Semidefinite Programming and Linear Equations vs. Homomorphism Problems

Lorenzo Ciardo
University of Oxford
Oxford, United Kingdom
lorenzo.ciardo@cs.ox.ac.uk

Stanislav Živný
University of Oxford
Oxford, United Kingdom
standa.zivny@cs.ox.ac.uk

ABSTRACT

We introduce a relaxation for homomorphism problems that combines semidefinite programming with linear Diophantine equations, and propose a framework for the analysis of its power based on the spectral theory of association schemes. We use this framework to establish an unconditional lower bound against the semidefinite programming + linear equations model, by showing that the relaxation does not solve the approximate graph homomorphism problem and thus, in particular, the approximate graph colouring problem.

CCS CONCEPTS

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

KEYWORDS

approximate graph colouring, approximate graph homomorphism, semidefinite programming relaxations, affine integer programming relaxations, Diophantine equations, promise constraint satisfaction

ACM Reference Format:
Lorenzo Ciardo and Stanislav Živný. 2024. Semidefinite Programming and Linear Equations vs. Homomorphism Problems. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3618260.3649635

1 INTRODUCTION

Semidefinite programming plays a central role in the design of efficient algorithms and in dealing with NP-hardness. For many fundamental problems, the best known (and sometimes provably best possible) approximation algorithms are achieved via relaxations based on semidefinite programs [3, 48, 59, 62, 78]. In this work, we focus on computational problems of the following general form: Given two structures (say, two digraphs) $\mathbf{X}$ and $\mathbf{A}$, is there a homomorphism from $\mathbf{X}$ to $\mathbf{A}$? A plethora of different computational problems – in particular, those involving satisfiability of constraints – can be cast in this form. The semidefinite programming paradigm is naturally applicable to this type of problems, and it yields relaxations that are *robust to noise*: They are able to find a near-satisfying assignment even when the instance is almost – but not perfectly – satisfiable [8] (see also [17]). On the other hand, certain homomorphism problems can be solved exactly in polynomial time but are inherently *fragile to noise* – the primary example being systems of linear equations, which are tractable via Gaussian elimination but whose noisy version is NP-hard [52]. Problems that behave like linear equations are hopelessly stubborn against the semidefinite programming model [23, 79, 84]. It is then natural, in the context of homomorphism problems, to consider stronger versions of semidefinite programming relaxations that are equipped with a built-in linear-equation solver.

Consider a homomorphism[1] $f : \mathbf{X} \to \mathbf{A}$. Letting $|V(\mathbf{X})| = p$, $|V(\mathbf{A})| = n$, we can encode f in a $pn \times pn$ matrix M_f containing blocks of size $n \times n$, where the blocks are indexed by pairs of vertices of $\mathbf{X}$, and the entries in a block by pairs of vertices of $\mathbf{A}$. For $x, y \in V(\mathbf{X})$ and $a, b \in V(\mathbf{A})$, the (a, b)-th entry of the (x, y)-th block is 1 if $a = f(x)$ and $b = f(y)$, and 0 otherwise. Let us explore the structure of M_f. Each block has nonnegative entries summing up to 1, and diagonal blocks are diagonal matrices. Since f is a homomorphism, the (a, b)-th entry of the (x, y)-th block is 0 when $(x, y) \in E(\mathbf{X})$ and $(a, b) \notin E(\mathbf{A})$. Finally, M_f is positive semidefinite since it is symmetric and, for a pn-vector $\mathbf{v}$, it satisfies $\mathbf{v}^T M_f \mathbf{v} = (\sum_x v_{x,f(x)})^2 \geq 0$. The *standard semidefinite programming relaxation* (SDP) of the homomorphism problem "$\mathbf{X} \to \mathbf{A}$?" consists in looking for a real matrix M with the properties described above. We write $\mathrm{SDP}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$ if such a matrix M exists.

Any *Constraint Satisfaction Problem* (CSP) may be expressed as the homomorphism problem of checking whether an *instance* structure $\mathbf{X}$ homomorphically maps to a *template* structure $\mathbf{A}$.[2] The power of semidefinite programming in the realm of CSPs is well understood:[3] The CSPs solved by SDP are exactly those having *bounded width* [8, 39]. Crucially, for CSPs, boosting SDP via the so called *lift-and-project technique* [71] does not increase its power: Any semidefinite programming relaxation of polynomial size – in particular, any constant number of rounds of the *Lasserre "Sum-of-Squares" hierarchy* [70] – solves precisely the same CSPs as SDP [8, 83]. The positive resolution of Feder–Vardi CSP Dichotomy Conjecture [39] by Bulatov [22] and Zhuk [85] implies that any tractable CSP is a certain (nontrivial) combination of (i) bounded-width CSPs and (ii) CSPs that can simulate linear equations (which

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649635

[1]Letting $V(\mathbf{X})$ and $E(\mathbf{X})$ denote the vertex and edge sets of $\mathbf{X}$, a map $f : V(\mathbf{X}) \to V(\mathbf{A})$ is a homomorphism if $(f(x), f(y)) \in E(\mathbf{A})$ whenever $(x, y) \in E(\mathbf{X})$. The expression "$\mathbf{X} \to \mathbf{A}$" denotes the existence of a homomorphism.

[2]Up to polynomial-time equivalence, $\mathbf{X}$ and $\mathbf{A}$ can be assumed to be digraphs without loss of generality [39]. As was shown in [15], a similar fact holds for the promise version of CSP, which we shall encounter in a while.

[3]In the CSP literature [8, 17, 29, 78, 83], the SDP relaxation is usually described in terms of real vectors meeting orthogonality constraints; see the full version [30] for the vector formulation of SDP.

have unbounded width). The aim to find a *universal solver* for all tractable CSPs has then driven a new generation of algorithms that combine (i) techniques suitable for exploiting bounded width with (ii) variants of Gaussian elimination (which solves linear equations). This line of work was pioneered by [14, 18], with the description of the algorithm BA mixing a linear-programming-based relaxation with Gaussian elimination. Variants of this algorithm were later considered in [28, 31, 33].

The algorithm we propose in this work (which we call SDA) can be described as follows. First, notice that the matrix M_f encoding a homomorphism $f : \mathbf{X} \to \mathbf{A}$ has entries in $\{0, 1\}$, and all of the properties of M_f highlighted above are in fact linear equations, with the exception of the nonnegativity of its entries and the positive semidefiniteness. Hence, a different relaxation can be obtained by looking for a matrix M' that respects the linear conditions, and whose entries are integers. We end up with a linear Diophantine system, that can be solved efficiently through integer variants of Gaussian elimination, see [80]. We write $\mathrm{SDA}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$ if both M and M' exist.[4]

The first main goal of our work is to *introduce a technique based on the spectral theory of association schemes for the analysis of this relaxation model*. Our approach aims to describe how the algorithm exploits the symmetry of the problem under relaxation. To that end, we gradually refine and abstract the way symmetry is expressed. Starting from *automorphisms*, which capture symmetry of $\mathbf{X}$ and $\mathbf{A}$, we lift the analysis to the *orbitals* of $\mathbf{X}$ and $\mathbf{A}$ under the action of the automorphism groups and, finally, we endow the orbitals with the algebraic structure of *association schemes*. The progressively more abstract language for expressing the symmetry of the problem yields a progressively cleaner description of the impact of symmetry on the relaxation. For the SDP part of SDA, the abstraction process "automorphisms $\to$ orbitals $\to$ association schemes" may be viewed in purely linear-algebraic terms, as the *quest for a convenient (i.e., low-dimensional) vector space where the output of the algorithm lives, and a suitable basis for this space*. The last stage of this metamorphosis of symmetry discloses a new algebraic perspective on the relaxation. In particular, for certain classes of digraphs, association schemes allow turning SDP into a linear program.[5] The non-convex nature of Diophantine equations makes the linear part of SDA process the symmetry of the inputs in a quite different way. We exploit the dihedral structure of the automorphism group of cycles to show that each associate in their scheme can be assigned an integral matrix with a small support; this, in turn, can be used to produce a solution M' to the linear system.

This approach allows for a direct transfer of the results available in algebraic combinatorics on association schemes to the study of relaxations of homomorphism problems. For example, the explicit expression for the *character table* of a specific scheme known as the *Johnson scheme* shall be crucial for establishing a lower bound against the SDA model. One peculiarity of this framework is that *it is not forgetful of the structure of the instance* $\mathbf{X}$. This contrasts with the techniques for describing relaxations of CSPs [8, 32, 67, 82] based on the *polymorphic approach* [21, 56, 57], whose gist is that the complexity of a CSP depends on the identities satisfied by the *polymorphisms* of the CSP template $\mathbf{A}$ [9]. The polymorphic approach yields elegant characterisations of the power of some relaxations, in the sense that a CSP is solved by a certain algorithm if and only if its polymorphisms satisfy identities typical of the algorithm.[6] These "instance-free" characterisations rely on knowing both the identities typical of the algorithm (not available in the case of SDP and, thus, SDA) and a succinct description of the polymorphisms of the template (which is missing in the case of the approximate homomorphism problems we shall see next). In contrast, the description based on association schemes does take the structure of the instance into account, which results in a higher control over the behaviour of the algorithm on certain highly symmetric instances.

