Electronic Colloquium on Computational Complexity, Report No. 70 (2005)

ECCC

On Matrix Rigidity and the Complexity of Linear Forms

Mahdi Cheraghchi*

February 2005

Abstract

The rigidity function of a matrix is defined as the minimum number of its entries that need to be changed in order to reduce the rank of the matrix to below a given parameter. Proving a strong enough lower bound on the rigidity of a matrix implies a nontrivial lower bound on the complexity of any linear circuit computing the set of linear forms associated with it. However, although it is shown that most matrices are rigid enough, no explicit construction of a rigid family of matrices is known.

In this survey report we review the concept of rigidity and some of its interesting variations as well as several notable results related to that. We also show the existence of highly rigid matrices constructed by evaluation of bivariate polynomials over finite fields.

Key words: Matrix Rigidity; Low Level Complexity; Circuit Complexity; Linear Forms.

1 Introduction

One of the major and the most fundamental open problems in theoretical computer science is proving nontrivial lower bounds on the size of algebraic circuits (circuits whose gates can perform algebraic operations such as addition, multiplication, etc.) computing an explicit function. Still there is nothing much to say even for interesting special cases.

Viewed as a directed acyclic graph, the size of a circuit is defined as the sum of the number of nodes and the number of edges in the graph. Moreover, the depth of the circuit is defined as the length of the longest path in the graph. Roughly speaking, there is a tradeoff between the size and the depth of any circuit computing a specific function, namely, as one decreases, the other one grows. A special case of interest is when the depth is restricted to be logarithmic in terms of the number of inputs. In fact, many natural algorithms (e.g., the FFT algorithm for computing Discrete Fourier Transform) achieve this depth.

Linear circuits can be regarded as an interesting restriction of the algebraic circuits. In a linear circuit, only two operations are allowed, namely, binary addition and multiplication by a constant (scalar). In fact, the function that a linear circuit computes can be expressed as a set of linear forms, i.e., the multiplication of a matrix by the input vector. The problem of proving lower bounds on the size of linear circuits deals with the problem of efficiently computing the corresponding matrix by vector multiplication. Unfortunately, proving a nontrivial lower bound on the size of the circuits computing an explicit (and *natural*) set of linear forms is a difficult task, even for the binary field where the circuit is only consisted of binary xor gates.

*Email: ⟨mahdi.cheraghchi@epfl.ch⟩. This report is on a term project the author has been working on as a Masters student of computer science in Laboratoire d'algorithmique (ALGO), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

ISSN 1433-8092

Much effort has been made to relate the complexity of a set of linear forms (i.e., the size and depth of the linear circuit computing it) to other problems that seem to be more tractable. In particular, translating such a computational problem to a combinatorial property (of, say, the corresponding matrix that defines the linear forms) has been of special interest. In 1977, Valiant introduced a combinatorial property of matrices, namely, the rigidity function of the matrix, and showed that proving a strong enough lower bound on this property translates to a nontrivial complexity bound.

The rigidity function of a matrix A, $\mathcal{R}_A(r)$ is defined as the minimum number of entries of A that must be changed to reduce the rank of A to r or less. Therefore, the concept of rigidity carefully combines two combinatorial properties, namely, rank and sparsity, none of which has a complexity implication (e.g., consider the identity matrix which has a full rank, and the all-one matrix that has no zero entries.). Valiant showed that if for any $n \times n$ matrix A, $\mathcal{R}_A(\epsilon n) = \Omega(n^{1+\delta})$, for some positive constants ϵ, δ, $\epsilon < 1$, then the set of linear forms defined by A cannot be computed by a linear circuit of linear size and logarithmic depth. He also showed that over any arbitrary field, *most* matrices satisfy a much stronger condition.

As most matrices are *highly rigid*, it seems that finding explicit families of rigid enough matrices is much easier than directly dealing with the linear circuits. Ironically, after nearly three decades, all efforts to find an explicit family of rigid matrices have failed. The best known lower bound for rigidity of explicit families of matrices is $\Omega((n^2/r)\log(n/r))$, due to several authors [2, 6, 10], that gives a trivial result for $r = \Theta(n)$.

2 Algebraic computation and highly connected graphs

2.1 Straight-line programs

First, we need to formalize our measure of complexity by defining a certain model of computation. Throughout this report we shall focus on an algebraic model called *straight-line programs*, as is the usual case for the arguments on low-level complexity. Informally, in a straight-line program we have a set of variables $\{X_1, \ldots, X_n\}$ taking values in a certain field $\mathbb{F}$, and a sequence of *instructions*. Two disjoint subsets of the variables are identified as *inputs* and *outputs* of the program, and the rest are considered as *intermediate variables*. Any instruction is an assignment in one of the following forms:

1. $X_i \leftarrow \mathrm{F}(X_{j1}, \ldots, X_{jk})$, where F is one of the four basic operations, namely, addition, substraction, multiplication, or division.
2. $X_i \leftarrow k \cdot X_j$, i.e., multiplication by a scalar $k \in \mathbb{F}$.
3. $X_i \leftarrow k$, for some constant $k \in \mathbb{F}$.

Indeed, input (output) variables can only appear on the right (left) hand side of any instruction. Furthermore, the sequence of the instructions must satisfy the property that no variable on the left-hand side of some assignment can occur earlier in the sequence. In other words, *multiple assignments* and *feedbacks* are not allowed. The following [1] is a formal definition of straight-line programs:

Definition 1. Let A be a K-algebra, and $\Omega = k^c \cup k \cup \{+, -, \times, \div\}$ be the set of operations, where the operations denoted by k and k^c stand for scalar multiplication and constant assignment, respectively. Let $a \in A^n$ be an *input of length* n, for some integer n. A *straight-line program* Γ (over K, expecting inputs of length n) is a sequence $(\Gamma_1, \ldots, \Gamma_r)$ of *instructions*

$$\Gamma_i = (\omega_i; u_{i1}, \ldots, u_{i\mathrm{ar}(\omega_i)}),$$

where $\omega_i \in \Omega$, the *arity* (also called *fan-in*) of ω_i is denoted by $\mathrm{ar}(\omega_i)$, and $u_{i\ell}$ are integers satisfying $-n < u_{i1}, \ldots, u_{i\mathrm{ar}(\omega_i)} < i$.

A straight-line program can be thought of as a combinational circuit consisting of a number of *gates* and *wires*. The connections between inputs and outputs of different instructions can thus be represented by a directed acyclic multigraph[1], where each vertex of the graph corresponds to a specific instruction, or equivalently, a combinational gate. We will focus our arguments on two major complexity measures of a straight-line program, namely, its *size* and its *depth*. The *size* of a program is defined as sum of the number of vertices and edges in the corresponding multigraph, and its *depth* denotes the length of the longest path in the multigraph. Observe that the depth of a program is always finite (as is the size), since there are no cycles in its multigraph.

In general, straight-line programs are able to compute any multivariate rational function, and thus may seem to be too strong for our concerns. Here we restrict ourselves to special cases of straight-line programs that seem to be easier to deal with, namely, the ones that compute *linear forms* over (finite or infinite) fields. A *linear form* in indeterminates $x_1, \ldots, x_n$ over a field $\mathbb{F}$ is any expression of the form $\sum_{i=1}^{n} \lambda_i x_i$, where each λ_i is in $\mathbb{F}$. Any linear form can be computed by a *linear program*, which is defined as follows:

Definition 2. A *linear program* over a field $\mathbb{F}$ is a straight-line program with the function set only consisted of scalar multiplication and binary addition over $\mathbb{F}$.

Observe that because of the bounded fan-in, the size of a linear program and the number of vertices and edges of its multigraph representation are of the same order, and thus, can be used interchangeably in asymptotic arguments. With some abuse of notation, we shall assume that any instruction in a linear program $\mathcal{L}$ is of the form $X_i \leftarrow \lambda X_j + \mu X_k$, where λ and μ are constants in $\mathbb{F}$, and the multigraph $\mathcal{G}$ that corresponds to $\mathcal{L}$ is a labeled directed acyclic graph. That is to say, any vertex in $\mathcal{G}$ is a binary addition gate. Scalar multiplications appear as labels of the edges of $\mathcal{G}$, as if they are carried out by the wires of the circuit. By another convention, we remove any edge in $\mathcal{G}$ that corresponds to a multiplication by zero. Note that for the case of a binary field, edge labels of $\mathcal{G}$ carry no information, as they are all one. Thus in that case, a linear program defines a combinational circuit consisting of exclusive-or gates only.

