Grothendieck-Type Inequalities in Combinatorial Optimization

SUBHASH KHOT
Courant Institute

ASSAF NAOR
Courant Institute

Abstract

We survey connections of the Grothendieck inequality and its variants to combinatorial optimization and computational complexity.

1 Introduction

The Grothendieck inequality asserts that there exists a universal constant $K \in (0,\infty)$ such that for every $m,n \in \mathbb{N}$ and every $m \times n$ matrix $A = (a_{ij})$ with real entries we have

$$\max\Big\{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\langle x_i, y_j\rangle : \{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq \mathbb{S}^{n+m-1}\Big\} \le K \max\Big\{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\varepsilon_i\delta_j : \{\varepsilon_i\}_{i=1}^m, \{\delta_j\}_{j=1}^n \subseteq \{-1,1\}\Big\}. \tag{1.1}$$

Here and in what follows, the standard scalar product on $\mathbb{R}^k$ is denoted $\langle x, y\rangle = \sum_{i=1}^k x_i y_i$ and the euclidean sphere in $\mathbb{R}^k$ is denoted

$$\mathbb{S}^{k-1} = \Big\{x \in \mathbb{R}^k : \sum_{i=1}^{k} x_i^2 = 1\Big\}.$$

We refer to [36, 58] for the simplest known proofs of the Grothendieck inequality; see Section 2.2 for a proof of (1.1) yielding the best-known bound on K. Grothendieck proved inequality (1.1) in [47], though it was stated there in a different but equivalent form. The formulation of the Grothendieck inequality appearing in (1.1) is due to Lindenstrauss and Pełczyński [85].

The Grothendieck inequality is of major importance to several areas, ranging from Banach space theory to C^* algebras and quantum information theory. We will

Communications on Pure and Applied Mathematics, Vol. LXV, 0992–1035 (2012)

not attempt to indicate here this wide range of applications of (1.1), and refer instead to [1, 20, 35, 36, 39, 42, 57, 85, 102, 103, 104, 116] and the references therein. The purpose of this survey is to focus solely on applications of the Grothendieck inequality and its variants to combinatorial optimization, and to explain their connections to computational complexity. The link between the Grothendieck inequality and combinatorial optimization stems from the fact that (1.1) can be viewed as a bound on a tractable vector relaxation (the left-hand side of (1.1)) of a certain integer program (the right-hand side of (1.1)); this will be explained in detail later.

The infimum over those $K \in (0, \infty)$ for which (1.1) holds for all $m, n \in \mathbb{N}$ and all $m \times n$ matrices $A = (a_{ij})$ is called the Grothendieck constant, and is denoted K_G. Evaluating the exact value of K_G remains a longstanding open problem, posed by Grothendieck in [47]. In fact, even the second digit of K_G is currently unknown, though clearly this is of lesser importance than the issue of understanding the structure of matrices A and spherical configurations $\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq \mathbb{S}^{n+m-1}$, which make the inequality (1.1) "most difficult." Following a series of investigations [47, 79, 80, 85, 109], the best-known upper bound [23] on K_G is

$$K_G < \frac{\pi}{2\log(1+\sqrt{2})} = 1.782\ldots, \tag{1.2}$$

and the best-known lower bound [107] on K_G is

$$K_G \geq \frac{\pi}{2} e^{\eta_0^2} = 1.676\ldots, \tag{1.3}$$

where $\eta_0 = 0.25573\ldots$ is the unique solution of the equation

$$1 - 2\sqrt{\frac{2}{\pi}} \int_0^{\eta} e^{-z^2/2}\, dz = \frac{2}{\pi} e^{-\eta^2}.$$

In [106] the problem of estimating K_G up to an additive error of $\varepsilon \in (0, 1)$ was reduced to an optimization over a compact space, and by exhaustive search over an appropriate net it was shown that there exists an algorithm that computes K_G up to an additive error of $\varepsilon \in (0, 1)$ in time $\exp(\exp(O(1/\varepsilon^3)))$. It does not seem likely that this approach can yield computer-assisted proofs of estimates such as (1.2) and (1.3), though to the best of our knowledge this has not been attempted.

Above we focused on the classical Grothendieck inequality (1.1). However, the literature contains several variants and extensions of (1.1) that have been introduced for various purposes and applications in the decades following Grothendieck's original work. In this survey we describe some of these variants, emphasizing relatively recent developments that yielded Grothendieck-type inequalities that are a useful tool in the design of polynomial-time algorithms for computing approximate solutions of computationally hard optimization problems. In doing so, we omit some important topics, including applications of the Grothendieck inequality to communication complexity and quantum information theory. While these research directions can be viewed as dealing with a type of optimization problem, they are of a different nature than the applications described here, which belong

to classical optimization theory. Connections to communication complexity have already been covered in the survey of Lee and Shraibman [83]; we refer in addition to [82, 86, 87, 88] for more information on this topic.

An explanation of the relation of the Grothendieck inequality to quantum mechanics is contained in section 19 of Pisier's survey [103], the pioneering work in this direction being that of Tsirelson [116]. An investigation of these questions from a computational complexity point of view was initiated in [30], where it was shown, for example, how to obtain a polynomial-time algorithm for computing the entangled value of an XOR game based on Tsirelson's work. We hope that the developments surrounding applications of the Grothendieck inequality in quantum information theory will eventually be surveyed separately by experts in this area. Interested readers are referred to [1, 24, 30, 39, 56, 63, 82, 88, 100, 103, 104, 108, 116].

Perhaps the most influential variants of the Grothendieck inequality are its noncommutative generalizations. The noncommutative versions in [51, 101] were conjectured by Grothendieck himself [47]; additional extensions to operator spaces are extensively discussed in Pisier's survey [103]. We will not describe these developments here, even though we believe that they might have applications to optimization theory.

Finally, multilinear extensions of the Grothendieck inequality have also been investigated in the literature; see, for example, [21, 22, 111, 114, 117] and especially Blei's book [20]. We will not cover this research direction since its relation to classical combinatorial optimization has not (yet?) been established, though there are recent investigations of multilinear Grothendieck inequalities in the context of quantum information theory [82, 100].

Being a mainstay of functional analysis, the Grothendieck inequality might attract to this survey readers who are not familiar with approximation algorithms and computational complexity. We wish to encourage such readers to persist beyond this introduction so that they will be exposed to, and hopefully eventually contribute to, the use of analytic tools in combinatorial optimization. For this reason we include Sections 1.1 and 1.2 below: two very basic introductory sections intended to quickly provide background on computational complexity and convex programming for nonexperts.

1.1 Assumptions from Computational Complexity

At present there are few unconditional results on the limitations of polynomial-time computation. The standard practice in this field is to frame an impossibility result in computational complexity by asserting that the polynomial-time solvability of a certain algorithmic task would contradict a benchmark hypothesis. We briefly describe below two key hypotheses of this type.

A graph $G = (V, E)$ is 3-colorable if there exists a partition $\{C_1, C_2, C_3\}$ of V such that for every $i \in \{1, 2, 3\}$ and $u, v \in C_i$ we have $\{u, v\} \notin E$. The P $\neq$ NP hypothesis asserts that there is no polynomial-time algorithm that takes an n-vertex

graph as input and determines whether it is 3-colorable. We are doing an injustice to this important question by stating it this way, since it has many far-reaching equivalent formulations. We refer to [33, 41, 110] for more information, but for nonexperts it suffices to keep the above simple formulation in mind.

When we say that assuming P $\neq$ NP no polynomial-time algorithm can perform a certain task $\mathcal{T}$ (e.g., evaluating the maximum of a certain function up to a predetermined error) we mean that given an algorithm ALG that performs the task $\mathcal{T}$, one can design an algorithm ALG' that determines whether any input graph is 3-colorable while making at most polynomially many calls to the algorithm ALG, with at most polynomially many additional Turing machine steps. Thus, if ALG were a polynomial-time algorithm, then the same would be true for ALG', contradicting the P $\neq$ NP hypothesis. Such results are called hardness results. The message that nonexperts should keep in mind is that a hardness result is nothing more than the design of a new algorithm for 3-colorability, and if one accepts the P $\neq$ NP hypothesis, then it implies that there must exist inputs on which ALG takes superpolynomial time to terminate.

The unique games conjecture (UGC) asserts that for every $\varepsilon \in (0, 1)$ there exists a prime $p = p(\varepsilon) \in \mathbb{N}$ such that no polynomial-time algorithm can perform the following task: The input is a system of m linear equations in n variables $x_1, \ldots, x_n$, each of which has the form $x_i - x_j \equiv c_{ij} \mod p$ (thus the input is $S \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ and $\{c_{ij}\}_{(i,j)\in S} \subseteq \mathbb{N}$). The algorithm must determine whether there exists an assignment of an integer value to each variable x_i such that at least $(1 - \varepsilon)m$ of the equations are satisfied, or whether no assignment of such values can satisfy more than εm of the equations. If neither of these possibilities occurs, then an arbitrary output is allowed.

As in the case of P $\neq$ NP, saying that assuming the UGC no polynomial-time algorithm can perform a certain task $\mathcal{T}$ is the same as designing a polynomial-time algorithm that solves the above linear equations problem while making at most polynomially many calls to a "black box" that can perform the task $\mathcal{T}$. The UGC was introduced in [64], though the above formulation of it, which is equivalent to the original one, is due to [66]. The use of the UGC as a hardness hypothesis has become popular over the past decade; we refer to the survey [65] for more information on this topic.

To simplify matters (while describing all the essential ideas), we allow polynomial-time algorithms to be randomized. Most (if not all) of the algorithms described here can be turned into deterministic algorithms, and corresponding hardness results can be stated equally well in the context of randomized or deterministic algorithms. We will ignore these distinctions, even though they are important. Moreover, it is widely believed that in our context these distinctions do not exist; i.e., randomness does not add computational power to polynomial-time algorithms; see, for example, the discussion of the NP $\not\subseteq$ BPP hypothesis in [11].

1.2 Convex and Semidefinite Programming

An important paradigm of optimization theory is that one can efficiently optimize linear functionals over compact convex sets that have a "membership oracle." A detailed exposition of this statement is contained in [48], but for the sake of completeness we now quote the precise formulation of the results that will be used in this article.

Let $K \subseteq \mathbb{R}^n$ be a compact convex set. We are also given a point $z \in \mathbb{Q}^n$ and two radii $r, R \in (0, \infty) \cap \mathbb{Q}$ such that $B(z, r) \subseteq K \subseteq B(z, R)$, where $B(z, t) = \{x \in \mathbb{R}^n : \|x - z\|_2 \leq t\}$. In what follows, stating that an algorithm is polynomial means that we allow the running time to grow at most polynomially in the number of bits required to represent the data (z, r, R). Thus, if, say, $z = 0$, $r = 2^{-n}$, and $R = 2^n$, then the running time will be polynomial in the dimension n. Assume that there exists an algorithm ALG with the following properties: The input of ALG is a vector $y \in \mathbb{Q}^n$ and $\varepsilon \in (0, 1) \cap \mathbb{Q}$. The running time of ALG is polynomial in n, and the number of bits required to represent the data (ε, y). The output of ALG is the assertion that either the distance of y from K is at most ε or that the distance of y from the complement of K is at most ε. Then there exists an algorithm ALG' that takes as input a vector $c = (c_1, \ldots, c_n) \in \mathbb{Q}^n$ and $\varepsilon \in (0, 1) \cap \mathbb{Q}$, and outputs a vector $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ that is at distance at most ε from K; for every $x = (x_1, \ldots, x_n) \in K$ that is at distance greater than ε from the complement of K, we have $\sum_{i=1}^n c_i y_i \geq \sum_{i=1}^n c_i x_i - \varepsilon$. The running time of ALG' is allowed to grow at most polynomially in n and the number of bits required to represent the data $(z, r, R, c, \varepsilon)$. This important result is due to [59]; we refer to [48] for an excellent account of this theory.

The above statement is a key tool in optimization, as it yields a polynomial-time method to compute the maximum of linear functionals on a given convex body with arbitrarily good precision. We note the following special case of this method, known as semidefinite programming. Assume that $n = k^2$ and think of $\mathbb{R}^n$ as the space of all $k \times k$ matrices. Assume that we are given a compact convex set $K \subseteq \mathbb{R}^n$ that satisfies the above assumptions, and that for a given $k \times k$ matrix (c_{ij}) we wish to compute in polynomial time (up to a specified additive error) the maximum of $\sum_{i=1}^k \sum_{j=1}^k c_{ij} x_{ij}$ over the set of symmetric positive semidefinite matrices (x_{ij}) that belong to K. This can indeed be done, since determining whether a given symmetric matrix is (approximately) positive semidefinite is an eigenvalue computation and hence can be performed in polynomial time.

The use of semidefinite programming to design approximation algorithms is by now a deep theory of fundamental importance to several areas of theoretical computer science. The Goemans-Williamson MAX-CUT algorithm [44] was a key breakthrough in this context. It is safe to say that after the discovery of this algorithm the field of approximation algorithms was transformed, and many subsequent results, including those presented in the present article, can be described

as attempts to mimic the success of the Goemans-Williamson approach in other contexts.

2 Applications of the Classical Grothendieck Inequality

The classical Grothendieck inequality (1.1) has applications to algorithmic questions of central interest. These applications will be described here in some detail. In Section 2.1 we discuss the cut norm estimation problem, whose relation to the Grothendieck inequality was first noted in [8]. This is a generic combinatorial optimization problem that contains well-studied questions as subproblems. Examples of its usefulness are presented in Sections 2.1.1, 2.1.2, 2.1.3, and 2.1.4. Section 2.2 is devoted to the rounding problem, including the (algorithmic) method behind the proof of the best-known upper bound on the Grothendieck constant.

2.1 Cut Norm Estimation

Let $A = (a_{ij})$ be an $m \times n$ matrix with real entries. The cut norm of A is defined as follows:

$$\|A\|_{\text{cut}} = \max_{\substack{S \subseteq \{1,\ldots,m\} \\ T \subseteq \{1,\ldots,n\}}} \left| \sum_{\substack{i \in S \\ j \in T}} a_{ij} \right|. \tag{2.1}$$

We will now explain how the Grothendieck inequality can be used to obtain a polynomial-time algorithm for the following problem. The input is an $m \times n$ matrix $A = (a_{ij})$ with real entries, and the goal of the algorithm is to output in polynomial time a number α that is guaranteed to satisfy

$$\|A\|_{\text{cut}} \le \alpha \le C\,\|A\|_{\text{cut}}, \tag{2.2}$$

where C is a (hopefully not too large) universal constant. A closely related algorithmic goal is to output in polynomial time two subsets $S_0 \subseteq \{1,\ldots,m\}$ and $T_0 \subseteq \{1,\ldots,n\}$ satisfying

$$\left| \sum_{\substack{i \in S_0 \\ j \in T_0}} a_{ij} \right| \ge \frac{1}{C}\|A\|_{\text{cut}}. \tag{2.3}$$

The link to the Grothendieck inequality is made via two simple transformations. First, define an $(m+1) \times (n+1)$ matrix $B = (b_{ij})$ as follows:

$$B = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & -\sum_{k=1}^{n} a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2n} & -\sum_{k=1}^{n} a_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & -\sum_{k=1}^{n} a_{mk} \\ -\sum_{\ell=1}^{m} a_{\ell 1} & -\sum_{\ell=1}^{m} a_{\ell 2} & \cdots & -\sum_{\ell=1}^{m} a_{\ell n} & \sum_{k=1}^{n}\sum_{\ell=1}^{m} a_{\ell k} \end{pmatrix}. \tag{2.4}$$

Observe that

$$\|A\|_{\mathrm{cut}} = \|B\|_{\mathrm{cut}}. \tag{2.5}$$

Indeed, for every $S \subseteq \{1,\ldots,m+1\}$ and $T \subseteq \{1,\ldots,n+1\}$, define $S^* \subseteq \{1,\ldots,m\}$ and $T^* \subseteq \{1,\ldots,n\}$ by

$$S^* = \begin{cases} S & \text{if } m+1 \notin S, \\ \{1,\ldots,m\} \setminus S & \text{if } m+1 \in S, \end{cases}$$

and

$$T^* = \begin{cases} T & \text{if } n+1 \notin T, \\ \{1,\ldots,n\} \setminus T & \text{if } n+1 \in T. \end{cases}$$

One checks that for all $S \subseteq \{1,\ldots,m+1\}$ and $T \subseteq \{1,\ldots,n+1\}$ we have

$$\Big|\sum_{\substack{i\in S\\ j\in T}} b_{ij}\Big| = \Big|\sum_{\substack{i\in S^*\\ j\in T^*}} a_{ij}\Big|,$$

implying (2.5).

We next claim that

$$\|B\|_{\mathrm{cut}} = \frac{1}{4}\|B\|_{\infty\to 1}, \tag{2.6}$$

where

$$\|B\|_{\infty\to 1} = \max\left\{\sum_{i=1}^{m+1}\sum_{j=1}^{n+1} b_{ij}\varepsilon_i\delta_j : \{\varepsilon_i\}_{i=1}^{m+1}, \{\delta_j\}_{j=1}^{n+1} \subseteq \{-1,1\}\right\}. \tag{2.7}$$

To explain this notation observe that $\|B\|_{\infty\to 1}$ is the norm of B when viewed as a linear operator from ℓ_∞^n to ℓ_1^m. Here and in what follows, for $p \in [1,\infty]$ and $k \in \mathbb{N}$ the space ℓ_p^k is $\mathbb{R}^k$ equipped with the ℓ_p-norm $\|\cdot\|_p$, where $\|x\|_p^p = \sum_{\ell=1}^k |x_\ell|^p$ for $x = (x_1,\ldots,x_k) \in \mathbb{R}^k$ (for $p=\infty$ we set as usual $\|x\|_\infty = \max_{i\in\{1,\ldots,n\}} |x_i|$). Though it is important, this operator-theoretic interpretation of the quantity $\|B\|_{\infty\to 1}$ will not have any role in this survey, so it may be harmlessly ignored at first reading.

