

Approximate Graph Colouring and the Hollow Shadow

Lorenzo Ciardo
lorenzo.ciardo@cs.ox.ac.uk
University of Oxford
Oxford, U.K.

Stanislav Živný
standa.zivny@cs.ox.ac.uk
University of Oxford
Oxford, U.K.

ABSTRACT

We show that approximate graph colouring is not solved by constantly many levels of the lift-and-project hierarchy for the combined basic linear programming and affine integer programming relaxation. The proof involves a construction of tensors whose fixed-dimensional projections are equal up to reflection and satisfy a sparsity condition, which may be of independent interest.

CCS CONCEPTS

• **Theory of computation** → **Approximation algorithms analysis; Problems, reductions and completeness.**

KEYWORDS

approximate graph colouring, Sherali-Adams linear programming relaxations, affine integer programming relaxations, Diophantine equations, promise constraint satisfaction

ACM Reference Format:

Lorenzo Ciardo and Stanislav Živný. 2023. Approximate Graph Colouring and the Hollow Shadow. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23)*, June 20–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3564246.3585112>

1 INTRODUCTION

The *approximate graph colouring* problem (AGC) consists in finding a d -colouring of a given c -colourable graph, where $3 \leq c \leq d$. There is a huge gap in our understanding of this problem. For an n -vertex graph and $c = 3$, the currently best known polynomial-time algorithm of Kawarabayashi and Thorup [51] finds a d -colouring of the graph with $d = O(n^{0.19996})$, building on a long line of works started by Wigderson [70]. It was conjectured by Garey and Johnson [42] 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 [50]. The case $c = 3$, $d = 4$ was only proved to be NP-hard in 2000 by Khanna, Linial, and Safra [52] (and a simpler proof was given by Guruswami and Khanna in [44]); more generally, [52] showed hardness of the case $d = c + 2\lfloor c/3 \rfloor - 1$. This was improved to $d = 2c - 2$ in 2016 by Brakensiek and Guruswami [14], and

recently to $d = 2c - 1$ by Barto, Bulín, Krokhn, and Opršal [8]. 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 the work of Khot [53] and Huang [48], Krokhn, Opršal, Wrochna, and Živný established NP-hardness for $d = \binom{c}{\lfloor c/2 \rfloor} - 1$ for $c \geq 4$ in [59]. NP-hardness of AGC was established for all constants $3 \leq c \leq d$ by Dinur, Mossel, and Regev in [39] under a non-standard variant of the Unique Games Conjecture, by Guruswami and Sandeep in [45] under the d -to-1 conjecture [54] for any fixed d , and (an even stronger statement of distinguishing 3-colourability from not having an independent set of significant size) by Braverman, Khot, Lifshitz, and Minzer in [21] under the rich 2-to-1 conjecture of Braverman, Khot, and Minzer [22].

AGC is a prominent example of so called *promise constraint satisfaction problems* (PCSPs), which we define next. A *directed graph* (digraph) A consists of a set $V(A)$ of elements called *vertices* and a set $E(A) \subseteq V(A)^2$ of pairs of vertices called *edges*. Given two digraphs A and B , a map $f : V(A) \rightarrow V(B)$ is a *homomorphism* from A to B if $(f(u), f(v)) \in E(B)$ for any $(u, v) \in E(A)$. We shall indicate the existence of a homomorphism from A to B by writing $A \rightarrow B$. Let A and B be two fixed finite digraphs with $A \rightarrow B$; we call the pair (A, B) a *template*. The PCSP parameterised by the template (A, B) , denoted by $\text{PCSP}(A, B)$, is the following decision problem: Given a finite digraph X as input, answer Yes if $X \rightarrow A$ and No if $X \not\rightarrow B$.¹ A p -colouring of a digraph X is precisely a homomorphism from X to the *clique* K_p – i.e., the digraph on vertex set $\{1, \dots, p\}$ such that any pair of distinct vertices is a (directed) edge. Hence, AGC is $\text{PCSP}(K_c, K_d)$.

By letting $A = B$ in the definition of a PCSP, one obtains the standard (non-promise) *constraint satisfaction problem* (CSP) [40]. PCSPs were introduced by Austrin, Guruswami, and Håstad [5] and Brakensiek and Guruswami [16] as a general framework for studying approximability of perfectly satisfiable CSPs and have emerged as a new exciting direction in constraint satisfaction that requires different techniques than CSPs.² Recent works on PCSPs include those using analytical methods [12, 13, 17, 22] and those building on algebraic methods [3, 7, 10, 15, 18, 19, 26, 33, 45, 63] developed in [8]. However, most basic questions are still left open, including complexity classifications and applicability of different types of algorithms.

Two main algorithmic techniques have been utilised for solving CSPs and their variants: enforcing (some type of) *local consistency*, and solving (generalisations of) *linear equations*. The first type of algorithms divides a given CSP into multiple small CSPs, each of which requires meeting *local* constraints on a portion of the instance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org).
STOC '23, June 20–23, 2023, Orlando, FL, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9913-5/23/06...\$15.00
<https://doi.org/10.1145/3564246.3585112>

¹The requirement $A \rightarrow B$ implies that the two cases cannot happen simultaneously, as homomorphisms compose; the *promise* is that one of the two cases always happens.

²It is customary to study (P)CSPs on more general objects than digraphs, known as *relational structures* [8], which consist of a collection of relations of arbitrary arities on a vertex set, cf. [8].

of bounded size, and then enforces *consistency* between all solutions (called partial homomorphisms); i.e., it requires that they agree on the intersection of their domains. Instead, the second type of algorithms seeks a *global* solution that satisfies a *linearised* version of the constraints. More precisely, it is always possible to formulate a CSP (and, in fact, any homomorphism problem) as a system of linear equations over $\{0, 1\}$; then, the algorithms of the second type work by suitably modifying the system (in particular, extending the domain of its variables) in a way that it can be efficiently solved through variants of Gaussian elimination.

Remarkably, all algorithms hitherto proposed in the literature on (variants of) CSPs can be broadly classified as instances of one of the two aforementioned techniques, or a combination of both. A primary example of the first type is the *bounded width* algorithm, which outputs YES if a consistent collection of partial homomorphisms exists [40]. More powerful versions of the local consistency technique require that the partial homomorphisms be sampled according to a probability distribution (which results in the *Sherali-Adams LP* hierarchy [67]), and that the probabilities be treated as vectors satisfying certain orthogonality requirements (which gives the *sum-of-squares* or *Lasserre SDP* hierarchy [60, 64, 68]). As for the second type, the linear-system formulation of a CSP can be efficiently solved in \mathbb{Z} by computing the Hermite or the Smith canonical forms of the corresponding coefficient matrix [66]; this results in the *affine integer programming* (AIP) relaxation, studied in the context of PCSPs in [8, 15, 16, 18]. A hierarchy of *linear Diophantine equations* was also studied by Berkholz and Grohe [11] in the context of the graph isomorphism problem.

Since polynomial-time algorithms are not expected to solve NP-hard problems, a well-established line of work has sought lower bounds on the efficacy of these algorithms; see [2, 20, 28, 43, 57] for lower bounds on LPs arising from lift-and-project hierarchies such as that of Sherali-Adams, [27, 62, 69] for lower bounds on SDPs, and [11] for lower bounds on linear Diophantine equations. If, as conjectured by Garey and Johnson [42], AGC is NP-hard and $P \neq NP$, neither of the two algorithmic techniques should be able to solve it. This is indeed the case! In a striking sequence of works by Dinur, Khot, Kindler, Minzer, and Safra [37, 38, 55, 56], the 2-to-2 conjecture of Khot [54] (with imperfect completeness) was resolved. As detailed in [56], this implies (together with [45]) that the sum-of-squares hierarchy does not solve AGC, which implies the same result for the weaker Sherali-Adams and bounded width hierarchies. Regarding the second type of algorithms, very recent work of Ciardo and Živný established that the AIP hierarchy does not solve AGC [32].