The second main goal of our work is to apply the association-scheme framework to *obtain an unconditional lower bound against SDA* (and, a fortiori, against SDP). We consider the *Approximate Graph Homomorphism* problem (AGH): Given two (undirected) graphs $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{A} \to \mathbf{B}$ and an instance $\mathbf{X}$, distinguish between the cases (i) $\mathbf{X} \to \mathbf{A}$ and (ii) $\mathbf{X} \not\to \mathbf{B}$.[7] This problem is commonly studied in the context of *Promise CSPs* [5, 7, 15, 69], and we shall thus denote it by $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$. If we let $\mathbf{A} = \mathbf{K}_n$ (the n-clique) and $\mathbf{B} = \mathbf{K}_{n'}$ where $n \leq n'$, AGH specialises to the *Approximate Graph Colouring* problem (AGC): Distinguish whether a given graph is n-colourable or not even n'-colourable. The computational complexity of these problems is a long-standing open question.[8] In 1976, Garey and Johnson conjectured that AGC is always NP-hard if $3 \leq n$:[9]

Conjecture ([41]). *Let* $3 \leq n \leq n'$ *be integers. Then* $\mathrm{PCSP}(\mathbf{K}_n, \mathbf{K}_{n'})$ *is NP-hard.*

More recently, Brakensiek and Guruswami proposed the stronger conjecture that even AGH may always be NP-hard except in trivial cases:[10]

Conjecture ([15]). *Let* $\mathbf{A}, \mathbf{B}$ *be non-bipartite loopless undirected graphs such that* $\mathbf{A} \to \mathbf{B}$. *Then* $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ *is NP-hard.*

Among the several papers making progress on the two conjectures above, we mention [7, 13, 16, 19, 38, 50, 55, 63, 64, 69]. However, they both remain wide open in their full generality. Given the apparent "hardness of proving hardness" surrounding these problems, significant efforts have been directed towards showing inapplicability of specific algorithmic models, following an established line of work on lower bounds against relaxations, e.g., [2, 10, 23, 24, 43, 68, 72, 84]. Non-solvability of AGC via sublinear levels of local consistency

[4] We also require that a technical refinement condition constraining the supports of M and M' should hold; see the full version [30] for the formal definition of the algorithm. The "A" in SDA stands for "affine" integer programming, the name by which the CSP relaxation based on linear Diophantine equations is sometimes referred to in the literature [7].

[5] On a high level, this is an instance of a general *invariant-theoretic* phenomenon: The presence of a rich group of symmetries makes it possible to reduce the size of semidefinite programs [34, 42, 58] and, in certain cases, to describe their feasible regions in terms of linear inequalities [35, 47], see also [36, 81].

[6] As established in [7], a similar approach also works for the promise version of CSP that we shall discuss shortly.

[7] This is the *decision* version of the problem. In the *search* version, the goal is to find an explicit homomorphism $\mathbf{X} \to \mathbf{B}$ assuming that $\mathbf{X} \to \mathbf{A}$. The former version reduces to the latter, so our non-solvability result applies to both.

[8] In contrast, the complexity of the non-approximate versions of AGC and AGH (i.e., the cases $n = n'$ and $\mathbf{A} = \mathbf{B}$, respectively) was already classified by Karp [61] and Hell–Nešetřil [53], respectively.

[9] The case $n = 2$ reduces to 2-colouring and is thus tractable.

[10] If either $\mathbf{A}$ or $\mathbf{B}$ has a loop or is bipartite, the problem is trivial or reduces to 2-colouring.

and via linear Diophantine equations was proved in [4] and [26], respectively. It was shown in [59] that the technique of vector colouring, based on a semidefinite program akin to Lovász's orthonormal representation [74], is inapplicable to solving AGC. It follows from [51, 66] that polynomial levels of the Sum-of-Squares hierarchy (and, in particular, SDP) are also not powerful enough to solve AGC. Very recently, [27] improved on the result in [26] by proving non-solvability of AGC via constant levels of the BA hierarchy, obtained by applying the *lift-and-project* technique to the BA relaxation of [18]. By leveraging the framework of association schemes, we establish that AGH is not solved by SDA:[11]

Theorem 1. *Let* $\mathbf{A}, \mathbf{B}$ *be non-bipartite loopless undirected graphs such that* $\mathbf{A} \to \mathbf{B}$*. Then* SDA *does not solve* $\text{PCSP}(\mathbf{A}, \mathbf{B})$.

The improvement on the state of the art is twofold: Theorem 1 yields (i) the first non-solvability result for the whole class of problems AGH, as opposed to the subclass AGC, and (ii) the first lower bound against the combined "SDP + linear equations" model (which is strictly stronger than both models individually).[12] Via Raghavendra's framework [78], the (SDP part of the) integrality gap in Theorem 1 directly yields a conditional hardness-of-approximation result for AGH: Assuming Khot's UGC [65] and P ≠ NP, AGH is not solved by any polynomial-time *robust* algorithm.

Related work on association schemes. The Johnson scheme and other association schemes such as the *Hamming scheme* have appeared in the analysis of the performance ratio of the Goemans–Williamson Max-Cut algorithm [48] based on semidefinite programming, see [1, 47, 60]. In [75], certain spectral properties of the Johnson scheme were used to obtain lower bounds against the Positivestellensatz proof system (and, thus, against the Sum-of-Squares hierarchy) applied to the planted clique problem, see also [37].

Notation. We let $\mathbb{N}$ be the set of positive integers, while $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For $t \in \mathbb{N}$, we let $[t] = \{1, \dots, t\}$. We view vectors in $\mathbb{R}^t$ as column vectors, but sometimes write them as tuples for typographical convenience. We denote by I_t and J_t the $t \times t$ identity and all-one matrices, by $O_{t,t'}$ the $t \times t'$ all-zero matrix, and by $\mathbf{1}_t$ and $\mathbf{0}_t$ the all-one and all-zero vectors of length t. (Indices shall sometimes be omitted when clear from the context.) We denote by $\mathbf{e}_i$ the i-th standard unit vector of length t (which shall be clear from the context); i.e., the vector in $\mathbb{R}^t$ all of whose entries are 0 except the i-th entry that is 1. Given a field $\mathbb{F}$ and a set $\mathscr{V}$ of vectors in $\mathbb{F}^t$, $\text{span}_{\mathbb{F}}(\mathscr{V})$ is the set of linear combinations over $\mathbb{F}$ of the vectors in $\mathscr{V}$. We write $\text{span}(\mathscr{V})$ for $\text{span}_{\mathbb{R}}(\mathscr{V})$.

A matrix is *Boolean* if its entries are in $\{0, 1\}$. Given a real matrix M, we write $M \geq 0$ if M is *entrywise nonnegative*, and we write $M \succcurlyeq 0$ if M is *positive semidefinite* (i.e., if M is symmetric and has a nonnegative spectrum). For two matrices M and M', we let $M \otimes M'$ denote their *Kronecker product*; if M and M' have equal size, we let $M \circ M'$ denote their *Schur product* (i.e., their entrywise product, also known as *Hadamard product*). We shall often use the fact that $(M \otimes M')(N \otimes N') = MN \otimes M'N'$, provided that the products are well defined (see [54]). The *support* of M, denoted by $\text{supp}(M)$, is the set of indices of nonzero entries of M; for two matrices M, M' of equal size, we write $M \triangleleft M'$ for $\text{supp}(M) \subseteq \text{supp}(M')$. Given a digraph $\mathbf{X}$, we let $\mathscr{A}(\mathbf{X})$ and $\text{Aut}(\mathbf{X})$ denote the *adjacency matrix* and the *automorphism group* of $\mathbf{X}$, respectively. We view undirected graphs as digraphs, by turning each undirected edge $\{x, y\}$ into a pair of directed edges (x, y) and (y, x). For $n \geq 3$, $\mathbf{C}_n$ denotes the *undirected cycle* on n vertices.

Given two digraphs $\mathbf{A}, \mathbf{B}$ such that $\mathbf{A} \to \mathbf{B}$, we say that SDP (resp., SDA) *solves* $\text{PCSP}(\mathbf{A}, \mathbf{B})$ if, for any digraph $\mathbf{X}$, $\text{SDP}(\mathbf{X}, \mathbf{A}) = \text{Yes}$ (resp., $\text{SDA}(\mathbf{X}, \mathbf{A}) = \text{Yes}$) implies $\mathbf{X} \to \mathbf{B}$. It follows from the definitions of the algorithms that $\mathbf{X} \to \mathbf{A}$ always implies $\text{SDP}(\mathbf{X}, \mathbf{A}) = \text{SDA}(\mathbf{X}, \mathbf{A}) = \text{Yes}$.

2 OVERVIEW OF RESULTS AND TECHNIQUES

In this section, we give an overview of our results and techniques. All details and proofs can be found in the full version of this paper [30].