It is immediate from the above definition that any linear program with n input and m output variables computes a set of m linear forms, and can be represented by an $m \times n$ *generator matrix*. Therefore, all that a linear program is capable to perform is nothing but a matrix by vector multiplication, namely, multiplication of its generator matrix by the input vector, which yields the vector of m linear forms at the output. Conversely, any $m \times n$ matrix can be represented by a linear program with n inputs and m outputs. Conveniently yet ambiguously, from now on we might use the term linear form when we actually mean a set of linear forms.

For a given depth d, the *complexity* of a set of linear forms $\mathcal{L}$ is measured by the smallest size of a linear program of depth d that computes $\mathcal{L}$. Thus, resolving the complexity of an explicit linear form answers the question on *how efficiently* one can compute the corresponding matrix by vector multiplication. Interestingly enough, it does not seem to be a weak assumption to restrict ourselves to linear programs when we try to resolve complexity of linear forms. This comes from the fact that for computing a linear form over certain fields (including real and complex numbers), linear programs are optimal within a constant factor as compared with straight-line programs with unrestricted use of all the four operations $\{+, -, \times, \div\}$. This is a special case of the result by Strassen [11].

[1] A multigraph is a graph in which more than one edge is allowed between a pair of vertices.

Typically, the number of input and output variables are the same, given as a parameter, and we wish to analyze the complexity of the linear form for infinite possibilities of n. To that end we will consider *families* of linear forms which are usually defined by a family of $n \times n$ generator matrices, for infinitely many $n \in \mathbb{N}$. This enables us to derive asymptotic lower bounds on the complexity of particular linear transformations.

The problem of proving nontrivial (superlinear) lower bound on the complexity of an explicit family of linear forms seems to be easier to deal with for a restricted model of linear programs where the scalar multipliers have bounded values. In fact, in a seminal work, Morgenstern [7] proved nontrivial lower bounds in such a model. He showed that any linear program for computing linear forms associated with a complex matrix A has size at least $\log_c |\det(A)|$, where c is defined as the maximum of the sum of the absolute values of the two coefficients that appear in any instruction of the program. If we restrict the coefficients to have an absolute value bounded by a constant δ, a lower bound $\log_{2\delta} |\det(A)|$ is obtained for the size of the corresponding restricted linear program. For the case of the Discrete Fourier Transform, this translates to a lower bound of $\frac{1}{2} n \log n$ on the size of any linear circuit computing DFT in which the absolute values of the scalars are restricted to be at most 1 (a condition which is satisfied by the Fast Fourier Transform algorithm.). This follows from the fact that for an $n \times n$ DFT matrix F, $|\det(F)| = n^{n/2}$. Unfortunately, Morgenstern's result can not be easily generalized to the unrestricted model, since we also get the same bound for certain *trivial* families of linear forms, say, nI_n, where I_n is the $n \times n$ identity matrix.

2.2 Examples of graphs with high connectivity

Intuitively, graphs with large sizes are *highly* connected. Thus, it seems natural to look for a formalized notion of high connectivity, inducing a lower bound on the size of the graph. Such a lower bound would immediately imply a lower bound on the complexity of the corresponding straight-line program. In other words, we wish to discover a graph connectivity property that is valid for any linear program computing a certain family of linear forms and subsequently obtain a lower bound on any graph satisfying such a property. To that end, we consider various families of graphs with high connectivity, based on [12].

Definition 3. Let $\mathcal{G}$ be a directed acyclic graph with n input nodes $a_0, \ldots, a_{n-1}$ and n output nodes $b_0, \ldots, b_{n-1}$, and σ be a permutation of the integers $1, \ldots, n$. Then $\mathcal{G}$ *implements* σ iff there are n mutually node disjoint paths joining the n pairs $\{a_i, b_{\sigma(i)} \mid 0 \leqslant i < n\}$.

Such families are known as *connection networks*. Clearly, there are $n!$ possibilities for the permutation vector σ and thus, any graph that implements all possible permutations has to realize $n!$ sets of paths and is of size at least $\log n! = \Omega(n \log n)$. In fact, this order of size can be achieved. A weaker requirement would be to implement only n distinct circular shifts $\{\sigma_i \mid \sigma_i(j) = j + 1 \bmod j,\ 0 \leqslant i \leqslant n-1\}$. In fact, such a graph would also need a size of at least $3n \log n$, as the following theorem suggests:

Theorem 4. [8] *If $\sigma_1, \ldots, \sigma_s$ are any permutations such that $\sigma_i(k) \neq \sigma_j(k)$ (for all $i \neq j, k$), then any graph that implements all the s permutations has to have size at least $3n \log_3 s$.*

However, a size of $3n \log n + O(n)$ turns out to be sufficient to construct such *shifting graphs* [8]. Another similar but important family of highly connected graphs are so-called *superconcentrators*, which are defined as follows:

Definition 5. A directed acyclic graph $\mathcal{G}$ with two disjoint sets of n nodes distinguished as input and output nodes, respectively, is an *n-superconcentrator* iff for all $1 \leqslant r < n$ and all sets A of r distinct input nodes and all sets B of r distinct output nodes, there are r mutually node-disjoint paths connecting nodes of A to nodes of B.

The above definition is similar to that of connection networks, except it is now immaterial which particular input-output pairs of nodes are connected. It has been shown that for any algorithm solving certain problems (notably computing convolutions) a superconcentrator is necessary. Surprisingly and unfortunately for our purpose, superconcentrators do not account for superlinear complexity in general:

Theorem 6. [13] *For all n, there exists a superconcentrator of size $\Theta(n)$.*

The above theorem has been used to disprove the plausible conjecture that the set of linear forms generated by any totally non regular square matrix cannot be computed in linear time:

Theorem 7. [12] *For all $n \in \mathbb{N}$, there exists an $n \times n$ integer matrix A with no singular minors of any size, but such that the n linear forms generated by A can be computed in $O(n)$ time.*

Thus an interesting problem would be to investigate minimal restrictions needed to be imposed on superconcentrators to ensure that they are of superlinear size.

The properties we have considered so far have been based on the existence of sufficient node disjoint paths in the graph. Now we examine a few properties which are based on removal of edges in the graph and may account for the complexity of algorithms.

Definition 8. A directed acyclic graph $\mathcal{G}$ is said to have the property $R(n, m)$ iff whichever set of n edges are removed from $\mathcal{G}$, some directed path of m edges remains in $\mathcal{G}$. We also define $S(n, m, d)$ to be the size of the smallest graph of depth at most d with the $R(n, m)$ property.

The following theorem states a lower bound on the size of graphs defined as above:

Theorem 9. [12] *The quantity $S(n, m, d)$ defined as above is bounded by $S(n, m, d) > \frac{n \log_2 d}{\log_2(d/m)}$.*

This immediately implies the following two corollaries:

Corollary 10. *For any $k > 0$, the depth d of any graph with q edges, $q \leqslant (n \log_2 d)/k$, can be reduced to $d/2^k$ by removing some set of n edges.*

Corollary 11. *The depth of any graph with $d = c(\log_2 n)^{c'}$ and $q < (n \log\log n)/\log\log\log n$ can be reduced to $d/\log\log n$ by removing some set of n edges.*

The special case of $d = \log n$ is of particular interest in complexity arguments, as most efficient algorithms known for certain interesting problem (e.g., discrete Fourier transform) achieve this depth.

The lower bound obtained in Theorem 9 can be improved by imposing an additional restriction on the structure of the graph, namely, restricting the graph to have a *series-parallel* structure. Roughly, series-parallel graphs can be constructed recursively from subgraphs placed in series or in parallel. The following formalizes this intuition.

Definition 12. For a directed acyclic graph $\mathcal{G} = (V, E)$ a *labeling* is a mapping $L: V \to \mathbb{N}$, such that for any directed edge $(u, v) \in E$, $L(v) > L(u)$.

Definition 13. A directed acyclic graph $\mathcal{G}$ with designated sets of input and output nodes is called a *series-parallel graph* (or an sp-graph) iff there is a labeling L for $\mathcal{G}$ such that for all pairs of edges (u, v) and (x, y) in $\mathcal{G}$ we have $(L(u) - L(x))(L(y) - L(v)) \geqslant 0$.

The properties $R(n, m)$ and $S(n, m, d)$ are redefined for sp-graphs as follows:

Definition 14. An sp-graph $\mathcal{G}$ has $R'(n,m)$ property iff whichever set of n edges are removed from $\mathcal{G}$, some directed path of length at least m remains from an input to an output. The quantity $S_{sp}(n,m)$ is defined as the size of the smallest sp-graph with the $R'(n,m)$ property.