Neither of the two techniques, alone, is powerful enough to solve all tractable CSPs, even in the non-promise variant and on Boolean domain. In fact, the elusive interaction between linear equations and consistency-checking methods was the major obstacle to the proof of the Feder-Vardi dichotomy conjecture [40], finally settled in [25, 71]. Hence, efforts have been directed to *blending the two techniques*, in order to design a stronger *local-global* algorithm [15, 18]. In [18], Brakensiek, Guruswami, Wrochna, and Živný proposed an algorithm that combines the first level of the Sherali-Adams LP hierarchy (known as the *basic linear programming* (BLP) relaxation) with the AIP relaxation, and characterised its power. Remarkably, that algorithm, which we call BA in this paper, solves all tractable

cases of Schaefer’s dichotomy of Boolean CSPs [65]. This has led some researchers to believe that the algorithmic hierarchy based on BA could be a *universal constraint-satisfaction solver* – i.e., a constant level of the hierarchy could solve all tractable CSPs, cf. [18, 33, 35, 58].

Contributions. We prove that the BA hierarchy does not solve approximate graph colouring. This substantially extends the known results on non-solvability of AGC by the two weaker Sherali-Adams LP and AIP hierarchies separately. Ruling out the first level of the BA hierarchy is trivial using the characterisation from [18], while the task is significantly more challenging for higher levels. The core of our proof is geometric. Using the framework recently developed by Ciardo and Živný in [34] to study algorithmic hierarchies, we reduce the problem of finding a “fooling instance” for the BA hierarchy applied to AGC to the following geometric problem: building a high-dimensional integral tensor whose projections onto hyperplanes of fixed dimension are equal up to reflection (i.e., up to permutation of the tensor modes) and satisfy a sparsity condition, which dictates that certain entries of the tensor should be set to zero. The main technical result of this work is a constructive proof of the existence of tensors having these features. An active research trend in combinatorial matrix theory investigates the conditions for the existence of matrices (i.e., 2-dimensional tensors) over a certain domain having prescribed row- and column-sums (i.e., 1-dimensional projections) and a fixed *pattern*, i.e., a fixed set of entries allowed (or required) to be nonzero. Examples include 0-1 matrices with zero trace (i.e., adjacency matrices of digraphs) [41], with at most one fixed zero in each column [1], or with a fixed zero block [23], real matrices with a fixed pattern [49], and integral matrices with fixed lower and upper bounds on each entry [30]; see also related work in [24, 29, 36]. We believe that our work will stimulate further progress within that trend since, for the first time to our best knowledge, it extends the investigation from matrices to tensors of arbitrary dimension.

2 OVERVIEW OF RESULTS AND TECHNIQUES

Let X and A be two digraphs. We can cast the question “Is X homomorphic to A ” as the question of checking whether a system of linear equations has a solution in the set $\{0, 1\}$. Indeed, introduce the variables $\lambda_{x,a}$ for all vertices $x \in V(X)$, $a \in V(A)$, and the variables $\mu_{y,b}$ for all edges $y \in E(X)$, $b \in E(A)$, and consider the equations

$$\begin{aligned}
 (\text{IP}_1) \quad \sum_{a \in V(A)} \lambda_{x,a} &= 1 \quad \forall x \in V(X) \\
 (\text{IP}_2) \quad \sum_{\substack{b \in E(A) \\ b_i = a}} \mu_{y,b} &= \lambda_{y_i, a} \quad \forall y \in E(X), \forall i \in \{1, 2\}, \\
 &\quad \forall a \in V(A).
 \end{aligned} \tag{IP}$$

One readily checks that $X \rightarrow A$ if and only if (IP) has a solution in $\{0, 1\}$. Unless $P=NP$, this system is not solvable in polynomial time over $\{0, 1\}$. Relaxing it by allowing that the variables can be assigned rational nonnegative values (resp. integer values) results in the so-called *basic linear programming* (BLP) relaxation (resp. *affine integer programming* (AIP) relaxation). Note that both these

relaxations (as well as all relaxations we shall use in this work) result in algorithms that are complete but not necessarily sound, in the sense that they always output YES if $X \rightarrow A$, but may fail to output NO if $X \not\rightarrow A$. The BA relaxation described in [18] combines the two relaxations mentioned above as follows: It outputs YES if and only if there exist a solution to BLP and a solution to AIP such that the following so-called *refinement condition* holds: Whenever a variable is zero in the first solution, it is zero in the second solution. It follows that BA is at least as strong as both BLP and AIP; in fact, as shown in [18], it is strictly stronger, in the sense that there exist templates that are solved by BA but not by BLP or AIP.

The system (IP) can be refined by replacing the variables $\lambda_{x,a}$ with variables $\lambda_{S,f}$, where S is a set of vertices of X of size at most k and f is a function from S to $V(A)$. Solving such refined system over the set of nonnegative rational numbers (resp. integer numbers) would then mean finding rational nonnegative (resp. integer) distributions over the set of partial assignments from portions of the instance of size at most k to A . The former choice results in the Sherali-Adams LP hierarchy [67], which we call the BLP hierarchy; the latter results in the affine integer programming hierarchy [32], which we call the AIP hierarchy. Crucially, the former but not the latter choice ensures local consistency: Each assignment receiving nonzero weight in the BLP hierarchy corresponds to a partial homomorphism, while the same is not true for the AIP hierarchy. Equivalently, the BLP hierarchy is at least as strong as the bounded-width algorithm [6, 9, 40] (and, in fact, strictly stronger, see [4]). It is also worth noting that these hierarchies are still complete but not necessarily sound, and they become progressively stronger as the *level* k increases. In particular, the BLP hierarchy is “sound in the limit”, in the sense that its k -th level correctly classifies all instances of size k or less – which is clear from the fact that a partial homomorphism over the whole domain is a homomorphism.

The BA hierarchy we consider in this work consists in applying the BA relaxation of [18] to progressively larger portions of the instances, in the same spirit as the BLP and AIP hierarchies. Equivalently, the BA hierarchy can be described as follows: Its k -th level, applied to two digraphs X and A , outputs YES if and only if (i) the k -th level of both BLP and AIP outputs YES when applied to X and A , and (ii) the two solutions they provide satisfy the refinement condition. In this case, we write $BA^k(X, A) = \text{YES}$. Given two digraphs A, B such that $A \rightarrow B$, we say that the k -th level of BA solves PCSP(A, B) if, for any instance X , $BA^k(X, A) = \text{YES}$ implies $X \rightarrow B$.

The main result of our work is that no constant level of the BA hierarchy solves the approximate graph colouring problem.

Theorem 1. *For any fixed $3 \leq c \leq d$, there is no $k \in \mathbb{N}$ such that BA^k solves PCSP(K_c, K_d).*

A way to prove that approximate graph colouring is not solved by the BA hierarchy is to present *fooling instances* – digraphs with a large chromatic number but yet whose structure meets all constraints of the hierarchy. More precisely, it would suffice to build, for every c, d , and k , a digraph G whose chromatic number is higher than d and such that $BA^k(G, K_c) = \text{YES}$. Thus our goal is the following:

“Find a fooling instance for the BA hierarchy applied to AGC.”

Instead of directly looking for instances that fool the hierarchy, our approach shall be to consider the following questions: How does a *certificate of acceptance* for the BA hierarchy look like? Can we tell, from the shape of such a certificate, what the *limits* of the hierarchy applied to AGC are? The first step of our analysis is to translate the problem of whether the BA hierarchy accepts an input into a problem having a different, *multilinear* nature. Building on the framework developed in [34], we find that BA acceptance is implied by the existence of a family of *tensors* having certain special characteristics. First of all, they need to satisfy (i) a *system of symmetries*. This is essentially the result of the marginality constraints that are enforced by all “lift-and-project” hierarchies such as the BLP, AIP, and Lasserre SDP hierarchies [61], and is common to all algorithmic hierarchies studied in [34] through the tensor approach. There is, however, a feature that is typical of the BA hierarchy. Not only does BA require that both a linear program and a system of Diophantine equations have a solution; it also requires that any variable that is assigned zero weight by the former should be also assigned zero weight by the latter. This refinement condition of the relaxation introduced in [18] blends together the consistency-enforcing and linear-equation-solving techniques, to produce an algorithm that, as discussed above, is provably strictly stronger than both. The translation of the refinement condition into the multilinear framework is (ii) a *hollowness* requirement: Each tensor certifying BA acceptance needs to be hollow; i.e., it needs to contain zeros in certain prescribed locations. Summarising, the original problem has now become the following:

“Produce a family of hollow tensors satisfying a system of symmetries.”

