n}] = [n_1] \times \dots \times [n_q]$. Given a tuple $\mathbf{b} = (b_1, \dots, b_q) \in [\mathbf{n}]$ and a tuple $\mathbf{i} = (i_1, \dots, i_p) \in [q]^p$ for $p, q \in \mathbb{N}$, the *projection* of \mathbf{b} onto \mathbf{i} is the tuple $\mathbf{b}_i = (b_{i_1}, \dots, b_{i_p})$. Notice that $\mathbf{b}_i \in [\mathbf{n}_i]$. For $\tilde{\mathbf{n}} \in \mathbb{N}^p$, the *concatenation* of two tuples $\mathbf{b} = (b_1, \dots, b_q) \in [\mathbf{n}]$ and $\mathbf{c} = (c_1, \dots, c_p) \in [\tilde{\mathbf{n}}]$ is the tuple $(\mathbf{b}, \mathbf{c}) = (b_1, \dots, b_q, c_1, \dots, c_p)$. Notice that $(\mathbf{b}, \mathbf{c}) \in [(\mathbf{n}, \tilde{\mathbf{n}})]$. It will be handy to extend the notation above to include tuples of length zero. For any set S , we define $S^\epsilon = \{\epsilon\}$, where ϵ denotes the empty tuple. For any tuple \mathbf{x} , we let $\mathbf{x}_\epsilon = \epsilon$ and $(\mathbf{x}, \epsilon) = (\epsilon, \mathbf{x}) = \mathbf{x}$. We also define $[\epsilon] = \{\epsilon\}$. For $n \in \mathbb{N}$, define the tuple $\langle n \rangle = (1, \dots, n)$. Also, let $\langle 0 \rangle = \epsilon$. Given a tuple \mathbf{x} , $\#(\mathbf{x})$ is the cardinality of the set of elements appearing in \mathbf{x} .

Let $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Take a set S , an integer $q \in \mathbb{N}_0$, and a tuple $\mathbf{n} \in \mathbb{N}^q$. We denote by $\mathcal{T}^{\mathbf{n}}(S)$ the set of *tensors* on q *modes* of sizes n_1, \dots, n_q whose entries are in S ; formally, $\mathcal{T}^{\mathbf{n}}(S)$ is the set of functions from $[\mathbf{n}]$ to S . We sometimes denote a tensor in $\mathcal{T}^{\mathbf{n}}(S)$ by $T = (t_i)_{i \in [\mathbf{n}]}$, where $t_i \in S$ is the image of \mathbf{i} under T . For example, $\mathcal{T}^n(S)$ and $\mathcal{T}^{(m,n)}(S)$ are the sets of n -vectors and $m \times n$ matrices, respectively, having entries in S . Notice that $\mathcal{T}^\epsilon(S)$ is the set of functions from $[\epsilon] = \{\epsilon\}$ to S , which we identify with S . We will often consider *cubical* tensors, all of whose modes have equal size; i.e., tensors in the set $\mathcal{T}^{n \cdot \mathbf{1}_q}$ for some $n \in \mathbb{N}$, where $\mathbf{1}_q$ is the all-one tuple of length q .

We shall usually consider tensors having entries in the ring of integers \mathbb{Z} . For $k, \ell, m \in \mathbb{N}_0$, take $\mathbf{n} \in \mathbb{N}^k$, $\mathbf{p} \in \mathbb{N}^\ell$, and $\mathbf{q} \in \mathbb{N}^m$. The *contraction* of two tensors $T = (t_i)_{i \in [(\mathbf{n}, \mathbf{p})]} \in \mathcal{T}^{(\mathbf{n}, \mathbf{p})}(\mathbb{Z})$ and $\tilde{T} = (\tilde{t}_i)_{i \in [(\mathbf{p}, \mathbf{q})]} \in \mathcal{T}^{(\mathbf{p}, \mathbf{q})}(\mathbb{Z})$, denoted by $T \overset{\ell}{*} \tilde{T}$, is the tensor in $\mathcal{T}^{(\mathbf{n}, \mathbf{q})}(\mathbb{Z})$ such that, for $\mathbf{i} \in [\mathbf{n}]$ and $\mathbf{j} \in [\mathbf{q}]$, the (\mathbf{i}, \mathbf{j}) -th entry of $T \overset{\ell}{*} \tilde{T}$ is given by $\sum_{\mathbf{z} \in [\mathbf{p}]} t_{(\mathbf{i}, \mathbf{z})} \tilde{t}_{(\mathbf{z}, \mathbf{j})}$. If at least one of k and m equals zero – i.e., if we are contracting over all modes of T or \tilde{T} , we write $T \overset{\ell}{*} \tilde{T}$ for $T \overset{\ell}{*} \tilde{T}$, to increase readability. It is not hard to see that tensor contraction is associative, in the sense that $(T \overset{\ell}{*} \tilde{T}) \overset{m}{*} \hat{T} = T \overset{\ell}{*} (\tilde{T} \overset{m}{*} \hat{T})$ for any $\hat{T} \in \mathcal{T}^{(\mathbf{q}, \mathbf{r})}(\mathbb{Z})$, where $\mathbf{r} \in \mathbb{N}^n$ for some $n \in \mathbb{N}_0$. On the other hand, the order of operations matters for the “ $*$ ” operator. For example, if $\bar{T} \in \mathcal{T}^{\mathbf{n}}(\mathbb{Z})$, the expression $(\bar{T} * T) * \tilde{T}$ is well defined but the expression $\bar{T} * (T * \tilde{T})$ is not, in general. For this reason, we define “ $*$ ” to be left-associative; i.e., $T_1 * T_2 * T_3$ means $(T_1 * T_2) * T_3$. The next example shows that contraction generalises various linear-algebraic products.

EXAMPLE 2.1. For $m, n, p \in \mathbb{N}$, consider the tensors $c \in \mathcal{T}^\epsilon(\mathbb{Z}) = \mathbb{Z}$, $\mathbf{u}, \mathbf{v} \in \mathcal{T}^m(\mathbb{Z})$, $\mathbf{w} \in \mathcal{T}^n(\mathbb{Z})$, $M, N \in \mathcal{T}^{(m,n)}(\mathbb{Z})$, and $Q \in \mathcal{T}^{(n,p)}(\mathbb{Z})$. The following products can be seen as examples of contraction: $c \overset{0}{*} \mathbf{u} = c * \mathbf{u} = c\mathbf{u}$, $c \overset{0}{*} M = c * M = cM$ (multiplication times scalar); $\mathbf{u} \overset{1}{*} \mathbf{v} = \mathbf{u} * \mathbf{v} = \mathbf{u}^T \mathbf{v}$ (inner product); $\mathbf{u} \overset{0}{*} \mathbf{w} = \mathbf{u} \mathbf{w}^T$ (outer product); $M \overset{1}{*} Q = MQ$ (standard matrix product); $M \overset{2}{*} N = \text{tr}(M^T N)$ (Frobenius inner product).