First of all, it will be convenient to encode the output of the algorithms into a matrix. Henceforth, we shall label the vertex sets of $\mathbf{X}$ and $\mathbf{A}$ as $V(\mathbf{X}) = [p]$ and $V(\mathbf{A}) = [n]$. We say that a real $pn \times pn$ matrix M is a *relaxation matrix* for $\mathbf{X}, \mathbf{A}$ if M satisfies the following requirements:[13]

- (r_1) $(\mathbf{e}_x \otimes I_n)^T M (\mathbf{e}_x \otimes I_n)$ is a diagonal matrix for each $x \in V(\mathbf{X})$;
- (r_2) $(\mathbf{e}_x \otimes \mathbf{e}_a)^T M (\mathbf{e}_y \otimes \mathbf{e}_b) = 0$ whenever $(x, y) \in E(\mathbf{X})$ and $(a, b) \in V(\mathbf{A})^2 \setminus E(\mathbf{A})$;
- (r_3) $M(\mathbf{e}_x \otimes \mathbf{1}_n) = M(\mathbf{e}_y \otimes \mathbf{1}_n)$ for each $(x, y) \in V(\mathbf{X})^2$;
- (r_4) $M^T(\mathbf{e}_x \otimes \mathbf{1}_n) = M^T(\mathbf{e}_y \otimes \mathbf{1}_n)$ for each $(x, y) \in V(\mathbf{X})^2$;
- (r_5) $\mathbf{1}_{pn}^T M \mathbf{1}_{pn} = p^2$.

Given a relaxation matrix M, we say that M is an SDP-*matrix* for $\mathbf{X}, \mathbf{A}$ if $M \succcurlyeq 0$ and $M \geq 0$, and we say that M is an AIP-*matrix* for $\mathbf{X}, \mathbf{A}$ if all of its entries are integral.

2.1 Automorphisms

If ξ and α are automorphisms of $\mathbf{X}$ and $\mathbf{A}$, respectively, we may permute the rows and columns of a relaxation matrix M according to ξ and α and the result would still be a relaxation matrix. By averaging over all pairs of automorphisms (ξ, α), we end up with a relaxation matrix that is invariant under automorphisms of $\mathbf{X}$ and $\mathbf{A}$. More formally, given two finite sets R and S and a function $f : R \to S$, we let Q_f be the $|R| \times |S|$ matrix whose (r, s)-th entry is 1 if $f(r) = s$, 0 otherwise. We say that a real $pn \times pn$ matrix M is *balanced* for $\mathbf{X}, \mathbf{A}$ if

$$(Q_\xi \otimes Q_\alpha) M (Q_\xi^T \otimes Q_\alpha^T) = M \quad \text{for each } \xi \in \text{Aut}(\mathbf{X}),\ \alpha \in \text{Aut}(\mathbf{A}). \tag{1}$$

The set of positive semidefinite, entrywise-nonnegative matrices is closed under simultaneous permutations of rows and columns and under convex combinations. It follows that we may always

[11] As we have seen, our framework requires a certain amount of symmetry in the inputs when applied to a specific problem. We manage to avoid a loss of generality by reducing to the general case of AGH from a more symmetric case.

[12] Very recently — and independently of our work — Chan, Ng, and Peng described a hierarchy based on SDP + linear equations for CSPs, and proved that linear levels do not solve random instances of a certain natural class of CSPs including k-SAT [25].

[13] Viewing M as a block matrix whose $n \times n$ blocks are indexed by pairs in $V(\mathbf{X})^2$, (r_1) states that diagonal blocks are diagonal matrices, (r_2) states that the supports of blocks corresponding to edges of $\mathbf{X}$ are included in $E(\mathbf{A})$, (r_3) and (r_4) state that the row-sum (resp. column-sum) vectors of blocks aligned horizontally (resp. vertically) are equal, and (r_5) is a normalisation condition. (Cf. the properties of the matrix M_f described in the Introduction.)

assume that SDP-matrices are balanced.[14] This yields the following characterisation.

Proposition 2. *Let* $\mathbf{X}, \mathbf{A}$ *be digraphs. Then*

(i) $\mathrm{SDP}(\mathbf{X}, \mathbf{A}) = \text{YES}$ *if and only if there exists a balanced* SDP-*matrix for* $\mathbf{X}, \mathbf{A}$*;*

(ii) *if* $\mathbf{X}$ *is loopless,* $\mathrm{SDA}(\mathbf{X}, \mathbf{A}) = \text{YES}$ *if and only if there exist a balanced* SDP-*matrix* M *and an* AIP-*matrix* N *for* $\mathbf{X}, \mathbf{A}$ *such that* $N \circ ((I_p + \mathscr{A}(\mathbf{X})) \otimes J_n) \quad \triangleleft \quad M$.

It follows that, instead of studying the outputs of SDP in $\mathbb{R}^{pn \times pn}$ with the basis of standard unit matrices $\mathbf{e}_i \mathbf{e}_j^T$ – as we have implicitly done so far – we may work without loss of generality in the real vector space $\mathscr{L}$ of balanced matrices for $\mathbf{X}, \mathbf{A}$.[15] As we see next, the concept of orbitals provides a natural basis for the space $\mathscr{L}$.

2.2 Orbitals

For a digraph $\mathbf{X}$, consider the action of the group $\mathrm{Aut}(\mathbf{X})$ onto the set $V(\mathbf{X})^2$ given by $(x, y)^\xi = (\xi(x), \xi(y))$ for $\xi \in \mathrm{Aut}(\mathbf{X})$, $x, y \in V(\mathbf{X})$. An *orbital* of $\mathbf{X}$ is an orbit of $V(\mathbf{X})^2$ with respect to this action; i.e., it is a minimal subset of $V(\mathbf{X})^2$ that is invariant under the action. We let $\mathscr{O}(\mathbf{X})$ be the set of orbitals of $\mathbf{X}$. Given an orbital $\omega \in \mathscr{O}(\mathbf{X})$, we let R_ω be the $p \times p$ matrix whose (x, y)-th entry is 1 if $(x, y) \in \omega$ and 0 otherwise. Orbitals provide an alternative description of balanced matrices: A block matrix M is balanced for $\mathbf{X}, \mathbf{A}$ if and only if the block structure of M is constant over the orbitals of $\mathbf{X}$, and each block is constant over the orbitals of $\mathbf{A}$. As stated next, it follows that we can find a basis for $\mathscr{L}$ by taking Kronecker products of the matrices R_ω.

Proposition 3. *Let* $\mathbf{X}, \mathbf{A}$ *be digraphs, and let* $\mathscr{L}$ *be the real vector space of balanced matrices for* $\mathbf{X}, \mathbf{A}$. *Then the set* $\mathscr{R} = \{R_\omega \otimes R_{\tilde{\omega}} : \omega \in \mathscr{O}(\mathbf{X}), \tilde{\omega} \in \mathscr{O}(\mathbf{A})\}$ *forms a basis for* $\mathscr{L}$.

As a consequence, given a balanced matrix M, there is a unique list of coefficients $v_{\omega\tilde{\omega}}$ such that

$$M = \sum_{\omega \in \mathscr{O}(\mathbf{X}),\ \tilde{\omega} \in \mathscr{O}(\mathbf{A})} v_{\omega\tilde{\omega}} \, R_\omega \otimes R_{\tilde{\omega}}. \tag{2}$$

We shall refer to the $|\mathscr{O}(\mathbf{X})| \times |\mathscr{O}(\mathbf{A})|$ matrix $V = (v_{\omega\tilde{\omega}})$ as the *orbital matrix* of M. Expressing a balanced matrix M in the new basis $\mathscr{R}$ rather than in the standard basis for $\mathbb{R}^{pn \times pn}$ is especially convenient when $\mathbf{X}$ and $\mathbf{A}$ are highly symmetric. Indeed, if $\mathrm{Aut}(\mathbf{X})$ and $\mathrm{Aut}(\mathbf{A})$ are large, $\mathscr{O}(\mathbf{X})$ and $\mathscr{O}(\mathbf{A})$ are small. Working with $\mathscr{R}$ allows then compressing the information of the $pn \times pn$ matrix M in the smaller $|\mathscr{O}(\mathbf{X})| \times |\mathscr{O}(\mathbf{A})|$ orbital matrix V. However, if we want to make use of V to certify acceptance of SDP, we need to be able to check if M is an SDP-matrix by only looking at V. While lifting the requirements defining an SDP-matrix to the orbital matrix, it should come with little surprise that the crucial one is positive semidefiniteness: How to translate the fact that $M \succcurlyeq 0$ into a condition on V? We shall see that the key for recovering the spectral properties of M from the orbital matrix is to endow the set of orbitals with a certain algebraic structure.

[14] Observe that the same is not true for AIP-matrices, as integral matrices are not closed under convex combinations.

[15] The fact that $\mathscr{L}$ is a real vector space easily follows from (1).