There is a natural way to produce a family $\{T_i\}$ of tensors satisfying such symmetries: One starts with a high-dimensional tensor C whose low-dimensional oriented projections (i.e., projections onto oriented hyperplanes) are equal. Then, the family of *all* (not necessarily oriented) low-dimensional projections of C satisfies the required symmetries. We call such a tensor C a *crystal* (in accordance with [32]), while the *shadow* of C is any of its oriented projections. We then reformulate the problem to its final form; the solution of this problem is the main technical result of the paper.

“Find a crystal whose shadow is hollow.”

The rest of the paper is conceptually organised in three parts, each corresponding to a different phase of the proof of Theorem 1: (1) a *pre-processing* phase, where BA^k acceptance is turned into a multilinear problem; (2) a *multilinear* phase, where the multilinear problem is solved (i.e., hollow-shadowed crystals are built); (3) a *post-processing* phase, where the solution of the previous problem is translated back to the algorithmic framework, and it is used to recover a fooling instance. Full details of the three phases are discussed in the full version of this paper [31]. Sections 2.1, 2.2, and 2.3 below give a more intuitive overview of the contents of each of the phases.

2.1 The BA Hierarchy through Tensors

All hitherto studied relaxation algorithms for (promise) CSPs, including the BLP, AIP, and BA algorithms, are captured algebraically through the notion of *linear minion* – an algebraic structure consisting of matrices having a fixed number of columns and a variable

number of rows, that is closed under the application of elementary row operations (summing up or swapping two rows, inserting an extra zero row). Given a linear minion \mathcal{M} and a digraph \mathbf{A} with n vertices and m edges, there exists a natural way of simulating the structure of \mathbf{A} in \mathcal{M} , by defining a new (potentially infinite) digraph $\mathbb{F}_{\mathcal{M}}(\mathbf{A})$ (the *free structure* of \mathcal{M} generated by \mathbf{A}) whose vertices are the matrices in \mathcal{M} having n rows, and whose edges are pairs of matrices (M, N) such that both M and N can be obtained from some matrix Q having m rows through certain elementary row operations. Then, the relaxation corresponding to \mathcal{M} works as follows: Given an instance \mathbf{X} , rather than directly checking whether $\mathbf{X} \rightarrow \mathbf{A}$, one checks whether $\mathbf{X} \rightarrow \mathbb{F}_{\mathcal{M}}(\mathbf{A})$. The reason for doing so is that, for certain linear minions, the latter homomorphism problem is always tractable. As an example, stochastic rational vectors form a linear minion (since they are preserved under elementary row operations) named $\mathcal{Q}_{\text{conv}}$, whose corresponding relaxation is BLP. Similarly, integer vectors whose entries sum up to 1 form the linear minion \mathcal{Z}_{aff} corresponding to AIP. The framework developed in [34] allows to systematically strengthen the relaxation corresponding to any linear minion, by making use of the notion of *tensor power* of a digraph: For $k \in \mathbb{N}$, the k -th tensor power of \mathbf{A} is the *hypergraph* $\mathbf{A}^{\otimes k}$ whose vertices are k -tuples of vertices of \mathbf{A} , and whose edges are k -dimensional tensors obtained by “scattering” the edges of \mathbf{A} in k dimensions. The k -th level of the hierarchy of the relaxation corresponding to some linear minion \mathcal{M} essentially consists in applying the relaxation to the *tensorised* digraphs rather than the original digraphs; in other words, one checks if there exists a homomorphism $\mathbf{X}^{\otimes k} \rightarrow \mathbb{F}_{\mathcal{M}}(\mathbf{A}^{\otimes k})$.³ In addition, the homomorphism needs to preserve the tensor structure of the two hypergraphs (intuitively, it must “behave well with respect to projections”) – in which case, we say that it is a *k-tensorial* homomorphism. The algorithm obtained in this way is progressively stronger as k increases, and it still runs in polynomial time (for a fixed k) since the size of the tensorised digraph is polynomial in the size of the original digraph. In particular, if the matrices of \mathcal{M} satisfy a certain positivity requirement – in which case we say that the linear minion is *conic* – the hierarchy is *sound in the limit*, in the sense that its k -th level correctly classifies all instances \mathbf{X} on at most k vertices. In fact, the hierarchies based on conic minions enforce local consistency [34].⁴

How does the BA hierarchy fit within this framework? Given two linear minions \mathcal{M} and \mathcal{N} such that \mathcal{M} is conic, we define their *semi-direct product* $\mathcal{M} \ltimes \mathcal{N}$ as a new linear minion that is essentially designed in a way to make the corresponding relaxation stronger than the relaxations associated with \mathcal{M} and \mathcal{N} . In particular, one easily checks that $\mathcal{M} \ltimes \mathcal{N}$ is in fact a conic minion. Hence, this operation can be viewed as a standard way of making a given linear minion \mathcal{N} conic; or, in other words, a way to make a given relaxation locally consistent. The minion introduced in [18] corresponding to the first level of BA is the semi-direct product of the conic minion $\mathcal{Q}_{\text{conv}}$ and the linear minion \mathcal{Z}_{aff} . We can then capture algebraically the BA hierarchy as follows.

³We note that now $\mathbb{F}_{\mathcal{M}}(\mathbf{A}^{\otimes k})$ is a hypergraph rather than a digraph; the definitions are analogous.

⁴Intuitively, a linear minion is conic if any matrix 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.

Proposition 2. *Let \mathbf{X}, \mathbf{A} be two digraphs and let $k \in \mathbb{N}$. Then $\text{BA}^k(\mathbf{X}, \mathbf{A}) = \text{YES}$ if and only if there exists a k -tensorial homomorphism $\vartheta : \mathbf{X}^{\otimes k} \rightarrow \mathbb{F}_{\mathcal{Q}_{\text{conv}} \ltimes \mathcal{Z}_{\text{aff}}}(\mathbf{A}^{\otimes k})$.*

Recall that the goal of this pre-processing phase is to come up with a multilinear criterion to check whether BA^k accepts an instance of AGC. From the way the semi-direct product is defined, it follows that a homomorphism ϑ from $\mathbf{X}^{\otimes k}$ to $\mathbb{F}_{\mathcal{Q}_{\text{conv}} \ltimes \mathcal{Z}_{\text{aff}}}(\mathbf{A}^{\otimes k})$ can be *decoupled* into a homomorphism ξ to $\mathbb{F}_{\mathcal{Q}_{\text{conv}}}(\mathbf{A}^{\otimes k})$ and a homomorphism ζ to $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{A}^{\otimes k})$. If \mathbf{A} is a clique – as it happens when the BA hierarchy is applied to AGC – one can design a simpler sufficient criterion, based on the fact that one can always assume ξ to be the homomorphism mapping a tuple of vertices of \mathbf{X} to a tensor in $\mathbb{F}_{\mathcal{Q}_{\text{conv}}}(\mathbf{A}^{\otimes k})$ that is uniform on its support. After dealing with some combinatorial technicalities, this fact produces the following criterion of acceptance.

Theorem 3. *Let $2 \leq k \leq n \in \mathbb{N}$, let \mathbf{X} be a loopless digraph, and let $\zeta : \mathbf{X}^{\otimes k} \rightarrow \mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k})$ be a k -tensorial homomorphism such that $E_{\mathbf{a}} * \zeta(\mathbf{x}) = 0$ for any $\mathbf{x} \in V(\mathbf{X})^k$ and $\mathbf{a} \in \{1, \dots, n\}^k$ for which $\mathbf{a} \not\prec \mathbf{x}$. Then $\text{BA}^k(\mathbf{X}, \mathbf{K}_n) = \text{YES}$.⁵*

2.2 Crystals with Hollow Shadows

The criterion of acceptance for BA^k stated in Theorem 3 is multilinear. Indeed, $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k})$ is a space of integer affine tensors (i.e., whose entries sum up to 1), and the existence of a k -tensorial homomorphism from $\mathbf{X}^{\otimes k}$ to $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_n^{\otimes k})$ corresponds to the existence of a family of tensors satisfying a specific system of symmetries. Letting q be the number of vertices in \mathbf{X} , such a family can be realised as the family of k -dimensional projections of a single affine q -dimensional *crystal* tensor, which we next informally define. We let $\mathcal{T}^{n \cdot 1 q}(\mathbb{Z})$ denote the set of all integer *cubical tensors of dimension q and width n* – i.e., $n \times n \times \dots \times n$ arrays of integer numbers, where n appears q times. The notion of projecting should intuitively be thought of as “summing up all entries of a tensor along a certain set of directions”; the formal definition shall make use of the operation of *tensor contraction*, which is defined in the full version [31]. By oriented projection we mean that the directions are considered to be ordered.