The proof of (2.6) is simple: for $\{\varepsilon_i\}_{i=1}^{m+1}, \{\delta_j\}_{j=1}^{n+1} \subseteq \{-1,1\}$ define $S^+, S^- \subseteq \{1,\ldots,m+1\}$ and $T^+, T^- \subseteq \{1,\ldots,n+1\}$ by setting $S^\pm = \{i \in \{1,\ldots,m+1\} : \varepsilon_i = \pm 1\}$ and $T^\pm = \{j \in \{1,\ldots,n+1\} : \delta_j = \pm 1\}$. Then

$$\sum_{i=1}^{m+1}\sum_{j=1}^{n+1} b_{ij}\varepsilon_i\delta_j = \sum_{\substack{i\in S^+\\ j\in T^+}} b_{ij} + \sum_{\substack{i\in S^-\\ j\in T^-}} b_{ij} - \sum_{\substack{i\in S^+\\ j\in T^-}} b_{ij} - \sum_{\substack{i\in S^-\\ j\in T^+}} b_{ij} \le 4\|B\|_{\mathrm{cut}}. \tag{2.8}$$

This shows that $\|B\|_{\infty\to 1} \le 4\|B\|_{\mathrm{cut}}$ (for any matrix B, actually, not just the specific choice in (2.4); we will use this observation later, in Section 2.1.3). In the

reverse direction, given $S \subseteq \{1,\dots,m+1\}$ and $T \subseteq \{1,\dots,n+1\}$, define for $i \in \{1,\dots,m+1\}$ and $j \in \{1,\dots,n+1\}$,

$$\varepsilon_i = \begin{cases} 1 & \text{if } i \in S, \\ -1 & \text{if } i \notin S, \end{cases} \quad \text{and} \quad \delta_j = \begin{cases} 1 & \text{if } j \in T, \\ -1 & \text{if } j \notin T. \end{cases}$$

Then, since the sum of each row and each column of B vanishes,

$$\sum_{\substack{i \in S \\ j \in T}} b_{ij} = \sum_{i=1}^{m+1}\sum_{j=1}^{n+1} b_{ij}\frac{1+\varepsilon_i}{2}\cdot\frac{1+\delta_j}{2} = \frac{1}{4}\sum_{i=1}^{m+1}\sum_{j=1}^{n+1} b_{ij}\varepsilon_i\delta_j \le \frac{1}{4}\|B\|_{\infty\to 1}.$$

This completes the proof of (2.6). We summarize the above simple transformations in the following lemma:

LEMMA 2.1. *Let $A = (a_{ij})$ be an $m \times n$ matrix with real entries, and let $B = (b_{ij})$ be the $(m+1)\times(n+1)$ matrix given in* (2.4)*. Then*

$$\|A\|_{\text{cut}} = \frac{1}{4}\|B\|_{\infty\to 1}.$$

A consequence of Lemma 2.1 is that the problem of approximating $\|A\|_{\text{cut}}$ in polynomial time is equivalent to the problem of approximating $\|A\|_{\infty\to 1}$ in polynomial time in the sense that any algorithm for one of these problems can be used to obtain an algorithm for the other problem with the same running time (up to constant factors) and the same (multiplicative) approximation guarantee.

Given an $m \times n$ matrix $A = (a_{ij})$, consider the following quantity:

$$\text{SDP}(A) = \max\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\langle x_i, y_j\rangle : \{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq S^{n+m-1}\right\}. \tag{2.9}$$

The maximization problem in (2.9) falls into the framework of semidefinite programming as discussed in Section 1.2. Therefore $\text{SDP}(A)$ can be computed in polynomial time with arbitrarily good precision. It is clear that $\text{SDP}(A) \ge \|A\|_{\infty\to 1}$, because the maximum in (2.9) is over a bigger set than the maximum in (2.7). The Grothendieck inequality says that $\text{SDP}(A) \le K_G\|A\|_{\infty\to 1}$, so we have

$$\|A\|_{\infty\to 1} \le \text{SDP}(A) \le K_G\|A\|_{\infty\to 1}.$$

Thus, the polynomial-time algorithm that outputs the number $\text{SDP}(A)$ is guaranteed to be within a factor of K_G of $\|A\|_{\infty\to 1}$. By Lemma 2.1, the algorithm that outputs the number $\alpha = \frac{1}{4}\text{SDP}(B)$, where the matrix B is as in (2.4), satisfies (2.2) with $C = K_G$.

Section 7 is devoted to algorithmic impossibility results. But it is worthwhile to make at this juncture two comments regarding hardness of approximation. First of all, unless P = NP, we need to introduce an error $C > 1$ in our requirement (2.2). This was observed in [8]: the classical MAXCUT problem from algorithmic graph theory was shown in [8] to be a special case of the problem of computing $\|A\|_{\text{cut}}$, and therefore by [53] we know that unless P = NP there does

not exist a polynomial-time algorithm that outputs a number α satisfying (2.2) with C strictly smaller than $\frac{17}{16}$. In fact, by a reduction to the MAX DICUT problem, one can show that C must be at least $\frac{13}{12}$ unless P = NP; we refer to Section 7 and [8] for more information on this topic.

Another (more striking) algorithmic impossibility result is based on the unique games conjecture (UGC). Clearly the above algorithm cannot yield an approximation guarantee strictly smaller than K_G (this is the definition of K_G). In fact, it was shown in [106] that unless the UGC is false, for every $\varepsilon \in (0,1)$ any polynomial-time algorithm for estimating $\|A\|_{\mathrm{cut}}$ whatsoever, and not only the specific algorithm described above, must make an error of at least $K_G - \varepsilon$ on some input matrix A. Thus, if we assume the UGC, then the classical Grothendieck constant has a complexity theoretic interpretation: it equals the best approximation ratio of polynomial-time algorithms for the cut norm problem. Note that [106] manages to prove this statement despite the fact that the value of K_G is unknown.

We have thus far ignored the issue of finding in polynomial time the subsets S_0 and T_0 satisfying (2.3); i.e., we only explained how the Grothendieck inequality can be used for polynomial-time estimation of the quantity $\|A\|_{\mathrm{cut}}$ without actually finding efficiently subsets at which $\|A\|_{\mathrm{cut}}$ is approximately attained. In order to do this, we cannot use the Grothendieck inequality as a black box: we need to look into its proof and argue that it yields a polynomial-time procedure that converts vectors $\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq S^{n+m-1}$ into signs $\{\varepsilon_i\}_{i=1}^m, \{\delta_j\}_{j=1}^n \subseteq \{-1,1\}$ (this is known as a rounding procedure). It is indeed possible to do so, as explained in Section 2.2. We postpone the explanation of the rounding procedure that hides behind the Grothendieck inequality in order to first give examples why one might want to efficiently compute the cut norm of a matrix.

2.1.1 Szemerédi Partitions

The Szemerédi regularity lemma [113] (see also [74]) is a general and very useful structure theorem for graphs, asserting (informally) that any graph can be partitioned into a controlled number of pieces that interact with each other in a pseudorandom way. The Grothendieck inequality, via the cut norm estimation algorithm, yields a polynomial-time algorithm that, when given a graph $G = (V,E)$ as input, outputs a partition of V that satisfies the conclusion of the Szemerédi regularity lemma.

To make the above statements formal, we need to recall some definitions. Let $G = (V,E)$ be a graph. For every disjoint $X, Y \subseteq V$, denote the number of edges joining X and Y by $e(X,Y) = |\{(u,v) \in X \times Y : \{u,v\} \in E\}|$. Let $X, Y \subseteq V$ be disjoint and nonempty, and fix $\varepsilon, \delta \in (0,1)$. The pair of vertex sets (X,Y) is called (ε,δ)-regular if for every $S \subseteq X$ and $T \subseteq Y$ that are not too small, the quantity $\frac{e(S,T)}{|S|\cdot|T|}$ (the density of edges between S and T) is essentially independent of the pair (S,T) itself. Formally, we require that for every $S \subseteq X$ with $|S| \geq \delta|X|$ and

every $T \subseteq Y$ with $|T| \geq \delta|Y|$ we have

$$\left| \frac{e(S,T)}{|S| \cdot |T|} - \frac{e(X,Y)}{|X| \cdot |Y|} \right| \leq \varepsilon. \tag{2.10}$$

The almost uniformity of the numbers $\frac{e(S,T)}{|S| \cdot |T|}$ as exhibited in (2.10) says that the pair (X, Y) is "pseudorandom," i.e., it is similar to a random bipartite graph where each $(x, y) \in X \times Y$ is joined by an edge independently with probability $\frac{e(X,Y)}{|X| \cdot |Y|}$.

The Szemerédi regularity lemma says that for all $\varepsilon, \delta, \eta \in (0, 1)$ and $k \in \mathbb{N}$ there exists $K = K(\varepsilon, \delta, \eta, k) \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ any n-vertex graph $G = (V, E)$ can be partitioned into m-sets $S_1, \ldots, S_m \subseteq V$ with the following properties:

- $k \leq m \leq K$,
- $|S_i| - |S_j| \leq 1$ for all $i, j \in \{1, \ldots, m\}$,
- the number of $i, j \in \{1, \ldots, m\}$ with $i < j$ such that the pair (S_i, S_j) is (ε, δ)-regular is at least $(1 - \eta)\binom{m}{2}$.

Thus every graph is almost a superposition of a bounded number of pseudorandom graphs, the key point being that K is independent of n and the specific combinatorial structure of the graph in question.

It would be of interest to have a way to produce a Szemerédi partition in polynomial time with K independent of n (this is a good example of an approximation algorithm: one might care to find such a partition into the minimum possible number of pieces, but producing any partition into boundedly many pieces is already a significant achievement). Such a polynomial-time algorithm was designed in [5] (see also [75]). We refer to [5, 75] for applications of algorithms for constructing Szemerédi partitions, and to [5] for a discussion of the computational complexity of this algorithmic task. We shall now explain how the Grothendieck inequality yields a different approach to this problem, which has some advantages over [5, 75] that will be described later. The argument below is due to [8].

Assume that X and Y are disjoint n-point subsets of a graph $G = (V, E)$. How can we determine in polynomial time whether the pair (X, Y) is close to being (ε, δ)-regular? It turns out that this is the main bottleneck in constructing Szemerédi partitions in polynomial time. To this end consider the following $n \times n$ matrix $A = (a_{xy})_{(x,y) \in X \times Y}$:

$$a_{xy} = \begin{cases} 1 - \frac{e(X,Y)}{|X| \cdot |Y|} & \text{if } \{x, y\} \in E, \\ -\frac{e(X,Y)}{|X| \cdot |Y|} & \text{if } \{x, y\} \notin E. \end{cases} \tag{2.11}$$

By the definition of A, if $S \subseteq X$ and $T \subseteq Y$, then

$$\left| \sum_{\substack{x \in S \\ y \in T}} a_{xy} \right| = |S| \cdot |T| \cdot \left| \frac{e(S,T)}{|S| \cdot |T|} - \frac{e(X,Y)}{|X| \cdot |Y|} \right|. \tag{2.12}$$

Hence if (X, Y) is not (ε, δ)-regular, then $\|A\|_{\text{cut}} \geq \varepsilon\delta^2 n^2$. The approximate cut norm algorithm based on the Grothendieck inequality, together with the rounding procedure in Section 2.2, finds in polynomial time subsets $S \subseteq X$ and $T \subseteq Y$ such that

$$\min\left\{n|S|, n|T|, n^2\left|\frac{e(S,T)}{|S|\cdot|T|} - \frac{e(X,Y)}{|X|\cdot|Y|}\right|\right\} \overset{(2.12)}{\geq} \left|\sum_{\substack{x\in S\\ y\in T}} a_{xy}\right| \geq \frac{1}{K_G}\varepsilon\delta^2 n^2 \geq \frac{1}{2}\,\varepsilon\delta^2 n^2.$$

This establishes the following lemma:

LEMMA 2.2. *There exists a polynomial-time algorithm that takes as input two disjoint n-point subsets X and Y of a graph and either decides that (X, Y) is (ε, δ)-regular or finds $S \subseteq X$ and $T \subseteq Y$ with*

$$|S|, |T| \geq \frac{1}{2}\,\varepsilon\delta^2 n \quad \text{and} \quad \left|\frac{e(S,T)}{|S|\cdot|T|} - \frac{e(X,Y)}{|X|\cdot|Y|}\right| \geq \frac{1}{2}\,\varepsilon\delta^2.$$

From Lemma 2.2 it is quite simple to design a polynomial algorithm that constructs a Szemerédi partition with bounded cardinality; compare Lemma 2.2 to corollary 3.3 in [5] and theorem 1.5 in [75]. We will not explain this deduction here since it is identical to the argument in [5]. We note that the quantitative bounds in Lemma 2.2 improve over the corresponding bounds in [5, 75], yielding, say, when $\varepsilon = \delta = \eta$, an algorithm with the best-known bound on K as a function of ε (this bound is nevertheless still huge, as must be the case due to [46]; see also [32]). See [8] for a precise statement of these bounds. In addition, the algorithms of [5, 75] worked only in the "dense case," i.e., when $\|A\|_{\text{cut}}$, for A as in (2.11) is of order n^2, while the above algorithm does not have this requirement. This observation can be used to design the only known polynomial-time algorithm for sparse versions of the Szemerédi regularity lemma [4] (see also [43]). We will not discuss the sparse version of the regularity lemma here and refer instead to [73, 74] for a discussion of this topic. We also refer to [4] for additional applications of the Grothendieck inequality in sparse settings.

2.1.2 Frieze-Kannan Matrix Decomposition

The cut norm estimation problem was originally raised in the work of Frieze and Kannan [40], which introduced a method to design polynomial-time approximation schemes for dense constraint satisfaction problems. The key tool for this purpose is a decomposition theorem for matrices that we now describe.

An $m \times n$ matrix $D = (d_{ij})$ is called a cut matrix if there exist subsets $S \subseteq \{1, \ldots, m\}$ and $T \subseteq \{1, \ldots, n\}$, and $d \in \mathbb{R}$ such that for all $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$ we have

$$d_{ij} = \begin{cases} d & \text{if } (i,j) \in S \times T, \\ 0 & \text{if } (i,j) \notin S \times T. \end{cases} \tag{2.13}$$

Denote the matrix D defined in (2.13) by $\mathrm{CUT}(S, T, d)$. In [40] it is proved that for every $\varepsilon > 0$ there exists an integer $s = O(1/\varepsilon^2)$ such that for any $m \times n$ matrix $A = (a_{ij})$ with entries bounded in absolute value by 1, there are cut matrices $D_1, \ldots, D_s$ satisfying

$$\Big\| A - \sum_{k=1}^{s} D_k \Big\|_{\mathrm{cut}} \le \varepsilon mn. \tag{2.14}$$

Moreover, these cut matrices $D_1, \ldots, D_s$ can be found in time $C(\varepsilon)(mn)^{O(1)}$. We shall now explain how this is done using the cut norm approximation algorithm of Section 2.1.

The argument is iterative. Set $A_0 = A$ and, assuming that the cut matrices $D_1, \ldots, D_r$ have already been defined, write $A_r = (a_{ij}(r)) = A - \sum_{k=1}^{r} D_k$. We are done if $\|A_r\|_{\mathrm{cut}} \le \varepsilon mn$, so we may assume that $\|A_r\|_{\mathrm{cut}} > \varepsilon mn$. By the cut norm approximation algorithm, we can find in polynomial time $S \subseteq \{1, \ldots, m\}$ and $T \subseteq \{1, \ldots, n\}$ satisfying

$$\Big| \sum_{\substack{i \in S \\ j \in T}} a_{ij}(r) \Big| \ge c \|A_r\|_{\mathrm{cut}} \ge c\varepsilon mn, \tag{2.15}$$

where $c > 0$ is a universal constant. Set

$$d = \frac{1}{|S| \cdot |T|} \sum_{\substack{i \in S \\ j \in T}} a_{ij}(r).$$

Define $D_{r+1} = \mathrm{CUT}(S, T, d)$ and $A_{r+1} = (a_{ij}(r+1)) = A_r - D_{r+1}$. Then by expanding the squares we have

$$\begin{aligned} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}(r+1)^2 &= \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}(r)^2 - \frac{1}{|S| \cdot |T|} \Big(\sum_{\substack{i \in S \\ j \in T}} a_{ij}(r) \Big)^2 \\ &\overset{(2.15)}{\le} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}(r)^2 - c^2 \varepsilon^2 mn. \end{aligned}$$

It follows inductively that if we can carry out this procedure r times then

$$0 \le \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}(r)^2 \le \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 - rc^2\varepsilon^2 mn \le mn - rc^2\varepsilon^2 mn,$$

where we used the assumption that $|a_{ij}| \le 1$. Therefore the above iteration must terminate after $\lceil 1/(c^2\varepsilon^2) \rceil$ steps, yielding (2.14). We note that the bound $s = O(1/\varepsilon^2)$ in (2.14) cannot be improved [6]; see also [32, 91] for related lower bounds.

The key step in the above algorithm was finding sets S and T as in (2.15). In [40] an algorithm was designed that, given an $m \times n$ matrix $A = (a_{ij})$ and

$\varepsilon > 0$ as input, produces in time $2^{1/\varepsilon^{O(1)}}(mn)^{O(1)}$ subsets $S \subseteq \{1,\ldots,m\}$ and $T \subseteq \{1,\ldots,n\}$ satisfying

$$\left|\sum_{\substack{i\in S\\ j\in T}} a_{ij}\right| \geq \|A\|_{\text{cut}} - \varepsilon mn. \tag{2.16}$$

The additive approximation guarantee in (2.16) implies (2.15) only if $\|A\|_{\text{cut}} \geq \varepsilon(c+1)mn$, and similarly the running time is not polynomial if, say, $\varepsilon = n^{-\Omega(1)}$. Thus the Kannan-Frieze method is relevant only to "dense" instances, while the cut norm algorithm based on the Grothendieck inequality applies equally well for all values of $\|A\|_{\text{cut}}$. This fact, combined with more work (and, necessarily, additional assumptions on the matrix A), was used in [31] to obtain a sparse version of (2.14): with εmn in the right-hand side of (2.14) replaced by $\varepsilon\|A\|_{\text{cut}}$ and $s = O(1/\varepsilon^2)$ (importantly, here s is independent of m and n).

We have indicated above how the cut norm approximation problem is relevant to Kannan-Frieze matrix decompositions, but we did not indicate the uses of such decompositions since this is beyond the scope of the current survey. We refer the reader to [6, 16, 31, 40] for a variety of applications of this methodology to combinatorial optimization problems.