Let $q \in \mathbb{N}_0$ and $\mathbf{n} \in \mathbb{N}^q$. Given $\mathbf{i} \in [\mathbf{n}]$, we denote by E_i the \mathbf{i} -th *standard unit tensor*; i.e., the tensor in $\mathcal{T}^{\mathbf{n}}(\mathbb{Z})$ all of whose entries are 0, except the \mathbf{i} -th entry that is 1. Observe that, for any $T \in \mathcal{T}^{\mathbf{n}}(\mathbb{Z})$, we may express the \mathbf{i} -th entry of T as $E_i * T$. If $q = 1$, $n \in \mathbb{N}$, and $i \in [n]$, notice that E_i is the i -th standard unit vector of length n . Let $\mathbf{i} \in [q]^p$ for some $p \in \mathbb{N}_0$. We associate with \mathbf{n} and \mathbf{i} the tensor $\Pi_i^{\mathbf{n}} \in \mathcal{T}^{(\mathbf{n}, \mathbf{i})}(\mathbb{Z})$ defined by

$$(2.1) \quad E_{\mathbf{a}} * \Pi_i^{\mathbf{n}} * E_{\mathbf{b}} = \begin{cases} 1 & \text{if } \mathbf{b}_i = \mathbf{a} \\ 0 & \text{otherwise} \end{cases} \quad \text{for each } \begin{matrix} \mathbf{a} \in [\mathbf{n}_i], \\ \mathbf{b} \in [\mathbf{n}]. \end{matrix}$$

We will need a few technical lemmas on the tensors defined above,⁴ whose proofs can be found in the full version of this paper [28]. The first concerns the “limit case” of the empty tuple ϵ .

LEMMA 2.1. $E_\epsilon = 1$. Moreover, given $q \in \mathbb{N}_0$ and $\mathbf{n} \in \mathbb{N}^q$, $\Pi_\epsilon^{\mathbf{n}}$ is the all-one tensor in $\mathcal{T}^{\mathbf{n}}(\mathbb{Z})$.

The following is a simple description of the entries of $\Pi_i^{\mathbf{n}}$.

LEMMA 2.2. Given $p, q \in \mathbb{N}_0$, $\mathbf{n} \in \mathbb{N}^q$, $\mathbf{i} \in [q]^p$, and $\mathbf{a} \in [\mathbf{n}_i]$, we have $E_{\mathbf{a}} * \Pi_i^{\mathbf{n}} = \sum_{\mathbf{b} \in [\mathbf{n}], \mathbf{b}_i = \mathbf{a}} E_{\mathbf{b}}$.

⁴The expression “ $x \stackrel{\bullet}{=} y$ ” shall mean “ $x = y$ by Lemma \bullet ”. Similarly, “ $x \stackrel{\text{D}\bullet}{=} y$ ” and “ $x \stackrel{(\bullet)}{=} y$ ” shall mean “ $x = y$ by Definition \bullet ” and “ $x = y$ by equation (\bullet) ”, respectively.

The assignment $\mathbf{i} \mapsto \Pi_{\mathbf{i}}^{\mathbf{p}}$ creates a correspondence between tuples and tensors. More specifically, Lemma 2.3 shows that, under this assignment, the operation of tuple projection is translated into the operation of tensor contraction, while Lemma 2.4 shows that the tuple $\langle q \rangle$, that acts by projection as the identity on the set of tuples of appropriate length, corresponds to a tensor that acts by contraction as the identity on the space of tensors of appropriate size.

LEMMA 2.3. *Let $m, p, q \in \mathbb{N}_0$, and consider two tuples $\mathbf{i} \in [q]^p$ and $\mathbf{j} \in [p]^m$. Then, for any $\mathbf{n} \in \mathbb{N}^q$, $\Pi_{\mathbf{i}\mathbf{j}}^{\mathbf{n}} = \Pi_{\mathbf{j}}^{\mathbf{n}\mathbf{i}} \stackrel{*}{*} \Pi_{\mathbf{i}}^{\mathbf{n}}$.*

LEMMA 2.4. *Let $q, q' \in \mathbb{N}_0$, $\mathbf{n} \in \mathbb{N}^q$, $\mathbf{n}' \in \mathbb{N}^{q'}$, and $T \in \mathcal{T}^{(\mathbf{n}, \mathbf{n}')}(\mathbb{Z})$. Then $\Pi_{\langle q \rangle}^{\mathbf{n}} \stackrel{*}{*} T = T$.*

In [30], the AIP hierarchy was characterised algebraically by using a multilinear construction. We state the characterisation in Theorem 2.2 below, after introducing the necessary terminology.

DEFINITION 2.1. ([6, 16]) *A minion \mathcal{M} is the disjoint union of nonempty sets $\mathcal{M}^{(p)}$ for $p \in \mathbb{N}$ equipped with operations $(\cdot)_{/\pi} : \mathcal{M}^{(p)} \rightarrow \mathcal{M}^{(q)}$ for all $\pi : [p] \rightarrow [q]$ that satisfy, for any $p, q, r \in \mathbb{N}$, $\pi : [p] \rightarrow [q]$, $\rho : [q] \rightarrow [r]$, $M \in \mathcal{M}^{(p)}$, the requirements (i) $(M_{/\pi})_{/\rho} = M_{/\rho \circ \pi}$, and (ii) $M_{/\text{id}} = M$.*

Let \mathbf{H} be a k -uniform hypergraph having n vertices and m hyperedges. The free hypergraph $\mathbb{F}_{\mathcal{M}}(\mathbf{H})$ of a minion \mathcal{M} generated by \mathbf{H} is the (potentially infinite) k -uniform hypergraph on the vertex set $\mathcal{V}(\mathbb{F}_{\mathcal{M}}(\mathbf{H})) = \mathcal{M}^{(n)}$ whose hyperedges are defined as follows: Given $M_1, \dots, M_k \in \mathcal{M}^{(n)}$, the tuple (M_1, \dots, M_k) belongs to $\mathcal{E}(\mathbb{F}_{\mathcal{M}}(\mathbf{H}))$ if and only if there exists some $Q \in \mathcal{M}^{(m)}$ such that $M_i = Q_{/\pi_i}$ for any $i \in [k]$, where $\pi_i : \mathcal{E}(\mathbf{H}) \rightarrow \mathcal{V}(\mathbf{H})$ maps a hyperedge \mathbf{h} to its i -th entry h_i .