Definition 4 (Informal). Let $q, n \in \mathbb{N}$ and $k \in \{0, \dots, q\}$. A cubical tensor $C \in \mathcal{T}^{n \cdot 1 q}(\mathbb{Z})$ is a *k-crystal* if all its k -dimensional oriented projections are equal. In this case, the *k-shadow* of C is this common oriented projection.

Equivalently, a k -crystal is required to have equal k -dimensional projections *up to reflection* – a reflection is a higher-dimensional analogue of the transpose operation. Let ζ_C be the map associated with an affine k -crystal C , which takes a k -tuple \mathbf{x} of vertices of \mathbf{X} and maps it to the projection of C onto the hyperplane generated by \mathbf{x} . By construction, ζ_C behaves well with respect to projections, so it is automatically k -tensorial. In order to yield a certificate of acceptance for $\text{BA}^k(\mathbf{X}, \mathbf{K}_n)$, according to Theorem 3, ζ_C also needs to be a homomorphism and satisfy the extra condition $\mathbf{a} \not\prec \mathbf{x} \Rightarrow E_{\mathbf{a}} * \zeta_C(\mathbf{x}) = 0$. It turns out that both these requirements translate

⁵Here, $E_{\mathbf{a}} * \zeta(\mathbf{x})$ denotes the \mathbf{a} -th entry of the tensor $\zeta(\mathbf{x})$, while $\mathbf{a} \not\prec \mathbf{x}$ means that there exist two indices i, j for which $a_i = a_j$ but $x_i \neq x_j$.

as a condition on the k -shadow S of C : The only entries of S allowed to be nonzero are those whose coordinates are all distinct. We say that a tensor having this property is *hollow*. As an example, if $k = 2$, the condition means that the $n \times n$ matrix S needs to have zero diagonal; if $k = 3$, three diagonal planes of the $n \times n \times n$ tensor S of the form (a, a, b) , (a, b, a) , (b, a, a) should be set to zero, and so on.

In summary, the discussion above indicates that an affine k -crystal of dimension q and width n whose k -shadow is hollow yields a certificate that $\text{BA}^k(\mathbf{X}, \mathbf{K}_n) = \text{Yes}$ for *any* loopless digraph \mathbf{X} with q vertices. The problem is now to verify whether hollow-shadowed crystals exist. It is not hard to check that such crystals cannot exist for all choices of k , q , and n ; this parallels the fact that the BA hierarchy is sound in the limit, so it cannot be the case that *any* \mathbf{X} is accepted by *any* level of BA applied to *any* clique \mathbf{K}_n . This is in sharp contrast with the weaker AIP hierarchy, for which a similar acceptance result holds, cf. [32]. Hence, unlike for AIP, one cannot simply take large cliques as fooling instances for BA. As we shall see in Section 2.3, a more refined family of digraphs can be shown to provide fooling instances for the BA hierarchy as long as one can produce hollow-shadowed crystals whose width n is *sub-exponential* in the level k . The main technical contribution of this work is a method for mining hollow-shadowed crystals whose width is *quadratic* in k , as stated next.

Theorem 5. For any $k \leq q \in \mathbb{N}$ there is an affine k -crystal $C \in \mathcal{T}^{\frac{k^2+k}{2} \cdot 1_q}(\mathbb{Z})$ with hollow k -shadow.

The key to establishing Theorem 5 is proving the following.

Theorem 6. For any $k \in \mathbb{N}$ there exists a hollow affine $(k-1)$ -crystal $C \in \mathcal{T}^{\frac{k^2+k}{2} \cdot 1_k}(\mathbb{Z})$.

We now discuss the main ideas of the proof of Theorem 6 for the case $k = 3$.

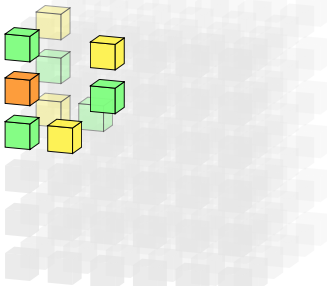


Figure 1: The crystal W .

Our goal is to find a hollow affine 2-crystal $C \in \mathcal{T}^{6 \cdot 1_3}(\mathbb{Z})$. In other words, C must be a three-dimensional cubical tensor of width 6, such that (i) C is hollow, i.e., the only entries allowed to be nonzero are the ones whose three coordinates are all distinct; (ii) C is affine, i.e., its entries sum up to 1;

and (iii) C is a 2-crystal, i.e., projecting it onto the xy -, yz -, and xz -planes results in the same 6×6 “shadow” matrix. By induction, we can assume that Theorem 6 holds for $k = 2$. In fact, it is not hard to find by inspection that the matrix⁶ $U = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \text{green} & \text{light grey} & \text{yellow} \\ \text{orange} & \text{light grey} & \text{light grey} \\ \text{light grey} & \text{light grey} & \text{light grey} \end{bmatrix}$ is a hollow affine 1-crystal in $\mathcal{T}^{3 \cdot 1_2}(\mathbb{Z})$.

⁶We indicate the numbers $-1, 0, 1$, and 2 by the colours green, light grey, yellow, and orange, respectively.

The next step is to build a (not necessarily hollow) 3-dimensional 2-crystal having shadow U . This can be done using the method developed in [32] as a black box, and it results in the tensor $V =$

$$\begin{bmatrix} -1 & 0 & 1 & | & 0 & 0 & 0 & | & 1 & 0 & 0 \\ 2 & 0 & -1 & | & 0 & 0 & 0 & | & -1 & 0 & 0 \\ -1 & 1 & 0 & | & 0 & 0 & 0 & | & 1 & -1 & 0 \end{bmatrix} = \begin{matrix} \text{green} & \text{light grey} & \text{yellow} & \text{orange} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} \\ \text{orange} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} \\ \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} & \text{light grey} \end{matrix}.$$

Clearly, V is not hollow – for example, its $(1, 1, 1)$ -th coordinate is $-1 \neq 0$. In fact, it is not hard to check that a hollow affine 2-crystal of dimension 3 and width 3 cannot exist.

We need to increase the width to “make more space”; we do so by padding V with three layers of zeros along each of the three dimensions. The tensor W we obtain in this way (see Figure 1) is clearly still a 2-crystal. We can view W as a block tensor with eight $3 \times 3 \times 3$ blocks; note that all

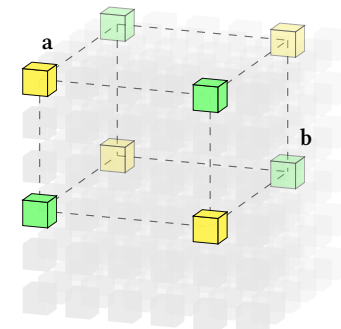


Figure 2: The quartz $Q_{a,b}$.

non-zero entries of W are in one block. The idea is now to “spread” these entries in the other blocks, in a way that they occupy positions whose indices have no repetitions. To this end, we make use of a particular class of “transparent” crystals that we call *quartzes*. Such crystals are designed in a way that the shadow they project is identically zero, meaning that we can freely add them (or their integer multiples) to a given crystal without changing its shadow and maintaining it a crystal.

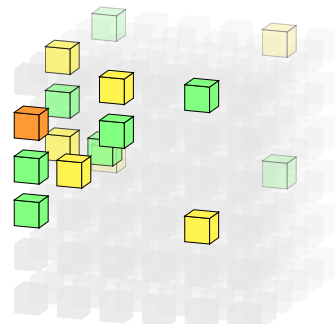


Figure 3: $W - w_{(1,1,1)} \cdot Q_{(1,1,1),b}$.