2.3 Association Schemes

An *association scheme*[16] is a set $\mathscr{S} = \{S_0, S_1, \dots, S_d\}$ of $p \times p$ Boolean matrices satisfying

(s_1) $S_0 = I_p$;
(s_2) $\sum_{i=0}^d S_i = J_p$;
(s_3) $S_i^T \in \mathscr{S} \quad \forall i$;
(s_4) $S_i S_j \in \mathrm{span}_{\mathbb{C}}(\mathscr{S}) \quad \forall i, j$;
(s_5) $S_i S_j = S_j S_i \quad \forall i, j$.

Observe that, if all S_i are permutation matrices, $\mathscr{S}$ is a finite group. Indeed, using association schemes one can develop a theory of symmetry that generalises character theory for group representations. The *Bose–Mesner algebra* $\mathfrak{B}$ of $\mathscr{S}$ is the vector space $\mathrm{span}_{\mathbb{C}}(\mathscr{S})$, which consists of all complex linear combinations of the matrices in $\mathscr{S}$ (see [12]). Since the matrices in $\mathscr{S}$ are Boolean and satisfy (s_2), they form a basis for $\mathfrak{B}$. Notice also that the set $\mathscr{S} \cup \{O_{p,p}\}$ is closed under the Schur product, and so is $\mathfrak{B}$. Moreover, the matrices in $\mathscr{S}$ are Schur-orthogonal and Schur-idempotent, in that $S_i \circ S_j$ equals S_i when $i = j$, and equals $O_{p,p}$ otherwise. Hence, we have the following.

Fact 4. *Let* $\mathscr{S}$ *be an association scheme. Then* $\mathscr{S}$ *forms a Schur-orthogonal basis of Schur-idempotents for its Bose–Mesner algebra* $\mathfrak{B}$.

Now, by (s_4), $\mathfrak{B}$ is also closed under the standard matrix product; in other words, it is a matrix algebra, thus justifying the name. It turns out that a different basis exists for $\mathfrak{B}$, whose members enjoy similar properties to those for the basis $\mathscr{S}$, but with a different product being involved.

Fact 5. *Let* $\mathscr{S}$ *be an association scheme. Then there exists an orthogonal basis* $\mathscr{E} = \{E_0, E_1, \dots, E_d\}$ *of Hermitian idempotents*[17] *for its Bose–Mesner algebra* $\mathfrak{B}$.

The interaction between the two bases $\mathscr{S}$ and $\mathscr{E}$ allows deriving several interesting features of association schemes. The change-of-basis matrix shall be particularly important for our purposes. More precisely, we can (uniquely) express the elements of $\mathscr{S}$ as

$$S_j = \sum_{i=0}^d p_{ij} E_i \tag{3}$$

for some numbers p_{ij}. The $(d+1) \times (d+1)$ matrix $P = (p_{ij})$ is known as the *character table* of the association scheme [6]. It turns out that the sets of orbitals for a certain class of digraphs form association schemes. We say that a digraph $\mathbf{X}$ is *generously transitive* if for any $x, y \in V(\mathbf{X})$ there exists $\xi \in \mathrm{Aut}(\mathbf{X})$ such that $\xi(x) = y$ and $\xi(y) = x$.

Theorem 6 ([6]). *Let* $\mathbf{X}$ *be a generously transitive digraph. Then the set* $\{R_\omega : \omega \in \mathscr{O}(\mathbf{X})\}$ *is a symmetric*[18] *association scheme.*

We shall refer to the character table of the association scheme $\{R_\omega : \omega \in \mathscr{O}(\mathbf{X})\}$ as the character table of $\mathbf{X}$. Note that this is a $|\mathscr{O}(\mathbf{X})| \times |\mathscr{O}(\mathbf{X})|$ matrix. Recall that our current objective is to

[16] The results on association schemes presented in this section can be found in any of the references [6, 20, 44, 45].

[17] I.e., the matrices in $\mathscr{E}$ are Hermitian, and $E_i E_j$ equals E_i when $i = j$, and equals $O_{p,p}$ otherwise.

[18] An association scheme $\mathscr{S}$ is *symmetric* if $\mathscr{S}$ consists of symmetric matrices.

decipher the spectral properties of a balanced matrix M from the corresponding orbital matrix V. The next result shows that the character table is precisely the dictionary we need to make the translation.

Theorem 7. *Let* $\mathbf{X}$ *and* $\mathbf{A}$ *be generously transitive digraphs, let* M *be a balanced matrix for* $\mathbf{X}, \mathbf{A}$*, let* V *be the orbital matrix of* M*, and let* P *and* $\tilde{P}$ *be the character tables of* $\mathbf{X}$ *and* $\mathbf{A}$*, respectively. Then the spectrum of* M *consists of the entries of the matrix* $PV\tilde{P}^T$.

Theorem 7 is proved by expressing a balanced matrix in a new, *third* basis given by the Kronecker product of the bases for $\mathscr{O}(\mathbf{X})$ and $\mathscr{O}(\mathbf{A})$ coming from Fact 5. One consequence of this result is that the semidefinite program applied to two generously transitive digraphs $\mathbf{X}$ and $\mathbf{A}$ can be turned into a *linear* program, whose constraints are in terms of the character tables of $\mathbf{X}$ and $\mathbf{A}$. This is made explicit in the next corollary. For a digraph $\mathbf{X}$, we let $\boldsymbol{\mu}^{\mathbf{X}}$ be the vector, indexed by the elements of $\mathscr{O}(\mathbf{X})$, whose ω-th entry is $|\omega|$. We say that an orbital ω is the *diagonal orbital* if R_ω is the identity matrix, and we say that ω is an *edge orbital* if $\omega \subseteq E(\mathbf{X})$; *non-diagonal* and *non-edge* orbitals are defined in the obvious way. Notice that the edge orbitals of $\mathbf{X}$ partition $E(\mathbf{X})$.

Corollary 8. *Let* $\mathbf{X}$ *and* $\mathbf{A}$ *be generously transitive digraphs. Furthermore, let* P *and* $\tilde{P}$ *be the character tables of* $\mathbf{X}$ *and* $\mathbf{A}$*, respectively. Then* $\mathrm{SDP}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$ *if and only if there exists a real entrywise-nonnegative* $|\mathscr{O}(\mathbf{X})| \times |\mathscr{O}(\mathbf{A})|$ *matrix* V *such that*

- (c_1) $PV\tilde{P}^T \geq 0$;
- (c_2) $V\boldsymbol{\mu}^{\mathbf{A}} = \mathbf{1}$;
- (c_3) $v_{\omega\tilde{\omega}} = 0$ *if* ω *is the diagonal orbital of* $\mathbf{X}$ *and* $\tilde{\omega}$ *is a non-diagonal orbital of* $\mathbf{A}$;
- (c_4) $v_{\omega\tilde{\omega}} = 0$ *if* ω *is an edge orbital of* $\mathbf{X}$ *and* $\tilde{\omega}$ *is a non-edge orbital of* $\mathbf{A}$.

In order to prove that SDA does not solve $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ for any pair of non-bipartite loopless undirected graphs such that $\mathbf{A} \to \mathbf{B}$, thus establishing Theorem 1, we seek a *fooling instance*: a digraph $\mathbf{X}$ such that $\mathrm{SDA}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$ but $\mathbf{X} \not\to \mathbf{B}$. If we wish to apply Corollary 8 and take advantage of the machinery developed so far for describing the output of SDP, we need both $\mathbf{X}$ and $\mathbf{A}$ to be generously transitive digraphs. Regarding $\mathbf{A}$, this requirement does not create problems. Indeed, it is not hard to check that it is enough to establish the result in the case that $\mathbf{A}$ is an odd undirected cycle and $\mathbf{B}$ is a clique. Since odd cycles happen to be generously transitive, Theorem 6 does apply.[19] The more challenging part is to come up with a digraph $\mathbf{X}$ that (i) is generously transitive, (ii) is not homomorphic to $\mathbf{B}$ (i.e., has high chromatic number), and (iii) is accepted by SDA. A promising candidate is the class of Kneser graphs, as they (i) are generously transitive and (ii) have unbounded chromatic number (that is easily derived from the parameters of the graphs through a classic result by Lovász [73]). Next, we look at the association schemes for Kneser graphs and odd cycles. The task is to collect the right amount of information on their character tables that will allow us to design an orbital matrix witnessing the fact that (iii) $\mathrm{SDA}(\mathbf{X}, \mathbf{A}) = \textsc{Yes}$.

[19] As we shall see, the structure of the scheme for odd cycles also allows dealing with the linear part of SDA.