EXAMPLE 2.2. ([6]) *For any $p \in \mathbb{N}$, let $\mathcal{Z}_{\text{aff}}^{(p)}$ be the set of integer vectors of length p whose entries sum up to one. Given $\pi : [p] \rightarrow [q]$ and $\mathbf{v} \in \mathcal{Z}_{\text{aff}}^{(p)}$, let $\mathbf{v}_{/\pi}$ be the q -vector whose j -th entry is $\sum_{\ell \in \pi^{-1}(j)} v_{\ell}$ for each $j \in [q]$. One easily shows that $\mathcal{Z}_{\text{aff}} = \bigcup_{p \in \mathbb{N}} \mathcal{Z}_{\text{aff}}^{(p)}$ is a minion.*

DEFINITION 2.2. ([30]) *Given $k \in \mathbb{N}$, the k -th tensor power⁵ of a digraph \mathbf{H} is the 2^k -uniform hypergraph $\mathbf{H}^{\otimes k}$ having vertex set $\mathcal{V}(\mathbf{H}^{\otimes k}) = \mathcal{V}(\mathbf{H})^k$ and hyperedge set $\mathcal{E}(\mathbf{H}^{\otimes k}) = \{\mathbf{h}^{\otimes k} : \mathbf{h} \in \mathcal{E}(\mathbf{H})\}$, where, for $\mathbf{h} \in \mathcal{E}(\mathbf{H})$, $\mathbf{h}^{\otimes k}$ is the tensor⁶ in $\mathcal{T}^{2 \cdot \mathbf{1}_k}(\mathcal{V}(\mathbf{H})^k)$ whose \mathbf{i} -th entry is $\mathbf{h}_{\mathbf{i}}$ for every $\mathbf{i} \in [2]^k$.*

EXAMPLE 2.3. *Let us describe the free hypergraph of \mathcal{Z}_{aff} generated by $\mathbf{H}^{\otimes k}$, where \mathbf{H} is a digraph on n vertices. $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{H}^{\otimes k})$ is a (potentially infinite) 2^k -uniform hypergraph whose vertex set is $\mathcal{Z}_{\text{aff}}^{(n^k)}$, which we identify with the set of (cubical) tensors in $\mathcal{T}^{n \cdot \mathbf{1}_k}(\mathbb{Z})$ whose entries sum up to one. Each hyperedge of $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{H}^{\otimes k})$ consists of 2^k vertices, i.e., 2^k elements of $\mathcal{Z}_{\text{aff}}^{(n^k)}$. It is convenient to visualise it as a block tensor T belonging to $\mathcal{T}^{2 \cdot \mathbf{1}_k}(\mathcal{T}^{n \cdot \mathbf{1}_k}(\mathbb{Z})) = \mathcal{T}^{2n \cdot \mathbf{1}_k}(\mathbb{Z})$. Using Definition 2.1, we see that $T \in \mathcal{E}(\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{H}^{\otimes k}))$ if and only if there exists some $Q \in \mathcal{Z}_{\text{aff}}^{(|\mathcal{E}(\mathbf{H}^{\otimes k})|)} = \mathcal{Z}_{\text{aff}}^{(|\mathcal{E}(\mathbf{H})|)}$ such that, for any $\mathbf{i} \in [2]^k$, the \mathbf{i} -th block of T is equal to $Q_{/\pi_{\mathbf{i}}}$, where $\pi_{\mathbf{i}} : \mathcal{E}(\mathbf{H}) \rightarrow \mathcal{V}(\mathbf{H})^k$ maps $\mathbf{h} \in \mathcal{E}(\mathbf{H})$ to $\mathbf{h}_{\mathbf{i}}$. It only remains to describe the entries of $Q_{/\pi_{\mathbf{i}}}$. According to Example 2.2, given any $\mathbf{h} \in \mathcal{V}(\mathbf{H})^k$, the \mathbf{h} -th entry of $Q_{/\pi_{\mathbf{i}}}$ is given by*

$$(2.2) \quad E_{\mathbf{h}} * Q_{/\pi_{\mathbf{i}}} = \sum_{\ell \in \pi_{\mathbf{i}}^{-1}(\mathbf{h})} E_{\ell} * Q = \sum_{\substack{\ell \in \mathcal{E}(\mathbf{H}) \\ \ell_{\mathbf{i}} = \mathbf{h}}} E_{\ell} * Q.$$

The following result characterises acceptance for the AIP hierarchy.⁷

THEOREM 2.2. ([30]) *Let \mathbf{G}, \mathbf{H} be two digraphs and let $k \geq 2$. Then $\text{AIP}^k(\mathbf{G}, \mathbf{H}) = \text{YES}$ if and only if there exists a homomorphism $\xi : \mathbf{G}^{\otimes k} \rightarrow \mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{H}^{\otimes k})$ such that $\xi(\mathbf{g}_{\mathbf{i}}) = \Pi_{\mathbf{i}}^{n \cdot \mathbf{1}_k} * \xi(\mathbf{g})$ for any $\mathbf{g} \in \mathcal{V}(\mathbf{G})^k$, $\mathbf{i} \in [k]^k$.*

⁵The expression ‘‘tensor product of digraphs’’ is sometimes used in the literature to indicate the direct or categorical product of digraphs. The tensor power used here is unrelated to that notion – in particular, as it is clear from Definition 2.2, the k -th tensor power of a digraph is not a digraph for $k > 1$.

⁶In particular, the number of hyperedges in $\mathbf{H}^{\otimes k}$ is equal to the number of edges in \mathbf{H} .

⁷The result in [30] is proved for arbitrary relational structures; for the purpose of this work, the less general version concerning digraphs is enough. Moreover, the definition of the AIP hierarchy and the other hierarchies characterised in [30] is formally different from the definition used here, in that it requires preprocessing the PCSP template and instance by ‘‘ k -enhancing’’ them, i.e., adding dummy constraints on k -tuples of variables. As proved in [30, Section A.1], that definition is equivalent to the more standard hierarchy definition used in [25], which we follow in this work.

2.2 The quest for crystals Theorem 2.1 is established by proving the existence of certain highly symmetric tensors (Theorem 2.3, our second main result) and using them to fool the AIP hierarchy (Proposition 2.2). The tensors we will build enjoy the remarkable property of looking identical when observed from any angle, which is why we shall refer to them as to *crystals*.

Given $p, q \in \mathbb{N}$, we let $[q]_{\rightarrow}^p$ denote the set of increasing tuples in $[q]^p$; i.e., $[q]_{\rightarrow}^p = \{(i_1, \dots, i_p) \in [q]^p \text{ s.t. } i_1 < i_2 < \dots < i_p\}$. Moreover, we let $[q]_{\rightarrow}^0 = \{\epsilon\}$ for any $q \in \mathbb{N}$. Observe that $[q]_{\rightarrow}^p \neq \emptyset$ if and only if $p \leq q$.