A quartz can be built by choosing two cells a and b having disjoint coordinates, considering the parallelepiped generated by a and b , assigning value 1 or -1 to its vertices in a way that two adjacent vertices get values of opposite sign, and assigning value 0 to all other cells; we refer to such quartz as to $Q_{a,b}$,

see Figure 2. (This construction is easily generalised to an arbitrary dimension.) Quartzes yield a method to relocate some nonzero entry of W , while leaving the rest of W *almost* untouched. More precisely, if the a -th entry of W has value $w_a \neq 0$, the a -th entry of $W - w_a \cdot Q_{a,b}$ is zero, and this operation modifies the value of only 8 cells of W . The idea is then to perturb W with suitable quartzes, so as to transfer all nonzero entries to positions where they do not violate the hollowness requirement.

To this end, we take as b a fixed cell that generates the smallest number of ties and that lies in the block of W opposite to the one

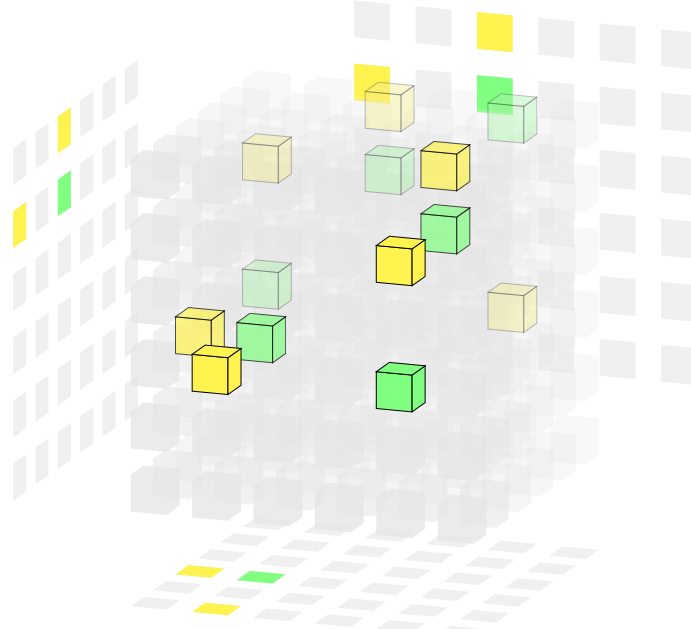


Figure 4: The hollow crystal C , and its shadows.

containing the nonzero entries – for example, the cell $\mathbf{b} = (4, 5, 6)$. Even with such a choice, it can happen that adding a multiple of a quartz introduces new nonzero entries in positions that violate hollowness. For example, Figure 3 shows the tensor $W - w_{(1,1,1)} \cdot Q_{(1,1,1),\mathbf{b}}$. The cell $(1, 1, 1)$ has become zero, as wanted, but three new forbidden cells $((1, 1, 6), (1, 5, 1), (4, 1, 1))$ now have nonzero values. However, the nonzero values in these forbidden cells cancel out once this procedure is applied to *all* entries in the nonzero block of W . In other words, the affine 2-crystal

$$C = W - \sum_{\mathbf{a} \in \{1,2,3\}^3} w_{\mathbf{a}} \cdot Q_{\mathbf{a},\mathbf{b}}$$

is hollow (see Figure 4).

2.3 Fooling the BA Hierarchy

Let C be an affine k -crystal of dimension q and width $\frac{k^2+k}{2}$ whose k -shadow is hollow, as in Theorem 5. Let \mathbf{X} be a loopless digraph on vertex set $V(\mathbf{X}) = \{1, \dots, q\}$. Consider the map ζ_C taking as input a tuple \mathbf{x} of k vertices of \mathbf{X} (i.e., a tuple of k numbers in $\{1, \dots, q\}$) and returning the k -dimensional projection of C onto the hyperplane corresponding to \mathbf{x} . As discussed earlier, ζ_C yields a k -tensorial homomorphism from $\mathbf{X}^{\otimes k}$ to $\mathbb{F}_{\mathcal{Z}_{\text{aff}}}(\mathbf{K}_{(k^2+k)/2}^{\otimes k})$, and the fact that the shadow of C is hollow translates as ζ_C satisfying the extra requirement of Theorem 3. Hence, we obtain the following.

Proposition 7. *Let $2 \leq k \in \mathbb{N}$ and let \mathbf{X} be a loopless digraph. Then $\text{BA}^k(\mathbf{X}, \mathbf{K}_{(k^2+k)/2}) = \text{YES}$.*

To prove Theorem 1, we need to show that BA^k does not solve $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$ for all choices of $k \in \mathbb{N}$ and $3 \leq c \leq d \in \mathbb{N}$. If $c = \frac{k^2+k}{2}$, any graph with chromatic number bigger than d (for example, the clique \mathbf{K}_{d+1}) would then yield a fooling instance. Since increasing c can only make AGC harder, this argument shows that

BA^k does not solve $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$ whenever $c \geq \frac{k^2+k}{2}$, and the fooling instances are simply cliques.

In order to establish Theorem 1 in full generality, however, we shall pick the fooling instances from a more refined class of digraphs: the so-called *shift digraphs* (see Figure 5).

Definition 8. The *line digraph* of a digraph \mathbf{X} is the digraph $\delta\mathbf{X}$ defined by $V(\delta\mathbf{X}) = E(\mathbf{X})$ and

$$E(\delta\mathbf{X}) = \{((x, y), (y, z)) : (x, y), (y, z) \in E(\mathbf{X})\}.$$

Definition 9. Let $q \in \mathbb{N}$ and $i \in \mathbb{N}_0$. The *shift digraph* $\mathbf{S}_{q,i}$ is recursively defined by setting $\mathbf{S}_{q,0} = \mathbf{K}_q$, $\mathbf{S}_{q,i} = \delta\mathbf{S}_{q,i-1}$ for each $i \geq 1$.

It is not hard to verify that the following non-recursive description of shift digraphs is equivalent to Definition 9 for $i \geq 1$: $\mathbf{S}_{q,i}$ is the digraph whose vertex set consists of all strings of length $i+1$ over the alphabet $\{1, \dots, q\}$ such that consecutive letters are distinct, and whose edge set consists of all pairs of strings of the form $(a_1 \dots a_k, a_2 \dots a_{k+1})$.⁷ In particular, it is clear from this description that the edge set of $\mathbf{S}_{q,i}$ is nonempty for $q \geq 2$.

The line digraph construction has been utilised in [45, 59] as a polynomial-time (and in fact log-space) reduction between PCSPs. In particular, the construction changes the chromatic number in a controlled way, as we now describe. Consider the integer functions a and b defined by $a(p) = 2^p$ and $b(p) = \binom{p}{\lfloor p/2 \rfloor}$ for $p \in \mathbb{N}$, and notice that $a(p) \geq b(p)$ for each p . Let also $a^{(i)}$ (resp. $b^{(i)}$) be the function obtained by iterating a (resp. b) i -many times, for $i \in \mathbb{N}$. The following result bounds the chromatic number of the line digraph in terms of that of the original digraph.

Theorem 10 ([46]). *Let \mathbf{X} be a digraph and let $p \in \mathbb{N}$. If $\delta\mathbf{X} \rightarrow \mathbf{K}_p$, then $\mathbf{X} \rightarrow \mathbf{K}_{a(p)}$; if $\mathbf{X} \rightarrow \mathbf{K}_{b(p)}$, then $\delta\mathbf{X} \rightarrow \mathbf{K}_p$.*

⁷In [47, § 2.5], a slightly different definition of shift digraphs is given, where the case $i = 0$ is a transitive tournament rather than a clique; equivalently, the vertex set of $\mathbf{S}_{q,i}$ only includes *monotonically increasing* strings.