The Johnson scheme. Given $s, t \in \mathbb{N}$ such that $s > 2t$, the *Kneser graph* $\mathbf{G}_{s,t}$ is the undirected graph whose vertices are all subsets of $[s]$ of size t, and whose edges are all disjoint pairs of such subsets. The automorphism group of $\mathbf{G}_{s,t}$ is isomorphic to the symmetric group Sym_s [46], as a consequence of the Erdős–Ko–Rado theorem [11, 45]. It is not hard to check that Kneser graphs are generously transitive. The association scheme corresponding to $\mathscr{O}(\mathbf{G}_{s,t})$ consists of the adjacency matrices of the *generalised Johnson graphs* $\mathbf{J}_{s,t,q}$ for $q = 0, \dots, t$, where $\mathbf{J}_{s,t,q}$ is the graph having the same vertex set as $\mathbf{G}_{s,t}$, with two vertices being adjacent if and only if their intersection has size $t - q$. This association scheme is known as the *Johnson scheme.*[20]

In order to design an orbital matrix witnessing that $\mathbf{G}_{s,t}$ is accepted by SDP (and, as we will see, SDA), it shall be useful to gain some insight into the behaviour of the character table of $\mathbf{G}_{s,t}$ when it is multiplied by column vectors (which, ultimately, will be the columns of the orbital matrix, cf. Corollary 8). We shall see that, if a column vector is interpolated by a polynomial of low degree, multiplying it by the character table yields a vector living in a fixed, low-dimensional subspace of $\mathbb{R}^{t+1}$. This observation leads us to choose an orbital matrix whose nonzero columns are polynomials of degree one, cf. the proof of Theorem 1. More precisely,[21] let $\mathbf{h}$ be the vector $(0, 1, \dots, t)$ and, given a univariate polynomial $f \in \mathbb{R}[x]$, let $\mathbf{h}^f$ be the vector $(f(0), f(1), \dots, f(t))$.

Theorem 9. *Let* $s, t \in \mathbb{N}$ *with* $s > 2t$*, and let* P *be the character table of* $\mathbf{G}_{s,t}$*. Then*

- (i) $P\mathbf{h}^f \in \mathrm{span}(\mathbf{e}_0, \dots, \mathbf{e}_d)$ *for any univariate polynomial* f *of degree* $d \leq t$;
- (ii) $P\mathbf{1} = \binom{s}{t}\mathbf{e}_0$;
- (iii) $P\mathbf{h} = \binom{s}{t}\left(\frac{st-t^2}{s}\mathbf{e}_0 - \frac{st-t^2}{s^2-s}\mathbf{e}_1\right)$.

To prove Theorem 9, we can take advantage of the explicit expression for the character table of the Johnson scheme obtained by Delsarte [35] (see also [45, § 6.5]) in terms of *Eberlein polynomials*.

Theorem 10 ([35])**.** *Let* $s, t \in \mathbb{N}$ *be such that* $s > 2t$*. Then the character table of the Kneser graph* $\mathbf{G}_{s,t}$ *is the* $(t+1) \times (t+1)$ *matrix whose* (j, q)*-th entry, for* $j, q \in \{0, \dots, t\}$*, is*[22]

$$\beta(s, t, q, j) = \sum_{i=0}^{\infty} (-1)^{i-q+j} \binom{i}{q} \binom{t-j}{i-j} \binom{s-i-j}{t-j}.$$

Our strategy consists in associating with the entries of the character table a family of bivariate generating functions (parameterised by t and j) defined by

$$\gamma_{t,j}(x, y) = \sum_{s,q \in \mathbb{N}_0} \beta(s, t, q, j) x^s y^q. \tag{4}$$

We find a closed formula for these generating functions.

[20] Note that $\mathbf{J}_{s,t,t} = \mathbf{G}_{s,t}$, while $\mathbf{J}_{s,t,0}$ consists of $\binom{s}{t}$ loops. Hence, the diagonal orbital corresponds to $q = 0$, while the (unique) edge orbital corresponds to $q = t$.

[21] As we have seen, the members of the Johnson scheme are naturally labelled by $0, 1, \dots, t$, where the q-th member is $\mathscr{A}(\mathbf{J}_{s,t,q})$. Hence, we label the entries of the character table of $\mathbf{G}_{s,t}$ and the standard unit vectors $\mathbf{e}_i$ in Theorem 9 accordingly, with indices ranging over $\{0, \dots, t\}$ rather than $\{1, \dots, t+1\}$. A similar labelling shall also be used for the cycle scheme, cf. Propositions 12 and 13.

[22] We use the conventions that $\binom{x}{y} = 0$ unless $0 \leq y \leq x$, and $\binom{0}{0} = 1$.

Proposition 11. *The identity*

$$\gamma_{t,j}(x,y) = x^{t+j}(1-x)^{j-t-1}(1-y)^j(1-x+xy)^{t-j}$$

holds for each $t, j \in \mathbb{N}_0$ and $x, y \in \mathbb{R}$ such that $j \le t$ and $-1 < x < 1$.

Theorem 9 is then proved by expressing the entries of the vector $P\mathbf{h}^f$ in terms of partial derivatives of the generating functions $\gamma_{t,j}$, and by finding analytic expressions for these partial derivatives through Proposition 11.

The cycle scheme. The automorphism group of the undirected cycle $\mathbf{C}_n$ is the dihedral group of order $2n$, and it is not hard to check that $\mathbf{C}_n$ is generously transitive. If $n = 2m+1$ is odd, $\mathscr{O}(\mathbf{C}_n)$ contains $m+1$ orbitals $\omega_0, \dots, \omega_m$, where ω_0 and ω_1 are the diagonal orbital and the (unique) edge orbital, respectively. Each orbital has size $2n$ except ω_0, which has size n. The Perron–Frobenius theorem for primitive matrices [40, 77] yields the next property of the corresponding character table.

Proposition 12. *Let $n \ge 3$ be an odd integer, and let P be the character table of $\mathbf{C}_n$. Then $P\mathbf{e}_0 = \mathbf{1}$, while $P\mathbf{e}_1$ contains exactly one entry equal to 2, and all other entries are strictly smaller than 2 in absolute value.*

The next proposition shall be crucial for taking care of the linear Diophantine part of the algorithm SDA when showing that Kneser graphs are fooling instances.

Proposition 13. *For any odd integer $n \ge 3$ there exists a function $f : \mathscr{O}(\mathbf{C}_n) \to \mathbb{Z}^{n\times n}$ such that $\operatorname{supp}(f(\omega)) \subseteq \omega$ and $f(\omega)\mathbf{1} = f(\omega)^T\mathbf{1} = \mathbf{e}_0$ for each $\omega \in \mathscr{O}(\mathbf{C}_n)$.*

2.4 A Lower Bound Against SDA

We now have all the ingredients for proving Theorem 1.

Proof of Theorem 1. Let $\mathbf{A}, \mathbf{B}$ be non-bipartite loopless undirected graphs such that $\mathbf{A} \to \mathbf{B}$, and notice that there exist $n, n' \ge 3$ with n odd such that $\mathbf{C}_n \to \mathbf{A}$ and $\mathbf{B} \to \mathbf{K}_{n'}$. (For example, we may choose n and n' as the odd girth of $\mathbf{A}$ and the chromatic number of $\mathbf{B}$, respectively.) Let $m = \frac{n-1}{2}$, and let $\tilde{P}$ be the character table of the association scheme corresponding to $\mathscr{O}(\mathbf{C}_n)$. By Proposition 12, there exists $0 < \delta < 2$ such that, up to a permutation of the rows, $\tilde{P}\mathbf{e}_0 = \mathbf{1}$ and $\tilde{P}\mathbf{e}_1 = \begin{bmatrix} 2 \\ \mathbf{z} \end{bmatrix}$ for some vector $\mathbf{z} \in \mathbb{R}^m$ all of whose entries have absolute value strictly smaller than δ. Without loss of generality, we can assume that δ is rational. Let $t \in \mathbb{N}$ be such that $t \ge \frac{2n'}{2-\delta}$ and $\frac{t}{\delta} \in \mathbb{N}$, and let $s = \frac{2t}{\delta} + t$. Observe that $s > 2t$. We claim that $\mathrm{SDA}(\mathbf{G}_{s,t}, \mathbf{C}_n) = \textsc{Yes}$. Since SDA is monotone with respect to the homomorphism preorder of the arguments, this would imply that $\mathrm{SDA}(\mathbf{G}_{s,t}, \mathbf{A}) = \textsc{Yes}$. However, using Lovász's formula for the chromatic number of Kneser graphs [73], we find

$$\chi(\mathbf{G}_{s,t}) = s - 2t + 2 = \frac{2t}{\delta} + t - 2t + 2 = \frac{t(2-\delta)}{\delta} + 2 \ge \frac{2n'}{\delta} + 2 > n' + 2.$$

This means that $\mathbf{G}_{s,t} \not\to \mathbf{K}_{n'}$ and, hence, $\mathbf{G}_{s,t} \not\to \mathbf{B}$. As a consequence, the truth of the claim would establish that SDA does not solve $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$, thus concluding the proof of the theorem.