DEFINITION 2.3. For $q, n \in \mathbb{N}$, let M be an $n \times n$ integer matrix. A tensor $C \in \mathcal{T}^{n \cdot 1_q}(\mathbb{Z})$ is a q -dimensional M -crystal if $\Pi_{\mathbf{i}}^{n \cdot 1_q} * C = M$ for each $\mathbf{i} \in [q]_{\rightarrow}^2$.

REMARK 2.1. For $\mathbf{n} \in \mathbb{N}^q$ and $\mathbf{i} \in [q]^p$, the tensor $\Pi_{\mathbf{i}}^{\mathbf{n}}$ introduced in Section 2.1 should be understood as a projection operator, that projects a given tensor T living in $\mathcal{T}^{\mathbf{n}}(\mathbb{Z})$ onto a new system of modes – namely, $\mathbf{n}_{\mathbf{i}}$. As an example, we have seen (cf. Lemma 2.4) that, if \mathbf{i} is the identity tuple (i.e., the tuple (q)), contracting by $\Pi_{\mathbf{i}}^{\mathbf{n}}$ leaves T unaffected. More in general, if \mathbf{i} is a permutation (i.e., $\#(\mathbf{i}) = p = q$), $\Pi_{\mathbf{i}}^{\mathbf{n}}$ simply rotates the tensor by rearranging its modes. For instance, for $p = q = 2$, $\Pi_{(1,2)}^{\mathbf{n}}$ is the identity operator, while $\Pi_{(2,1)}^{\mathbf{n}}$ is the transpose operator. Indeed, letting $\mathbf{n} = (n_1, n_2) \in \mathbb{N}^2$ and considering an $n_1 \times n_2$ matrix M , $\Pi_{(1,2)}^{\mathbf{n}} * M = M$ and $\Pi_{(2,1)}^{\mathbf{n}} * M = M^T$. If $p \leq q$, as it is the case for Definition 2.3, $\Pi_{\mathbf{i}}^{\mathbf{n}}$ projects a tensor T having q modes onto a smaller, p -dimensional space. In other words, $\Pi_{\mathbf{i}}^{\mathbf{n}} * T$ is a “ p -dimensional picture” of T .

THEOREM 2.3. Let $q, n \in \mathbb{N}$, and let M be an $n \times n$ integer matrix satisfying $M\mathbf{1}_n = M^T\mathbf{1}_n$. Then there exists a q -dimensional M -crystal.

Our approach to prove Theorem 2.3 will be to show something slightly more general: Given a collection \mathcal{C} of pictures that is *realistic* – i.e., such that each pair of pictures is “locally compatible” with each other – one can always produce a tensor C such that photographing C from all angles results in the pictures in \mathcal{C} . After establishing this result (Proposition 2.1), Theorem 2.3 will easily follow, by letting all pictures be the same matrix M . We note that, even if the pictures in the definition of a crystal are two-dimensional objects (matrices), the results we shall prove are more conveniently phrased in terms of arbitrary-dimensional pictures.

DEFINITION 2.4. For $p, q \in \mathbb{N}$ and $\mathbf{n} \in \mathbb{N}^q$, a (p, \mathbf{n}) -album of pictures is a set $\mathcal{C} = \{C_{\mathbf{i}}\}_{\mathbf{i} \in [q]_{\rightarrow}^p}$ such that $C_{\mathbf{i}} \in \mathcal{T}^{\mathbf{n}_{\mathbf{i}}}(\mathbb{Z})$ for each $\mathbf{i} \in [q]_{\rightarrow}^p$. \mathcal{C} is a realistic album if

$$(2.3) \quad \Pi_{\mathbf{r}}^{\mathbf{n}_{\mathbf{i}}} * C_{\mathbf{i}} = \Pi_{\mathbf{s}}^{\mathbf{n}_{\mathbf{j}}} * C_{\mathbf{j}} \quad \text{for any} \quad \mathbf{i}, \mathbf{j} \in [q]_{\rightarrow}^p, \mathbf{r}, \mathbf{s} \in [p]_{\rightarrow}^{p-1} \text{ such that } \mathbf{i}_{\mathbf{r}} = \mathbf{j}_{\mathbf{s}}.$$

\mathcal{C} is a realisable album if there exists a tensor $C \in \mathcal{T}^{\mathbf{n}}(\mathbb{Z})$ such that $\Pi_{\mathbf{i}}^{\mathbf{n}} * C = C_{\mathbf{i}}$ for each $\mathbf{i} \in [q]_{\rightarrow}^p$.

REMARK 2.2. Crucially, the pictures in Definition 2.4 are oriented; this is enforced by taking $\mathbf{i} \in [q]_{\rightarrow}^p$ instead of $\mathbf{i} \in [q]^p$. Similarly, in Definition 2.3, we only require that “oriented pictures” of a crystal C should look identical. If we strengthened this requirement by asking that $\Pi_{\mathbf{i}}^{n \cdot 1_q} * C = M$ for all $\mathbf{i} \in [q]_{\rightarrow}^2$, an M -crystal could only exist for a symmetric matrix M . Indeed, applying this strengthened requirement to the tuples $\mathbf{i} = (i_1, i_2)$ and $\mathbf{i}_{(2,1)} = (i_2, i_1)$, we would find

$$M = \Pi_{\mathbf{i}}^{n \cdot 1_q} * C = \Pi_{\mathbf{i}_{(2,1)}}^{n \cdot 1_q} * C \stackrel{\text{L.2.3}}{=} \Pi_{(2,1)}^{n \cdot 1_2} * \Pi_{\mathbf{i}}^{n \cdot 1_q} * C = \Pi_{(2,1)}^{n \cdot 1_2} * (\Pi_{\mathbf{i}}^{n \cdot 1_q} * C) = \Pi_{(2,1)}^{n \cdot 1_2} * M = M^T,$$

where the last equality follows from the discussion in Remark 2.1. This is not a sacrifice we are willing to make, as the crystal we shall need in Proposition 2.2 to fool AIP corresponds to an integer matrix having zero diagonal and whose entries sum up to one – which, as a consequence, cannot be symmetric, see (2.4).

It is not difficult to show that, if the pictures in an album are indeed photographs of some unique tensor, then they must be compatible. In other words, a realisable album must be realistic (cf. the beginning of the proof of Proposition 2.1). Proving that a realistic album is always realisable shall require some more work. We start by showing that the problem of checking if a realistic album is realisable does not change if we rotate the space where the tensors live.

LEMMA 2.5. Let $p, q \in \mathbb{N}$, let $\ell \in [q]^q$ be such that $\#(\ell) = q$, and let $\mathbf{n} \in \mathbb{N}^q$. If every realistic (p, \mathbf{n}_{ℓ}) -album of pictures is realisable then every realistic (p, \mathbf{n}) -album of pictures is realisable.