According to the definitions above, the following lower bound holds in sp-graphs:

Theorem 15. [12] *For some constnt $c > 0$, $S_{sp}(n,m) \geqslant c \cdot n \cdot \log\log_2 m$.*

Finally, we consider a family of highly connected graphs, namely, *grates*. It turns out later that this family relates the complexity of linear forms to a combinatorial property of matrices.

Definition 16. Let $\mathcal{G}$ be a directed acyclic graph with and f be a mapping over nonnegative integers, $f\colon \mathbb{Z}_{\geqslant 0} \to \mathbb{Z}_{\geqslant 0}$. Then $\mathcal{G}$ is called an *f-grate* iff there exists two disjoint subsets $A = \{a_1 \ldots, a_s\}$ and $B = \{b_1 \ldots, b_t\}$ of the nodes in $\mathcal{G}$ such that the following holds: "If any r nodes (and adjacent edges) are removed from $\mathcal{G}$ then for at least $f(r)$ of the $s \cdot t$ distinct pairs (a_i, b_j) there remains a directed path from a_i to b_j."

If we choose specific values for s and t in the above definition, the restriction will be called (f,s,t)-grate. The following theorem shows a tradeoff between the size and the depth of any grate:

Theorem 17. [12] *For all positive constants ϵ, c, k and all sufficiently large n, any f-grate of indegree two and depth $k\log_2 n$ with $f(n) > cn^{1+\epsilon}$ has size at least $\frac{n\log\log n}{\log\log\log n}$.*

Proof. Assume the contrary. By Corollary 11, a set of n nodes can be removed from any graph of size $\frac{n\log\log n}{\log\log\log n}$ and depth $k\log_2 n$ such that no path of length more than $\frac{k\log n}{\log\log n}$ remains. Therefore, after deletion, each output will be connected to at most $n^{k/\log\log n} = o(n^{\epsilon})$ inputs. This implies that for sufficiently large n the graph is not an f-grate, a contradiction. ∎

In the next section we consider the application of such graphs in proving complexity lower bounds.

3 Rigid matrices

3.1 Rigidity and grates

The concept of grates introduced in [12] is followed by an interesting combinatorial property of the corresponding generator matrix that seems to be more tractable than the direct analysis of straight-line programs, namely, *rigidity*.

Definition 18. The *rigidity* of an $n \times n$ matrix A with entries in a field $\mathbb{F}$ is the function $\mathcal{R}_A(r)\colon \{0,1,\ldots,n\} \to \{0,1,\ldots,n^2\}$ defined by:

$$\mathcal{R}_A(r) = \min\{i \mid \exists B \in \mathbb{F}^{n\times n} \text{ with } \mathrm{wt}(B) = i \text{ and } \mathrm{rank}(A+B) \leqslant r\},$$

where $\mathrm{wt}(B)$ is the number of nonzero entries in B, i.e., $\mathrm{wt}(B) \stackrel{\text{def}}{=} \#\{(i,j) \mid b_{ij} \neq 0\}$.

In other words, the rigidity of a matrix A is defined as the minimum number of changes that need to occur in entries of A to reduce its rank to at most a certain value r. The following theorem relates rigidity of matrices to the grates:

Theorem 19. [12] *The graph of any linear program for computing a set of linear forms Ax is an $(\mathcal{R}_A, n, n)$-grate.*

Proof. Let $\mathcal{G}$ be the graph of a linear program, and let A and B be the set of input and output nodes of $\mathcal{G}$, respectively. We wish to show that the two subsets A and B satisfy the conditions of Definition 16 and thus $\mathcal{G}$ is an $(\mathcal{R}_A, n, n)$-grate. Assume for the sake of contradiction that if a certain set of r nodes ($1 \leqslant r \leqslant n$) are removed from $\mathcal{G}$, then fewer than $\mathcal{R}_A(r)$ input-output pairs remain connected. Define the *weight* of a directed path π in G be the product of scalar multipliers (labels) that appear on the edges of π. Observe that an entry a_{ij} of the generator matrix A is equal to the sum of the weights of all directed paths in $\mathcal{G}$ connecting the input node i to the output node j. Thus the above assumptions imply that if the multipliers at the r removed nodes are changed to zero then the matrix B of the linear form computed by the modified program has a weight less than $\mathcal{R}_A(r)$. On the other hands, it is easy to see that the rows of B differ from the corresponding ones of A only by linear combinations of the forms computed by the original program at the removed nodes. Thus, $A - B = X$, where X is an $n \times n$ matrix of rank at most r. This immediately implies that $\mathrm{wt}(B) \geqslant \mathcal{R}_A(r)$, a contradiction. It follows that $\mathcal{G}$ is an $(\mathcal{R}_A, n, n)$-grate. ∎

Moreover, the assertion in Theorem 3.1 is tight, as the following theorem implies:

Theorem 20. [12] *Let A be any $n \times n$ matrix and a f be any function defined as $f\colon \{0, \ldots, n\} \to \{0, \ldots, n^2\}$. If for some r, $f(r) > \mathcal{R}_A(r)$, then there exists a linear program P for computing Ax whose graph is not an f-grate.*

Proof. Fix a value r for which $f(r) > \mathcal{R}_A(r)$. Let $A + B = C$, where $\mathrm{wt}(B) = \mathcal{R}_A(r)$, and $\mathrm{rank}(C) = r$. Let X be a set of r linearly independent forms defined by C. Construct P such that it first computes $n + r$ linear forms defined by B and X in a naive manner, by forming $n + r$ independent trees. Finally, P computes Ax using these $n + r$ linear forms. Clearly, if the r nodes corresponding to X are removed, n disjoint trees remain in the graph, with the outputs as roots, and $\mathrm{wt}(B)$ input-output connections. Since $\mathrm{wt}(B) = \mathcal{R}_A(r) < f(r)$, it follows that the graph corresponding to P is not an f-grate. ∎

In fact, the two theorems above can be used to relate a computational property of linear programs to a noncomputational problem on matrices.

Theorem 21. [12] *Let $A_1, A_2, \ldots$ be an infinite family of square matrices, where A_n is an $n \times n$ matrix, and for some $c, \epsilon > 0$, $\mathcal{R}_A(\frac{n}{2}) \geqslant cn^{1+\epsilon}$. Then there does not exist a family of linear programs for the corresponding sets of linear forms that (i) achieve linear size and logarithmic depth simultaneously, or (ii) are series-parallel and of size linear in n.*

Proof. The first part is immediate from Theorem 19, Theorem 17 and the second part from Theorem 15. ∎

If the family of matrices is defined over special fields (e.g., real or complex numbers) where the translation from straight-line programs to linear programs changes the size and depth by a constant factor only, the above theorem immediately generalizes to unrestricted straight-line programs.

Definition 22. An infinite family $A_1, A_2, \ldots$ of square matrices (where A_n is an $n \times n$ matrix over some field $\mathbb{F}$) is called *rigid* iff for some constants ϵ and δ such that $0 < \epsilon < 1$ and $\delta > 0$ and for sufficiently large n, $\mathcal{R}_{A_n}(\epsilon n) = \Omega(n^{1+\delta})$.

3.2 Existence of rigid matrices

3.2.1 Generic rigidity

It is immediate from Theorem 21 that computation of any rigid family of matrices by linear programs of logarithmic depth needs a superlinear size. Thus proving superlinear rigidity for

any family of matrices immediately translates to a nontrivial lower bound on the complexity of any family of programs computing the corresponding set of linear forms. At this stage, an important question is whether rigid matrices really exist or not. It can be easily shown that for any $n \times n$ matrix A, $\mathcal{R}_A(r) \leqslant (n-r)^2$, but we are specifically interested to know how close we can get to this upper bound. The following result by Valiant [12] shows that in fact most matrices are highly rigid:

Theorem 23. *For any $n \in \mathbb{N}$ and any field $\mathbb{F}$, there exists an $n \times n$ matrix A over $\mathbb{F}$ such that:*

1. *For $\mathbb{F}$ infinite, $\mathcal{R}_A(r) = (n-r)^2$.*
2. *For $\mathbb{F}$ finite with q elements,*

$$\mathcal{R}_A(r) \geqslant \frac{(n-r)^2 - 2n\log_q 2 - \log_2 n}{2\log_q n + 1}, \text{ for all } r < n - \sqrt{2n\log_q 2 + \log_2 n}.$$

Proof. First, consider a few definitions. A *mask* σ is defined as any subset of s pairs from the set $\{(i,j) \mid 1 \leqslant i,j \leqslant n\}$. In other words, a mask defines an $n \times n$ matrix with the entries chosen from the binary set $\{0,1\}$. A *minor* τ is any pair of nonempty subsets of $\{i \mid 1 \leqslant i \leqslant n\}$. In fact, a minor of a matrix A is obtained by removing any combination of the rows and the columns of A. The quantity $M(\sigma,\tau)$ is also defined to be the set of all $n \times n$ matrices A for which there exists an $n \times n$ matrix B such that all nonzero entries of B are indexed by σ, $\operatorname{rank}(A+B) = t$, and, τ specifies a $t \times t$ minor of maximal rank in C, where $C = A + B$.