Let P be the character table of $\mathbf{G}_{s,t}$, and recall that $\mathbf{h}$ denotes the vector $(0, 1, \dots, t)$ (which, as usual, we view as a column vector). Consider the matrices

$$W = \begin{bmatrix} \mathbf{1} - \frac{1}{t}\mathbf{h} & \frac{1}{t}\mathbf{h} & O_{t+1,m-1} \end{bmatrix} \in \mathbb{R}^{(t+1)\times(m+1)},$$

and

$$K = \frac{1}{2n}\operatorname{diag}(2, 1, \dots, 1) \in \mathbb{R}^{(m+1)\times(m+1)},$$

and $V = WK$. We now show that V meets the conditions in Corollary 8 and, thus, it is the orbital matrix for a balanced SDP-matrix. Recall that the diagonal orbitals of $\mathbf{G}_{s,t}$ and $\mathbf{C}_n$ are those having index 0, while the (unique) edge orbitals of $\mathbf{G}_{s,t}$ and $\mathbf{C}_n$ are those having index t and 1, respectively. Since $v_{i,j} = 0$ whenever $i = 0, j \neq 0$ or $i = t, j \neq 1$, the conditions (c_3) and (c_4) are satisfied. Observe that $\boldsymbol{\mu}^{\mathbf{C}_n}$ is the vector $2n\mathbf{1} - n\mathbf{e}_0$. Therefore, $V\boldsymbol{\mu}^{\mathbf{C}_n} = WK\boldsymbol{\mu}^{\mathbf{C}_n} = W\mathbf{1} = \mathbf{1}$, so (c_2) holds, too. Theorem 9 yields

$$\begin{aligned} PW &= \begin{bmatrix} P\mathbf{1} - \frac{1}{t}P\mathbf{h} & \frac{1}{t}P\mathbf{h} & O_{t+1,m-1} \end{bmatrix} \\ &= \binom{s}{t}\begin{bmatrix} \mathbf{e}_0 - (1-\frac{t}{s})\mathbf{e}_0 + \frac{s-t}{s^2-s}\mathbf{e}_1 & (1-\frac{t}{s})\mathbf{e}_0 - \frac{s-t}{s^2-s}\mathbf{e}_1 & O_{t+1,m-1} \end{bmatrix} \\ &= \binom{s}{t}\begin{bmatrix} \frac{t}{s} & 1-\frac{t}{s} & 0 & \dots & 0 \\ \frac{s-t}{s^2-s} & \frac{t-s}{s^2-s} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \end{aligned}$$

It follows that

$$\begin{aligned} PV\tilde{P}^T = PWK\tilde{P}^T &= \frac{\binom{s}{t}}{2n}\begin{bmatrix} \frac{2t}{s} & 1-\frac{t}{s} \\ \frac{2(s-t)}{s^2-s} & \frac{t-s}{s^2-s} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & \mathbf{1}_m^T \\ 2 & \mathbf{z}^T \end{bmatrix} \\ &= \frac{\binom{s}{t}}{2n}\begin{bmatrix} 2 & \frac{2t}{s}\mathbf{1}_m^T + (1-\frac{t}{s})\mathbf{z}^T \\ 0 & \frac{s-t}{s^2-s}(2\cdot\mathbf{1}_m - \mathbf{z})^T \end{bmatrix}. \end{aligned}$$

We have $2\cdot\mathbf{1}_m - \mathbf{z} > 0$. Using that $\frac{t}{s} = \frac{\delta}{2+\delta}$ and $z_i > -\delta$ for each $i \in [m]$, we find that

$$\frac{2t}{s} + \left(1 - \frac{t}{s}\right)z_i = \frac{2\delta}{2+\delta} + \left(1 - \frac{\delta}{2+\delta}\right)z_i > \frac{2\delta}{2+\delta} + \left(1 - \frac{\delta}{2+\delta}\right)(-\delta) = 0,$$

thus showing that $\frac{2t}{s}\mathbf{1}_m + (1-\frac{t}{s})\mathbf{z} > 0$. It follows that $PV\tilde{P}^T \ge 0$, which means that (c_1) is met. Applying Corollary 8, we deduce that $\mathrm{SDP}(\mathbf{G}_{s,t}, \mathbf{C}_n) = \textsc{Yes}$ and that the matrix $M = \sum_{\omega,\tilde{\omega}} v_{\omega\tilde{\omega}} R_\omega \otimes R_{\tilde{\omega}}$ (where ω ranges over $\mathscr{O}(\mathbf{G}_{s,t})$ and $\tilde{\omega}$ ranges over $\mathscr{O}(\mathbf{C}_n)$) is a balanced SDP-matrix for $\mathbf{G}_{s,t}, \mathbf{C}_n$, cf. Proposition 3.

The next step is to add AIP. For each $\mathbf{x} \in V(\mathbf{G}_{s,t})^2$, let $\omega^{(\mathbf{x})}$ be the orbital of $\mathbf{G}_{s,t}$ containing $\mathbf{x}$, and choose an orbital $\tilde{\omega}^{(\mathbf{x})}$ of $\mathbf{C}_n$ satisfying $v_{\omega^{(\mathbf{x})}\tilde{\omega}^{(\mathbf{x})}} \neq 0$. Letting $f : \mathscr{O}(\mathbf{C}_n) \to \mathbb{Z}^{n\times n}$ be the function from Proposition 13, we consider the $\binom{s}{t}n \times \binom{s}{t}n$ matrix N defined by $N_\mathbf{x} = f(\tilde{\omega}^{(\mathbf{x})})$ for each $\mathbf{x}$ (where $N_\mathbf{x} = (\mathbf{e}_{x_1} \otimes I_n)^T N(\mathbf{e}_{x_2} \otimes I_n)$ is the $\mathbf{x}$-th block of N). We claim that N is an AIP-matrix for $\mathbf{G}_{s,t}, \mathbf{C}_n$. Note that, if $\mathbf{x} = (x, x) \in V(\mathbf{G}_{s,t})^2$, we have $\omega^{(\mathbf{x})} = \omega_0$ and, thus, $\tilde{\omega}^{(\mathbf{x})} = \tilde{\omega}_0$, which gives $\operatorname{supp}(N_\mathbf{x}) = \operatorname{supp}(f(\tilde{\omega}_0)) \subseteq \tilde{\omega}_0$.

Similarly, if $\mathbf{x} \in E(\mathbf{G}_{s,t})$, then $\omega^{(\mathbf{x})} = \omega_t$ and, thus, $\tilde{\omega}^{(\mathbf{x})} = \tilde{\omega}_1$, which gives $\operatorname{supp}(N_{\mathbf{x}}) = \operatorname{supp}(f(\tilde{\omega}_1)) \subseteq \tilde{\omega}_1 = E(\mathbf{C}_n)$. This yields the conditions (r_1) and (r_2). Moreover, for $\mathbf{x} = (x_1, x_2) \in V(\mathbf{G}_{s,t})^2$, we find

$$\begin{aligned}(\mathbf{e}_{x_1} \otimes I_n)^T N(\mathbf{e}_{x_2} \otimes \mathbf{1}_n) &= (\mathbf{e}_{x_1} \otimes I_n)^T N(\mathbf{e}_{x_2} \otimes I_n)(1 \otimes \mathbf{1}_n) \\ &= N_{\mathbf{x}}\mathbf{1}_n = f(\tilde{\omega}^{(\mathbf{x})})\mathbf{1}_n\end{aligned}$$

which, by the properties of f, is constant over the orbitals of $\mathbf{C}_n$; this gives (r_3). Similarly, using that $f(\tilde{\omega}^{(\mathbf{x})})^T\mathbf{1}_n$ is constant over the orbitals, we obtain (r_4). Finally, (r_5) follows by observing that $\mathbf{1}_n^T N_{\mathbf{x}} \mathbf{1}_n = \mathbf{1}_n^T f(\tilde{\omega}^{(\mathbf{x})})\mathbf{1}_n = \mathbf{1}_n^T \mathbf{e}_0 = 1$ for any $\mathbf{x}$. As a consequence, N is a relaxation matrix; since its entries are integral, it is an AIP-matrix. For any $\mathbf{x} \in V(\mathbf{G}_{s,t})^2$, the $\mathbf{x}$-th block of M satisfies

$$\begin{aligned}M_{\mathbf{x}} &= (\mathbf{e}_{x_1} \otimes I_n)^T M(\mathbf{e}_{x_2} \otimes I_n) \\ &= \sum_{\substack{\omega \in \mathscr{O}(\mathbf{G}_{s,t}) \\ \tilde{\omega} \in \mathscr{O}(\mathbf{C}_n)}} v_{\omega\tilde{\omega}}\, (\mathbf{e}_{x_1} \otimes I_n)^T (R_\omega \otimes R_{\tilde{\omega}})(\mathbf{e}_{x_2} \otimes I_n) \\ &= \sum_{\substack{\omega \in \mathscr{O}(\mathbf{G}_{s,t}) \\ \tilde{\omega} \in \mathscr{O}(\mathbf{C}_n)}} v_{\omega\tilde{\omega}}\, (\mathbf{e}_{x_1}^T R_\omega \mathbf{e}_{x_2}) R_{\tilde{\omega}} = \sum_{\tilde{\omega} \in \mathscr{O}(\mathbf{C}_n)} v_{\omega^{(\mathbf{x})}\tilde{\omega}} R_{\tilde{\omega}}.\end{aligned}$$

Since $v_{\omega^{(\mathbf{x})}\tilde{\omega}^{(\mathbf{x})}} \neq 0$, using that the orbitals of a graph are disjoint, we deduce that $R_{\tilde{\omega}^{(\mathbf{x})}} \lhd M_{\mathbf{x}}$. On the other hand, we have $\operatorname{supp}(N_{\mathbf{x}}) = \operatorname{supp}(f(\tilde{\omega}^{(\mathbf{x})})) \subseteq \tilde{\omega}^{(\mathbf{x})} = \operatorname{supp}(R_{\tilde{\omega}^{(\mathbf{x})}})$, which means that $N_{\mathbf{x}} \lhd R_{\tilde{\omega}^{(\mathbf{x})}}$. It follows that $N \circ ((I_{\binom{s}{t}} + \mathscr{A}(\mathbf{G}_{s,t})) \otimes J_n) \lhd N \lhd M$. Applying Proposition 2, we conclude that $\mathrm{SDA}(\mathbf{G}_{s,t}, \mathbf{C}_n) = \textsc{Yes}$, as required. □

We note that the SDP part of the integrality gap in Theorem 1 may be directly converted into Unique-Games approximation hardness of AGH through Raghavendra's framework [78]. Indeed, as observed in [17], it follows from [78] that any PCSP admitting a polynomial-time robust algorithm[23] is solved by SDP, assuming the Unique Games Conjecture (UGC) of [65]. Thus, Theorem 1 implies the following conditional hardness result for AGH.