2.1.3 Maximum Acyclic Subgraph

In the maximum acyclic subgraph problem we are given as input an n-vertex directed graph $G = (\{1,\ldots,n\}, E)$. Thus E consists of a family of *ordered* pairs of distinct elements in $\{1,\ldots,n\}$. We are interested in the maximum of

$$\begin{aligned}&|\{(i,j) \in \{1,\ldots,n\}^2 : \sigma(i) < \sigma(j)\} \cap E|\\ &\quad - |\{(i,j) \in \{1,\ldots,n\}^2 : \sigma(i) > \sigma(j)\} \cap E|\end{aligned}$$

over all possible permutations $\sigma \in S_n$ (S_n denotes the group of permutations of $\{1,\ldots,n\}$). In words, the quantity of interest is the maximum over all orderings of the vertices of the number of edges going "forward" minus the number of edges going "backward." Note that it is trivial to get at least half of the edges to go forward by considering a random permutation, so in essence we are measuring here the advantage of the best possible ordering over a random ordering. The best-known approximation algorithm for this problem was discovered in [28] as an application of the cut norm approximation algorithm.

It is most natural to explain the algorithm of [28] for a weighted version of the maximum acyclic subgraph problem. Let $W : \{1,\ldots,n\} \times \{1,\ldots,n\} \to \mathbb{R}$ be skew-symmetric, i.e., $W(u,v) = -W(v,u)$ for all $u, v \in \{1,\ldots,n\}$. For $\sigma \in S_n$ define

$$W(\sigma) = \sum_{\substack{u,v\in\{1,\ldots,n\}\\ u<v}} W(\sigma(u),\sigma(v)).$$

Thus $W(\sigma)$ is the sum of the entries of W that lie above the diagonal after the rows and columns of W have been permuted according to the permutation σ. We are

interested in the quantity $M_W = \max_{\sigma \in S_n} W(\sigma)$. The case of a directed graph $G = (\{1, \dots, n\}, E)$ described above corresponds to the matrix

$$W(u, v) = \mathbb{1}_{\{(u,v) \in E\}} - \mathbb{1}_{\{(v,u) \in E\}}.$$

THEOREM 2.3 ([28]). *There exists a polynomial-time algorithm that takes as input an $n \times n$ skew-symmetric $W : \{1, \dots, n\} \times \{1, \dots, n\} \to \mathbb{R}$ and outputs a permutation $\sigma \in S_n$ satisfying*[1]

$$W(\sigma) \gtrsim \frac{M_W}{\log n}.$$

PROOF. The proof below is a slight variant of the reasoning of [28]. By the cut norm approximation algorithm one can find in polynomial time two subsets $S, T \subseteq \{1, \dots, n\}$ satisfying

$$\sum_{\substack{u \in S \\ v \in T}} W(u, v) \geq c \|W\|_{\text{cut}}, \tag{2.17}$$

where $c \in (0, \infty)$ is a universal constant. Note that we do not need to take the absolute value of the left-hand side of (2.17) because W is skew-symmetric. Observe also that since W is skew-symmetric we have $\sum_{u,v \in S \cap T} W(u, v) = 0$ and therefore

$$\sum_{\substack{u \in S \\ v \in T}} W(u, v) = \sum_{\substack{u \in S \setminus T \\ v \in T \setminus S}} W(u, v) + \sum_{\substack{u \in S \setminus T \\ v \in S \cap T}} W(u, v) + \sum_{\substack{u \in S \cap T \\ v \in T \setminus S}} W(u, v).$$

By replacing the pair of subsets (S, T) by one of

$$\{(S \setminus T, T \setminus S),\ (S \setminus T, S \cap T),\ (S \cap T, T \setminus S)\}$$

and replacing the constant c in (2.17) by $\frac{c}{3}$, we may assume without loss of generality that (2.17) holds with S and T disjoint. Set $R = \{1, \dots, n\} \setminus (S \cup T)$ and write $S = \{s_1, \dots, s_{|S|}\}$, $T = \{t_1, \dots, t_{|T|}\}$, and $R = \{r_1, \dots, r_{|R|}\}$, where $s_1 < \cdots < s_{|S|}$, $t_1 < \cdots < t_{|T|}$, and $r_1 < \cdots < r_{|R|}$.

Define two permutations $\sigma^1, \sigma^2 \in S_n$ as follows:

$$\sigma^1(u) = \begin{cases} s_u & \text{if } u \in \{1, \dots, |S|\}, \\ t_{u-|S|} & \text{if } u \in \{|S|+1, \dots, |S|+|T|\}, \\ r_{u-|S|-|T|} & \text{if } u \in \{|S|+|T|+1, \dots, n\}, \end{cases}$$

and

$$\sigma^2(u) = \begin{cases} r_{|R|-u+1} & \text{if } u \in \{1, \dots, |R|\}, \\ s_{|R|+|S|-u+1} & \text{if } u \in \{|R|+1, \dots, |R|+|S|\}, \\ t_{n-u+1} & \text{if } u \in \{|R|+|S|+1, \dots, n\}. \end{cases}$$

[1] Here and in what follows, the relations $\gtrsim$ and $\lesssim$ indicate the corresponding inequalities up to an absolute factor. The relation $\asymp$ stands for $\gtrsim \wedge \lesssim$.

In words, σ^1 orders $\{1,\ldots,n\}$ by starting with the elements of S in increasing order, then the elements of T in increasing order, and finally the elements of R in increasing order. At the same time, σ^2 orders $\{1,\ldots,n\}$ by starting with the elements of R in decreasing order, then the elements of S in decreasing order, and finally the elements of T in decreasing order. The quantity $W(\sigma^1)+W(\sigma^2)$ consists of a sum of terms of the form $W(u,v)$ for $u,v\in\{1,\ldots,n\}$, where if $(u,v)\in(S\times S)\cup(T\times T)\cup(R\times\{1,\ldots,n\})$, then both $W(u,v)$ and $W(v,u)$ appear exactly once in this sum, and if $(u,v)\in S\times T$, then $W(u,v)$ appears twice in this sum and $W(v,u)$ does not appear in this sum at all. Therefore, using the fact that W is skew-symmetric we have the following identity:

$$W(\sigma^1)+W(\sigma^2)=2\sum_{\substack{u\in S\\ v\in T}}W(u,v).$$

It follows that for some $\ell\in\{1,2\}$ we have

$$W(\sigma^\ell)\geq\sum_{\substack{u\in S\\ v\in T}}W(u,v)\overset{(2.17)}{\geq}c\|W\|_{\mathrm{cut}}.$$

The output of the algorithm will be the permutation σ^ℓ, so it suffices to prove that

$$\|W\|_{\mathrm{cut}}\gtrsim\frac{M_W}{\log n}. \tag{2.18}$$

We will prove below that

$$\|W\|_{\mathrm{cut}}\gtrsim\frac{1}{\log n}\sum_{\substack{u,v\in\{1,\ldots,n\}\\ u<v}}W(u,v). \tag{2.19}$$

Inequality (2.18) follows by applying (2.19) to $W'(u,v)=W(\sigma(u),\sigma(v))$ for every $\sigma\in S_n$.

To prove (2.19) first note that $\|W\|_{\mathrm{cut}}\geq\frac{1}{4}\|W\|_{\infty\to1}$; we have already proved this inequality as a consequence of the simple identity (2.8). Moreover, we have

$$\|W\|_{\infty\to1}\gtrsim \max\Big\{\sum_{u=1}^n\sum_{v=1}^n W(u,v)\sin(\alpha_u-\beta_v):\{\alpha_u\}_{u=1}^n,\{\beta_v\}_{v=1}^n\subseteq\mathbb{R}\Big\}. \tag{2.20}$$

Inequality (2.20) is a special case of (1.1) with the choice of vectors

$$x_u=(\sin\alpha_u,\cos\alpha_u)\in\mathbb{R}^2\quad\text{and}\quad y_v=(\cos\beta_v,-\sin\beta_v)\in\mathbb{R}^2.$$

We note that this two-dimensional version of the Grothendieck inequality is trivial with the constant in the right-hand side of (2.20) being $\frac{1}{2}$, and it is shown in [80] that the best constant in the right-hand side of (2.20) is actually $1/\sqrt{2}$.

For every choice of $\theta_1, \dots, \theta_n \in \mathbb{R}$, an application of (2.20) when $\alpha_u = \beta_u = \theta_u$ and $\alpha_u = \beta_u = -\theta_u$ yields the inequality

$$\begin{aligned} \|W\|_{\mathrm{cut}} &\gtrsim \Big| \sum_{u=1}^{n} \sum_{v=1}^{n} W(u,v) \sin(\theta_u - \theta_v) \Big| \\ &= 2 \Big| \sum_{\substack{u,v \in \{1,\dots,n\} \\ u<v}} W(u,v) \sin(\theta_u - \theta_v) \Big|, \end{aligned} \tag{2.21}$$

where for the equality in (2.21) we used the fact that W is skew-symmetric. Consequently, for every $k \in \mathbb{N}$ we have

$$\|W\|_{\mathrm{cut}} \gtrsim \left| \sum_{\substack{u,v \in \{1,\dots,n\} \\ u<v}} W(u,v) \sin\left(\frac{\pi(v-u)k}{n} \right) \right|. \tag{2.22}$$

By the standard orthogonality relation for the sine function, for every $u, v \in \{1, \dots, n\}$ such that $u < v$ we have

$$\frac{2}{n} \sum_{k=1}^{n-1} \sum_{\ell=1}^{n-1} \sin\left(\frac{\pi(v-u)k}{n} \right) \sin\left(\frac{\pi k \ell}{n} \right) = 1. \tag{2.23}$$

We refer to the derivation of (2.23) in the appendix of [28]; it can be proved by substituting

$$\sin\left(\frac{\pi(v-u)k}{n} \right) = \frac{e^{i\pi(v-u)k/n} - e^{-i\pi(v-u)k/n}}{2i}$$

and

$$\sin\left(\frac{\pi k \ell}{n} \right) = \frac{e^{i\pi k\ell/n} - e^{-i\pi k\ell/n}}{2i}$$

into the left-hand side of (2.23) and computing the resulting geometric sums explicitly. Now,

$$\begin{aligned} &\sum_{\substack{u,v \in \{1,\dots,n\} \\ u<v}} W(u,v) \\ &\overset{(2.23)}{=} \frac{2}{n} \sum_{\substack{u,v \in \{1,\dots,n\} \\ u<v}} W(u,v) \sum_{k=1}^{n-1} \sum_{\ell=1}^{n-1} \sin\left(\frac{\pi(v-u)k}{n} \right) \sin\left(\frac{\pi k \ell}{n} \right) \\ &\leq \frac{2}{n} \sum_{k=1}^{n-1} \left| \sum_{\ell=1}^{n-1} \sin\left(\frac{\pi k \ell}{n} \right) \right| \cdot \left| \sum_{\substack{u,v \in \{1,\dots,n\} \\ u<v}} W(u,v) \sin\left(\frac{\pi(v-u)k}{n} \right) \right| \\ &\overset{(2.22)}{\lesssim} \frac{\sum_{k=1}^{n-1} \left| \sum_{\ell=1}^{n-1} \sin\left(\frac{\pi k \ell}{n} \right) \right|}{n} \|W\|_{\mathrm{cut}}. \end{aligned}$$

The desired inequality (2.19) will follow from $\sum_{k=1}^{n-1} |\sum_{\ell=1}^{n-1} \sin(\pi k\ell/n)| \lesssim n \log n$. To establish this estimate, note that by writing

$$\sin\left(\frac{\pi k\ell}{n}\right) = \frac{e^{i\pi k\ell/n} - e^{-i\pi k\ell/n}}{2i}$$

and computing geometric sums explicitly, one sees that $\sum_{\ell=1}^{n-1} \sin(\pi k\ell/n) = 0$ if k is even and $\sum_{\ell=1}^{n-1} \sin(\pi k\ell/n) = \cot(\pi k/(2n))$ if k is odd (see the appendix of [28] for the details of this computation). Hence, since $\cot(\theta) < \frac{1}{\theta}$ for every $\theta \in (0, \pi/2)$, we have

$$\sum_{k=1}^{n-1}\left|\sum_{\ell=1}^{n-1} \sin\left(\frac{\pi k\ell}{n}\right)\right| = \sum_{j=0}^{\lfloor \frac{n}{2}-1 \rfloor} \cot\left(\frac{\pi(2j+1)}{2n}\right) \le \frac{2n}{\pi} \sum_{j=0}^{\lfloor \frac{n}{2}-1 \rfloor} \frac{1}{2j+1} \lesssim n \log n.$$

□

2.1.4 Linear Equations Modulo 2

Consider a system $\mathcal{E}$ of N linear equations modulo 2 in n Boolean variables $z_1, \ldots, z_n$ such that in each equation only three distinct variables appear. Define $\mathrm{MAXSAT}(\mathcal{E})$ to be the maximum number of equations in $\mathcal{E}$ that can be satisfied simultaneously. A random $\{0,1\}$ assignment of these variables satisfies in expectation $N/2$ equations, so it is natural to ask for a polynomial-time approximation algorithm to the quantity $\mathrm{MAXSAT}(\mathcal{E}) - N/2$. We describe below the best-known [67] approximation algorithm for this problem, which uses the Grothendieck inequality in a crucial way. The approximation guarantee thus obtained is $O(\sqrt{n/\log n})$. While this allows for a large error, it is shown in [54] that for every $\varepsilon \in (0,1)$ if there were a polynomial-time algorithm that approximates $\mathrm{MAXSAT}(\mathcal{E}) - N/2$ to within a factor of $2^{(\log n)^{1-\varepsilon}}$ in time $2^{(\log n)^{O(1)}}$, then there would be an algorithm for 3-colorability that runs in time $2^{(\log n)^{O(1)}}$, a conclusion that is widely believed to be impossible.

Let $\mathcal{E}$ be a system of linear equations as described above. Write $a_{ijk} = 1$ if the equation $z_i + z_j + z_k = 0$ is in the system $\mathcal{E}$. Similarly, write $a_{ijk} = -1$ if the equation $z_i + z_j + z_k = 1$ is in $\mathcal{E}$. Finally, write $a_{ijk} = 0$ if no equation in $\mathcal{E}$ corresponds to $z_i + z_j + z_k$. Assume that the assignment $(z_1, \ldots, z_n) \in \{0,1\}^n$ satisfies m of the equations in $\mathcal{E}$. Then

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk}(-1)^{z_i+z_j+z_k} = m - (N-m) = 2\left(m - \frac{N}{2}\right).$$

It follows that

$$\max\left\{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk}\varepsilon_i\varepsilon_j\varepsilon_k : \{\varepsilon_i\}_{i=1}^n \subseteq \{-1,1\}\right\} = 2\left(\mathrm{MAXSAT}(\mathcal{E}) - \frac{N}{2}\right) \stackrel{\text{def}}{=} M. \tag{2.24}$$

We will now present a randomized polynomial algorithm that outputs a number $\alpha \in \mathbb{R}$ that satisfies with probability at least $\frac{2}{3}$

$$\frac{1}{20K_G}\sqrt{\frac{\log n}{n}}\,M \le \alpha \le M. \tag{2.25}$$

Fix $m \in \mathbb{N}$, which will be determined later. Choose $\varepsilon^1, \dots, \varepsilon^m \in \{-1,1\}^n$ independently and uniformly at random and consider the following random variable:

$$\alpha = \frac{1}{10K_G}\max_{\ell\in\{1,\dots,m\}}\max\Big\{\sum_{i=1}^n\sum_{j=1}^n\sum_{k=1}^n a_{ijk}\varepsilon_i^\ell\langle y_j, z_k\rangle : \{y_j\}_{j=1}^n, \{z_k\}_{k=1}^n \subseteq S^{2n-1}\Big\}. \tag{2.26}$$

By the Grothendieck inequality we know that

$$\alpha \le \frac{1}{10}\max\Big\{\sum_{i=1}^n\sum_{j=1}^n\sum_{k=1}^n a_{ijk}\varepsilon_i\delta_j\zeta_k : \{\varepsilon_i\}_{i=1}^n, \{\delta_j\}_{j=1}^n, \{\zeta_k\}_{k=1}^n \subseteq \{-1,1\}\Big\} \le M. \tag{2.27}$$

The final step in (2.27) follows from an elementary decoupling argument; see [67, lemma 2.1].

We claim that

$$\Pr\left[\alpha \ge \frac{1}{20K_G}\sqrt{\frac{\log n}{n}}M\right] \ge 1 - e^{-cm/\sqrt[4]{n}}. \tag{2.28}$$

Once (2.28) is established, it would follow that for $m \asymp \sqrt[4]{n}$ we have

$$\alpha \ge \frac{1}{20K_G}\sqrt{\frac{\log n}{n}}\,M$$

with probability at least $\frac{2}{3}$. This combined with (2.27) would complete the proof of (2.25) since α as defined in (2.26) can be computed in polynomial time, being the maximum of $O(\sqrt[4]{n})$ semidefinite programs.

To check (2.28), let $\|\cdot\|$ be the norm on $\mathbb{R}^n$ defined for every $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ by

$$\|x\| = \max\Big\{\sum_{i=1}^n\sum_{j=1}^n\sum_{k=1}^n a_{ijk}x_i\langle y_j, z_k\rangle : \{y_j\}_{j=1}^n, \{z_k\}_{k=1}^n \subseteq S^{2n-1}\Big\}.$$

Define $K = \{x \in \mathbb{R}^n : \|x\| \le 1\}$ and let $K^\circ = \{w \in \mathbb{R}^n : \sup_{x\in K}\langle x, w\rangle \le 1\}$ be the polar of K. Then $\max\{\|w\|_1 : w \in K^\circ\} = \max\{\|x\| : \|x\|_\infty \le 1\} \ge M$, where the first equality is straightforward duality and the final inequality is a

consequence of the definition of $\|\cdot\|$ and M. It follows that there exists $w \in K^\circ$ with $\|w\|_1 \geq M$. Hence, recalling that $\alpha = \frac{1}{10K_G}\max_{\ell\in\{1,\ldots,m\}}\|\varepsilon^\ell\|$, we have

$$\Pr\left[\alpha \geq \frac{1}{20K_G}\sqrt{\frac{\log n}{n}}\,M\right] \overset{(2.26)}{=} 1 - \prod_{\ell=1}^{m}\Pr\left[\|\varepsilon^\ell\| < \frac{1}{2}\sqrt{\frac{\log n}{n}}\,M\right]$$
$$\geq 1 - \left(\Pr\left[\sum_{i=1}^{n}\varepsilon_i^1 w_i < \frac{1}{2}\sqrt{\frac{\log n}{n}}\sum_{i=1}^{n}|w_i|\right]\right)^m.$$

In order to prove (2.28) it therefore suffices to prove that if ε is chosen uniformly at random from $\{-1,1\}^n$ and $a \in \mathbb{R}^n$ satisfies $\|a\|_1 = 1$, then

$$\Pr\left[\sum_{i=1}^{n}\varepsilon_i a_i \geq \sqrt{\frac{\log n}{4n}}\right] \geq 1 - \frac{c}{\sqrt[4]{n}},$$

where $c \in (0,\infty)$ is a universal constant. This probabilistic estimate for i.i.d. Bernoulli sums can be proved directly; see [67, lemma 3.2].