Assume without loss of generality that the minor specified by τ is at the top left corner. For any $n \times n$ matrix X, consider writing it in the following block form:

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix},$$

where X_{11} is a $t \times t$ minor.

It is easy to see that for any matrix X of rank t where X_{11} is nonsingular, the entries of X_{22} can be expressed as rational functions in the entries of X_{11}, X_{12}, and X_{21}. Therefore, any possibility of C is uniquely determined by having $(n^2 - (n-t)^2)$ of its entries known. It follows that the entries of any $A \in M(\sigma,\tau)$ are given by a set of n^2 rational functions in terms of $(n^2 - (n-t)^2)$ arguments, i.e., the entries of C_{11}, C_{12}, C_{21}, and the nonzero entries of B. Hence, $M(\sigma,\tau)$ is the image of $\mathbb{F}^{2nt-t^2+s}$ into $\mathbb{F}^{n^2}$ under some rational mapping. For the case of $t = (n-r)^2 - 1$, this implies that all matrices whose rank can be reduced to r by changing $(n-r)^2 - 1$ of their entries belong to the union of the images of a finite number of rational mappings from $\mathbb{F}^{n^2-1}$ to $\mathbb{F}^{n^2}$. It is known that for an infinite field $\mathbb{F}$ and any parameter u, the finite union of the images of $\mathbb{F}^u$ under rational mappings into $\mathbb{F}^{u+1}$ cannot fill $\mathbb{F}^{u+1}$. Thus for an infinite field $\mathbb{F}$ matrices of maximal rigidity $(n-r)^2$ exist, and the result follows.

For the case of a finite field $\mathbb{F}$ with q elements we devise a simple counting argument to show the assertion. Observe that the number of possible choices of the mask σ for fixed values of s and t is $\binom{n^2}{s}$, which is upper bounded by $2^{2s\log_2 n}$. Moreover, the number of choices of the minor τ is $\binom{n}{t}^2$, which is upper bounded by 2^{2n}. Therefore, for fixed s and t, the number of matrices in the union of $M(\sigma,\tau)$ over all choices of σ and τ is upper bounded by $q^{2nt-t^2+s+2s\log_q n + 2n\log_q 2}$. Now picking $t = r$ within the range of our assumption, i.e.,

$$t < n - \sqrt{2n\log_q 2 + \log_2 n},$$

and for any s satisfying the desired bound on the rigidity of A, that is,

$$0 \leqslant s < \frac{(n-r)^2 - 2n\log_q 2 - \log_2 n}{2\log_q n + 1},$$

it is clearly seen that the number of possible matrices is upper bounded by

$$q^{n^2 - \log_q n} = \frac{q^{n^2}}{n},$$

that does not fill up the entire space $\mathbb{F}^{n^2}$. Therefore, highly rigid matrices exist over any finite field. ∎

For the case of finite fields, the above theorem can be generalized as follows:

Theorem 24. *Let $\mathcal{S}_n$ be any set of $n \times n$ matrices defined over a finite field $\mathbb{F}_q$ (q might depend on n) such that $|\mathcal{S}_n| \geqslant q^{pn^2}$, for some constant $0 < p \leqslant 1$. Then $\mathcal{S}_n$ contains highly rigid matrices. More precisely, for large enough n, there exists $A \in \mathcal{S}_n$ and a constant $0 < \epsilon < 1$ such that $\mathcal{R}_A(\epsilon n) = \omega(n^{1+\delta})$, for any positive constant δ such that $0 \leqslant \delta < 1$.*

Proof. The number of $n \times n$ matrices with weight at most w, $W(w)$, is equal to

$$W(w) = \binom{n^2}{w}(q-1)^w \leqslant \left(\frac{eqn^2}{w}\right)^w, \tag{1}$$

where the inequality follows from the fact that $\binom{n}{k} \leqslant \left(\frac{ne}{k}\right)^k$. Similarly, let $R(r)$ be the number of $n \times n$ matrices with rank at most r. Note that any matrix of rank one can be written as the product of a column vector by a row vector, and conversely, such a product produces a rank one matrix. Therefore, for $r = 1$, the following holds:

$$R(1) \leqslant q^{2n}. \tag{2}$$

Moreover, any matrix of rank $r > 0$ can be expressed as the summation of r matrices of rank one. It follows that:

$$R(r) \leqslant \binom{q^{2n}}{r} \leqslant \left(\frac{q^{2n}e}{r}\right)^r. \tag{3}$$

For some positive constants ϵ, c, δ where $\epsilon, \delta < 1$, let $N(\epsilon, c, \delta)$ be the number of $n \times n$ matrices whose rank can be reduced to $r = \epsilon n$ by $w = cn^{1+\delta}$ number of changes. Replacing these values of r and w in (1) and (3) it follows that,

$$\begin{aligned} N(\epsilon, c, \delta) &\leqslant \left(\frac{eqn^2}{w}\right)^w \left(\frac{q^{2n}e}{r}\right)^r \\ &= \left(\frac{eqn^{1-\delta}}{c}\right)^{cn^{1+\delta}} \left(\frac{q^{2n}e}{\epsilon n}\right)^{\epsilon n} \\ &= q^{cn^{1+\delta}\left(1+\log_q \frac{\epsilon n^{1-\delta}}{c}\right)+\epsilon n\left(2n+\log_q \frac{e}{\epsilon n}\right)} \\ &= q^{2\epsilon n^2 + o(n^2)}. \end{aligned} \tag{4}$$

Now let ϵ be any constant less than $\frac{p}{2}$. It follows from (4) that for large enough n and arbitrary positive constants c, δ, $N(\epsilon, c, \delta) < q^{pn^2}$ if $\delta < 1$. Thus, there exists a matrix $A \in \mathcal{S}_n$ which needs arbitrarily close to a quadratic number of changes to have its rank reduced to below ϵn, i.e., $\mathcal{R}_A(\epsilon n) = \omega(n^{1+\delta})$ for any $0 \leqslant \delta < 1$. ∎

An immediate application of Theorem 24 is to matrices defined over a large number of indeterminates. Consider a set of indeterminates $X = \{x_1, x_2, \ldots, x_m\}$ over a finite field $\mathbb{F}_q$, and a set of bijections $F = \{f_1, f_2, \ldots, f_{n^2}\}$, where $m = \Omega(n^2)$, and each f_i is defined as $f_i : \mathbb{F}_q^m \to \mathbb{F}_q$. Then the $n \times n$ matrix A constructed as $a_{ij} = f_{ij}(x_1, \ldots, x_m)$ is highly rigid, i.e., $\mathcal{R}_A(\epsilon n) = \omega(n^{1+\delta})$, for some constant $0 \leqslant \epsilon \leqslant 1$ and any $0 \leqslant \delta < 1$. For the special case that for all i, $f_i(x_1, \ldots, x_m) \stackrel{\text{def}}{=} x_j$, this implies that we can obtain a rigid matrix simply by arranging $\Omega(n^2)$ indeterminates in an $n \times n$ matrix.

3.2.2 Evaluation matrices of low degree polynomials

Here we introduce a special case of Reed-Muller codes and consider its relation to the matrix rigidity.

Definition 25. The *flattening* operator M is defined as a mapping $\mathrm{M} : \mathbb{F}_q^{n \times n} \to \mathbb{F}_q^{n^2}$ such that for any $n \times n$ matrix $A = (a_{ij})_{n \times n}$, $\mathrm{M}(A) \stackrel{\text{def}}{=} (a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, a_{22}, \ldots, a_{2n}, \ldots\ldots a_{nn})$.