Proposition 2.1 is proved through a nested induction, first on the dimension of the pictures (i.e., p), and second on the sum of the sizes of the modes of the tensor C that the pictures claim to depict (i.e., $\mathbf{n}^T \mathbf{1}_q$). Lemmas 2.6 and 2.7 contain the base cases for the second and the first inductions, respectively.

LEMMA 2.6. *A realistic $(p, \mathbf{1}_q)$ -album of pictures is realisable for any $p, q \in \mathbb{N}$.*

LEMMA 2.7. *A realistic $(1, \mathbf{n})$ -album of pictures is realisable for any $q \in \mathbb{N}$ and $\mathbf{n} \in \mathbb{N}^q$.*

PROPOSITION 2.1. *Let $p, q \in \mathbb{N}$ and $\mathbf{n} \in \mathbb{N}^q$. A (p, \mathbf{n}) -album of pictures is realistic if and only if it is realisable.*

Proof. [Proof of Theorem 2.3] Consider the $(2, n \cdot \mathbf{1}_q)$ -album of pictures $\mathcal{C} = \{C_i\}_{i \in [q]^2}$, given by $C_i = M$ for each $i \in [q]^2$. To check that \mathcal{C} is a realistic album, we only need to notice that $\Pi_1^{n \cdot \mathbf{1}_2} * M = M \mathbf{1}_n$ and $\Pi_2^{n \cdot \mathbf{1}_2} * M = M^T \mathbf{1}_n$ and use that, by hypothesis, $M \mathbf{1}_n = M^T \mathbf{1}_n$. It then follows from Proposition 2.1 that \mathcal{C} is a realisable album. Hence, there exists a tensor $C \in \mathcal{T}^{n \cdot \mathbf{1}_q}(\mathbb{Z})$ such that $\Pi_i^{n \cdot \mathbf{1}_q} * C = M$ for each $i \in [q]^2$. By Definition 2.3, C is a q -dimensional M -crystal. \square

The results in this section are proved in in the full version of this paper [28]. We point out that the proofs of Proposition 2.1 and of the lemmas needed to establish it are constructive, in that they allow to explicitly build a tensor whose projections are prescribed by a realistic album of pictures. As a consequence, the proof of Theorem 2.3 on the existence of crystals is constructive, too. We now give an example to illustrate the proof strategy.

EXAMPLE 2.4. *Throughout this example (and Example 2.5), we shall indicate the numbers $-2, -1, 0, 1, 2,$ and 3 by the colours blue, green, light grey, yellow, orange, and red, respectively. The goal is to build a 4-dimensional M -crystal, where*

$$M = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}.$$

To this end, we consider the $(2, 3 \cdot \mathbf{1}_4)$ album of pictures \mathcal{C} such that all pictures are equal to $\begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$. It is easy to check that \mathcal{C} is realistic (cf. the proof of Theorem 2.3); the goal is to show that \mathcal{C} is realisable, as the tensor $C \in \mathcal{T}^{3 \cdot \mathbf{1}_4}(\mathbb{Z})$ witnessing this fact would be the crystal we are looking for.

Following the proof of Proposition 2.1, we create two auxiliary albums $\hat{\mathcal{C}}$ and $\tilde{\mathcal{C}}$ from \mathcal{C} . $\hat{\mathcal{C}}$ is a $(1, 3 \cdot \mathbf{1}_3)$ -album – i.e., both the pictures and the tensor that $\hat{\mathcal{C}}$ claims to depict have one fewer dimension than those for the original album \mathcal{C} . In particular, we see from the proof that all pictures in $\hat{\mathcal{C}}$ are the same vector $\begin{array}{|c|c|} \hline \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$. Again, it is not hard to check that $\hat{\mathcal{C}}$ is a realistic album. To check that it is realisable, we only need to find a 3-dimensional tensor such that summing its entries along all three modes yields $\begin{array}{|c|c|} \hline \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$. Either by inspection or using the proof of Lemma 2.7,

we find that $\hat{C} = \begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array} \in \mathcal{T}^{3 \cdot \mathbf{1}_3}(\mathbb{Z})$ satisfies these conditions. The

second album we build is the $(2, (3, 3, 3, 2))$ -album $\tilde{\mathcal{C}}$ defined as follows: $\tilde{C}_{1,4} = \tilde{C}_{2,4} = \tilde{C}_{3,4} = \begin{array}{|c|c|} \hline \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$ (i.e., the matrix obtained by slicing off the rightmost column of $\begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$); each other picture in the album is obtained by taking the corresponding picture in \mathcal{C} and subtracting from it the corresponding projection of \hat{C} (i.e., $\tilde{C}_i = C_i - \Pi_i^{(3 \cdot \mathbf{1}_3)} * \hat{C}$). In this way, we obtain $\tilde{C}_{(1,2)} = \tilde{C}_{(1,3)} = \tilde{C}_{(2,3)} = \begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$. This album is also realistic, and it is such that the sum of the dimensions is strictly smaller than the sum of the dimensions for the album \mathcal{C} . At this point, we simply iterate the process, by repeatedly “slicing” $\tilde{\mathcal{C}}$ into an album of 1-dimensional pictures (which we handle through

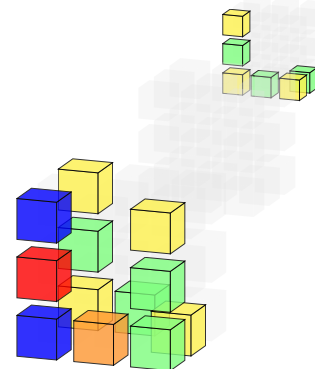




Figure 1:
A 4-dimensional $\begin{array}{|c|c|c|} \hline \color{yellow} & \color{yellow} & \color{yellow} \\ \hline \color{green} & \color{green} & \color{green} \\ \hline \color{lightgrey} & \color{lightgrey} & \color{lightgrey} \\ \hline \end{array}$ -crystal.

Lemma 2.7) and a smaller album of 2-dimensional pictures, until we end up with an album such that all dimensions are shrunk to one, so that the tensor it depicts is a single number (see Lemma 2.6). Throughout this process, Lemma 2.5 guarantees that the tensors can be rotated in a way that we slice along the rightmost mode, thus avoiding complications with the orientations of the pictures. In this way, we find that the album \tilde{C} depicts the

tensor \tilde{C} whose two blocks are  and the all-zero $3 \times 3 \times 3$ tensor, respectively. Finally, to obtain a tensor depicted by the initial album C (i.e., a 4-dimensional -crystal), we glue together \tilde{C} and \hat{C} . The result is shown in Figure 1.



2.3 Approximate graph colouring

In this section, we prove the following result.