Corollary 14. *Let* $\mathbf{A}, \mathbf{B}$ *be non-bipartite loopless undirected graphs such that* $\mathbf{A} \to \mathbf{B}$. *Then, assuming the* UGC *and* $\mathrm{P} \neq \mathrm{NP}$, $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ *does not admit a polynomial-time robust algorithm.*

2.5 Incomparability to the BA Hierarchy

It was shown in [27] that no constant level of the BA hierarchy – obtained by combining the Sherali–Adams LP hierarchy with linear Diophantine equations[24] – solves approximate graph colouring. It is then natural to ask how SDA compares to such hierarchy. In particular, we investigate whether SDA is "dominated" by some level BA^k of the BA hierarchy, in the sense that, for any instance on which the former algorithm gives the correct answer, the latter also does.[25] If this were the case, non-solvability of approximate graph colouring via SDA would be a corollary of the result in [27]. We give a negative answer to the above question by showing that not even SDP (in fact, not even a weaker version SDP^ϵ of SDP that can be implemented in polynomial time, see the paragraph below) is dominated by the BA hierarchy. Formally, we establish the following result.

Theorem 15. *For each* $k \in \mathbb{N}$ *there exists* $\epsilon > 0$ *such that* $\mathrm{SDP}^\epsilon \not\preceq \mathrm{BA}^k$.[26]

In order to prove Theorem 15, we use cliques as the separating instances. Indeed, a result in [27] implies that the BA hierarchy is not sound on cliques; the situation is different for SDP^ϵ, which, as stated next and proved via the framework of association schemes, is able to correctly classify cliques provided that ϵ is small enough. (Of course, it follows that the same result holds for the stronger relaxations SDP and SDA.)

Proposition 16. *Let* $p, n \geq 2$ *and* $0 < \epsilon < \frac{1}{n^3}$. *Then* $\mathrm{SDP}^\epsilon(\mathbf{K}_p, \mathbf{K}_n) = \textsc{Yes}$ *if and only if* $p \leq n$.

A tale of two polytopes. A few years ago, O'Donnell noted that polynomial-time solvability of certain semidefinite programming relaxations, assumed in several papers in the context of the Sum-of-Squares proof system, is in fact not true in general [76]. To the best of the authors' knowledge, details of how the semidefinite program SDP can be solved in polynomial time to near-optimality (if at all) have not been made explicit in the literature. This motivates us to give a formal argument showing that this is indeed possible. As we shall discuss in detail in the full version [30], the issue is rather subtle and requires unexpected matrix-theoretic considerations.

For $\epsilon > 0$, we define SDP^ϵ by turning SDP into an optimisation problem, and then solving it up to precision ϵ via the semidefinite program solver of [49] based on the ellipsoid method. The optimisation problem consists in minimising a linear objective function (given by the sum of the inner products in (SDP_3), see the SDP formulation in the full version [30]) over the intersection of a polytope with the cone of positive semidefinite matrices. There are two natural candidates for choosing such a polytope. The first, $\mathscr{W}$, is obtained by discarding (SDP_3) and positive semidefiniteness from the constraints describing the feasible region of the system (SDP); the second, $\mathscr{U}$, is obtained by discarding (r_2) – which is the matrix-analogue of (SDP_3) – and positive semidefiniteness from the conditions defining SDP-matrices. The intersections of either of these two polytopes with the cone of positive semidefinite matrices coincide, so if we were able to solve the corresponding programs to optimality we would obtain the same output using either formulation. However, the two polytopes are different, and we prove that $\mathscr{U}$ (and not $\mathscr{W}$!) meets *Slater condition*, which ensures that the corresponding semidefinite program can be solved in polynomial time to near-optimality in the Turing model of computation. Thus, we formulate SDP^ϵ using the polytope $\mathscr{U}$, and we obtain the following result.

Theorem 17. *For each* $\epsilon > 0$, SDP^ϵ *is a complete, polynomial-time test. Moreover,* $\mathrm{SDP}^\epsilon \preceq \mathrm{SDP}$.

[23] Given two digraphs $\mathbf{X}, \mathbf{X}'$ and a real number $0 \leq \epsilon \leq 1$, an *ϵ-homomorphism* from $\mathbf{X}$ to $\mathbf{X}'$ is a map $f : V(\mathbf{X}) \to V(\mathbf{X}')$ that preserves at least $(1 - \epsilon)$ fraction of the edges of $\mathbf{X}$. A *robust* algorithm for $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ is an algorithm that finds a $g(\epsilon)$-homomorphism from $\mathbf{X}$ to $\mathbf{B}$ whenever the instance $\mathbf{X}$ is such that there exists an ϵ-homomorphism from $\mathbf{X}$ to $\mathbf{A}$, where g is some monotone, nonnegative function satisfying $g(\epsilon) \to 0$ as $\epsilon \to 0$.

[24] The BA hierarchy may alternatively be described as the result of applying the *lift-and-project* technique to the algorithm introduced in [18], see also [29].

[25] For two algorithms T_1, T_2, we write $T_1 \preceq T_2$ to mean that T_2 dominates T_1; see the formal definition in the full version [30]

[26] Since $\mathrm{SDP}^\epsilon \preceq \mathrm{SDP} \preceq \mathrm{SDA}$, it follows that $\mathrm{SDP} \not\preceq \mathrm{BA}^k$ and $\mathrm{SDA} \not\preceq \mathrm{BA}^k$. It is, however, an open question how the algorithms compare in terms of *solvability* of PCSPs, see the full version [30].

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. We thank the referees for their comments.

REFERENCES

[1] Noga Alon and Benny Sudakov. 2000. Bipartite subgraphs and the smallest eigenvalue. *Comb. Probab. Comput.* 9, 1 (2000), 1–12. https://doi.org/10.1017/S0963548399004071

[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] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. 2009. Expander flows, geometric embeddings and graph partitioning. *J. ACM* 56, 2 (2009). https://doi.org/10.1145/1502793.1502794

[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 Håstad. 2017. $(2+\epsilon)$-Sat Is NP-hard. *SIAM J. Comput.* 46, 5 (2017), 1554–1573. https://doi.org/10.1137/15M1006507

[6] Eiichi Bannai and Tatsuro Ito. 1984. *Algebraic Combinatorics I. Association Schemes.* Benjamin/Cummings Publishing.

[7] 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

[8] Libor Barto and Marcin Kozik. 2016. Robustly Solvable Constraint Satisfaction Problems. *SIAM J. Comput.* 45, 4 (2016), 1646–1669. https://doi.org/10.1137/130915479 arXiv:1512.01157

[9] Libor Barto, Jakub Opršal, and Michael Pinsker. 2018. The wonderland of reflections. *Isr. J. Math* 223, 1 (Feb 2018), 363–398. https://doi.org/10.1007/s11856-017-1621-9 arXiv:1510.04521

[10] 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

[11] Béla Bollobás. 1986. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability.* Cambridge University Press.

[12] Raj Chandra Bose and Dale M. Mesner. 1959. On Linear Associative Algebras Corresponding to Association Schemes of Partially Balanced Designs. *Ann. Math. Statist.* 30, 1 (1959), 21–38. https://doi.org/10.1214/aoms/1177706356

[13] 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

[14] 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

[15] 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

[16] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2023. Conditional Dichotomy of Boolean Ordered Promise CSPs. *TheoretiCS* 2 (2023). https://doi.org/10.46298/theoretics.23.2 arXiv:2102.11854

[17] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2023. SDPs and Robust Satisfiability of Promise CSP. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 609–622. https://doi.org/10.1145/3564246.3585180 arXiv:2211.08373

[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] 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 arXiv:2110.10725

[20] A.E. Brouwer, A.M. Cohen, and A. Neumaier. 1989. *Distance-Regular Graphs.* Springer, Heidelberg.