2.2 Rounding

Let $A = (a_{ij})$ be an $m \times n$ matrix. In Section 2.1 we described a polynomial-time algorithm for approximating $\|A\|_{\mathrm{cut}}$ and $\|A\|_{\infty\to 1}$. For applications it is also important to find in polynomial time signs $\varepsilon_1,\ldots,\varepsilon_m,\delta_1,\ldots,\delta_n \in \{-1,1\}$ for which $\sum_{i=1}^m\sum_{j=1}^n a_{ij}\varepsilon_i\delta_j$ is at least a constant multiple of $\|A\|_{\infty\to 1}$. This amounts to a "rounding problem": we need to find a procedure that, given input vectors $x_1,\ldots,x_m,y_1,\ldots,y_n \in \mathbb{S}^{m+n-1}$, produces signs $\varepsilon_1,\ldots,\varepsilon_m,\delta_1,\ldots,\delta_n \in \{-1,1\}$ whose existence is ensured by the Grothendieck inequality, namely, $\sum_{i=1}^m\sum_{j=1}^n a_{ij}\varepsilon_i\delta_j$, is at least a constant multiple of $\sum_{i=1}^m\sum_{j=1}^n a_{ij}\langle x_i,y_j\rangle$.

For this purpose one needs to examine proofs of the Grothendieck inequality, as done in [8]. We will now describe the rounding procedure that gives the best-known approximation guarantee. This procedure yields a randomized algorithm that produces the desired signs; it is also possible to obtain a deterministic algorithm, as explained in [8].

The argument below is based on a clever two-step rounding method due to Krivine [79]. Fix $k \in \mathbb{N}$ and assume that we are given two centrally symmetric measurable partitions of $\mathbb{R}^k$, or, equivalently, two odd measurable functions $f,g : \mathbb{R}^k \to \{-1,1\}$. Let $G_1,G_2 \in \mathbb{R}^k$ be independent random vectors that are distributed according to the standard Gaussian measure on $\mathbb{R}^k$, i.e., the measure with density $x \mapsto e^{-\|x\|_2^2/2}/(2\pi)^{k/2}$. For $t \in (-1,1)$ define

$$H_{f,g}(t) \overset{\mathrm{def}}{=} \mathbb{E}\left[f\left(\frac{1}{\sqrt{2}}G_1\right)g\left(\frac{t}{\sqrt{2}}G_1 + \frac{\sqrt{1-t^2}}{\sqrt{2}}G_2\right)\right] = \tag{2.29}$$
$$\frac{1}{\pi^k(1-t^2)^{k/2}}\int_{\mathbb{R}^k}\int_{\mathbb{R}^k} f(x)g(y)\exp\left(\frac{-\|x\|_2^2 - \|y\|_2^2 + 2t\langle x,y\rangle}{1-t^2}\right)dx\,dy.$$

Then $H_{f,g}$ extends to an analytic function on the strip $\{z \in \mathbb{C} : \Re(z) \in (-1,1)\}$. The pair of functions $\{f,g\}$ is called a Krivine rounding scheme if $H_{f,g}$ is invertible on a neighborhood of the origin, and if we consider the Taylor expansion $H_{f,g}^{-1}(z) = \sum_{j=0}^{\infty} a_{2j+1} z^{2j+1}$, then there exists $c = c(f,g) \in (0,\infty)$ satisfying $\sum_{j=0}^{\infty} |a_{2j+1}| c^{2j+1} = 1$.

For (f,g) as above and unit vectors $\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq \mathbb{S}^{m+n-1}$, one can find new unit vectors $\{u_i\}_{i=1}^m, \{v_j\}_{j=1}^n \subseteq \mathbb{S}^{m+n-1}$ satisfying the identities

$$\forall (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}, \quad \langle u_i, v_j \rangle = H_{f,g}^{-1}(c(f,g)\langle x_i, y_j \rangle). \tag{2.30}$$

We refer to [23] for the proof that $\{u_i\}_{i=1}^m$ and $\{v_j\}_{j=1}^n$ exist. This existence proof is not via an efficient algorithm, but as explained in [8]; once we know that they exist the new vectors can be computed efficiently provided $H_{f,g}^{-1}$ can be computed efficiently; this simply amounts to computing a Cholesky decomposition or, alternatively, solving a semidefinite program corresponding to (2.30). This completes the first (preprocessing) step of a generalized Krivine rounding procedure. The next step is to apply a random projection to the new vectors thus obtained, as in Grothendieck's original proof [47] or the Goemans-Williamson algorithm [44].

Let $G : \mathbb{R}^{m+n} \to \mathbb{R}^k$ be a random $k \times (m+n)$ matrix whose entries are i.i.d. standard Gaussian random variables. Define random signs $\{\varepsilon_i\}_{i=1}^m$ and $\{\delta_j\}_{j=1}^n \subseteq \{-1,1\}$ by

$$\forall (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}, \quad \varepsilon_i \stackrel{\text{def}}{=} f\left(\frac{1}{\sqrt{2}} G u_i\right) \quad \text{and} \quad \delta_j \stackrel{\text{def}}{=} g\left(\frac{1}{\sqrt{2}} G v_j\right). \tag{2.31}$$

Now,

$$\begin{aligned} \mathbb{E}\left[\sum_{i=1}^m \sum_{j=1}^n a_{ij} \varepsilon_i \delta_j\right] &\stackrel{(*)}{=} \mathbb{E}\left[\sum_{i=1}^m \sum_{j=1}^n a_{ij} H_{f,g}\Big(\langle u_i, v_j \rangle\Big)\right] \\ &\stackrel{(2.30)}{=} c(f,g) \sum_{i=1}^m \sum_{j=1}^n a_{ij} \langle x_i, y_j \rangle, \end{aligned} \tag{2.32}$$

where $(*)$ follows by rotation invariance from (2.31) and (2.29). The identity (2.32) yields the desired polynomial-time randomized rounding algorithm, provided one can bound $c(f,g)$ from below. Identity (2.32) also gives a systematic way to bound the Grothendieck constant from above: for every Krivine rounding scheme $f, g : \mathbb{R}^k \to \{-1,1\}$ we have $K_G \le 1/c(f,g)$. Krivine used this reasoning to obtain the bound $K_G \le \pi/(2\log(1+\sqrt{2}))$ by considering the case $k = 1$ and $f_0(x) = g_0(x) = \operatorname{sign}(x)$. With $H_{f_0,g_0}(t) = \frac{2}{\pi} \arcsin(t)$ (Grothendieck's identity) and $c(f_0,g_0) = \frac{2}{\pi} \log(1+\sqrt{2})$, one checks that $\{f_0, g_0\}$ is a Krivine rounding scheme.

Since the goal of the above discussion is to round vectors $\{x_i\}_{i=1}^m$ and $\{y_j\}_{j=1}^n \subseteq \mathbb{S}^{m+n-1}$ to signs $\{\varepsilon_i\}_{i=1}^m, \{\delta_j\}_{j=1}^n \subseteq \{-1,1\}$, it seems natural to expect that the

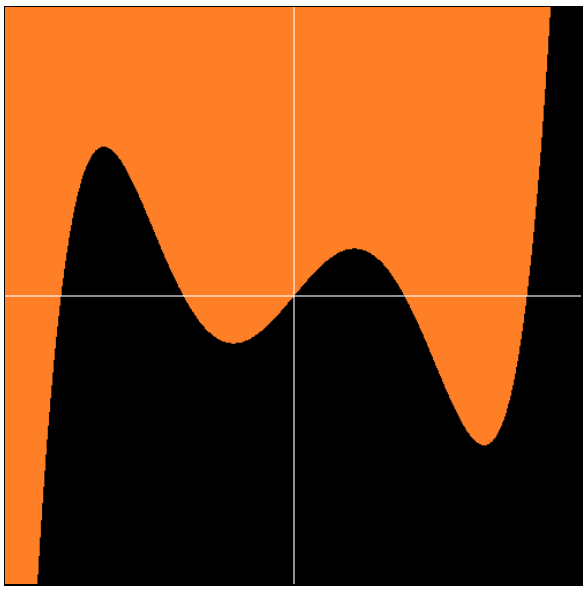

FIGURE 2.1. The partition of $\mathbb{R}^2$ used in [23] to show that K_G is smaller than Krivine's bound; the shaded regions are separated by the graph $y = c(x^5 - 10x^3 + 15x)$.

best possible Krivine rounding scheme occurs when $k = 1$ and $f(x) = g(x) = \text{sign}(x)$. If true, this would imply that $K_G = \pi/(2\log(1+\sqrt{2}))$; a longstanding conjecture of Krivine [79]. Over the years additional evidence supporting Krivine's conjecture was discovered, and a natural analytic conjecture was made in [78] as a step towards proving it. We will not discuss these topics here since in [23] it was shown that actually $K_G \leq \pi/(2\log(1+\sqrt{2})) - \varepsilon_0$ for some effective constant $\varepsilon_0 > 0$.

It is known [23, lemma 2.4] that among all *one-dimensional* Krivine rounding schemes $f, g : \mathbb{R} \to \{-1, 1\}$, we indeed have $c(f, g) \leq \frac{2}{\pi}\log(1+\sqrt{2})$; i.e., it does not pay off to take partitions of $\mathbb{R}$ that are more complicated than the half-line partitions. Unexpectedly, it was shown in [23] that a certain two-dimensional Krivine rounding scheme $f, g : \mathbb{R}^2 \to \{-1, 1\}$ satisfies $c(f, g) > \frac{2}{\pi}\log(1+\sqrt{2})$. The proof of [23] uses a Krivine rounding scheme $f, g : \mathbb{R}^2 \to \{-1, 1\}$ when $f = g$ corresponds to the partition of $\mathbb{R}^2$ as the subgraph and supergraph of the polynomial $y = c(x^5 - 10x^3 + 15x)$, where $c > 0$ is an appropriately chosen constant. This partition is depicted in Figure 2.1.

As explained in [23, sec. 3], there is a natural guess for the "best" two-dimensional Krivine rounding scheme based on a certain numerical computation, which we will not discuss here. For this (conjectural) scheme we have $f \neq g$, and the planar partition corresponding to f is depicted in Figure 2.2. Of course, once Krivine's conjecture has been disproved and the usefulness of higher-dimensional

FIGURE 2.2. The "tiger partition" restricted to the square $[-20, 20]^2$. This is the conjectured [23] optimal partition of $\mathbb{R}^2$ for the purpose of Krivine-type rounding.

rounding schemes has been established, there is no reason to expect that the situation won't improve as we consider k-dimensional Krivine rounding schemes for $k \geq 3$. A positive solution to an analytic question presented in [23] might even lead to an exact computation of K_G; see [23, sec. 3] for the details.

3 Grothendieck Constant of a Graph

Fix $n \in \mathbb{N}$ and let $G = (\{1, \ldots, n\}, E)$ be a graph. We assume throughout that G does not contain any self-loops, i.e., $E \subseteq \{S \subseteq \{1, \ldots, n\} : |S| = 2\}$. Following [7], define the Grothendieck constant of G, denoted $K(G)$, to be the smallest constant $K \in (0, \infty)$ such that every $n \times n$ matrix (a_{ij}) satisfies

$$\max_{x_1,\ldots,x_n \in \mathbb{S}^{n-1}} \sum_{\substack{i,j \in \{1,\ldots,n\} \\ \{i,j\} \in E}} a_{ij} \langle x_i, x_j \rangle \leq K \max_{\varepsilon_1,\ldots,\varepsilon_n \in \{-1,1\}} \sum_{\substack{i,j \in \{1,\ldots,n\} \\ \{i,j\} \in E}} a_{ij} \varepsilon_i \varepsilon_j. \tag{3.1}$$

Inequality (3.1) is an extension of the Grothendieck inequality since (1.1) is the special case of (3.1) when G is a bipartite graph. Thus

$$K_G = \sup_{n \in \mathbb{N}} \{K(G) : G \text{ is an } n\text{-vertex bipartite graph}\}. \tag{3.2}$$

The opposite extreme of bipartite graphs is $G = K_n$, the n-vertex complete graph. In this case (3.1) boils down to the following inequality:

$$\max_{x_1,\ldots,x_n\in\mathbb{S}^{n-1}} \sum_{\substack{i,j\in\{1,\ldots,n\}\\ i\neq j}} a_{ij}\langle x_i, x_j\rangle \le K(K_n) \max_{\varepsilon_1,\ldots,\varepsilon_n\in\{-1,1\}} \sum_{\substack{i,j\in\{1,\ldots,n\}\\ i\neq j}} a_{ij}\varepsilon_i\varepsilon_j. \tag{3.3}$$

It turns out that $K(K_n) \asymp \log n$. The estimate $K(K_n) \lesssim \log n$ was proved in [29, 62, 93, 96]. In fact, as shown in [7, theorem 3.7], the following stronger inequality holds true for every $n \times n$ matrix (a_{ij}); it implies that $K(K_n) \lesssim \log n$ by the Cauchy-Schwartz inequality:

$$\max_{x_1,\ldots,x_n\in\mathbb{S}^{n-1}} \sum_{\substack{i,j\in\{1,\ldots,n\}\\ i\neq j}} a_{ij}\langle x_i, x_j\rangle \lesssim \log\left(\frac{\sum_{i\in\{1,\ldots,n\}}\sum_{j\in\{1,\ldots,n\}\setminus\{i\}}|a_{ij}|}{\sqrt{\sum_{i\in\{1,\ldots,n\}}\sum_{j\in\{1,\ldots,n\}\setminus\{i\}}a_{ij}^2}}\right) \max_{\varepsilon_1,\ldots,\varepsilon_n\in\{-1,1\}} \sum_{\substack{i,j\in\{1,\ldots,n\}\\ i\neq j}} a_{ij}\varepsilon_i\varepsilon_j.$$

The matching lower bound $K(K_n) \gtrsim \log n$ is due to [7], which improves on a result of [62].

How can we interpolate between the two extremes contained in inequalities (3.2) and (3.3)? The Grothendieck constant $K(G)$ depends on the combinatorial structure of the graph G, but at present our understanding of this dependence is incomplete. The following general bounds are known:

$$\log\omega \lesssim K(G) \lesssim \log\vartheta \tag{3.4}$$

and

$$K(G) \le \frac{\pi}{2\log\left(\frac{1+\sqrt{(\vartheta-1)^2+1}}{\vartheta-1}\right)}, \tag{3.5}$$

where (3.4) is due to [7] and (3.5) is due to [25]. Here ω is the clique number of G, i.e., the largest $k \in \{2,\ldots,n\}$ such that there exists $S \subseteq \{1,\ldots,n\}$ of cardinality k satisfying $\{i,j\} \in E$ for all distinct $i, j \in S$, and

$$\vartheta = \min\left\{\max_{i\in\{1,\ldots,n\}} \frac{1}{\langle x_i, y\rangle^2} : x_1,\ldots,x_n, y \in \mathbb{S}^n \ \wedge\ \forall\{i,j\}\in E,\ \langle x_i, x_j\rangle = 0\right\}. \tag{3.6}$$

The parameter ϑ is known as the Lovász theta function of the complement of G, an important graph parameter that was introduced in [89]. We refer to [61] and [7, theorem 3.5] for alternative characterizations of ϑ. It suffices to say here that it was shown in [89] that $\vartheta \le \chi$, where χ is the chromatic number of G, i.e., the smallest integer k such that there exists a partition $\{A_1,\ldots,A_k\}$ of $\{1,\ldots,n\}$ such that

$\{i,j\} \notin E$ for all $(i,j) \in \bigcup_{\ell=1}^{k} A_\ell \times A_\ell$. Note that the upper bound in (3.4) is superior to (3.5) when ϑ is large, but when $\vartheta = 2$ the bound (3.5) implies Krivine's classical bound [79] $K_G \le \pi/(2\log(1+\sqrt{2}))$.

The upper and lower bounds in (3.4) are known to match up to absolute constants for a variety of graph classes. Several such sharp Grothendieck-type inequalities are presented in sections 5.2 and 5.3 of [7]. For example, as explained in [7], it follows from (3.4), combined with combinatorial results of [9, 89], that for every $n \times n \times n$ 3-tensor (a_{ijk}) we have

$$\max_{\{x_{ij}\}_{i,j=1}^{n} \subseteq \mathbb{S}^{n^2-1}} \sum_{\substack{i,j,k \in \{1,\ldots,n\} \\ i \neq j \neq k}} a_{ijk} \langle x_{ij}, x_{jk} \rangle \lesssim \max_{\{\varepsilon_{ij}\}_{i,j=1}^{n} \subseteq \{-1,1\}} \sum_{\substack{i,j,k \in \{1,\ldots,n\} \\ i \neq j \neq k}} a_{ijk} \varepsilon_{ij} \varepsilon_{jk}.$$

While (3.4) is often a satisfactory asymptotic evaluation of $K(G)$, this isn't always the case. In particular, it is unknown whether $K(G)$ can be bounded from below by a function of ϑ that tends to ∞ as $\vartheta \to \infty$. An instance in which (3.4) is not sharp is the case of Erdős-Rényi [38] random graphs $G(n, \frac{1}{2})$. For such graphs we have $\omega \asymp \log n$ almost surely as $n \to \infty$; see [92] and [10, sec. 4.5]. At the same time, for $G(n, \frac{1}{2})$ we have [60] $\vartheta \asymp \sqrt{n}$ almost surely as $n \to \infty$. Thus (3.4) becomes in this case the rather weak estimate $\log\log n \lesssim K(G(n, \frac{1}{2})) \lesssim \log n$. It turns out [3] that $K(G(n, \frac{1}{2})) \asymp \log n$ almost surely as $n \to \infty$; we refer to [3] for additional computations of this type of the Grothendieck constant of random and pseudorandom graphs. An explicit evaluation of the Grothendieck constant of certain graph families can be found in [81]; for example, if G is a graph of girth g that is not a forest and does not admit K_5 as a minor, then $K(G) = g\cos(\pi/g)/(g-2)$.