THEOREM 2.4. (THEOREM 2.1 RESTATED) *No level of the AIP hierarchy solves approximate graph colouring; i.e., for any fixed $3 \leq c \leq d$, there is no k such that the k -th level of AIP solves $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$.*

The next proposition shows that the crystals we mined in Section 2.2 are able to fool the affine integer programming hierarchy. After establishing this result, Theorem 2.1 will easily follow.

PROPOSITION 2.2. *Let $k, n \in \mathbb{N}$ with $k \geq 2$, $n \geq 3$, and let \mathbf{G} be a loopless digraph. Then $\text{AIP}^k(\mathbf{G}, \mathbf{K}_n) = \text{YES}$.*

EXAMPLE 2.5. *We first illustrate Proposition 2.2 and its proof for the case $k = n = 3$ and $\mathbf{G} = \mathbf{K}_4$. Take the 4-dimensional -crystal C in Figure 1, and consider the map $\xi : [4]^3 \rightarrow \mathcal{T}^{3 \cdot 1_3}(\mathbb{Z})$ defined by $\mathbf{g} \mapsto \Pi_{\mathbf{g}}^{3 \cdot 1_4} * C$; i.e., ξ applied to a triplet \mathbf{g} of modes is the projection of the 4-dimensional crystal onto the 3-dimensional hyperplane corresponding to \mathbf{g} . In particular, $\xi(\mathbf{g})$ is a $3 \times 3 \times 3$ cube. According to Theorem 2.2, to show that $\text{AIP}^3(\mathbf{K}_4, \mathbf{K}_3) = \text{YES}$, we need to prove that ξ is a homomorphism from $\mathbf{K}_4^{(3)}$ to $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_3^{(3)})$; i.e., that ξ maps hyperedges of $\mathbf{K}_4^{(3)}$ to hyperedges of $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_3^{(3)})$. (The extra condition $\xi(\mathbf{g}_i) = \Pi_{\mathbf{i}}^{3 \cdot 1_3} * \xi(\mathbf{g})$ easily follows from the definition of ξ). Take, for example, the hyperedge $(1, 2)^{(3)} \in \mathcal{E}(\mathbf{K}_4^{(3)})$. Applying ξ entrywise to the $2^3 = 8$ entries of $(1, 2)^{(3)}$ yields the tensor $T \in \mathcal{T}^{2 \cdot 1_3}(\mathcal{T}^{3 \cdot 1_3}(\mathbb{Z})) = \mathcal{T}^{6 \cdot 1_3}(\mathbb{Z})$ in Figure 2. According to Example 2.3, to conclude that $T \in \mathcal{E}(\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_3^{(3)}))$, we need to exhibit some $Q \in \mathcal{Z}_{\text{aff}}^{(|\mathcal{E}(\mathbf{K}_3)|)} = \mathcal{Z}_{\text{aff}}^{(6)}$ (i.e., some integer distribution over the edges of \mathbf{K}_3) such that, for any $\mathbf{i} \in [2]^3$, the \mathbf{i} -th block of T is $Q_{/\pi_{\mathbf{i}}}$. Here it is where we use that the two-dimensional pictures of a crystal are all identical: The \mathbf{i} -th block of T is $\xi((1, 2)_{\mathbf{i}}) = \Pi_{(1, 2)_{\mathbf{i}}}^{3 \cdot 1_4} * C \stackrel{\text{L. 2.3}}{=} \Pi_{\mathbf{i}}^{3 \cdot 1_2} * (\Pi_{(1, 2)}^{3 \cdot 1_4} * C) = \Pi_{\mathbf{i}}^{3 \cdot 1_2} * \begin{smallmatrix} \square & \square \\ \square & \square \end{smallmatrix}$. As a consequence, we can let Q be the distribution encoded by the picture ; i.e., the distribution assigning weight 1 to the edges (1, 3) and (2, 1), and weight -1 to the edge (2, 3).*

Proof. [Proof of Proposition 2.2] Suppose, without loss of generality, that $\mathcal{V}(\mathbf{G}) = [q]$ for some $q \in \mathbb{N}$. If $q = 1$, the proposition is trivially true, so we can assume $q \geq 2$. Consider the matrices

$$(2.4) \quad \hat{M} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \in \mathcal{T}^{3 \cdot 1_2}(\mathbb{Z}) \quad \text{and} \quad M = \begin{bmatrix} \hat{M} & O \\ O & O \end{bmatrix} \in \mathcal{T}^{n \cdot 1_2}(\mathbb{Z}),$$

where O denotes the all-zero matrix of suitable size. Notice that $M\mathbf{1}_n = M^T\mathbf{1}_n = E_1$. (Recall that E_i is the i -th standard unit vector of length n for any $i \in [n]$.) Then, Theorem 2.3 provides us with a q -dimensional M -crystal $C \in \mathcal{T}^{n \cdot 1_q}(\mathbb{Z})$. Consider the map

$$(2.5) \quad \begin{aligned} \xi : \mathcal{V}(\mathbf{G})^k &\rightarrow \mathcal{T}^{n \cdot 1_k}(\mathbb{Z}) \\ \mathbf{g} &\mapsto \Pi_{\mathbf{g}}^{n \cdot 1_q} * C, \end{aligned}$$

which is well defined since $\Pi_{\mathbf{g}}^{n \cdot 1_q} \in \mathcal{T}^{(n \cdot 1_k, n \cdot 1_q)}(\mathbb{Z}) = \mathcal{T}^{n \cdot 1_{k+q}}(\mathbb{Z})$ for any $\mathbf{g} \in \mathcal{V}(\mathbf{G})^k$. We claim that ξ yields a

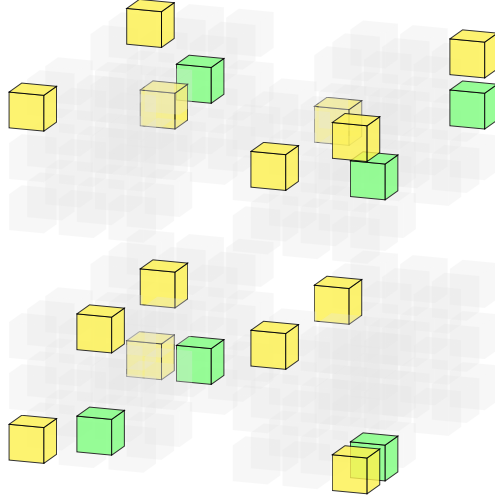


Figure 2: The tensor $\xi((1, 2)^{\otimes 3})$. Each of the 8 blocks is obtained by projecting the 4-dimensional crystal from Figure 1 onto a 3-dimensional hyperplane.