In other words, M is an operator that *flattens* a matrix by putting all its elements in order into a vector. Now, our code $\mathcal{C}$ is defined over bivariate polynomial as follows:

Definition 26. Let q be a prime power and $\mathbb{F}_q$ be a field of q elements, namely, $\{e_1 \ldots, e_q\}$. For some constant parameter $0 < p \leqslant 1$, let $d = pq$. For any vector $u = (u_1, \ldots, u_{d^2}) \in \mathbb{F}_q^{d^2}$, define $P_u(x, y) \in \mathbb{F}_q[x, y]$ to be the bivariate polynomial

$$P_u(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{d} \sum_{j=1}^{d} u_{(i \cdot j)} x^{i-1} y^{j-1}$$

of degree $(d-1, d-1)$. For any vector u, define the matrix $G_u = (g_{ij})_{q \times q}$ such that $g_{ij} = P_u(e_i, e_j)$. Similarly, define the matrix $G'_u = (g'_{ij})_{q \times q}$, where

$$g'_{ij} = \begin{cases} P_u(e_i, e_j) & \text{if } i > d \text{ or } j > d \\ u_{ij} & \text{else.} \end{cases}$$

Then the code $\mathcal{C}$ is defined as the mapping $\mathcal{C} : \mathbb{F}_q^{d^2} \to \mathbb{F}_q^{q^2}$ such that $\mathcal{C}(u) = \mathrm{M}(G_u)$. Similarly, consider the code $\mathcal{C}'$ as the mapping $\mathcal{C}' : \mathbb{F}_q^{d^2} \to \mathbb{F}_q^{q^2}$ such that $\mathcal{C}'(u) = \mathrm{M}(G'_u)$.

The following lemma indicates that the codes $\mathcal{C}$ and $\mathcal{C}'$ defined as above (that are clearly linear) have the maximal dimension d^2:

Lemma 27. *For $\mathcal{C}$ and $\mathcal{C}'$ defined as above,* $\dim \ker(\mathcal{C}) = \dim \ker(\mathcal{C}') = 0$.

Proof. This is immediate for $\mathcal{C}'$, as by definition it is a systematic code. Now, consider a vector $u \in \mathbb{F}_q^{d^2}$ such that $\mathcal{C}(u) = 0$. Therefore, u defines a bivariate polynomial P_u such that $P_u(x, y) = 0$ for all values of x and y. If $u \neq 0$, P_u must have the factors $(x - e_i)$ and $(x - e_i)$ for all $i = 1, 2, \ldots, q$ and thus must be of degree at least (q, q). However, this is not possible since $d \leqslant q$ and P_u is of degree $(d-1, d-1)$. It follows that $u = 0$. Therefore only the zero vector is mapped to zero by $\mathcal{C}$ and $\mathcal{C}'$ and their null spaces have zero dimension. ∎

Corollary 28. *If for two vectors u and v, $\mathcal{C}(u) = \mathcal{C}(v)$, then $u = v$. Similarly, if $\mathcal{C}'(u) = \mathcal{C}'(v)$, then $u = v$.*

Proof. Immediate from Lemma 27 and the fact that $\mathcal{C}$ and $\mathcal{C}'$ are linear codes. ∎

Recall that the codewords of $\mathcal{C}$ and $\mathcal{C}'$ were initially defined as matrices, namely, G_u and G'_u. It turns out that the codes $\mathcal{C}$ and $\mathcal{C}'$ suggest a way of constructing possibly rigid matrices.

Theorem 29. *For a finite field $\mathbb{F}_q$ with q elements $\{e_1, \ldots, e_q\}$, and a fixed constant $0 < p \leqslant 1$, consider the set $\mathcal{S}$ of $q \times q$ matrices over $\mathbb{F}_q$, where any $A = (a_{ij})_{q\times q} \in \mathcal{S}$ is defined in such a way that $a_{ij} = P(e_i, e_j)$ for some bivariate polynomial P of degree $(d-1, d-1)$ over $\mathbb{F}_q$, where $d = pq$. Then $\mathcal{S}$ contains highly rigid matrices, i.e., there exists $A \in \mathcal{S}$ such that $\mathcal{R}_A(\epsilon n) = \omega(n^{1+\delta})$, for some constant $0 \leqslant \epsilon \leqslant 1$ and any $0 \leqslant \delta < 1$.*

Proof. Observe that the set $\mathcal{S}$ defined here is in fact the set of matrices G_u (over all possible choices of u) defined in Definition 26, i.e., $\mathcal{S} = \{G_u \mid u \in \mathbb{F}_q^{d^2}\}$. Therefore, Corollary 28 implies that the number of elements in $\mathcal{S}$ is equal to the number of elements in $\mathbb{F}_q^{d^2}$, i.e., $q^{d^2} = q^{p^2q^2}$. Then the result follows immediately from Theorem 24. ∎

Note that a similar assertion as Theorem 29 can be made by considering the set of matrices that correspond to the codewords of $\mathcal{C}'$, i.e., the set $\mathcal{S}' = \{G'_u \mid u \in \mathbb{F}_q^{d^2}\}$ instead of the codewords of $\mathcal{C}$.

One may wonder if we can improve the above result by allowing smaller (and even constant) values for d. Unfortunately, this is not the case, as the following lemma suggests:

Lemma 30. *For any $u = (u_1, \ldots, u_{d^2}) \in \mathbb{F}_q^{d^2}$ and any positive integer $d \leqslant q$, let the corresponding $d \times d$ coefficient matrix $C_u = (c_{ij})_{d\times d}$ be defined such that $c_{ij} = u_{ij}$. Then* $\operatorname{rank}(G_u) = \operatorname{rank}(C_u)$.

Proof. Let the matrix V be defined as

$$V = \begin{pmatrix} 1 & e_1 & e_1^2 & \ldots & e_1^{d-1} \\ 1 & e_2 & e_2^2 & \ldots & e_2^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e_q & e_q^2 & \ldots & e_q^{d-1} \end{pmatrix}_{q\times d},$$

where $\{e_1, \ldots, e_q\}$ is the set of elements in $\mathbb{F}_q$. It is easy to verify that $G_u = V \cdot C_u \cdot V^T$. Note the following inequality for the product of any $n \times m$ matrix A by any $m \times l$ matrix B:

$$\operatorname{rank}(A) + \operatorname{rank}(B) - m \leqslant \operatorname{rank}(A \cdot B) \leqslant \min\{\operatorname{rank}(A), \operatorname{rank}(B)\}.$$

Now the result follows by considering this inequality and observing that the columns of V are linearly independent, i.e., $\operatorname{rank}(V) = d$. ∎

This immediately implies the following corollary:

Corollary 31. *The rank of the matrix G_u defined as above can be at most d.*

Thus, for any fixed $d = o(q)$, as q gets large, the rigidity of G_u tends to zero. More precisely, for any $0 < \epsilon \leqslant 1$ and large enough q, we have that $\operatorname{rank}(G_u) \leqslant d < \epsilon q$ and therefore, $\mathcal{R}_{G_u}(\epsilon q) = 0$. Therefore, there is no hope to find rigid matrices defined over bivariate polynomials of degree d unless $d = \Omega(q)$.

3.3 Variations of rigidity

Unfortunately, obtaining a lower bound on the rigidity of explicit families of matrices that translates to a nontrivial lower bound on the complexity of the corresponding family of linear forms turns out to be much more difficult than what it seems to be. Thus many attempts have been made to define similar but more tractable properties that would also lead to interesting (but probably weaker) complexity results. In this section we review a few of such alternative approaches.

In the following problem, we consider the *norm* of the difference matrix instead of its weight:

Definition 32. Let A be an $n \times n$ complex matrix. The *norm rigidity* $\Delta_A(r)$ is defined as the minimum norm of changes needed in entries of A to reduce its rank to at most r. More precisely,

$$\Delta_A^2(r) \stackrel{\text{def}}{=} \inf_B \left\{ \sum_{i,j} |b_{ij}|^2 : \text{rank}(A+B) \leqslant r \right\}.$$

An interesting restricted version of rigidity is *bounded* rigidity, where the absolute value of the changes are bounded by a given parameter θ, i.e.,

Definition 33. The *bounded rigidity* $\mathcal{R}_A(r,\theta)$ of an $n \times n$ matrix A is defined as

$$\mathcal{R}_A(r,\theta) \stackrel{\text{def}}{=} \min_B \{\text{wt}(B) \mid \text{rank}(A+B) \leqslant r, \forall i,j : |b_{ij}| \leqslant \theta\}.$$

For many interesting explicit examples, a maximal lower bound on the rigidity has been shown when the absolute value of changes is bounded by some constant.