3.1 Algorithmic Consequences

Other than being a natural variant of the Grothendieck inequality, and hence of intrinsic mathematical interest, (3.1) has ramifications to discrete optimization problems, which we now describe.

3.1.1 Spin Glasses

Perhaps the most natural interpretation of (3.1) is in the context of solid state physics, specifically the problem of efficient computation of ground states of Ising spin glasses. The graph G represents the interaction pattern of n particles; thus $\{i,j\} \notin E$ if and only if the particles i and j cannot interact with each other. Let a_{ij} be the magnitude of the interaction of i and j (the sign of a_{ij} corresponds to attraction/repulsion). In the Ising model each particle $i \in \{1,\ldots,n\}$ has a spin $\varepsilon_i \in \{-1,1\}$, and the total energy of the system is given by the quantity $-\sum_{\{i,j\} \in E} a_{ij}\varepsilon_i\varepsilon_j$. A spin configuration $(\varepsilon_1,\ldots,\varepsilon_n) \in \{-1,1\}^n$ is called a ground state if it minimizes the total energy. Thus the problem of finding a ground

state is precisely that of computing the maximum appearing in the right-hand side of (3.1). For more information on this topic, see [90, pp. 352–355].

Physical systems seek to settle at a ground state, and therefore it is natural to ask whether it is computationally efficient (i.e., polynomial time computable) to find such a ground state, at least approximately. Such questions have been studied in the physics literature for several decades; see [13, 15, 17, 19]. In particular, it was shown in [17] that if G is a planar graph, then one can find a ground state in polynomial time, but in [13] it was shown that when G is the three-dimensional grid, then this computational task is NP-hard.

Since the quantity in the left-hand side of (3.1) is a semidefinite program and therefore can be computed in polynomial time with arbitrarily good precision, a good bound on $K(G)$ yields a polynomial-time algorithm that computes the energy of a ground state with correspondingly good approximation guarantee. Moreover, as explained in [7], the proof of the upper bound in (3.4) yields a polynomial-time algorithm that finds a spin configuration $(\sigma_1, \ldots, \sigma_n) \in \{-1, 1\}^n$ for which

$$\sum_{\substack{i,j \in \{1,\ldots,n\} \\ \{i,j\} \in E}} a_{ij}\sigma_i\sigma_j \gtrsim \frac{1}{\log \vartheta} \cdot \max_{\{\varepsilon_i\}_{i=1}^n \subseteq \{-1,1\}} \sum_{\substack{i,j \in \{1,\ldots,n\} \\ \{i,j\} \in E}} a_{ij}\varepsilon_i\varepsilon_j. \tag{3.7}$$

An analogous polynomial-time algorithm corresponds to the bound (3.5). These algorithms yield the best-known efficient methods for computing a ground state of Ising spin glasses on a variety of interaction graphs.

3.1.2 Correlation Clustering

A different interpretation of (3.1) yields the best-known polynomial-time approximation algorithm for the correlation clustering problem [14, 27]; this connection is due to [29]. Interpret the graph $G = (\{1, \ldots, n\}, E)$ as the "similarity/dissimilarity graph" for the items $\{1, \ldots, n\}$ in the following sense: For $\{i, j\} \in E$ we are given a sign $a_{ij} \in \{-1, 1\}$, for which if $a_{ij} = 1$, then i and j are deemed to be similar, and if $a_{ij} = -1$ then i and j are deemed to be different. If $\{i, j\} \notin E$, then we do not express any judgment on the similarity or dissimilarity of i and j.

Assume that $A_1, \ldots, A_k$ is a partition (or "clustering") of $\{1, \ldots, n\}$. An agreement between this clustering and our similarity/dissimilarity judgments is a pair $i, j \in \{1, \ldots, n\}$ such that $a_{ij} = 1$ and $i, j \in A_r$ for some $r \in \{1, \ldots, k\}$ or $a_{ij} = -1$ and $i \in A_r$, $j \in A_s$, for distinct $r, s \in \{1, \ldots, k\}$. A disagreement between this clustering and our similarity/dissimilarity judgments is a pair $i, j \in \{1, \ldots, n\}$ such that $a_{ij} = 1$ and $i \in A_r$, $j \in A_s$, for distinct $r, s \in \{1, \ldots, k\}$ or $a_{ij} = -1$ and $i, j \in A_r$ for some $r \in \{1, \ldots, k\}$. Our goal is to cluster the items while encouraging agreements and penalizing disagreements. Thus, we wish to find a clustering of $\{1, \ldots, n\}$ into an unspecified number of clusters that maximizes the total number of agreements minus the total number of disagreements.

It was proved in [29] that the case of clustering into two parts is the bottleneck for this problem. Namely, assume that there were a polynomial-time algorithm that finds a clustering into two parts for which the total number of agreements minus the total number of disagreements is at least a fraction $\alpha \in (0,1)$ of the maximum possible (over all bipartitions) total number of agreements minus the total number of disagreements. Then one could find in polynomial time a clustering that is at least a fraction $\frac{\alpha}{2+\alpha}$ of the analogous maximum that is defined without specifying the number of clusters.

One checks that the problem of finding a partition into two clusters that maximizes the total number of agreements minus the total number of disagreements is the same as the problem of computing the maximum in the right-hand side of (3.1). Thus the upper bound in (3.4) yields a polynomial-time algorithm for correlation clustering with approximation guarantee $O(\log \vartheta)$, which is the best-known approximation algorithm for this problem. Note that when G is the complete graph, then the approximation ratio is $O(\log n)$. As will be explained in Section 7, it is known [71] that for every $\gamma \in (0, \frac{1}{6})$, if there were a polynomial-time algorithm for correlation clustering that yields an approximation guarantee of $(\log n)^{\gamma}$, then there would be an algorithm for 3-colorability that runs in time $2^{(\log n)^{O(1)}}$, a conclusion that is widely believed to be impossible.

4 Kernel Clustering and the Propeller Conjecture

Here we describe a large class of Grothendieck-type inequalities that is motivated by algorithmic applications to a combinatorial optimization problem called "kernel clustering." This problem originates in machine learning [112], and its only known rigorous approximation algorithms follow from Grothendieck inequalities (these algorithms are sharp assuming the UGC). We will first describe the inequalities and then the algorithmic application.

Consider the special case of the Grothendieck inequality (1.1) where $A = (a_{ij})$ is an $n \times n$ positive semidefinite matrix. In this case we may assume without loss of generality that in (1.1) $x_i = y_i$ and $\varepsilon_i = \delta_i$ for every $i \in \{1, \dots, n\}$ since this holds for the maxima on either side of (1.1) (see also the explanation in [8, sec. 5.2]). It follows from [47, 109] (see also [97]) that for every $n \times n$ symmetric positive semidefinite matrix $A = (a_{ij})$ we have

$$\max_{x_1,\dots,x_n \in \mathbb{S}^{n-1}} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \langle x_i, x_j \rangle \le \frac{\pi}{2} \cdot \max_{\varepsilon_1,\dots,\varepsilon_n \in \{-1,1\}} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} \varepsilon_i \varepsilon_j, \tag{4.1}$$

and that $\frac{\pi}{2}$ is the best possible constant in (4.1).

A natural variant of (4.1) is to replace the numbers -1 and 1 by general vectors $v_1, \dots, v_k \in \mathbb{R}^k$; namely, one might ask for the smallest constant $K \in (0, \infty)$

such that for every symmetric positive semidefinite $n \times n$ matrix (a_{ij}) we have

$$\max_{x_1,\ldots,x_n \in \mathbb{S}^{n-1}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle x_i, x_j \rangle \le K \max_{u_1,\ldots,u_n \in \{v_1,\ldots,v_k\}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle u_i, u_j \rangle. \tag{4.2}$$

The best constant K in (4.2) can be characterized as follows: Let

$$B = (b_{ij} = \langle v_i, v_j \rangle)$$

be the Gram matrix of $v_1, \ldots, v_k$. Let $C(B)$ be the maximum over all partitions $\{A_1, \ldots, A_k\}$ of $\mathbb{R}^{k-1}$ into measurable sets of the quantity

$$\sum_{i=1}^k \sum_{j=1}^k b_{ij} \langle z_i, z_j \rangle,$$

where for $i \in \{1, \ldots, k\}$ the vector $z_i \in \mathbb{R}^{k-1}$ is the Gaussian moment of A_i, i.e.,

$$z_i = \frac{1}{(2\pi)^{(k-1)/2}} \int_{A_i} x e^{-\|x\|_2^2/2} \, dx.$$

It was proved in [69] that (4.2) holds with $K = 1/C(B)$ and that this constant is sharp.

Inequality (4.2) with $K = 1/C(B)$ is proved via the following *rounding procedure*: Fix unit vectors $x_1, \ldots, x_n \in \mathbb{S}^{n-1}$. Let $G = (g_{ij})$ be a $(k-1) \times n$ random matrix whose entries are i.i.d. standard Gaussian random variables. Let $A_1, \ldots, A_k \subseteq \mathbb{R}^{k-1}$ be a measurable partition of $\mathbb{R}^{k-1}$ at which $C(B)$ is attained (for a proof that the maximum defining $C(B)$ is indeed attained, see [69]). Define a random choice of $u_i \in \{v_1, \ldots, v_k\}$ by setting $u_i = v_\ell$ for the unique $\ell \in \{1, \ldots, k\}$ such that $Gx_i \in A_\ell$. The fact that (4.2) holds with $K = 1/C(B)$ is a consequence of the following fact, whose proof we skip (the full details are in [69]):

$$\mathbb{E}\left[\sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle u_i, u_j \rangle\right] \ge C(B) \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle x_i, x_j \rangle. \tag{4.3}$$

Determining the partition of $\mathbb{R}^{k-1}$ that achieves the value $C(B)$ is a nontrivial problem in general, even in the special case when $B = I_k$ is the $k \times k$ identity matrix. Note that in this case one desires a partition $\{A_1, \ldots, A_k\}$ of $\mathbb{R}^{k-1}$ into measurable sets so as to maximize the quantity

$$\sum_{i=1}^k \left\| \frac{1}{(2\pi)^{(k-1)/2}} \int_{A_i} x e^{-\|x\|_2^2/2} \, dx \right\|_2^2.$$

As shown in [68, 69], the optimal partition is given by simplicial cones centered at the origin. When $B = I_2$ we have $C(I_2) = \frac{1}{\pi}$, and the optimal partition of $\mathbb{R}$ into two cones is the positive and the negative axes. When $B = I_3$, it was shown in [68] that $C(I_3) = \frac{9}{8\pi}$, and the optimal partition of $\mathbb{R}^2$ into three cones is the *propeller* partition, i.e., into three cones with angular measure 120° each.

Though it might be surprising at first sight, the authors posed in [68] the *propeller conjecture*: for any $k \geq 4$, the optimal partition of $\mathbb{R}^{k-1}$ into k parts is $\mathcal{P} \times \mathbb{R}^{k-3}$, where $\mathcal{P}$ is the propeller partition of $\mathbb{R}^2$. In other words, even if one is allowed to use k parts, the propeller conjecture asserts that the best partition consists of only three nonempty parts. Recently this conjecture was solved positively [55] for $k = 4$, i.e., for partitions of $\mathbb{R}^3$ into four measurable parts. The proof of [55] reduces the problem to a concrete finite set of numerical inequalities that are then verified with full rigor in a computer-assisted fashion. Note that this is the first nontrivial (surprising?) case of the propeller conjecture; i.e., this is the first case in which we indeed drop one of the four allowed parts in the optimal partition.

We now describe an application of (4.2) to the kernel clustering problem; a general framework for clustering massive statistical data so as to uncover a certain hypothesized structure [112]. The problem is defined as follows. Let $A = (a_{ij})$ be an $n \times n$ symmetric positive semidefinite matrix that is usually normalized to be centered, i.e., $\sum_{i=1}^n \sum_{j=1}^n a_{ij} = 0$. The matrix A is often thought of as the correlation matrix of random variables $(X_1, \ldots, X_n)$ that measure attributes of certain empirical data, i.e., $a_{ij} = \mathbb{E}[X_i X_j]$. We are also given another symmetric positive semidefinite $k \times k$ matrix $B = (b_{ij})$ that functions as a hypothesis, or test, matrix. Think of n as huge and k as a small constant. The goal is to cluster A so as to obtain a smaller matrix that most resembles B. Formally, we wish to find a partition $\{S_1, \ldots, S_k\}$ of $\{1, \ldots, n\}$ so that if we write $c_{ij} = \sum_{(p,q) \in S_i \times S_j} a_{pq}$, then the resulting clustered version of A has the maximum correlation $\sum_{i=1}^k \sum_{j=1}^k c_{ij} b_{ij}$ with the hypothesis matrix B. In words, we form a $k \times k$ matrix $C = (c_{ij})$ by summing the entries of A over the blocks induced by the given partition, and we wish to produce in this way a matrix that is most correlated with B. Equivalently, the goal is to evaluate the number

$$\textbf{Clust}(A|B) = \max_{\sigma : \{1,\ldots,n\} \to \{1,\ldots,k\}} \sum_{i=1}^k \sum_{j=1}^k a_{ij} b_{\sigma(i)\sigma(j)}. \tag{4.4}$$

The strength of this generic clustering framework is based in part on the flexibility of adapting the matrix B to the problem at hand. Various particular choices of B lead to well-studied optimization problems, while other specialized choices of B are based on statistical hypotheses that have been applied with some empirical success. We refer to [68, 112] for additional background and a discussion of specific examples.

In [68] it was shown that there exists a randomized polynomial-time algorithm that takes as input two positive semidefinite matrices A and B and outputs a number α that satisfies $\mathbf{Clust}(A|B) \leq \mathbb{E}[\alpha] \leq (1 + \frac{3\pi}{2})\,\mathbf{Clust}(A|B)$. There is no reason to believe that the approximation factor of $1 + \frac{3\pi}{2}$ is sharp, but nevertheless prior to this result, which is based on (4.2), no constant-factor, polynomial-time approximation algorithm for this problem was known.

Sharper results can be obtained if we assume that the input matrices are normalized appropriately. Specifically, assume that $k \geq 3$ and consider only inputs A that are centered, i.e., $\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} = 0$, and inputs B that are either the identity matrix I_k or satisfy $\sum_{i=1}^{k}\sum_{j=1}^{k} b_{ij} = 0$ (B is centered as well) and $b_{ii} = 1$ for all $i \in \{1,\ldots,k\}$ (B is "spherical"). Under these assumptions the output of the algorithm of [68] satisfies $\mathbf{Clust}(A|B) \leq \mathbb{E}[\alpha] \leq \frac{8\pi}{9}(1-\frac{1}{k})\,\mathbf{Clust}(A|B)$. Moreover, it was shown in [68] that if the propeller conjecture and the UGC are assumed, no polynomial-time algorithm can achieve an approximation guarantee that is strictly smaller than $\frac{8\pi}{9}(1-\frac{1}{k})$ (for input matrices normalized as above). Since the propeller conjecture is known to hold true for $k = 3$ [68] and $k = 4$ [55], we know that the UGC hardness threshold for the above problem is exactly $\frac{16\pi}{27}$ when $k = 3$ and $\frac{2\pi}{3}$ when $k = 4$.

A finer, and perhaps more natural, analysis of the kernel clustering problem can be obtained if we fix the matrix B and let the input be only the matrix A, with the goal being, as before, to approximate the quantity $\mathbf{Clust}(A|B)$ in polynomial time. Since B is symmetric and positive semidefinite, we can find vectors $v_1,\ldots,v_k \in \mathbb{R}^k$ such that B is their Gram matrix, i.e., $b_{ij} = \langle v_i, v_j\rangle$ for all $i, j \in \{1,\ldots,k\}$. Let $R(B)$ be the smallest possible radius of a euclidean ball in $\mathbb{R}^k$ that contains $\{v_1,\ldots,v_k\}$, and let $w(B)$ be the center of this ball. We note that both $R(B)$ and $w(B)$ can be efficiently computed by solving an appropriate semidefinite program. Let $C(B)$ be the parameter defined above.

It is shown in [69] that for every fixed symmetric positive semidefinite $k \times k$ matrix B there exists a randomized polynomial-time algorithm which, given an $n \times n$ symmetric positive semidefinite centered matrix A, outputs a number $\mathrm{Alg}(A)$ such that

$$\mathbf{Clust}(A|B) \leq \mathbb{E}[\mathrm{Alg}(A)] \leq \frac{R(B)^2}{C(B)}\,\mathbf{Clust}(A|B).$$

As we will explain in Section 7, if we assume the UGC, no polynomial-time algorithm can achieve an approximation guarantee strictly smaller than $R(B)^2/C(B)$.

The algorithm of [69] uses semidefinite programming to compute the value

$$\begin{aligned}\mathrm{SDP}(A|B) &= \max\Big\{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\langle x_i, x_j\rangle : x_1,\ldots,x_n \in \mathbb{R}^n \wedge \|x_i\|_2 \leq 1\ \forall i \in \{1,\ldots,n\}\Big\}\\ &\overset{(\clubsuit)}{=} \max\Big\{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\langle x_i, x_j\rangle : x_1,\ldots,x_n \in \mathbb{S}^{n-1}\Big\},\end{aligned}$$

where (♣) holds since the function $(x_1, \ldots, x_n) \mapsto \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle x_i, x_j \rangle$ is convex (by virtue of the fact that A is positive semidefinite).