homomorphism from $\mathbf{G}^{\otimes k}$ to $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k})$. First, observe that, for any $\mathbf{g} \in \mathcal{V}(\mathbf{G})^k$,

$$\begin{aligned} \sum_{\mathbf{a} \in [n]^k} E_{\mathbf{a}} * \xi(\mathbf{g}) &\stackrel{(2.5)}{=} \sum_{\mathbf{a} \in [n]^k} E_{\mathbf{a}} * \Pi_{\mathbf{g}}^{n \cdot \mathbf{1}_q} * C \stackrel{L.2.1}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_k} * \Pi_{\mathbf{g}}^{n \cdot \mathbf{1}_q} * C \stackrel{L.2.3}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_q} * C \\ &\stackrel{L.2.3}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_2} * \Pi_{\langle 2 \rangle}^{n \cdot \mathbf{1}_q} * C \stackrel{D.2.3}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_2} * M \stackrel{L.2.1}{=} \sum_{\mathbf{b} \in [n]^2} E_{\mathbf{b}} * M = \mathbf{1}_n^T M \mathbf{1}_n = 1. \end{aligned}$$

Hence, $\xi(\mathbf{g}) \in \mathcal{Z}_{\text{aff}}^{(n^k)} = \mathcal{V}(\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k}))$, as required.

We now show that ξ preserves the hyperedges of $\mathbf{G}^{\otimes k}$. Recall from Definition 2.2 that $\mathcal{E}(\mathbf{G}^{\otimes k}) = \{\mathbf{g}^{\otimes k} : \mathbf{g} \in \mathcal{E}(\mathbf{G})\}$. Take $\mathbf{g} \in \mathcal{E}(\mathbf{G})$; we need to prove that $\xi(\mathbf{g}^{\otimes k}) \in \mathcal{E}(\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k}))$. Observe first that $\xi(\mathbf{g}^{\otimes k}) = (\xi(\mathbf{g}_i))_{i \in [2]^k} \in \mathcal{T}^{2 \cdot \mathbf{1}_k}(\mathcal{T}^{n \cdot \mathbf{1}_k}(\mathbb{Z}))$. Let $\alpha \in [2]^2$ be such that $\mathbf{g}_{\alpha} \in [q]^2_{\rightarrow}$ (which is possible since $\#(\mathbf{g}) = 2$ as \mathbf{G} is loopless). Notice that $\alpha_{\alpha} = \langle 2 \rangle$. Consider the vector $Q \in \mathcal{T}^{n^2 - n}(\mathbb{Z})$ whose entries are indexed by the edges of \mathbf{K}_n and are defined as follows: For each $\mathbf{a} \in \mathcal{E}(\mathbf{K}_n)$, the \mathbf{a} -th entry of Q is $E_{\mathbf{a}} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M$. Observe that, for any $a \in [n]$, we have

$$(2.6) \quad E_{(a,a)} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M \stackrel{L.2.2}{=} \sum_{\substack{\mathbf{b} \in [n]^2 \\ \mathbf{b}_{\alpha} = (a,a)}} E_{\mathbf{b}} * M = \sum_{\substack{\mathbf{b} \in [n]^2 \\ \mathbf{b} = (a,a)_{\alpha}}} E_{\mathbf{b}} * M = E_{(a,a)} * M = 0,$$

where we have used that α is an involution and the diagonal entries of M are zero. We find

$$\begin{aligned} \sum_{\mathbf{a} \in \mathcal{E}(\mathbf{K}_n)} E_{\mathbf{a}} * Q &= \sum_{\mathbf{a} \in \mathcal{E}(\mathbf{K}_n)} E_{\mathbf{a}} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M \stackrel{(2.6)}{=} \sum_{\mathbf{a} \in [n]^2} E_{\mathbf{a}} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M \stackrel{L.2.1}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_2} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M \\ &\stackrel{L.2.3}{=} \Pi_{\epsilon}^{n \cdot \mathbf{1}_2} * M \stackrel{L.2.1}{=} \sum_{\mathbf{b} \in [n]^2} E_{\mathbf{b}} * M = \mathbf{1}_n^T M \mathbf{1}_n = 1, \end{aligned}$$

which means that $Q \in \mathcal{Z}_{\text{aff}}^{(\mathcal{E}(\mathbf{K}_n))}$. We now aim to show that $\xi(\mathbf{g}_i) = Q_{/\pi_i}$ for any $i \in [2]^k$. We obtain

$$\begin{aligned} \xi(\mathbf{g}_i) &\stackrel{(2.5)}{=} \Pi_{\mathbf{g}_i}^{n \cdot \mathbf{1}_q} * C = \Pi_{\mathbf{g}_{\alpha \alpha_i}}^{n \cdot \mathbf{1}_q} * C \stackrel{L.2.3}{=} \Pi_i^{n \cdot \mathbf{1}_2} * \Pi_{\mathbf{g}_{\alpha \alpha}}^{n \cdot \mathbf{1}_q} * C \stackrel{L.2.3}{=} \Pi_i^{n \cdot \mathbf{1}_2} * (\Pi_{\alpha}^{n \cdot \mathbf{1}_2} * \Pi_{\mathbf{g}_{\alpha}}^{n \cdot \mathbf{1}_q}) * C \\ &= \Pi_i^{n \cdot \mathbf{1}_2} * (\Pi_{\alpha}^{n \cdot \mathbf{1}_2} * (\Pi_{\mathbf{g}_{\alpha}}^{n \cdot \mathbf{1}_q} * C)) \stackrel{D.2.3}{=} \Pi_i^{n \cdot \mathbf{1}_2} * (\Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M) = \Pi_i^{n \cdot \mathbf{1}_2} * \Pi_{\alpha}^{n \cdot \mathbf{1}_2} * M. \end{aligned}$$

Hence, for any $\mathbf{a} \in [n]^k$,

$$\begin{aligned} E_{\mathbf{a}} * \xi(\mathbf{g}_{\mathbf{i}}) &= E_{\mathbf{a}} * \Pi_{\mathbf{i}}^{n \cdot 1_2} * \Pi_{\alpha}^{n \cdot 1_2} * M \stackrel{L.2.2}{=} \sum_{\substack{\mathbf{b} \in [n]^2 \\ \mathbf{b}_i = \mathbf{a}}} E_{\mathbf{b}} * \Pi_{\alpha}^{n \cdot 1_2} * M \stackrel{(2.6)}{=} \sum_{\substack{\mathbf{b} \in \mathcal{E}(\mathbf{K}_n) \\ \mathbf{b}_i = \mathbf{a}}} E_{\mathbf{b}} * \Pi_{\alpha}^{n \cdot 1_2} * M \\ &= \sum_{\substack{\mathbf{b} \in \mathcal{E}(\mathbf{K}_n) \\ \mathbf{b}_i = \mathbf{a}}} E_{\mathbf{b}} * Q \stackrel{(2.2)}{=} E_{\mathbf{a}} * Q_{/\pi_{\mathbf{i}}}. \end{aligned}$$

It follows that $\xi(\mathbf{g}_{\mathbf{i}}) = Q_{/\pi_{\mathbf{i}}}$ for any $\mathbf{i} \in [2]^k$, as wanted, so $\xi(\mathbf{g}^{(k)}) \in \mathcal{E}(\mathbb{F}_{\mathcal{L}_{\text{aff}}}(\mathbf{K}_n^{\otimes}))$, which means that ξ is indeed a homomorphism.