Friedman [2] has proposed a modified definition of rigid matrices, namely *strong rigidity*, by imposing a restriction on the number of entries that can be changed on each row of the matrix. According to his definition, a matrix A is (k,t)-rigid if whenever no more than k entries in each row of A are altered, then A's rank remains at least t. This definition can be rephrased in terms of the linear spaces spanned by linear codes, that is,

Definition 34. A subspace $\mathcal{C} \subset \mathbb{F}^n$ of dimension c is called (k,t)-strongly rigid if every subspace $\mathcal{B}$ spanned by any c vectors $b_1, \ldots, b_c$, each of weight at most k, has $\dim(\mathcal{B} \cap \mathcal{C}) \leqslant \dim(\mathcal{C}) - t$.

Because of the additional restriction, strong rigidity is a simpler property than the original notion of rigidity and seems to be easier to use in giving explicit constructions.

Another variation of rigidity is proposed in [9]. They have generalized the notion of rigidity to a set of matrices of the same size, stacked together as a tensor (three-dimensional matrix). The precise definition is given by the following definitions:

Definition 35. Let t be an $l \times m \times n$ tensor over some fixed field $\mathbb{F}$. For a positive integer k, let $\{e_i^k\}_{i=1}^k$ denote the standard basis of $(\mathbb{F}^k)^*$, i.e., $e_i^k(j) = 1$ for $j = 1$, and $= 0$ for $j \neq i$. The symbol $t_{i,*,*}$ denotes the matrix (slice) consisting of all entries of t with the first coordinate i. Similarly, $t_{i,j,*}$ denotes the vector of all entries with the first two coordinates i and j, i.e., the j^{th} row of the matrix $t_{i,*,*}$. For a triple of vectors $u \in \mathbb{F}^l$, $v \in F^m$, and $w \in F^n$, the product $u \otimes v \otimes w$ is the tensor t with $t_{i,j,k} = u_i v_j w_k$ for $1 \leqslant i \leqslant l$, $1 \leqslant j \leqslant m$, $1 \leqslant k \leqslant n$. The *rank* of a tensor is defined to be the minimal number of rank 1 tensors t^i such that $t = \sum_i t^i$ and 0 if t is the zero tensor.

There are several ways to define a tensor of rank 1. The standard definition is the following:

Definition 36. A tensor t has rank 1 iff for some nonzero vectors u, v, w, $t = u \otimes v \otimes w$.

However, a modified notion of rank-1 tensors has been proposed in [9], as follows:

Definition 37. A tensor t has the *rigidity rank* of 1 iff for some nonzero vectors u, v, w and some i, j, k, either $t = e_i^l \otimes v \otimes w$ (i.e., there exists an i such that all nonzero entries of t are in $t_{i,*,*}$, and matrix rank of $t_{i,*,*}$ is 1.) or $t = u \otimes e_j^m \otimes e_k^n$ (i.e., there exists j, k such that all nonzero entries of t are in $t_{*,j,k}$.).

Now, based on the rigidity rank, define the *rigidity of a tensor* t as follows:

Definition 38. The rigidity of a tensor t is defined as

$$\mathcal{R}_t(r) \stackrel{\text{def}}{=} \min \left\{ |S| : \begin{array}{l} S \subseteq \{1,\ldots,m\} \times \{1,\ldots,n\}, \exists \text{ tensor } s \text{ such that} \\ \forall i, \operatorname{rank}(t_{i,*,*} + s_{i,*,*}) \leqslant r \text{ and} \\ \forall (i,j,k), s_{i,j,k} \neq 0 \Rightarrow (j,k) \in S. \end{array} \right\}.$$

In other words, the rigidity of a tensor is the minimal number of columns in which we have to change the tensor in order to reduce the rank of each slice to r or less. For the special case of a single matrix ($l = 1$) we just get the original notion of matrix rigidity. It seems more likely to obtain a larger lower bound for tensor rigidity than for the original rigidity, since $\mathcal{R}_t(r)$ may be substantially larger than the individual rigidity of the slices, i.e., $\max_i \mathcal{R}_{t_{i,*,*}}$.

4 Rigidity of explicit families of matrices

4.1 Highly regular matrices

Roughly speaking, the rigidity of a matrix is a measure that determines how much its rank *resists* upon an arbitrary sequence of changes in its entries. So in order to find rigid matrices, it seems reasonable to look for inherent and useful properties of the minors. In particular, a plausible conjecture is that any totally regular matrix (i.e., a matrix that contains no singular square minor) is highly rigid. In fact, this conjecture turns out to be false. It is well known that any linear program for computing the set of linear forms defined by a totally regular matrix must be a superconcentrator. The actual reason that makes the conjecture false is the existence of linear sized superconcentrators. More specifically, we have the following:

Theorem 39. [12] *For each n there is an $n \times n$ totally regular matrix A such that,*

$$\mathcal{R}_A\left(\frac{n \log\log\log n}{\log\log n}\right) \leqslant n^{1+O(1/\log\log n)}.$$

One such matrix is the one specified by Theorem 7. In other words, the rank of such a matrix can be reduced to $o(n)$ by changing $o(n^{1+\epsilon})$ elements.

Despite the fact that nonrigid totally regular matrices do exist, still there remains hope to find explicit examples of rigid matrices over the class of totally regular matrices. An interesting attempt has been made in [10] to find explicit examples of totally regular matrices with fairly high rigidity. Their bounds are based on the following combinatorial lemma:

Lemma 40. [10] *Let $\log^2 n \leqslant r \leqslant n/2$, and let n be sufficiently large. If in an $n \times n$ matrix fewer than $(n^2/4r) \cdot \log(n/(r-1))$ entries are marked, then there exists an $r \times r$ submatrix that has not been marked.*

Therefore we have the following corollary:

Corollary 41. *Let A be an $n \times n$ matrix for which the rank of all $t \times t$ minors is $\Omega(t)$ and n be sufficiently large. Then for all $\log^2 n \leqslant r \leqslant n/2$,*

$$\mathcal{R}_A(r) = \Omega\left(\frac{n^2}{r} \log \frac{n}{r}\right).$$

Here is a few examples of explicit families of matrices for which the conditions of Corollary 41 hold:

1. Cauchy matrices: For any n, Consider a field K_n that contains at least $2n$ distinct elements, namely, $x_1,\ldots,x_n,y_1,\ldots,y_n$ such that $x_i + y_j \neq 0$, for all $1 \leqslant i,j \leqslant n$. Then the Cauchy matrix $C_n = (c_{ij})_{n\times n}$ is defined as $c_{ij} \stackrel{\text{def}}{=} 1/(x_i + y_j)$. The Cauchy matrix is known to be totally regular, and hence the Corollary 41 holds for this family.

2. The family of matrices obtained from asymptotically good algebraic geometric codes.

3. The family of $p \times p$ discrete fourier transform (DFT) matrices, where p is constrained to be a prime integer and

$$\mathrm{DFT}_p = \left(\zeta_p^{(i-1)(j-1)}\right)_{1 \leqslant i,j \leqslant n}.$$

Here, ζ_p is defined as the primitive p^{th} complex root of unity.

A slightly weaker lower bound can also be obtained for the family of matrices that are highly regular on average. More precisely, we have the following lemma:

Lemma 42. *Consider any $n \times n$ matrix A for which there exists a constant $\epsilon > 1$ such that for any positive integer $t < n$, a $t \times t$ minor of A picked uniformly at random has the expected rank of at least $\frac{t}{\epsilon}$. Then for any $r \leqslant \frac{n}{2\epsilon}$, $\mathcal{R}_A(r) = \Omega\left(\frac{n^2}{r}\right)$.*

Proof. Let $A + B = C$, where $\mathrm{rank}(C) \leqslant r$, and $\mathrm{wt}(B) = \mathcal{R}_A(r)$. Let $t = 2r\epsilon$. Pick a $t \times t$ minor A_0 of A uniformly at random. Let B_0 and C_0 be the the $t \times t$ minors of B and C that correspond to A_0, respectively. Clearly, $A_0 + B_0 = C_0$. Thus we have,

$$\begin{aligned}
\mathrm{rank}(A_0) &\leqslant \mathrm{rank}(C_0) + \mathrm{rank}(B_0) \\
&\leqslant \mathrm{rank}(C) + \mathrm{rank}(B_0) \\
&\leqslant r + \mathrm{rank}(B_0) \\
&\leqslant r + \mathrm{wt}(B_0).
\end{aligned}$$

Taking expectations, we get:

$$E[\mathrm{rank}(A_0)] \leqslant r + E[\mathrm{wt}(B_0)].$$

But by the assumption, $E[\mathrm{rank}(A_0)] \geqslant t/\epsilon = 2r$. On the other hand, $E[\mathrm{wt}(B_0)] = (t/n)^2 \mathrm{wt}(B)$. Thus,

$$2r \leqslant E[\mathrm{rank}(A_0)] \leqslant r + \frac{4\epsilon^2 r^2}{n^2}\mathcal{R}_A(r).$$

It follows that:

$$\mathcal{R}_A(r) \geqslant \frac{n^2}{4\epsilon^2 r},$$

and the claim follows. ∎

4.2 Generalized Hadamard matrices

Hadamard matrices seem to be good candidates for high rigidity. The usual notion of Hadamard matrices (also called *Sylvester matrices* in this case) is recursively defined as follows:

- $H_1 \stackrel{\text{def}}{=} \begin{pmatrix} 1 \end{pmatrix}$,
- $H_{2n} \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes H_n = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}$.