We claim that

$$\frac{\mathbf{Clust}(A|B)}{R(B)^2} \leq \mathrm{SDP}(A|B) \leq \frac{\mathbf{Clust}(A|B)}{C(B)}, \tag{4.5}$$

which implies that if we output the number $R(B)^2 \mathrm{SDP}(A|B)$ we will obtain a polynomial-time algorithm that will approximate $\mathbf{Clust}(A|B)$ up to a factor of $R(B)^2/C(B)$. To verify (4.5), let $x_1^*, \ldots, x_n^* \in \mathbb{S}^{n-1}$ and $\sigma^* : \{1, \ldots, n\} \to \{1, \ldots, k\}$ be such that

$$\mathrm{SDP}(A|B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle x_i^*, x_j^* \rangle \quad \text{and} \quad \mathbf{Clust}(A|B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\sigma^*(i)\sigma^*(j)}.$$

Write $(a_{ij})_{i,j=1}^n = (\langle u_i, u_j \rangle)_{i,j=1}^n$ for some $u_1, \ldots, u_n \in \mathbb{R}^n$. The assumption that A is centered means that $\sum_{i=1}^n u_i = 0$. The rightmost inequality in (4.5) is just the Grothendieck inequality (4.2). The leftmost inequality in (4.5) follows from the fact that $\frac{v_{\sigma^*(i)} - w(B)}{R(B)}$ has norm at most 1 for all $i \in \{1, \ldots, n\}$. Indeed, these norm bounds imply that

$$\begin{aligned}
\mathrm{SDP}(A|B) &\geq \sum_{i=1}^n \sum_{j=1}^n a_{ij} \left\langle \frac{v_{\sigma^*(i)} - w(B)}{R(B)}, \frac{v_{\sigma^*(j)} - w(B)}{R(B)} \right\rangle \\
&= \frac{1}{R(B)^2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle v_{\sigma^*(i)}, v_{\sigma^*(j)} \rangle \\
&\quad - \frac{2}{R(B)^2} \sum_{i=1}^n \langle w(B), v_{\sigma^*(i)} \rangle \left\langle u_i, \sum_{j=1}^n u_j \right\rangle + \frac{\|w(B)\|_2^2}{R(B)^2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \\
&= \frac{\mathbf{Clust}(A|B)}{R(B)^2}.
\end{aligned}$$

This completes the proof that the above algorithm approximates efficiently the number $\mathbf{Clust}(A|B)$ but does not address the issue of how to efficiently compute an assignment $\sigma : \{1, \ldots, n\} \to \{1, \ldots, k\}$ for which the induced clustering of A has the required value. The issue here is to find efficiently a conical simplicial partition $A_1, \ldots, A_k$ of $\mathbb{R}^{k-1}$ at which $C(B)$ is attained. Such a partition exists and may be assumed to be hardwired into the description of the algorithm. Alternately, the partition that achieves $C(B)$ up to a desired degree of accuracy can be found by brute force for fixed k (or $k = k(n)$ growing sufficiently slowly as a function of n); see [69]. For large values of k the problem of computing $C(B)$ efficiently remains open.

5 The L_p Grothendieck Problem

Fix $p \in [1,\infty]$ and consider the following algorithmic problem. The input is an $n \times n$ matrix $A = (a_{ij})$ whose diagonal entries vanish, and the goal is to compute (or estimate) in polynomial time the quantity

$$M_p(A) = \max_{\substack{t_1,\ldots,t_n \in \mathbb{R} \\ \sum_{k=1}^n |t_k|^p \le 1}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} t_i t_j = \max_{\substack{t_1,\ldots,t_n \in \mathbb{R} \\ \sum_{k=1}^n |t_k|^p = 1}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} t_i t_j. \tag{5.1}$$

The second equality in (5.1) follows from a straightforward convexity argument since the diagonal entries of A vanish. Some of the results described below hold true without the vanishing diagonal assumption, but we will tacitly make this assumption here since the second equality in (5.1) makes the problem become purely combinatorial when $p = \infty$. Specifically, if $G = (\{1,\ldots,n\}, E)$ is the complete graph, then

$$M_\infty(A) = \max_{\varepsilon_1,\ldots,\varepsilon_n \in \{-1,1\}} \sum_{\{i,j\} \in E} a_{ij} \varepsilon_i \varepsilon_j.$$

The results described in Section 3 therefore imply that there is a polynomial-time algorithm that approximates $M_\infty(A)$ up to an $O(\log n)$ factor, and that it is computationally hard to achieve an approximation guarantee smaller than $(\log n)^\gamma$ for all $\gamma \in (0, \frac{1}{6})$.

There are values of p for which the above problem can be solved in polynomial time. When $p = 2$ the quantity $M_2(A)$ is the largest eigenvalue of A and hence can be computed in polynomial time [45, 84]. When $p = 1$ it was shown in [2] that it is possible to approximate $M_1(A)$ up to a factor of $1 + \varepsilon$ in time $n^{O(1/\varepsilon)}$. It was also shown in [2] that the problem of $(1+\varepsilon)$-approximately computing $M_1(A)$ is $W[1]$ complete; we refer to [37] for the definition of this type of hardness result and just say here that it indicates that a running time of $c(\varepsilon)n^{O(1)}$ is impossible.

The algorithm of [2] proceeds by showing that for every $m \in \mathbb{N}$ there exist $y_1,\ldots,y_n \in \frac{1}{m}\mathbb{Z}$ with $\sum_{i=1}^n |y_i| \le 1$ and

$$\sum_{i=1}^n \sum_{j=1}^n a_{ij} y_i y_j \ge \left(1 - \frac{1}{m}\right) M_1(A).$$

One checks that the number of such vectors y is

$$1 + \sum_{k=1}^m \sum_{\ell=1}^k 2^\ell \binom{n}{\ell} \binom{k-1}{\ell-1} \le 4n^m.$$

An exhaustive search over all such vectors will then approximate $M_1(A)$ to within a factor of $m/(m-1)$ in time $O(n^m)$. To prove the existence of y, fix $t_1,\ldots,t_n \in \mathbb{R}$ with $\sum_{k=1}^n |t_k| = 1$ and $\sum_{i=1}^n \sum_{j=1}^n a_{ij} t_i t_j = M_1(A)$. Let $X \in \mathbb{R}^n$ be a random vector given by $\Pr[X = \operatorname{sign}(t_j) e_j] = |t_j|$ for every $j \in \{1,\ldots,n\}$. Here $e_1,\ldots,e_n$ is the standard basis of $\mathbb{R}^n$. Let $\{X_s = (X_{s1},\ldots,X_{sn})\}_{s=1}^m$ be

independent copies of X and set $Y = (Y_1, \ldots, Y_n) = \frac{1}{m}\sum_{s=1}^{m} X_s$. Note that if $s, t \in \{1, \ldots, m\}$ are distinct, then for all $i, j \in \{1, \ldots, n\}$ we have $\mathbb{E}[X_{si} X_{tj}] = \operatorname{sign}(t_i)\operatorname{sign}(t_j)|t_i| \cdot |t_j| = t_i t_j$. Also, for every $s \in \{1, \ldots, m\}$ and every distinct $i, j \in \{1, \ldots, n\}$, we have $X_{si} X_{sj} = 0$. Since the diagonal entries of A vanish, it follows that

$$\mathbb{E}\Big[\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} Y_i Y_j\Big] = \frac{1}{m^2} \sum_{\substack{s,t \in \{1,\ldots,m\} \\ s \neq t}} \sum_{\substack{i,j \in \{1,\ldots,n\} \\ i \neq j}} a_{ij}\, \mathbb{E}[X_{si} X_{tj}] = \left(1 - \frac{1}{m}\right) M_1(A). \tag{5.2}$$

Noting that the vector Y has ℓ_1-norm at most 1 and all of its entries are integer multiples of $\frac{1}{m}$, it follows from (5.2) that with positive probability Y will have the desired properties.

How can we interpolate between the above results for $p \in \{1, 2, \infty\}$? It turns out that there is a satisfactory answer for $p \in (2, \infty)$, but the range $p \in (1, 2)$ remains a mystery. To explain this write $\gamma_p = (\mathbb{E}[|G|^p])^{1/p}$, where G is a standard Gaussian random variable. One computes that

$$\gamma_p = \sqrt{2}\left(\frac{\Gamma(\frac{p+1}{2})}{\sqrt{\pi}}\right)^{1/p}. \tag{5.3}$$

Also, Stirling's formula implies that $\gamma_p^2 = \frac{p}{e} + O(1)$ as $p \to \infty$. It follows from [50, 94] that for every fixed $p \in [2, \infty)$ there exists a polynomial-time algorithm that approximates $M_p(A)$ to within a factor of γ_p^2, and that for every $\varepsilon \in (0, 1)$ the existence of a polynomial-time algorithm that approximates $M_p(A)$ to within a factor $\gamma_p^2 - \varepsilon$ would imply that P = NP. These results improve over the earlier work [72], which designed a polynomial-time algorithm for $M_p(A)$ whose approximation guarantee is $(1 + o(1))\gamma_p^2$ as $p \to \infty$, and which proved a $\gamma_p^2 - \varepsilon$ hardness results if the UGC is assumed rather than P $\neq$ NP.

The following Grothendieck-type inequality was proved in [94] and independently in [50]. For every $n \times n$ matrix $A = (a_{ij})$ and every $p \in [2, \infty)$ we have

$$\max_{\substack{x_1,\ldots,x_n \in \mathbb{R}^n \\ \sum_{k=1}^n \|x_k\|_2^p \le 1}} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} \langle x_i, x_j \rangle \le \gamma_p^2 \max_{\substack{t_1,\ldots,t_n \in \mathbb{R} \\ \sum_{k=1}^n |t_k|^p \le 1}} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} t_i t_j. \tag{5.4}$$

The constant γ_p^2 in (5.4) is sharp. The validity of (5.4) implies that $M_p(A)$ can be computed in polynomial time to within a factor γ_p^2. This follows since the left-hand side of (5.4) is the maximum of $\sum_{i=1}^n \sum_{j=1}^n a_{ij} X_{ij}$, which is a linear functional in the variables (X_{ij}), given the constraint that (X_{ij}) is a symmetric positive semidefinite matrix and $\sum_{i=1}^n X_{ii}^{p/2} \le 1$. The latter constraint is convex since $p \ge 2$, and therefore this problem falls into the framework of convex programming that

was described in Section 1.2. Thus the left-hand side of (5.4) can be computed in polynomial time with arbitrarily good precision.

Choosing the specific value $p = 3$ in order to illustrate the current satisfactory state of affairs concretely, the NP-hardness threshold of computing

$$\max_{\sum_{i=1}^n |x_i|^3 \le 1} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

equals $2/\sqrt[3]{\pi}$. Such a sharp NP-hardness result (with transcendental hardness ratio) is quite remarkable, since it shows that the geometric algorithm presented above probably yields the best possible approximation guarantee even when one allows any polynomial-time algorithm whatsoever. Results of this type have been known to hold under the UGC, but this NP-hardness result of [50] seems to be the first time that such an algorithm for a simple-to-state problem was shown to be optimal assuming P $\neq$ NP.

When $p \in [1, 2]$ one can easily show [94] that

$$\max_{\substack{x_1,\ldots,x_n \in \mathbb{R}^n \\ \sum_{k=1}^n \|x_k\|_2^p \le 1}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \langle x_i, x_j \rangle = \max_{\substack{t_1,\ldots,t_n \in \mathbb{R} \\ \sum_{k=1}^n |t_k|^p \le 1}} \sum_{i=1}^n \sum_{j=1}^n a_{ij} t_i t_j. \tag{5.5}$$

While the identity (5.5) seems to indicate the problem of computing $M_p(A)$ in polynomial time might be easy for $p \in (1, 2)$, the above argument fails since the constraint $\sum_{i=1}^n X_{ii}^{p/2} \le 1$ is no longer convex. This is reflected by the fact that despite (5.5) the problem of $(1 + \varepsilon)$-approximately computing $M_1(A)$ is $W[1]$ complete [2]. It remains open whether for $p \in (1, 2)$ one can approximate $M_p(A)$ in polynomial time up to a factor $O(1)$, and no hardness-of-approximation result is known for this problem as well.

Remark 5.1. If $p \in [2, \infty]$, then for positive semidefinite matrices (a_{ij}) the constant γ_p^2 in the right-hand side of (5.4) can be improved [94] to $\gamma_{p^*}^{-2}$, where here and in what follows $p^* = p/(p-1)$. For $p = \infty$ this estimate coincides with the classical bound [47, 109] that we have already encountered in (4.1), and it is sharp in the entire range $p \in [2, \infty]$. Moreover, this bound shows that there exists a polynomial-time algorithm that takes as input a positive semidefinite matrix A and outputs a number that is guaranteed to be within a factor $\gamma_{p^*}^{-2}$ of $M_p(A)$. Conversely, the existence of a polynomial-time algorithm for this problem whose approximation guarantee is strictly smaller than $\gamma_{p^*}^{-2}$ would contradict the UGC [94].

Remark 5.2. The natural bilinear variant of (5.4) is an immediate consequence of the Grothendieck inequality (1.1). Specifically, assume that $p, q \in [1, \infty]$ and $x_1, \ldots, x_m, y_1, \ldots, y_n \in \mathbb{R}^{m+n}$ satisfy $\sum_{i=1}^m \|x_i\|_2^p \le 1$ and $\sum_{j=1}^n \|y_j\|_2^q \le 1$. Write $\alpha_i = \|x_i\|_2$ and $\beta_j = \|y_j\|_2$. For an $m \times n$ matrix (a_{ij}) the Grothendieck

inequality provides $\varepsilon_1, \dots, \varepsilon_m, \delta_1, \dots, \delta_n \in \{-1, 1\}$ such that

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\langle x_i, y_j\rangle \le K_G \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\alpha_i\beta_j\varepsilon_i\delta_j.$$

This establishes the following inequality:

$$\max_{\substack{\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq \mathbb{R}^{n+m} \\ \sum_{i=1}^m \|x_i\|_2^p \le 1 \\ \sum_{j=1}^n \|y_j\|_2^q \le 1}} \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\langle x_i, y_j\rangle \le K_G \cdot \max_{\substack{\{s_i\}_{i=1}^m, \{t_j\}_{j=1}^n \subseteq \mathbb{R} \\ \sum_{i=1}^m |s_i|^p \le 1 \\ \sum_{j=1}^n |t_j|^q \le 1}} \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} s_i t_j. \tag{5.6}$$

Observe that the maximum on the right-hand side of (5.6) is $\|A\|_{p\to q^*}$, the operator norm of A acting as a linear operator from $(\mathbb{R}^m, \|\cdot\|_p)$ to $(\mathbb{R}^n, \|\cdot\|_{q^*})$. Moreover, if $p, q \ge 2$, then the left-hand side of (5.6) can be computed in polynomial time. Thus, for $p \ge 2 \ge r \ge 1$, the generalized Grothendieck inequality (5.6) yields a polynomial-time algorithm that takes as input an $m \times n$ matrix $A = (a_{ij})$ and outputs a number that is guaranteed to be within a factor K_G of $\|A\|_{p\to r}$. This algorithmic task has been studied in [98] (see also [95, sec. 4.3.2]), where for $p \ge 2 \ge r \ge 1$ a polynomial-time algorithm was designed that approximates $\|A\|_{p\to r}$ up to a factor $3\pi/(6\sqrt{3} - 2\pi) \in [2.293, 2.294]$.

The above argument yields the approximation factor $K_G < 1.783$ as a formal consequence of the Grothendieck inequality. The complexity of the problem of approximating $\|A\|_{p\to r}$ has been studied in [18], where it is shown that if either $p \ge r > 2$ or $2 > p \ge r$, then it is NP-hard to approximate $\|A\|_{p\to r}$ up to any constant factor, and unless 3-colorability can be solved in time $2^{(\log n)^{O(1)}}$, for any $\varepsilon \in (0, 1)$ no polynomial-time algorithm can approximate $\|A\|_{p\to r}$ up to $2^{(\log n)^{1-\varepsilon}}$.

Remark 5.3. Let $K \subseteq \mathbb{R}^n$ be a compact and convex set that is invariant under reflections with respect to the coordinate hyperplanes. Denote by C_K the smallest $C \in (0, \infty)$ such that for every $n \times n$ matrix (a_{ij}) we have

$$\max_{\substack{x_1,\dots,x_n \in \mathbb{R}^n \\ (\|x_1\|_2,\dots,\|x_n\|_2) \in K}} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\langle x_i, x_j\rangle \le C \max_{\substack{t_1,\dots,t_n \in \mathbb{R} \\ (t_1,\dots,t_n) \in K}} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} t_i t_j. \tag{5.7}$$

Such generalized Grothendieck inequalities are investigated in [94], where bounds on C_K are obtained under certain geometric assumptions on K. These assumptions are easy to verify when $K = \{x \in \mathbb{R}^n : \|x\|_p \le 1\}$, yielding (5.4). More subtle inequalities of this type for other convex bodies K are discussed in [94], but we will not describe them here. The natural bilinear version of (5.7) is: if $K \subseteq \mathbb{R}^m$ and $L \subseteq \mathbb{R}^n$ are compact and convex sets that are invariant under reflections with

respect to the coordinate hyperplanes, then let $C_{K,L}$ denote the smallest constant $C \in (0,\infty)$ such that for every $m \times n$ matrix (a_{ij}) we have

$$\max_{\substack{\{x_i\}_{i=1}^m, \{y_j\}_{j=1}^n \subseteq \mathbb{R}^{n+m} \\ (\|x_1\|_2,\ldots,\|x_m\|_2)\in K \\ (\|y_1\|_2,\ldots,\|y_n\|_2)\in L}} \sum_{i=1}^m \sum_{j=1}^n a_{ij} \langle x_i, y_j \rangle \le C \max_{\substack{\{s_i\}_{i=1}^m, \{t_j\}_{j=1}^n \subseteq \mathbb{R} \\ (s_1,\ldots,s_m)\in K \\ (t_1,\ldots,t_n)\in L}} \sum_{i=1}^m \sum_{j=1}^n a_{ij} s_i t_j . \tag{5.8}$$

The argument in Remark 5.2 shows that $C_{K,L} \le K_G$. Under certain geometric assumptions on K and L, this bound can be improved [94].