[21] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM J. Comput.* 34, 3 (2005), 720–742. https://doi.org/10.1137/S0097539700376676

[22] 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

[23] 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

[24] Siu On Chan, James R. Lee, Prasad Raghavendra, 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

[25] Siu On Chan, Hiu Tsun Ng, and Sijin Peng. 2024. How Random CSPs Fool Hierarchies. In *Proc. 56th Annual ACM Symposium on Theory of Computing (STOC'24)*. ACM.

[26] 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

[27] Lorenzo Ciardo and Stanislav Živný. 2023. Approximate Graph Colouring and the Hollow Shadow. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 623–631. https://doi.org/10.1145/3564246.3585112 arXiv:2211.03168

[28] 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

[29] 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

[30] Lorenzo Ciardo and Stanislav Živný. 2023. Semidefinite programming and linear equations vs. homomorphism problems. arXiv:2311.00882

[31] 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

[32] Víctor Dalmau, Andrei A. Krokhin, and Rajsekar Manokaran. 2018. Towards a characterization of constant-factor approximable finite-valued CSPs. *J. Comput. Syst. Sci.* 97 (2018), 14–27. https://doi.org/10.1016/j.jcss.2018.03.003 arXiv:1610.01019

[33] Victor Dalmau and Jakub Opršal. 2023. Local consistency as a reduction between constraint satisfaction problems. (2023). arXiv:2301.05084

[34] Etienne de Klerk, Dmitrii V. Pasechnik, and Alexander Schrijver. 2007. Reduction of symmetric semidefinite programs using the regular *-representation. *Math. Program.* 109, 2-3 (2007), 613–624. https://doi.org/10.1007/S10107-006-0039-7

[35] Philippe Delsarte. 1973. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.* 10 (1973), 1–97. https://users.wpi.edu/~martin/RESEARCH/philips.pdf

[36] Harm Derksen and Gregor Kemper. 2015. *Computational Invariant Theory.* Springer.

[37] Yash Deshpande and Andrea Montanari. 2015. Improved Sum-of-Squares Lower Bounds for Hidden Clique and Hidden Submatrix Problems. In *Proc. 28th Conference on Learning Theory (COLT'15) (JMLR Workshop and Conference Proceedings, Vol. 40)*. JMLR.org, 523–562. http://proceedings.mlr.press/v40/Deshpande15.html

[38] 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

[39] 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

[40] G.F. Frobenius. 1912. Über Matrizen aus nicht negativen Elementen. *Königliche Akademie der Wissenschaften Berlin* (1912), 456–477. https://doi.org/10.3931/e-rara-18865

[41] 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

[42] Karin Gatermann and Pablo A Parrilo. 2004. Symmetry groups, semidefinite programs, and sums of squares. *J. Pure Appl. Algebra* 192, 1-3 (2004), 95–128. https://doi.org/10.1016/j.jpaa.2003.12.011

[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 arXiv:1608.00497

[44] Chris Godsil. 2010. Association Schemes. https://www.math.uwaterloo.ca/~cgodsil/pdfs/assoc2.pdf

[45] Christopher Godsil and Karen Meagher. 2016. *Erdős-Ko-Rado Theorems: Algebraic Approaches.* Number 149 in Cambridge studies in advanced mathematics. Cambridge University Press.

[46] Chris Godsil and Gordon F. Royle. 2001. *Algebraic Graph Theory.* Vol. 207. Springer Science & Business Media.

[47] Michel X. Goemans and Franz Rendl. 1999. Semidefinite Programs and Association Schemes. *Computing* 63, 4 (1999), 331–340. https://doi.org/10.1007/

s006070050038

[48] Michel X. Goemans and David P. Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* 42, 6 (1995), 1115–1145. https://doi.org/10.1145/227683.227684

[49] Martin Grötschel, László Lovász, and Alexander Schrijver. 1981. The ellipsoid method and its consequences in combinatorial optimization. *Comb.* 1 (1981), 169–197. https://doi.org/10.1007/BF02579273

[50] Venkatesan Guruswami and Sanjeev Khanna. 2004. On the Hardness of 4-Coloring a 3-Colorable Graph. *SIAM J. Discret. Math.* 18, 1 (2004), 30–40. https://doi.org/10.1137/S0895480100376794

[51] 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

[52] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (2001), 798–859. https://doi.org/10.1145/502090.502098

[53] Pavol Hell and Jaroslav Nešetřil. 1990. On the complexity of H-coloring. *J. Comb. Theory, Ser. B* 48, 1 (1990), 92–110. https://doi.org/10.1016/0095-8956(90)90132-J

[54] Roger A. Horn and Charles R. Johnson. 1994. *Topics in Matrix Analysis*. Cambridge University Press.

[55] 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

[56] Peter G. Jeavons. 1998. On the Algebraic Structure of Combinatorial Problems. *Theor. Comput. Sci.* 200, 1-2 (1998), 185–204. https://doi.org/10.1016/S0304-3975(97)00230-2

[57] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. 1997. Closure Properties of Constraints. *J. ACM* 44, 4 (1997), 527–548. https://doi.org/10.1145/263867.263489

[58] Yoshihiro Kanno, Makoto Ohsaki, Kazuo Murota, and Naoki Katoh. 2001. Group symmetry in interior-point methods for semidefinite program. *Optim. Eng.* 2 (2001), 293–320.

[59] David R. Karger, Rajeev Motwani, and Madhu Sudan. 1998. Approximate Graph Coloring by Semidefinite Programming. *J. ACM* 45, 2 (1998), 246–265. https://doi.org/10.1145/274787.274791

[60] Howard J. Karloff. 1999. How Good is the Goemans-Williamson MAX CUT Algorithm? *SIAM J. Comput.* 29, 1 (1999), 336–350. https://doi.org/10.1137/S0097539797321481

[61] 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

[62] 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

[63] 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

[64] 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

[65] 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

[66] 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

[67] Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. 2015. The power of linear programming for general-valued CSPs. *SIAM J. Comput.* 44, 1 (2015), 1–36. https://doi.org/10.1137/130945648 arXiv:1311.4219

[68] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. 2022. Approximating Rectangles by Juntas and Weakly Exponential Lower Bounds for LP Relaxations of CSPs. *SIAM J. Comput.* 51, 2 (2022), 17–305. https://doi.org/10.1137/17m1152966 arXiv:1610.02704

[69] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. 2023. Topology and adjunction in promise constraint satisfaction. *SIAM J. Comput.* 52, 1 (2023), 37–79. https://doi.org/10.1137/20M1378223 arXiv:2003.11351

[70] 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

[71] 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

[72] 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 arXiv:1411.6317

[73] László Lovász. 1978. Kneser's conjecture, chromatic number, and homotopy. *J. Comb. Theory, Ser. A* 25, 3 (1978), 319–324. https://doi.org/10.1016/0097-3165(78)90022-5

[74] László Lovász. 1979. On the Shannon capacity of a graph. *IEEE Trans. Inf. Theory* 25, 1 (1979), 1–7. https://doi.org/10.1109/TIT.1979.1055985

[75] Raghu Meka, Aaron Potechin, and Avi Wigderson. 2015. Sum-of-squares Lower Bounds for Planted Clique. In *Proc. 47th Annual ACM Symposium on Theory of Computing (STOC'15)*. ACM, 87–96. https://doi.org/10.1145/2746539.2746600

[76] Ryan O'Donnell. 2017. SOS Is Not Obviously Automatizable, Even Approximately. In *Proc. 8th Innovations in Theoretical Computer Science Conference (ITCS'17) (LIPIcs, Vol. 67)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 59:1–59:10. https://doi.org/10.4230/LIPIcs.ITCS.2017.59

[77] Oskar Perron. 1907. Zur Theorie der Matrizen. *Math. Ann.* 64 (1907), 248–263. https://doi.org/10.1007/BF01449896

[78] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP?. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC'08)*. ACM, 245–254. https://doi.org/10.1145/1374376.1374414

[79] Grant Schoenebeck. 2008. Linear Level Lasserre Lower Bounds for Certain k-CSPs. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*. IEEE Computer Society, 593–602. https://doi.org/10.1109/FOCS.2008.74

[80] Alexander Schrijver. 1998. *Theory of Linear and Integer Programming*. John Wiley & Sons.

[81] Bernd Sturmfels. 2008. *Algorithms in Invariant Theory*. Springer Science & Business Media.

[82] Johan Thapper and Stanislav Živný. 2017. The power of Sherali–Adams relaxations for general-valued CSPs. *SIAM J. Comput.* 46, 4 (2017), 1241–1279. https://doi.org/10.1137/16M1079245 arXiv:1606.02577

[83] Johan Thapper and Stanislav Živný. 2018. The Limits of SDP Relaxations for General-Valued CSPs. *ACM Trans. Comput. Theory* 10, 3 (2018), 12:1–12:22. https://doi.org/10.1145/3201777 arXiv:1612.01147

[84] 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

[85] 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 01-NOV-2023; accepted 2024-02-11