A generalized Hadamard matrix is defined as follows:

Definition 43. An $n \times n$ matrix H over the field of complex numbers is called a *generalized Hadamard matrix* iff:

1. $|h_{ij}| = 1$, for all $1 \leqslant i, j \leqslant n$,
2. $HH^* = nI_n$, where H^* is the conjugate transpose of H and I_n denotes the $n \times n$ identity matrix. (i.e., $\frac{1}{\sqrt{n}}H$ is unitary.)

In other words, in a generalized Hadamard matrix, all entries have the absolute value of one, and the rows (columns) of the matrix are pairwise orthogonal. Observe that Sylvester matrices are special cases of the Generalized Hadamard matrices, namely, when H has only real entries. Another interesting special case is the Discrete Fourier Transform (DFT) matrix. An $n \times n$ DFT matrix $F_n = (f_{ij})_{n\times n}$ is defined as $f_{ij} \stackrel{\text{def}}{=} \zeta_n^{(i-1)(j-1)}$, where ζ_n is the primitive n^{th} root of unity.

By using spectral methods, the following lower bounds have been obtained in [5] for the rigidity of a generalized Hadamard matrix:

Theorem 44. [5] *Let H be an $n \times n$ generalized Hadamard matrix. Then,*

(i) $\mathcal{R}_H(r) \geqslant max\{n - r, \frac{n^2}{(r+1)^2}\}$.

(ii) *For $\theta \leqslant \frac{n}{r-1}$, $\mathcal{R}_H(r, \theta) \geqslant \frac{n^2\theta}{4(\theta+1)^2}$*

(iii) $\Delta_H(r) = n(n - r)$.

Again based on spectral techniques, Kashin and Razborov [3] improved the above result as follows:

Theorem 45. [3] *Let H be an $n \times n$ generalized Hadamard matrix. Then,*

(i) $\mathcal{R}_H(r) \geqslant \Omega(\frac{n^2}{r})$.

(ii) *For $\theta \geqslant \frac{n}{r}$, $\mathcal{R}_H(r, \theta) \geqslant \Omega(\frac{n^3}{r\theta^2})$.*

They have also shown the following interesting property of the minors of a generalized Hadamard matrix:

Proposition 46. [3] *Let H be an $n \times n$ generalized Hadamard matrix, and H_0 be a random $q \times q$ submatrix of H. Then $E[\text{rank}(H_0)] \geqslant q/8$.*

In fact the first part of Theorem 45 immediately follows from the above proposition and Lemma 42. Moreover, the lower bound obtained in [5] is based on a similar proposition on minors of the Hadamard matrix:

Proposition 47. [5] *For any $u \times v$ submatrix B of an $n \times n$ generalized Hadamard matrix H, $\text{rank}(B) \geqslant uv/n$.*

Now, it is worthy to review several spectral properties of matrices that have been employed in [5, 3] to obtain lower bounds on the rigidity of Hadamard matrices, as it seems that the results can still be improved by using a similar technique.

Definition 48. The *Frobenious norm* $||A||_F$ of a complex matrix A is defined by

$$||A||_F \stackrel{\text{def}}{=} \sqrt{\sum_{i,j} |a_{ij}|^2}.$$

Definition 49. The *Spectral norm* $||A||$ of a matrix A is defined by

$$||A|| \stackrel{\text{def}}{=} \max_{x\neq 0} \frac{||Ax||}{|x|}.$$

Definition 50. For any $n \times n$ matrix A, The i^{th} *singular value*, $\sigma_i(A)$ is defined by $\sigma_i(A) = \sqrt{\lambda_i(AA^*)}$, $1 \leqslant i \leqslant n$, where $\lambda_i(AA^*)$ denotes the i^{th} largest eigenvalue of AA^*.

It is immediate from the definition that for a generalized Hadamard matrix H, $\sigma_i(H) = \sqrt{n}$ for all $1 \leqslant i \leqslant n$.

Proposition 51. [5, 3] *For any $n \times n$ complex matrix A,*

1. *There exist unitary matrices $U, V \in \mathbb{C}^{n\times n}$ such that $U^*AV = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$.*
2. *For $i = 1, \ldots, n$,*
$$\sigma_i(A) = \max_{\dim(S)=i} \min_{0\neq x\in S} \frac{||Ax||}{|x|},$$
where S is an i-dimensional subspace of $\mathbb{C}^n$.
3. *$\operatorname{rank}(A) = r$ iff $\sigma_1(A) \geqslant \cdots \geqslant \sigma_r(A) > \sigma_{r+1}(A) = \cdots = \sigma_n(A) = 0$.*
4. *$||A||_F^2 = \sigma_1^2(A) + \cdots + \sigma_n^2(A)$.*
5. *$||A|| = \sigma_1(A)$.*
6. *For any submatrix B of the matrix A, $\operatorname{rank}(B) \geqslant ||B||_F^2/||A||^2$.*
7. *(Hoffman-Wielandt inequality) For any $n \times n$ complex matrix B,*
$$\sum_{i=1}^{n} (\sigma_i(A) - \sigma_i(B))^2 \leqslant ||A - B||_F^2.$$
8. *If A is symmetric, then*
   (a) *All eigenvalues of A are real,*
   (b) *$\operatorname{Tr}(A) = \lambda_1(A) + \cdots + \lambda_n(A)$,*
   (c) *$||A||_F^2 = \lambda_1^2(A) + \cdots + \lambda_n^2(A)$,*
   (d) *$\operatorname{rank}(A) \geqslant \operatorname{Tr}(A)^2/||A||_F^2$.*

The following propositions on the determinant of the Hadamard matrices may also be useful:

Proposition 52. *(Hadamard Inequality) For any $n \times n$ complex matrix A,*
$$|\det(A)| \leqslant \prod_{j=1}^{n} \sqrt{\sum_{i=1}^{n} |a_{ij}|^2},$$
which is sharp for Hadamard marices, i.e., the absolute value of the determinant of any $n \times n$ generalized Hadamard matrix is equal to $n^{n/2}$.

Proposition 53. *Let H be an $n \times n$ generalized Hadamard matrix, and H_0 be any $(n-1) \times (n-1)$ minor of H. Then H_0 is nonsingular and $|\det(H_0)| = n^{n/2-1}$.*

Proof. Assume that H_0 is obtained by removing the i^{th} row and the j^{th} column of H, $1 \leqslant i, j \leqslant n$. For all $1 \leqslant k \leqslant n$, divide the k^{th} row of H by h_{kj}. Subsequently, for all $1 \leqslant \ell \leqslant n$, $\ell \neq j$, divide the ℓ^{th} column of the matrix by $h_{i\ell}/h_{ij}$. Then the resulting matrix

H' is still a generalized Hadamard matrix for which the i^{th} row and the j^{th} column are all-one vectors, and $\det(H') = \alpha_{ij}\det(H)$, where

$$\alpha_{ij} = \frac{\left(\prod_{k=1}^{n} h_{ij}\right)^n}{\prod_{k=1}^{n}(h_{kj}h_{ik})}.$$