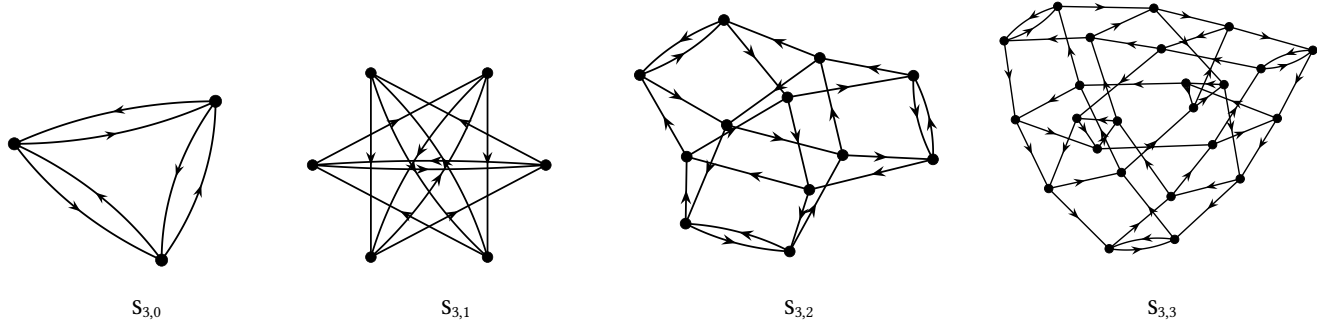


Figure 5: Shift digraphs.

An interesting feature of the line digraph operator is that it preserves acceptance by hierarchies of relaxations corresponding to conic minions, at the only cost of halving the level. As stated next, this in particular holds for the BA hierarchy, whose corresponding minion $\mathcal{Q}_{\text{conv}} \times \mathcal{Z}_{\text{aff}}$ is conic.

Proposition 11. *Let $2 \leq k \in \mathbb{N}$, let X, A be digraphs, and suppose that $\text{BA}^{2k}(X, A) = \text{YES}$ and $E(\delta A) \neq \emptyset$. Then $\text{BA}^k(\delta X, \delta A) = \text{YES}$.*

The key point is that, under the application of the line digraph operator, a digraph decreases *exponentially* fast in terms of chromatic number, but only *polynomially* fast in terms of BA acceptance level. Intuitively, our strategy to fool BA^k as an algorithm to solve $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$ will be to take as the fooling instance a shift digraph $S_{q,i}$ where $q \sim \exp^{(i)}(d+1)$, rather than the clique \mathbf{K}_{d+1} .⁸ Chromatically, this digraph is similar to \mathbf{K}_{d+1} by Theorem 10, so it is not d -colourable. On the other hand, for large enough i , the difference in speed decrease guarantees that $\text{BA}^{\text{pol}^{(i)}(k)}(\mathbf{K}_q, \mathbf{K}_{\exp^{(i)}(c)}) = \text{YES}$ by Proposition 7 – which, through Proposition 11, eventually implies $\text{BA}^k(S_{q,i}, \mathbf{K}_c) = \text{YES}$. We note that this argument crucially depends on the fact that the size $\frac{k^2+k}{2}$ of the clique in Proposition 7 – i.e., the width of the hollow-shadowed crystals mined in Section 2.2 – is sub-exponential in k . Before proving Theorem 1 in full detail, we present a result, true for all linear minions, stating that acceptance of some instance X by some level of the hierarchy is preserved under homomorphisms of the template.

Proposition 12. *Let $k \in \mathbb{N}$, let X, A, B be digraphs such that $A \rightarrow B$, and suppose that $\text{BA}^k(X, A) = \text{YES}$. Then $\text{BA}^k(X, B) = \text{YES}$.*

PROOF OF THEOREM 1. Since BA^2 is at least as powerful as BA^1 , we can assume that $k \geq 2$. Suppose first that $c \geq 4$. In this case, we can find $i \in \mathbb{N}$ such that $b^{(i)}(c) \geq k^2 4^i$. Take $q > a^{(i)}(d)$. We claim that the shift digraph $S_{q,i}$ is a fooling instance for the k -th level of BA applied to $\text{PCSP}(\mathbf{K}_c, \mathbf{K}_d)$; in other words, we claim that (i) $\text{BA}^k(S_{q,i}, \mathbf{K}_c) = \text{YES}$ and (ii) $S_{q,i} \not\rightarrow \mathbf{K}_d$.

For (i), we start by applying Proposition 7 to find that

$$\text{BA}^{k2^i}(\mathbf{K}_q, \mathbf{K}_{(k^2 4^i + k 2^i)/2}) = \text{YES}.$$

Observe that $\frac{k^2 4^i + k 2^i}{2} \leq k^2 4^i \leq b^{(i)}(c)$, so

$$\mathbf{K}_{(k^2 4^i + k 2^i)/2} \rightarrow \mathbf{K}_{k^2 4^i} \rightarrow \mathbf{K}_{b^{(i)}(c)}.$$

⁸Here by $\exp^{(i)}(\cdot)$ (resp., $\text{pol}^{(i)}(\cdot)$) we mean a function obtained by iterating i -many times an exponential (resp. polynomial) function.

By Proposition 12, we deduce that $\text{BA}^{k2^i}(\mathbf{K}_q, \mathbf{K}_{b^{(i)}(c)}) = \text{YES}$. Applying Proposition 11 repeatedly, we obtain $\text{BA}^k(S_{q,i}, \mathbf{K}_{b^{(i)}(c),i}) = \text{YES}$. Noticing that $\mathbf{K}_{b^{(i)}(c)} \rightarrow \mathbf{K}_{b^{(i)}(c)}$ and applying the second part of Theorem 10 repeatedly, we find $\mathbf{K}_{b^{(i)}(c),i} \rightarrow \mathbf{K}_c$. Again by Proposition 12, we conclude that $\text{BA}^k(S_{q,i}, \mathbf{K}_c) = \text{YES}$, as required. For (ii), we first note that $\mathbf{K}_q \not\rightarrow \mathbf{K}_{a^{(i)}(d)}$ as $q > a^{(i)}(d)$. Applying the (contrapositive of the) first part of Theorem 10 repeatedly, we deduce that $S_{q,i} \not\rightarrow \mathbf{K}_d$, as required.

Suppose now that $c = 3$. Assume, for the sake of contradiction, that the k -th level of BA solves $\text{PCSP}(\mathbf{K}_3, \mathbf{K}_d)$. Let X be a digraph such that $\text{BA}^{4k}(X, \mathbf{K}_4) = \text{YES}$. Applying Proposition 11 twice, we find that $\text{BA}^k(\delta(\delta X), S_{4,2}) = \text{YES}$. We now use the fact, observed in [59, Lemma 4.19], that $S_{4,2} \rightarrow \mathbf{K}_3$; combining this with Proposition 12 yields $\text{BA}^k(\delta(\delta X), \mathbf{K}_3) = \text{YES}$. Since we are assuming that BA^k solves $\text{PCSP}(\mathbf{K}_3, \mathbf{K}_d)$, we must have $\delta(\delta X) \rightarrow \mathbf{K}_d$, whence it follows, through a double application of the first part of Theorem 10, that $X \rightarrow \mathbf{K}_{a^{(2)}(d)}$. Thus, we have shown that the $(4k)$ -th level of BA solves $\text{PCSP}(\mathbf{K}_4, \mathbf{K}_{a^{(2)}(d)})$, which is a PCSP template as $d \geq c = 3$ implies $a^{(2)}(d) = 2^{2^d} \geq 2^{2^3} \geq 4$, so $\mathbf{K}_4 \rightarrow \mathbf{K}_{a^{(2)}(d)}$. This contradicts the argument above establishing the case of $c \geq 4$. \square

ACKNOWLEDGMENTS

This research was funded in whole by UKRI EP/X024431/1. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper. The authors are grateful to Jakub Opršal, who suggested to us a connection between the line digraph reduction and the Sherali-Adams LP hierarchy for approximate graph colouring.