To be able to apply Theorem 2.2 and conclude that $\text{AIP}^k(\mathbf{G}, \mathbf{K}_n) = \text{YES}$, we are only left to observe that, for any $\mathbf{g} \in \mathcal{V}(\mathbf{G})^k$ and any $\mathbf{i} \in [k]^k$,

$$\xi(\mathbf{g}_{\mathbf{i}}) \stackrel{D.2.5}{=} \Pi_{\mathbf{g}_{\mathbf{i}}}^{n \cdot 1_q} * C \stackrel{L.2.3}{=} \Pi_{\mathbf{i}}^{n \cdot 1_k} * \Pi_{\mathbf{g}}^{n \cdot 1_q} * C = \Pi_{\mathbf{i}}^{n \cdot 1_k} * (\Pi_{\mathbf{g}}^{n \cdot 1_q} * C) \stackrel{(2.5)}{=} \Pi_{\mathbf{i}}^{n \cdot 1_k} * \xi(\mathbf{g}),$$

as desired. \square

We remark that Proposition 2.2 does *not* hold for $n = 2$, cf. the discussion in [28].

Proof. [Proof of Theorem 2.1] Consider three integers c, d, k such that $3 \leq c \leq d$ and $2 \leq k$. Taking \mathbf{K}_{d+1} as \mathbf{G} in Proposition 2.2, we find that $\text{AIP}^k(\mathbf{K}_{d+1}, \mathbf{K}_c) = \text{YES}$; however, clearly, $\mathbf{K}_{d+1} \not\rightarrow \mathbf{K}_d$. Hence, the k -th level of AIP does not solve $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$. \square

References

- [1] S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. Proving Integrality Gaps without Knowing the Linear Program. *Theory Comput.*, 2(2):19–51, 2006.
- [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 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.
- [4] P. Austrin, V. Guruswami, and J. Håstad. $(2+\epsilon)$ -Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017, eccc:2013/159.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] A. Barvinok. Matrices with prescribed row and column sums. *Linear Algebra Appl.*, 436(4):820–844, 2012.
- [9] C. Berkholz and M. Grohe. Linear Diophantine Equations, Group CSPs, and Graph Isomorphism. In *Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, pages 327–339. SIAM, 2017, arXiv:1607.04287.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.

- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] G. Braun, S. Pokutta, and D. Zink. Inapproximability of Combinatorial Problems via Small LPs and SDPs. In *Proc. 47th Annual ACM on Symposium on Theory of Computing (STOC'15)*, pages 107–116. ACM, 2015.
- [19] 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.
- [20] 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.
- [21] R. A. Brualdi and G. Dahl. Alternating sign matrices and hypermatrices, and a generalization of Latin squares. *Adv. in Appl. Math.*, 95:116–151, 2018.
- [22] R. A. Brualdi and G. Dahl. Sign-restricted matrices of 0's, 1's, and -1 's. *Linear Algebra and its Applications*, 615:77–103, 2021.
- [23] R. A. Brualdi and H. J. Ryser. *Combinatorial matrix theory*, volume 39. Springer, 1991.
- [24] 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.
- [25] 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.
- [26] S. O. Chan. Approximation Resistance from Pairwise-Independent Subgroups. *J. ACM*, 63(3):27:1–27:32, 2016.
- [27] 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.
- [28] L. Ciardo and S. Živný. Approximate graph colouring and crystals. Technical report, 2022, arXiv:2210.08293.
- [29] L. Ciardo and S. Živný. CLAP: A New Algorithm for Promise CSPs. *SIAM Journal on Computing*, 2022, arXiv:2107.05018.
- [30] L. Ciardo and S. Živný. Hierarchies of minion tests for PCSPs through tensors. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*, 2023, arXiv:2207.02277. To appear.
- [31] C. M. da Fonseca and R. Mamede. On $(0, 1)$ -matrices with prescribed row and column sum vectors. *Discrete mathematics*, 309(8):2519–2527, 2009.
- [32] I. Dinur, S. Khot, G. Kindler, D. Minzer, and M. Safra. On non-optimally expanding sets in Grassmann graphs. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 940–951. ACM, 2018.
- [33] I. Dinur, S. Khot, G. Kindler, D. Minzer, and M. Safra. Towards a proof of the 2-to-1 games conjecture? In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 376–389. ACM, 2018.
- [34] I. Dinur, E. Mossel, and O. Regev. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.
- [35] 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.
- [36] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976.
- [37] M. Ghosh and M. Tulsiani. From Weak to Strong Linear Programming Gaps for All Constraint Satisfaction Problems. *Theory Comput.*, 14(1):1–33, 2018.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] R. Kannan and A. Bachem. Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an

- Integer Matrix. *SIAM Journal on Computing*, 8(4):499–507, Nov. 1979.
- [42] R. M. Karp. Reducibility Among Combinatorial Problems. In *Proc. Complexity of Computer Computations*, pages 85–103, 1972.
- [43] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. *Comb.*, 20(3):393–415, 2000.
- [44] S. Khot. Improved Inapproximability 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.
- [45] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 767–775. ACM, 2002.
- [46] S. Khot, D. Minzer, and M. Safra. On independent sets, 2-to-2 games, and Grassmann graphs. In *Proc. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17)*, pages 576–589. ACM, 2017.
- [47] S. Khot, D. Minzer, and M. Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. In *Proc. 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS'18)*, pages 592–601. IEEE Computer Society, 2018.
- [48] 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.
- [49] J. B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769, 2002.
- [50] J. R. Lee, P. Raghavendra, and D. Steurer. Lower Bounds on the Size of Semidefinite Programming Relaxations. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC'15)*, pages 567–576. ACM, 2015.
- [51] 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.
- [52] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.
- [53] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canadian Journal of Mathematics*, 9:371–377, 1957.
- [54] 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.
- [55] N. Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.
- [56] 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.
- [57] M. Wrochna and S. Živný. Improved hardness for H -colourings of G -colourable graphs. In *Proc. 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1426–1435, 2020, arxiv:1907.00872.
- [58] D. Zeilberger. Proof of the alternating sign matrix conjecture. *Electron. J. Combin.*, 3(2), 1996.
- [59] D. Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020, arXiv:1704.01914.