6 Higher-Rank Grothendieck Inequalities

We have already seen several variants of the classical Grothendieck inequality (1.1), including the Grothendieck inequality for graphs (3.1), the variant of the positive semidefinite Grothendieck inequality arising from the kernel clustering problem (4.2), and Grothendieck inequalities for convex bodies other than the cube (5.4), (5.6), (5.7), (5.8). The literature contains additional variants of the Grothendieck inequality, some of which will be described in this section.

Let $G = (\{1,\ldots,n\}, E)$ be a graph and fix $q, r \in \mathbb{N}$. Following [25], define $K(q \to r, G)$ to be the smallest constant $K \in (0,\infty)$ such that for every $n \times n$ matrix $A = (a_{ij})$ we have

$$\max_{x_1,\ldots,x_n \in \mathbb{S}^{q-1}} \sum_{\substack{i,j\in\{1,\ldots,n\} \\ \{i,j\}\in E}} a_{ij} \langle x_i, x_j \rangle \le K \max_{y_1,\ldots,y_n \in \mathbb{S}^{r-1}} \sum_{\substack{i,j\in\{1,\ldots,n\} \\ \{i,j\}\in E}} a_{ij} \langle y_i, y_j \rangle. \tag{6.1}$$

Set also $K(r, G) = \sup_{q\in\mathbb{N}} K(q \to r, G)$. We similarly define $K^+(q \to r, G)$ to be the smallest constant $K \in (0,\infty)$ satisfying (6.1) for all positive semidefinite matrices A, and correspondingly $K^+(r, G) = \sup_{q\in\mathbb{N}} K^+(q \to r, G)$.

To link these definitions to what we have already seen in this article, observe that K_G is the supremum of $K(1, G)$ over all finite bipartite graphs G, and due to the results described in Section 4 we have

$$\sup_{n\in\mathbb{N}} K^+(r, K_n^{\circlearrowleft}) = \sup_{n\in\mathbb{N}} \sup_{x_1,\ldots,x_n\in\mathbb{S}^{r-1}} \frac{1}{C(\langle x_i, x_j\rangle)_{i,j=1}^n)}, \tag{6.2}$$

where $K_n^{\circlearrowleft}$ is the complete graph on n-vertices with self-loops. Recall that the definition of $C(B)$ for a positive semidefinite matrix B is given in the paragraph following (4.2).

An important special case of (6.1) is $r = 2$, since the supremum of $K(2, G)$ over all finite bipartite graphs G is at most the complex Grothendieck constant $K_G^{\mathbb{C}}$ (defined analogously to K_G, but over the complex scalar field), a fundamental quantity whose value has been investigated in [47, 52, 76, 85, 101]. The best-known bounds on $K_G^{\mathbb{C}}$ are $1.338 < K_G^{\mathbb{C}} < 1.4049$; see [103, sec. 4] for more information on this topic. We also refer to [34, 115] for information of the constants $K(2q \to 2, G)$ where G is a bipartite graph. The supremum of $K(q \to r, G)$ over all bipartite graphs G was investigated in [80] for $r = 1$ and in [76] for $r = 2$; see also [77] for a unified treatment of these cases. The higher-rank constants $K(q \to r, G)$ when G is bipartite were introduced in [24]. Definition (6.1) in full generality is due to [25], where several estimates on $K(q \to r, G)$ are given. One of the motivations of [25] is the case $r = 3$ (and G a subgraph of the grid $\mathbb{Z}^3$), based on the connection to the polynomial-time approximation of ground states of spin glasses as described in Section 3.1.1; the case $r = 1$ was discussed in Section 3.1.1 in connection with the Ising model, but the case $r = 3$ corresponds to the more physically realistic Heisenberg model of vector-valued spins. The parameter $\sup_{n\in\mathbb{N}} K^+(r, K_n^{\mathfrak{O}})$ (recall (6.2)) was studied in [24] in the context of quantum information theory, and in [26] it was shown that

$$K^+(1, K_n^{\mathfrak{O}}) \le \frac{\pi}{n}\left(\frac{\Gamma((n+1)/2)}{\Gamma(n/2)}\right)^2 = \frac{\pi}{2} - \frac{\pi}{4n} + O\left(\frac{1}{n^2}\right), \tag{6.3}$$

and

$$\sup_{n\in\mathbb{N}} K^+(r, K_n^{\mathfrak{O}}) = \frac{r}{2}\left(\frac{\Gamma(r/2)}{\Gamma((r+1)/2)}\right)^2 = 1 + \frac{1}{2r} + O\left(\frac{1}{r^2}\right).$$

We refer to [26] for a corresponding UGC hardness result. Note that (6.3) improves over (4.1) for fixed $n \in \mathbb{N}$.

7 Hardness of Approximation

We have seen some examples of how Grothendieck-type inequalities yield upper bounds on the best possible polynomial-time approximation ratio of certain optimization problems. From the algorithmic and computational complexity viewpoint, it is interesting to prove computational lower bounds as well, i.e., results that rule out the existence of efficient algorithms achieving a certain approximation guarantee. Such results are known as hardness or inapproximability results, and as explained in Section 1.1, at present the state of the art allows one to prove such results while relying on complexity theoretic assumptions such as P $\neq$ NP or the unique games conjecture. A nice feature of the known hardness results for problems in which a Grothendieck-type inequality has been applied is that often the hardness results (lower bounds) exactly match the approximation ratios (upper bounds). In this section we briefly review the known hardness results for optimization problems associated with Grothendieck-type inequalities.

Let $K_{n,n}$-QP denote the optimization problem corresponding to the classical Grothendieck inequality (the acronym QP stands for "quadratic programming").

Thus, in the problem $K_{n,n}$-QP we are given an $n \times n$ real matrix (a_{ij}), and the goal is to determine the quantity

$$\max\Big\{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\varepsilon_i\delta_j : \{\varepsilon_i\}_{i=1}^{m}, \{\delta_j\}_{j=1}^{n} \subseteq \{-1,1\}\Big\}.$$

As explained in [8], the MAX DICUT problem can be framed as a special case of the problem $K_{n,n}$-QP. Hence, as a consequence of [53], we know that for every $\varepsilon \in (0,1)$, assuming P $\neq$ NP there is no polynomial-time algorithm that approximates the $K_{n,n}$-QP problem within ratio $\frac{13}{12} - \varepsilon$. In [70] it is shown that the lower bound (1.3) on the Grothendieck constant can be translated into a hardness result, albeit relying on the unique games conjecture. Namely, letting η_0 be as in (1.3), for every $\varepsilon \in (0,1)$ assuming the UGC there is no polynomial-time algorithm that approximates the $K_{n,n}$-QP problem within a ratio $\frac{\pi}{2}e^{\eta_0^2} - \varepsilon$.

We note that all the hardness results cited here rely on the well-known paradigm of *dictatorship testing*. A lower bound on the integrality gap of a semidefinite program, such as the estimate $K_G \geq \frac{\pi}{2}e^{\eta_0^2}$, can be translated into a probabilistic test to check whether a function $f : \{-1,1\}^n \mapsto \{-1,1\}$ is a dictatorship, i.e., of the form $f(x) = x_i$ for some fixed $i \in \{1,\ldots,n\}$. If f is indeed a dictatorship, then the test passes with probability c and if f is "far from a dictator" (in a formal sense that we do not describe here), the test passes with probability at most s. The ratio c/s corresponds exactly to the UGC-based hardness lower bound. It is well-known how to prove a UGC-based hardness result once we have the appropriate dictatorship test; see the survey [65].

The above-quoted result of [70] relied on explicitly knowing the lower-bound construction of [107] leading to the estimate $K_G \geq \frac{\pi}{2}e^{\eta_0^2}$. On the other hand, in [106], building on the earlier work [105], it is shown that *any* lower bound on the Grothendieck constant can be translated into a UGC-based hardness result, even without explicitly knowing the construction! Thus, modulo the UGC, the best polynomial-time algorithm to approximate the $K_{n,n}$-QP problem is via the Grothendieck inequality, even though we do not know the precise value of K_G. Formally, for every $\varepsilon \in (0,1)$, given the UGC there is no polynomial-time algorithm that approximates the $K_{n,n}$-QP problem within a factor $K_G - \varepsilon$.

Let $K_{n,n}$-QP$_{\mathsf{PSD}}$ be the special case of the $K_{n,n}$-QP problem where the input matrix (a_{ij}) is assumed to be positive semidefinite. By considering matrices that are Laplacians of graphs, one sees that the MAX CUT problem is a special case of the problem $K_{n,n}$-QP$_{\mathsf{PSD}}$ (see [68]). Hence, due to [53], we know that for every $\varepsilon \in (0,1)$, assuming P $\neq$ NP there is no polynomial-time algorithm that approximates the $K_{n,n}$-QP$_{\mathsf{PSD}}$ problem within ratio $\frac{17}{16} - \varepsilon$. Moreover, it is proved in [68] that for every $\varepsilon \in (0,1)$, if the UGC is assumed, there is no polynomial-time algorithm that approximates the $K_{n,n}$-QP$_{\mathsf{PSD}}$ problem within ratio $\frac{\pi}{2} - \varepsilon$, an optimal hardness result due to the positive semidefinite Grothendieck inequality (4.1). This follows from the more general results for the kernel clustering problem described later.

Let (a_{ij}) be an $n \times n$ real matrix with zeroes on the diagonal. The K_n-QP problem seeks to determine the quantity

$$\max\Big\{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\varepsilon_i\varepsilon_j : \{\varepsilon_i\}_{i=1}^{m} \subseteq \{-1,1\}\Big\}.$$

In [71] it is proved that for every $\gamma \in (0, \frac{1}{6})$, assuming that NP does not have a $2^{(\log n)^{O(1)}}$-time deterministic algorithm, there is no polynomial-time algorithm that approximates the K_n-QP problem within ratio $(\log n)^{\gamma}$. This improves over [12], where a hardness factor of $(\log n)^c$ was proved, under the same complexity assumption, for an unspecified universal constant $c > 0$.

Recall that, as explained in Section 3, there is an algorithm for K_n-QP that achieves a ratio of $O(\log n)$, so there remains an asymptotic gap in our understanding of the complexity of the K_n-QP problem. For the maximum acyclic subgraph problem, as discussed in Section 2.1.3, the gap between the upper and lower bounds is even larger. We have already seen that an approximation factor of $O(\log n)$ is achievable, but from the hardness perspective we know due to [99] that there exists $\varepsilon_0 > 0$ such that assuming P $\neq$ NP there is no polynomial-time algorithm for the maximum acyclic subgraph problem that achieves an approximation ratio less than $1 + \varepsilon_0$. In [49] it was shown that assuming the UGC there is no polynomial-time algorithm for the maximum acyclic subgraph problem that achieves any constant approximation ratio.

Fix $p \in (0, \infty)$. As discussed in Section 5, the L_p Grothendieck problem is as follows: given an $n \times n$ real matrix $A = (a_{ij})$ with zeros on the diagonal, the goal is to determine the quantity $M_p(A)$ defined in (5.1). For $p \in (2, \infty)$ it was shown in [50] that for every $\varepsilon \in (0,1)$, if P $\neq$ NP is assumed, there is no polynomial-time algorithm that approximates the L_p Grothendieck problem within a ratio $\gamma_p^2 - \varepsilon$. Here γ_p is defined as in (5.3). This result (nontrivially) builds on the previous result of [72] that obtained the same conclusion while assuming the UGC rather than P $\neq$ NP.

For the kernel clustering problem with a $k \times k$ hypothesis matrix B, an optimal hardness result is obtained in [69] in terms of the parameters $R(B)$ and $C(B)$ described in Section 4. Specifically, for a fixed $k \times k$ symmetric positive semidefinite matrix B and for every $\varepsilon \in (0,1)$, assuming the UGC there is no polynomial-time algorithm that, given an $n \times n$ matrix A, approximates the quantity $\mathbf{Clust}(A|B)$ within ratio $\frac{R(B)^2}{C(B)} - \varepsilon$. When $B = I_k$ is the $k \times k$ identity matrix, the following hardness result is obtained in [68]: Let $\varepsilon > 0$ be an arbitrarily small constant. Assuming the UGC, there is no polynomial-time algorithm that approximates $\mathbf{Clust}(A|I_2)$ within ratio $\frac{\pi}{2} - \varepsilon$. Similarly, assuming the UGC there is no polynomial-time algorithm that approximates $\mathbf{Clust}(A|I_3)$ within ratio $\frac{16\pi}{27} - \varepsilon$, and, if the solution of the propeller conjecture in $\mathbb{R}^3$ given in [55] is also used, there is no polynomial-time algorithm that approximates $\mathbf{Clust}(A|I_4)$ within ratio $\frac{2\pi}{3} - \varepsilon$. Furthermore, for $k \geq 5$, if the propeller conjecture and the UGC are

assumed, there is no polynomial-time algorithm that approximates $\mathbf{Clust}(A|I_k)$ within ratio $\frac{8\pi}{9}(1-\frac{1}{k})-\varepsilon$.

Acknowledgment. We are grateful to Oded Regev for many helpful suggestions.

S.K. was partially supported by NSF CAREER Grant CCF-0833228, NSF Expeditions Award CCF-0832795, an NSF Waterman award, and BSF Grant 2008059. A.N. was partially supported by NSF Expeditions Award CCF-0832795, BSF Grant 2006009, and the Packard Foundation.

Bibliography

[1] Acín, A.; Gisin, N.; Toner, B. Grothendieck's constant and local models for noisy entangled quantum states. *Phys. Rev. A (3)* **73** (2006), no. 6, part A, 062 105, 5. doi:10.1103/PhysRevA.73.062105

[2] Alon, N. Maximizing a quadratic form on the ℓ_1^n unit ball. Unpublished manuscript, 2006.

[3] Alon, N.; Berger, E. The Grothendieck constant of random and pseudo-random graphs. *Discrete Optim.* **5** (2008), no. 2, 323–327. doi:10.1016/j.disopt.2006.06.004

[4] Alon, N.; Coja-Oghlan, A.; Hàn, H.; Kang, M.; Rödl, V.; Schacht, M. Quasi-randomness and algorithmic regularity for graphs with general degree distributions. *SIAM J. Comput.* **39** (2010), no. 6, 2336–2362. doi:10.1137/070709529

[5] Alon, N.; Duke, R. A.; Lefmann, H.; Rödl, V.; Yuster, R. The algorithmic aspects of the regularity lemma. *J. Algorithms* **16** (1994), no. 1, 80–109. doi:10.1006/jagm.1994.1005

[6] Alon, N.; Fernandez de la Vega, W.; Kannan, R.; Karpinski, M. Random sampling and approximation of MAX-CSPs. *J. Comput. System Sci.* **67** (2003), no. 2, 212–243. doi:10.1016/S0022-0000(03)00008-4

[7] Alon, N.; Makarychev, K.; Makarychev, Y.; Naor, A. Quadratic forms on graphs. *Invent. Math.* **163** (2006), no. 3, 499–522.

[8] Alon, N.; Naor, A. Approximating the cut-norm via Grothendieck's inequality. *SIAM J. Comput.* **35** (2006), no. 4, 787–803 (electronic).

[9] Alon, N.; Orlitsky, A. Repeated communication and Ramsey graphs. *IEEE Trans. Inform. Theory* **41** (1995), no. 5, 1276–1289. doi:10.1109/18.412676

[10] Alon, N.; Spencer, J. H. *The probabilistic method*. 2nd ed. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[11] Arora, S.; Barak, B. *Computational complexity: A modern approach*. Cambridge University Press, Cambridge, 2009.

[12] Arora, S.; Berger, E.; Kindler, G.; Hazan, E.; Safra, S. On non-approximability for quadratic programs. *46th Annual Symposium on Foundations of Computer Science*, 206–215. IEEE Computer Society, Los Alamitos, Calif., 2005.

[13] Bachas, C. P. Computer-intractability of the frustration model of a spin glass. *J. Phys. A* **17** (1984), no. 13, L709–L712. Available at: `http://iopscience.iop.org/0305-4470/17/13/006/pdf/0305-4470_17_13_006.pdf`

[14] Bansal, N.; Blum, A.; Chawla, S. Correlation Clustering. *43rd Annual IEEE Symposium on Foundations of Computer Science*, 238–247. IEEE Computer Society, Los Alamitos, Calif., 2002.

[15] Bansal, N.; Bravyi, S.; Terhal, B. M. Classical approximation schemes for the ground-state energy of quantum and classical Ising spin Hamiltonians on planar graphs. *Quantum Inf. Comput.* **9** (2009), no. 7-8, 701–720.

[16] Bansal, N.; Williams, R. Regularity lemmas and combinatorial algorithms. *2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, 745–754. IEEE Computer Society, Los Alamitos, Calif., 2009. doi:10.1109/FOCS.2009.76

[17] Barahona, F. On the computational complexity of Ising spin glass models. *J. Phys. A* **15** (1982), no. 10, 3241–3253. Available at: http://stacks.iop.org/0305-4470/15/3241

[18] Bhaskara, A.; Vijayaraghavan, A. Approximating matrix p-norms. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, 497–511. SIAM, Philadelphia, 2011.

[19] Bieche, I.; Maynard, R.; Rammal, R.; Uhry, J.-P. On the ground states of the frustration model of a spin glass by a matching method of graph theory. *J. Phys. A* **13** (1980), no. 8, 2553–2576. Available at: http://stacks.iop.org/0305-4470/13/2553

[20] Blei, R. *Analysis in integer and fractional dimensions*. Cambridge Studies in Advanced Mathematics, 71. Cambridge University Press, Cambridge, 2001. doi:10.1017/CBO9780511543012

[21] Blei, R. The Grothendieck inequality revisited. Preprint, 2011. arXiv:1111.7304v1 [math.FA]

[22] Blei, R. C. Multidimensional extensions of the Grothendieck inequality and applications. *Ark. Mat.* **17** (1979), no. 1, 51–68. doi:10.1007/BF02385457

[23] Braverman, M.; Makarychev, K.; Makarychev, Y.; Naor, A. The Grothendieck constant is strictly smaller than Krivine's bound. Preprint, 2011. arXiv:1103.6161v3 [math.FA]

[24] Briët, J.; Buhrman, H.; Toner, B. A generalized Grothendieck inequality and entanglement in XOR games. Preprint, 2009. arXiv:0901.2009v1 [quant-ph]

[25] Briët, J.; de Oliveira Filho, F. M.; Vallentin, F. Grothendieck inequalities for semidefinite programs with rank constraint. Preprint, 2010. arXiv:1011.1754v1 [math.OC]

[26] Briët, J.; de Oliveira Filho, F. M.; Vallentin, F. The positive semidefinite Grothendieck problem with rank constraint. *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, 31–42. Lecture Notes in Computer Science, 6198. Springer, Berlin–Heidelberg–New York, 2010.