REFERENCES

- [1] Richard P. Anstee. 1982. Properties of a class of $(0, 1)$ -matrices covering a given matrix. *Canadian Journal of Mathematics* 34, 2 (1982), 438–453. <https://doi.org/10.4153/CJM-1982-029-3>
- [2] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. 2006. Proving Integrality Gaps without Knowing the Linear Program. *Theory Comput.* 2, 2 (2006), 19–51. <https://doi.org/10.4086/toc.2006.v002a002>
- [3] Kristina Asimi and Libor Barto. 2021. Finitely Tractable Promise Constraint Satisfaction Problems. In *Proc. 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21) (LIPIcs, Vol. 202)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 11:1–11:16. <https://doi.org/10.4230/LIPIcs.MFCS.2021.11>

- [4] Albert Atserias and Víctor Dalmau. 2022. Promise Constraint Satisfaction and Width. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. 1129–1153. <https://doi.org/10.1137/1.9781611977073.48> arXiv:2107.05886
- [5] Per Austrin, Venkatesan Guruswami, and Johan Hästad. 2017. $(2+\epsilon)$ -Sat Is NP-hard. *SIAM J. Comput.* 46, 5 (2017), 1554–1573. <https://doi.org/10.1137/15M1006507> eccc:2013/159
- [6] Libor Barto. 2016. The collapse of the bounded width hierarchy. *J. Log. Comput.* 26, 3 (2016), 923–943. <https://doi.org/10.1093/logcom/exu070>
- [7] Libor Barto, Diego Battistelli, and Kevin M. Berg. 2021. Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case. In *Proc. 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21) (LIPIcs, Vol. 187)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 10:1–10:16. <https://doi.org/10.4230/LIPIcs.STACS.2021.10> arXiv:2010.04623
- [8] Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. 2021. Algebraic approach to promise constraint satisfaction. *J. ACM* 68, 4 (2021), 28:1–28:66. <https://doi.org/10.1145/3457606> arXiv:1811.00970
- [9] Libor Barto and Marcin Kozik. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM* 61, 1 (2014). <https://doi.org/10.1145/2556646> Article No. 3.
- [10] Libor Barto and Marcin Kozik. 2022. Combinatorial Gap Theorem and Reductions between Promise CSPs. In *Proc. 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. 1204–1220. <https://doi.org/10.1137/1.9781611977073.50> arXiv:2107.09423
- [11] Christoph Berkholz and Martin Grohe. 2017. Linear Diophantine Equations, Group CSPs, and Graph Isomorphism. In *Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*. SIAM, 327–339. <https://doi.org/10.1137/1.9781611974782.21> arXiv:1607.04287
- [12] Amey Bhargale and Subhash Khot. 2021. Optimal Inapproximability of Satisfiable k -LIN over Non-Abelian Groups. In *Proc. 53rd Annual ACM Symposium on Theory of Computing (STOC'21)*. ACM, 1615–1628. <https://doi.org/10.1145/3406325.3451003> arXiv:2009.02815
- [13] Amey Bhargale, Subhash Khot, and Don Minzer. 2022. On Inapproximability of Satisfiable k -CSPs: I. In *Proc. 54th Annual ACM Symposium on Theory of Computing (STOC'22)*. ACM, 976–988. <https://doi.org/10.1145/3519935.3520028>
- [14] Joshua Brakensiek and Venkatesan Guruswami. 2016. New Hardness Results for Graph and Hypergraph Colorings. In *Proc. 31st Conference on Computational Complexity (CCC'16) (LIPIcs, Vol. 50)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 14:1–14:27. <https://doi.org/10.4230/LIPIcs.CCC.2016.14>
- [15] Joshua Brakensiek and Venkatesan Guruswami. 2019. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*. 436–455. <https://doi.org/10.1137/1.9781611975482.28> arXiv:1807.05194
- [16] Joshua Brakensiek and Venkatesan Guruswami. 2021. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.* 50, 6 (2021), 1663–1700. <https://doi.org/10.1137/19M128212X> arXiv:1704.01937
- [17] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2021. Conditional Dichotomy of Boolean Ordered Promise CSPs. In *Proc. 48th International Colloquium on Automata, Languages, and Programming (ICALP'21) (LIPIcs, Vol. 198)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 37:1–37:15. <https://doi.org/10.4230/LIPIcs.ICALP.2021.37> arXiv:2102.11854
- [18] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. 2020. The power of the combined basic LP and affine relaxation for promise CSPs. *SIAM J. Comput.* 49 (2020), 1232–1248. Issue 6. <https://doi.org/10.1137/20M1312745> arXiv:1907.04383
- [19] Alex Brandts, Marcin Wrochna, and Stanislav Živný. 2021. The complexity of promise SAT on non-Boolean domains. *ACM Trans. Comput. Theory* 13, 4 (2021), 26:1–26:20. <https://doi.org/10.1145/3470867> arXiv:1911.09065
- [20] Gábor Braun, Sebastian Pokutta, and Daniel Zink. 2015. Inapproximability of Combinatorial Problems via Small LPs and SDPs. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC'15)*. ACM, 107–116. <https://doi.org/10.1145/2746539.2746550>
- [21] Mark Braverman, Subhash Khot, Noam Lifshitz, and Dor Minzer. 2021. An Invariance Principle for the Multi-slice, with Applications. In *Proc. 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'21)*. IEEE, 228–236. <https://doi.org/10.1109/FOCS52979.2021.00030>
- [22] Mark Braverman, Subhash Khot, and Dor Minzer. 2021. On Rich 2-to-1 Games. In *Proc. 12th Innovations in Theoretical Computer Science Conference (ITCS'21) (LIPIcs, Vol. 185)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 27:1–27:20. <https://doi.org/10.4230/LIPIcs.ITCS.2021.27>
- [23] Richard A. Brualdi and Geir Dahl. 2003. Matrices of zeros and ones with given line sums and a zero block. *Linear Algebra Appl.* 371 (2003), 191–207. [https://doi.org/10.1016/S0024-3795\(03\)00429-4](https://doi.org/10.1016/S0024-3795(03)00429-4)
- [24] R. A. Brualdi and G. Dahl. 2007. Constructing $(0, 1)$ -matrices with given line sums and certain fixed zeros. In *Advances in discrete tomography and its applications*. Birkhäuser Boston, Boston, MA, 113–123. https://doi.org/10.1007/978-0-8176-4543-4_6
- [25] Andrei A. Bulatov. 2017. A Dichotomy Theorem for Nonuniform CSPs. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*. 319–330. <https://doi.org/10.1109/FOCS.2017.37> arXiv:1703.03021
- [26] Silvia Butti and Víctor Dalmau. 2021. 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) (LIPIcs, Vol. 202)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 27:1–27:19. <https://doi.org/10.4230/LIPIcs.MFCS.2021.27> arXiv:2107.02956
- [27] Siu On Chan. 2016. Approximation Resistance from Pairwise-Independent Subgroups. *J. ACM* 63, 3 (2016), 27:1–27:32. <https://doi.org/10.1145/2873054>
- [28] Siu On Chan, James R. Lee, Prasad Ravhendra, and David Steurer. 2016. Approximate Constraint Satisfaction Requires Large LP Relaxations. *J. ACM* 63, 4 (2016), 34:1–34:22. <https://doi.org/10.1145/2811255>
- [29] Wei Chen, Yanfang Mo, Li Qiu, and Pravin Varaiya. 2016. Constrained $(0, 1)$ -matrix completion with a staircase of fixed zeros. *Linear Algebra Appl.* 510 (2016), 171–185. <https://doi.org/10.1016/j.laa.2016.08.020>
- [30] William Y. C. Chen. 1992. Integral matrices with given row and column sums. *J. Combin. Theory Ser. A* 61, 2 (1992), 153–172. [https://doi.org/10.1016/0097-3165\(92\)90015-M](https://doi.org/10.1016/0097-3165(92)90015-M)
- [31] Lorenzo Ciardo and Stanislav Živný. 2022. *Approximate Graph Colouring and the Hollow Shadow*. Technical Report. arXiv:2211.03168
- [32] Lorenzo Ciardo and Stanislav Živný. 2023. Approximate Graph Colouring and Crystals. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. 2256–2267. <https://doi.org/10.1137/1.9781611977554.ch86> arXiv:2210.08293
- [33] Lorenzo Ciardo and Stanislav Živný. 2023. CLAP: A New Algorithm for Promise CSPs. *SIAM J. Comput.* 52, 1 (2023), 1–37. <https://doi.org/10.1137/22M1476435> arXiv:2107.05018