Observe that $|\alpha_{ij}| = 1$. It follows from the orthogonality of the rows (columns) of H' and the fact that the i^{th} row (the j^{th} column) of H_0 is the all-one vector that the sum of the entries of any row (column) of H' is zero, except for the i^{th} row and the j^{th} column where the sum is n. Now, add up all rows of H' (except the i^{th} row) with the i^{th} row to get a matrix H''. Observe that $h''_{ij} = n$, $h''_{ik} = 0$ $(\forall k \neq j)$, $h''_{kj} = 0$ $(\forall k \neq i)$, and the minor H''_0 of H'' obtained by removing its i^{th} row and j^{th} column is in fact H_0. Therefore, $\det(H'') = (-1)^{i+j} n \det(H_0)$. But $\det(H'') = \det(H') = \alpha_{ij}\det(H)$. Therefore,

$$\begin{aligned} |\det(H_0)| &= \left|\tfrac{(-1)^{i+j}\alpha_{ij}\det(H)}{n}\right| \\ &= \left|(-1)^{i+j}\alpha_{ij}\right|\frac{n^{n/2}}{n} \qquad (5) \\ &= n^{n/2-1}, \end{aligned}$$

where (5) follows from Proposition 53. ∎

4.3 Vandermonde matrices

For any vector $x = (x_1, \ldots, x_n)$ over some field $\mathbb{F}$, the Vandermonde matrix V is defined as an $n \times n$ matrix $V = (v_{ij})_{n\times n}$ over $\mathbb{F}$ such that:

$$v_{ij} \stackrel{\text{def}}{=} x_i^{(j-1)}, \quad 1 \leqslant i, j \leqslant n.$$

An interesting special case is when $\mathbb{F} = \mathbb{C}$ and $x_i = \zeta_n^{(i-1)}$, where ζ_n is the n^{th} primitive root of unity and thus we get the Discrete Fourier Transform matrix.

To analyze the rigidity of a Vandermonde matrix, one needs to consider two cases separately, namely, when the x_i are algebraically independent and when they are not. The following is the best known lower bound for the either cases:

Theorem 54. [6] *If the x_i are restricted to be algebraically independent over $\mathbb{Q}$, then, $\mathcal{R}_V(r) \geqslant n(n - cr^2)/2$, where c is a positive constant. In particular, for any arbitrary constant $\delta < 1$, there exists an $\epsilon > 0$ such that for every $r \leqslant \epsilon\sqrt{n}$, $\mathcal{R}_V(r) \geqslant \delta n^2$.*

Theorem 55. [6] *If the x_i are arbitrary but distinct, $\mathcal{R}_V(r) \geqslant (n-r)^2/(r+1)$.*

4.4 Constructive approaches

For certain special families of matrices, it is possible to resolve the rigidity by a constructive argument, i.e., by giving an efficient (polynomial time) algorithm to compute (or bound) the rigidity function. However, such an approach does not lead to a complexity result. Unless proven to be optimal, an algorithm can only be useful for showing low rigidity of a certain family, as it gives an upper bound on the rigidity function.

A constructive method is given in [4] to reduce the rank of the lower triangular all-ones matrix. Let A_n be the $n \times n$ all 1's lower triangular matrix defined over an arbitrary field $\mathbb{F}$. The construction to reduce the rank of A_n to $1 \leqslant r \leqslant n$ is as follows:

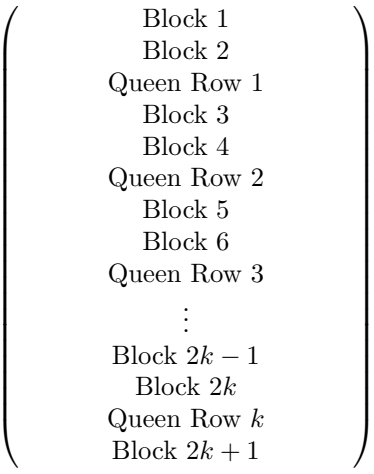

Figure 1: Partitioning the rows of a lower triangular all-ones matrix into blocks and Queen rows.

1. Let $t = (n-k)/(2k+1)$. If t is integral, divide $n-k$ rows of A_n into $2k+1$ groups of t consecutive rows, according to the pattern shown in Figure 1. The rows specified as *Queen rows* will be left unaltered. If t happens to be fractional, divide the $n-k$ non-queen rows as evenly as possible into $2k+1$ blocks, so that any block contains either $\lfloor t \rfloor$ or $\lceil t \rceil$ rows.
2. Change all rows in block 1 to zero.
3. For each row v not in block 1, change v so that it is identical to the closest Queen row (leave the Queen rows unchanged).

For integral t, the above algorithm requires a total of $\frac{1}{2}t(t+1)(2k+1)$ changes. It has been shown in [4] that the method is optimal for integral t, and at least close to optimal otherwise. Thus it follows that,

Corollary 56. *Reducing the rank of the $n \times n$ lower triangular all-ones matrix to $\sqrt{n}$ needs $\Omega(n^{3/2})$ changes.*

Now, it is natural to ask, whether a similar technique can be used to obtain nontrivial statements on the rigidity of other interesting families of matrices. Another question would be, if a *universal* construction exists or not, i.e., whether the problem of computing (or approximating) the rigidity of an arbitrary matrix (over the field of rational numbers or some finite field) is NP-complete or not.

5 Conclusion

The following major question, proposed by Valiant [12], regarding the rigidity function remains open:

Question. [12] *For some natural $n \times n$ matrix A prove that $\mathcal{R}_A(r)$ is large. A bound of $k(n-r)^2$ is one aim. A weaker aim would be one on the value of $\mathcal{R}_A(n/2)$ alone, of kn^2, $kn^{1+\epsilon}$, or some other superlinear function in n. Natural candidates for A are: (i) For the*

integers some Vandermonde matrix, (ii) For the complex numbers the discrete Fourier transform matrix, and (iii) For GF(2)*, the* 0-1 *matrix associated with a finite projective plane.*

We reviewed several explicit candidates for high rigidity. However, one may think of many other families that seem to be as good for high rigidity. For instance, we can mention a Toeplitz matrix (where the entries along any negative-sloping diagonal are the same), or a circulant matrix (where the i^{th} row ($i > 1$) is obtained by a circular right shift of the $(i-1)^{\text{th}}$ row by one.).

However, a weaker question would be to improve upon the best known lower bound $(n^2/r)\log(n/r)$ for the rigidity of explicit matrices. A careful observation reveals that most current analyses are merely based on a high rank assumption on the minors of the matrix. However, Theorem 39 says that there exists totally regular matrices with low rigidity. Thus, having minors with high (or even full) ranks by itself implies nothing useful on the rigidity function. So to obtain a strong enough bound on the rigidity, one needs to make a sharp analysis by investigating the special structure of the matrix more carefully and deeply.

Finally, another approach could be to revisit the original problem of proving nontrivial lower bounds on the complexity of linear forms and to see if it can be related to a combinatorial property which is potentially more promising and more tractable than matrix rigidity.

Acknowledgement

I am indebted to Amin Shokrollahi for introducing the exciting problem of matrix rigidity to me and guiding me through my work on this project.

References

[1] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory,* Grundlehren der mathematischen Wissenschaften, Vol. 315, Springer, Berlin, 1997.

[2] J. Friedman. A note on matrix rigidity. *Combinatorica,* 13(2);235-239, 1993.

[3] B. Kashin and A. A. Razborov. Improved lower bounds on the rigidity of Hadamard matrices. *Mathematical Notes,* 63(4):471-475, 1998.

[4] P. Kimmel and A. Settle. Reducing the rank of lower triangular all-ones matrices. *Technical Report,* CS 92-21, University of Chicago, 1992.

[5] S. V. Lokam. Spectral methods for matrix rigidity with applications to size-depth tradeoffs and communication complexity. *Journal of Computer and System Sciences,* 63:449-473, 2001.

[6] S. V. Lokam. Note on the rigidity of Vandermonde matrices. *Theoretical Computer Science,* 237(1-2):477-483, 2000.

[7] J. Morgenstern. The linear complexity of computation. *Journal of the ACM,* 22(2):184-194, 1975.

[8] N. Pippenger and L. G. Valiant. Shifting graphs and their applications. *Journal of the ACM,* 23:423-432, 1976.

[9] P. Pudlák and V. Rödl. Modified ranks of tensors and the size of circuits. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC),* pages 523-531, 1993.

[10] M. A. Shokrollahi, D. A. Spielman, and V. Stemann. A remark on matrix rigidity. *Information Processing Letters,* 64(6):283-285, 1997.

[11] V. Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.,* 264:184-202, 1973.

[12] L. G. Valiant. Graph theoretic arguments in low-level complexity. *Lecture Notes in Computer Science, Springer, Berlin,* 53:162-176, 1977.

[13] L. G. Valiant. On nonlinear lower bounds in computational complexity. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing (STOC),* pages 45-53, 1975.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'