[27] Charikar, M.; Guruswami, V.; Wirth, A. Clustering with qualitative information. *J. Comput. System Sci.* **71** (2005), no. 3, 360–383. doi:10.1016/j.jcss.2004.10.012

[28] Charikar, M.; Makarychev, K.; Makarychev, Y. On the advantage over random for maximum acyclic subgraph. *48th Annual IEEE Symposium on Foundations of Computer Science*, 625–633. IEEE Computer Society, Los Alamitos, Calif., 2007.

[29] Charikar, M.; Wirth, A. Maximizing quadratic programs: extending Grothendieck's inequality. *45th Annual Symposium on Foundations of Computer Science*, 54–60. IEEE Computer Society, Los Alamitos, Calif., 2004.

[30] Cleve, R.; Høyer, P.; Toner, B.; Watrous, J. Consequences and limits of nonlocal strategies. *19th Annual IEEE Conference on Computational Complexity*, 236–249. IEEE Computer Society, Los Alamitos, Calif., 2004.

[31] Coja-Oghlan, A.; Cooper, C.; Frieze, A. An efficient sparse regularity concept. *SIAM J. Discrete Math.* **23** (2009/10), no. 4, 2000–2034. doi:10.1137/080730160

[32] Conlon, D.; Fox, J. Bounds for graph regularity and removal lemmas. Preprint, 2011. arXiv:1107.4829v1 [math.CO]

[33] Cook, S. The P versus NP problem. *The millennium prize problems*, 87–104. Clay Mathematics Institute, Cambridge, Mass., 2006.

[34] Davie, A. M. Matrix norms related to Grothendieck's inequality. *Banach spaces (Columbia, Mo., 1984)*, 22–26. Lecture Notes in Mathematics, 1166. Springer, Berlin, 1985. doi:10.1007/BFb0074689

[35] Diestel, J.; Fourie, J. H.; Swart, J. *The metric theory of tensor products: Grothendieck's résumé revisited*. American Mathematical Society, Providence, R.I., 2008.

[36] Diestel, J.; Jarchow, H.; Tonge, A. *Absolutely summing operators*. Cambridge Studies in Advanced Mathematics, 43. Cambridge University Press, Cambridge, 1995. doi:10.1017/CBO9780511526138

[37] Downey, R. G.; Fellows, M. R. *Parameterized complexity*. Monographs in Computer Science. Springer, New York, 1999.

[38] Erdős, P.; Rényi, A. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **5** (1960), 17–61.

[39] Fishburn, P. C.; Reeds, J. A. Bell inequalities, Grothendieck's constant, and root two. *SIAM J. Discrete Math.* **7** (1994), no. 1, 48–56. doi:10.1137/S0895480191219350

[40] Frieze, A.; Kannan, R. Quick approximation to matrices and applications. *Combinatorica* **19** (1999), no. 2, 175–220. doi:10.1007/s004930050052

[41] Garey, M. R.; Johnson, D. S. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.

[42] Garling, D. J. H. *Inequalities: a journey into linear analysis*. Cambridge University Press, Cambridge, 2007. doi:10.1017/CBO9780511755217

[43] Gerke, S.; Steger, A. A characterization for sparse ϵ-regular pairs. *Electron. J. Combin.* **14** (2007), no. 1, Research Paper 4, 12 pp. (electronic). Available at: `http://www.combinatorics.org/Volume_14/Abstracts/v14i1r4.html`

[44] Goemans, M. X.; Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.* **42** (1995), no. 6, 1115–1145. doi:10.1145/227683.227684

[45] Golub, G. H.; Van Loan, C. F. *Matrix computations*. 3rd ed. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, Md., 1996.

[46] Gowers, W. T. Lower bounds of tower type for Szemerédi's uniformity lemma. *Geom. Funct. Anal.* **7** (1997), no. 2, 322–337. doi:10.1007/PL00001621

[47] Grothendieck, A. Résumé de la théorie métrique des produits tensoriels topologiques. *Bol. Soc. Mat. São Paulo* **8** (1953), 1–79.

[48] Grötschel, M.; Lovász, L.; Schrijver, A. *Geometric algorithms and combinatorial optimization: Algorithms and combinatorics*, vol. 2. 2nd ed. Springer, Berlin, 1993.

[49] Guruswami, V.; Manokaran, R.; Raghavendra, P. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. *49th Annual IEEE Symposium on Foundations of Computer Science*, 573–582. IEEE Computer Society, Los Alamitos, Calif., 2008.

[50] Guruswami, V.; Raghavendra, P.; Saket, R.; Wu, Y. Bypassing UGC from some optimal geometric inapproximability results. *Electronic Colloquium on Computational Complexity (ECCC)* **17** (2010), 177.

[51] Haagerup, U. The Grothendieck inequality for bilinear forms on C^*-algebras. *Adv. in Math.* **56** (1985), no. 2, 93–116. doi:10.1016/0001-8708(85)90026-X

[52] Haagerup, U. A new upper bound for the complex Grothendieck constant. *Israel J. Math.* **60** (1987), no. 2, 199–224.

[53] Håstad, J. Some optimal inapproximability results. *J. ACM* **48** (2001), no. 4, 798–859 (electronic).

[54] Håstad, J.; Venkatesh, S. On the advantage over a random assignment. *Random Structures Algorithms* **25** (2004), no. 2, 117–149. doi:10.1002/rsa.20031

[55] Heilman, S.; Jagannath, A.; Naor, A. Solution of the propeller conjecture in $\mathbb{R}^3$. Unpublished manuscript, 2011.

[56] Heydari, H. Quantum correlation and Grothendieck's constant. *J. Phys. A* **39** (2006), no. 38, 11 869–11 875. doi:10.1088/0305-4470/39/38/012

[57] Jameson, G. J. O. *Summing and nuclear norms in Banach space theory*. London Mathematical Society Student Texts, 8. Cambridge University Press, Cambridge, 1987. doi:10.1017/CBO9780511569166

[58] Johnson, W. B.; Lindenstrauss, J. Basic concepts in the geometry of Banach spaces. *Handbook of the geometry of Banach spaces*, vol. I, 1–84. North-Holland, Amsterdam, 2001. doi:10.1016/S1874-5849(01)80003-6

[59] Judin, D. B.; Nemirovskiĭ, A. S. Informational complexity and effective methods for the solution of convex extremal problems. *Èkonom. i Mat. Metody* **12** (1976), no. 2, 357–369.

[60] Juhász, F. The asymptotic behaviour of Lovász' θ function for random graphs. *Combinatorica* **2** (1982), no. 2, 153–155. doi:10.1007/BF02579314

[61] Karger, D.; Motwani, R.; Sudan, M. Approximate graph coloring by semidefinite programming. *J. ACM* **45** (1998), no. 2, 246–265. doi:10.1145/274787.274791

[62] Kashin, B. S.; Sharek, S. Ĭ. On the Gram matrices of systems of uniformly bounded functions. *Tr. Mat. Inst. Steklova* **243** (2003), Funkts. Prostran., Priblizh., Differ. Uravn., 237–243.

[63] Kempe, J.; Kobayashi, H.; Matsumoto, K.; Toner, B.; Vidick, T. Entangled games are hard to approximate. *49th Annual IEEE Symposium on Foundations of Computer Science*, 447–456. IEEE Computer Society, Los Alamitos, Calif., 2008.

[64] Khot, S. On the power of unique 2-prover 1-round games. *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, 767–775. ACM, New York, 2002 (electronic).

[65] Khot, S. On the unique games conjecture (invited survey). *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, 99–121. IEEE Computer Society, Los Alamitos, Calif., 2010.

[66] Khot, S.; Kindler, G.; Mossel, E.; O'Donnell, R. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.* **37** (2007), no. 1, 319–357 (electronic).

[67] Khot, S.; Naor, A. Linear equations modulo 2 and the L_1 diameter of convex bodies. *SIAM J. Comput.* **38** (2008), no. 4, 1448–1463. doi:10.1137/070691140

[68] Khot, S.; Naor, A. Approximate kernel clustering. *Mathematika* **55** (2009), no. 1-2, 129–165. doi:10.1112/S002557930000098X

[69] Khot, S.; Naor, A. Sharp kernel clustering algorithms and their associated Grothendieck inequalities. *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, 664–683. SIAM, Philadelphia, 2011.

[70] Khot, S.; O'Donnell, R. SDP gaps and UGC-hardness for max-cut-gain. *Theory Comput.* **5** (2009), 83–117.

[71] Khot, S.; Safra, S. A two prover one round game with strong soundness. In *52nd Annual IEEE Symposium on Foundations of Computer Science*, in press.

[72] Kindler, G.; Naor, A.; Schechtman, G. The UGC hardness threshold of the L_p Grothendieck problem. *Math. Oper. Res.* **35** (2010), no. 2, 267–283. doi:10.1287/moor.1090.0425

[73] Kohayakawa, Y. Szemerédi's regularity lemma for sparse graphs. *Foundations of computational mathematics (Rio de Janeiro, 1997)*, 216–230. Springer, Berlin, 1997.

[74] Kohayakawa, Y.; Rödl, V. Szemerédi's regularity lemma and quasi-randomness. *Recent advances in algorithms and combinatorics*, 289–351. CMS Books in Mathematics, 11. Springer, New York, 2003. doi:10.1007/0-387-22444-0_9

[75] Kohayakawa, Y.; Rödl, V.; Thoma, L. An optimal algorithm for checking regularity. *SIAM J. Comput.* **32** (2003), no. 5, 1210–1235 (electronic). doi:10.1137/S0097539702408223

[76] König, H. On the complex Grothendieck constant in the n-dimensional case. *Geometry of Banach spaces (Strobl, 1989)*, 181–198. London Mathematical Society Lecture Note Series, 158. Cambridge University Press, Cambridge, 1990.

[77] König, H. Some remarks on the Grothendieck inequality. *General inequalities, 6 (Oberwolfach, 1990)*, 201–206. International Series of Numerical Mathematics, 103. Birkhäuser, Basel, 1992.

[78] König, H. On an extremal problem originating in questions of unconditional convergence. *Recent progress in multivariate approximation (Witten-Bommerholz, 2000)*, 185–192. International Series of Numerical Mathematics, 137. Birkhäuser, Basel, 2001.

[79] Krivine, J.-L. Sur la constante de Grothendieck. *C. R. Acad. Sci. Paris Sér. A-B* **284** (1977), no. 8, A445–A446.

[80] Krivine, J.-L. Constantes de Grothendieck et fonctions de type positif sur les sphères. *Adv. in Math.* **31** (1979), no. 1, 16–30. doi:10.1016/0001-8708(79)90017-3

[81] Laurent, M.; Varvitsiotis, A. Computing the Grothendieck constant of some graph classes. Preprint, 2011. arXiv:1106.2735v1 [math.CO]

[82] Lee, T.; Schechtman, G.; Shraibman, A. Lower bounds on quantum multiparty communication complexity. *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 254–262. IEEE Computer Society, Los Alamitos, Calif., 2009.

[83] Lee, T.; Shraibman, A. Lower bounds in communication complexity. *Found. Trends Theor. Comput. Sci.* **3** (2007), no. 4, front matter, 263–399 (2009). doi:10.1561/0400000040

[84] Lewis, A. S.; Overton, M. L. Eigenvalue optimization. *Acta numerica, 1996*, 149–190. Acta Numerica, 5. Cambridge University Press, Cambridge, 1996.

[85] Lindenstrauss, J.; Pełczyński, A. Absolutely summing operators in L_p-spaces and their applications. *Studia Math.* **29** (1968), 275–326.

[86] Linial, N.; Mendelson, S.; Schechtman, G.; Shraibman, A. Complexity measures of sign matrices. *Combinatorica* **27** (2007), no. 4, 439–463. doi:10.1007/s00493-007-2160-5

[87] Linial, N.; Shraibman, A. Learning complexity vs. communication complexity. *Combin. Probab. Comput.* **18** (2009), no. 1-2, 227–245. doi:10.1017/S0963548308009656

[88] Linial, N.; Shraibman, A. Lower bounds in communication complexity based on factorization norms. *Random Structures Algorithms* **34** (2009), no. 3, 368–394. doi:10.1002/rsa.20232

[89] Lovász, L. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory* **25** (1979), no. 1, 1–7. doi:10.1109/TIT.1979.1055985

[90] Lovász, L.; Plummer, M. D. *Matching theory*. North-Holland Mathematics Studies, 121. Annals of Discrete Mathematics, 29. North-Holland, Amsterdam, 1986.

[91] Lovász, L.; Szegedy, B. Szemerédi's lemma for the analyst. *Geom. Funct. Anal.* **17** (2007), no. 1, 252–270. doi:10.1007/s00039-007-0599-6

[92] Matula, D. W. On the complete subgraphs of a random graph. *Proc. Second Chapel Hill Conf. on Combinatorial Mathematics and Its Applications (Univ. North Carolina, Chapel Hill, N.C., 1970)*, 356–369. University of North Carolina, Chapel Hill, N.C., 1970.

[93] Megretski, A. Relaxations of quadratic programs in operator theory and system analysis. *Systems, approximation, singular integral operators, and related topics (Bordeaux, 2000)*, 365–392. Operator Theory: Advances and Applications, 129. Birkhäuser, Basel, 2001.

[94] Naor, A.; Schechtman, G. An approximation scheme for quadratic form maximization on convex bodies. Unpublished manuscript, 2009.

[95] Nemirovski, A. Advances in convex optimization: Conic programming. *International Congress of Mathematicians, Vol. I*, 413–444. European Mathematical Society, Zürich, 2007. doi:10.4171/022-1/17

[96] Nemirovski, A.; Roos, C.; Terlaky, T. On maximization of quadratic form over intersection of ellipsoids with common center. *Math. Program.* **86** (1999), no. 3, Ser. A, 463–473.

[97] Nesterov, Y. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.* **9** (1998), no. 1-3, 141–160. doi:10.1080/10556789808805690

[98] Nesterov, Y.; Wolkowicz, H.; Ye, Y. Semidefinite programming relaxations of nonconvex quadratic optimization. *Handbook of semidefinite programming*, 361–419. International Series in Operations Research & Management Science, 27. Kluwer, Boston, 2000.

[99] Papadimitriou, C. H.; Yannakakis, M. Optimization, approximation, and complexity classes. *J. Comput. System Sci.* **43** (1991), no. 3, 425–440. doi:10.1016/0022-0000(91)90023-X

[100] Pérez-García, D.; Wolf, M. M.; Palazuelos, C.; Villanueva, I.; Junge, M. Unbounded violation of tripartite Bell inequalities. *Comm. Math. Phys.* **279** (2008), no. 2, 455–486. doi:10.1007/s00220-008-0418-4

[101] Pisier, G. Grothendieck's theorem for noncommutative C^*-algebras, with an appendix on Grothendieck's constants. *J. Funct. Anal.* **29** (1978), no. 3, 397–415.

[102] Pisier, G. *Factorization of linear operators and geometry of Banach spaces*. CBMS Regional Conference Series in Mathematics, 60. American Mathematical Society, Providence, R.I., 1986.

[103] Pisier, G. Grothendieck's theorem, past and present. Preprint, 2011. arXiv:1101.4195v3 [math.FA]

[104] Pitowsky, I. New Bell inequalities for the singlet state: going beyond the Grothendieck bound. *J. Math. Phys.* **49** (2008), no. 1, 012 101, 11. doi:10.1063/1.2826227

[105] Raghavendra, P. Optimal algorithms and inapproximability results for every CSP? *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 245–254. ACM, New York, 2008.

[106] Raghavendra, P.; Steurer, D. Towards computing the Grothendieck constant. *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 525–534. SIAM, Philadelphia, 2008.

[107] Reeds, J. A. A new lower bound on the real Grothendieck constant. Unpublished manuscript, 1991. Available at: `http://www.dtc.umn.edu/reedsj/bound2.dvi`

[108] Regev, O.; Toner, B. Simulating quantum correlations with finite communication. *SIAM J. Comput.* **39** (2009/10), no. 4, 1562–1580. doi:10.1137/080723909

[109] Rietz, R. E. A proof of the Grothendieck inequality. *Israel J. Math.* **19** (1974), 271–276.

[110] Sipser, M. *Introduction to the theory of computation*. PWS Publishing, Boston, 1997.

[111] Smith, R. R. Completely bounded multilinear maps and Grothendieck's inequality. *Bull. London Math. Soc.* **20** (1988), no. 6, 606–612. doi:10.1112/blms/20.6.606

[112] Song, L.; Smola, A.; Gretton, A.; Borgwardt, K. A. A dependence maximization view of clustering. *Proceedings of the 24th International Conference on Machine Learning*, 815–822. ACM, New York, 2007. Available at: `http://www.machinelearning.org/proceedings/icml2007/papers/243.pdf`.

[113] Szemerédi, E. Regular partitions of graphs. *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, 399–401. Colloques Internationaux du Centre National de la Recherche Scientifique, 260. CNRS, Paris, 1978.

[114] Tonge, A. The von Neumann inequality for polynomials in several Hilbert-Schmidt operators. *J. London Math. Soc. (2)* **18** (1978), no. 3, 519–526. doi:10.1112/jlms/s2-18.3.519

[115] Tonge, A. The complex Grothendieck inequality for 2×2 matrices. *Bull. Soc. Math. Grèce (N.S.)* **27** (1986), 133–136.

[116] Tsirelson, B. S. Quantum analogues of Bell's inequalities. The case of two spatially divided domains. Problems of the theory of probability distributions, IX. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)* **142** (1985), 174–194, 200.

[117] Varopoulos, N. T. On an inequality of von Neumann and an application of the metric theory of tensor products to operators theory. *J. Functional Analysis* **16** (1974), 83–100.

SUBHASH KHOT
Courant Institute
251 Mercer St.
New York, NY 10012
E-mail: khot@cims.nyu.edu

ASSAF NAOR
Courant Institute
251 Mercer St.
New York, NY 10012
E-mail: naor@cims.nyu.edu

Received May 2011.