- [34] Lorenzo Ciardo and Stanislav Živný. 2023. Hierarchies of minion tests for PCSPs through tensors. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. 568–580. <https://doi.org/10.1137/1.9781611977554.ch25> arXiv:2207.02277
- [35] Adam Ó Conghaile. 2022. Cohomology in Constraint Satisfaction and Structure Isomorphism. In *Proc. 47th International Symposium on Mathematical Foundations of Computer Science (MFCS'22) (LIPIcs, Vol. 241)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 75:1–75:16. <https://doi.org/10.4230/LIPIcs.MFCS.2022.75> arXiv:2206.15253
- [36] Geir Dahl. 2008. Transportation matrices with staircase patterns and majorization. *Linear Algebra Appl.* 429, 7 (2008), 1840–1850. <https://doi.org/10.1016/j.laa.2008.05.019>
- [37] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. 2018. On non-optimally expanding sets in Grassmann graphs. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*. ACM, 940–951. <https://doi.org/10.1145/3188745.3188806>
- [38] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. 2018. Towards a proof of the 2-to-1 games conjecture?. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*. ACM, 376–389. <https://doi.org/10.1145/3188745.3188804>
- [39] Irit Dinur, Elchanan Mossel, and Oded Regev. 2009. Conditional Hardness for Approximate Coloring. *SIAM J. Comput.* 39, 3 (2009), 843–873. <https://doi.org/10.1137/07068062X>
- [40] Tomás Feder and Moshe Y. Vardi. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.* 28, 1 (1998), 57–104. <https://doi.org/10.1137/S0097539794266766>
- [41] D. R. Fulkerson. 1960. Zero-one matrices with zero trace. *Pacific J. Math.* 10 (1960), 831–836. <https://doi.org/10.2140/pjm.1960.10.831>
- [42] M. R. Garey and David S. Johnson. 1976. The Complexity of Near-Optimal Graph Coloring. *J. ACM* 23, 1 (1976), 43–49. <https://doi.org/10.1145/321921.321926>
- [43] Mrinalkanti Ghosh and Madhur Tulsiani. 2018. From Weak to Strong Linear Programming Gaps for All Constraint Satisfaction Problems. *Theory Comput.* 14, 1 (2018), 1–33. <https://doi.org/10.4086/toc.2018.v014a010>
- [44] Venkatesan Guruswami and Sanjeev Khanna. 2004. On the Hardness of 4-Coloring a 3-Colorable Graph. *SIAM Journal on Discrete Mathematics* 18, 1 (2004), 30–40. <https://doi.org/10.1137/S0895480100376794>
- [45] Venkatesan Guruswami and Sai Sandeep. 2020. 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) (LIPIcs, Vol. 168)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 62:1–62:12. <https://doi.org/10.4230/LIPIcs.ICALP.2020.62>
- [46] C.C. Harner and R.C. Entringer. 1972. Arc Colorings of Digraphs. *J. Comb. Theory, Ser. B* 13, 3 (1972), 219–225. [https://doi.org/10.1016/0095-8956\(72\)90057-3](https://doi.org/10.1016/0095-8956(72)90057-3)
- [47] Pavol Hell and Jaroslav Nešetřil. 2004. *Graphs and homomorphisms*. Oxford Lecture Series in Mathematics and its Applications, Vol. 28. OUP Oxford.
- [48] Sangxia Huang. 2013. 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)*. Springer, 233–243. https://doi.org/10.1007/978-3-642-40328-6_17 arXiv:1301.5216
- [49] Charles R. Johnson and David P. Stanford. 2000. Patterns that allow given row and column sums. *Linear Algebra Appl.* 311, 1-3 (2000), 97–105. [https://doi.org/10.1016/S0024-3795\(00\)00071-9](https://doi.org/10.1016/S0024-3795(00)00071-9)
- [50] Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proc. Complexity of Computer Computations*. 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9
- [51] Ken-ichi Kawarabayashi and Mikkel Thorup. 2017. Coloring 3-Colorable Graphs with Less than $n^{1/5}$ Colors. *J. ACM* 64, 1 (2017), 4:1–4:23. <https://doi.org/10.1145/3001582>
- [52] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. 2000. On the Hardness of Approximating the Chromatic Number. *Comb.* 20, 3 (2000), 393–415. <https://doi.org/10.1007/s004930070013>
- [53] Subhash Khot. 2001. Improved Inapproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*. IEEE Computer Society, 600–609. <https://doi.org/10.1109/SFCS.2001.959936>
- [54] Subhash Khot. 2002. On the power of unique 2-prover 1-round games. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*. ACM, 767–775. <https://doi.org/10.1145/509907.510017>
- [55] Subhash Khot, Dor Minzer, and Muli Safra. 2017. On independent sets, 2-to-2 games, and Grassmann graphs. In *Proc. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17)*. ACM, 576–589. <https://doi.org/10.1145/3055399.3055432>
- [56] Subhash Khot, Dor Minzer, and Muli Safra. 2018. Pseudorandom Sets in Grassmann Graph Have Near-Perfect Expansion. In *Proc. 59th IEEE Annual Symposium on Foundations of Computer Science (FOCS'18)*. IEEE Computer Society, 592–601. <https://doi.org/10.1109/FOCS.2018.00062>
- [57] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. 2017. 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)*. ACM, 590–603. <https://doi.org/10.1145/3055399.3055438>
- [58] Andrei Krokhnin and Jakub Opršal. 2022. An invitation to the promise constraint satisfaction problem. *ACM SIGLOG News* 9, 3 (2022), 30–59. arXiv:2208.13538
- [59] Andrei A. Krokhnin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. 2022. Topology and adjunction in promise constraint satisfaction. *SIAM J. Comput.* (2022). arXiv:2003.11351
- [60] Jean B. Lasserre. 2002. An Explicit Equivalent Positive Semidefinite Program for Nonlinear 0-1 Programs. *SIAM J. Optim.* 12, 3 (2002), 756–769. <https://doi.org/10.1137/S1052623400380079>
- [61] Monique Laurent. 2003. A Comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre Relaxations for 0-1 Programming. *Math. Oper. Res.* 28, 3 (2003), 470–496. <https://doi.org/10.1287/moor.28.3.470.16391>
- [62] James R. Lee, Prasad Raghavendra, and David Steurer. 2015. Lower Bounds on the Size of Semidefinite Programming Relaxations. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC'15)*. ACM, 567–576. <https://doi.org/10.1145/2746539.2746599>
- [63] Tamio-Vesa Nakajima and Stanislav Živný. 2022. Linearly Ordered Colourings of Hypergraphs. In *Proc. 49th International Colloquium on Automata, Languages, and Programming (ICALP'22) (LIPIcs, Vol. 229)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 128:1–128:18. <https://doi.org/10.4230/LIPIcs.ICALP.2022.128> arXiv:2204.05628
- [64] Pablo A Parrilo. 2000. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology. <http://www.cds.caltech.edu/~doyle/hot/thesis.pdf>
- [65] Thomas Schaefer. 1978. The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on the Theory of Computing (STOC'78)*. 216–226. <https://doi.org/10.1145/800133.804350>
- [66] Alexander Schrijver. 1986. *Theory of linear and integer programming*. John Wiley & Sons, Ltd., Chichester. xii+471 pages. A Wiley-Interscience Publication.
- [67] H. D. Sherali and W. P. Adams. 1990. A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM J. Discret. Math.* 3, 3 (1990), 411–430. <https://doi.org/10.1137/0403036>
- [68] Naum Z Shor. 1987. Class of global minimum bounds of polynomial functions. *Cybernetics* 23, 6 (1987), 731–734. <https://doi.org/10.1007/BF01070233>
- [69] Madhur Tulsiani. 2009. CSP gaps and reductions in the Lasserre hierarchy. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC'09)*. ACM, 303–312. <https://doi.org/10.1145/1536414.1536457>
- [70] Avi Wigderson. 1983. Improving the Performance Guarantee for Approximate Graph Coloring. *J. ACM* 30, 4 (1983), 729–735. <https://doi.org/10.1145/2157.2158>
- [71] Dmitriy Zhuk. 2020. A Proof of the CSP Dichotomy Conjecture. *J. ACM* 67, 5 (2020), 30:1–30:78. <https://doi.org/10.1145/3402029> arXiv:1704.01914

Received 2022-11-07; accepted 2023-02-06