

Matrix Rigidity from the Viewpoint of Parameterized Complexity

Fedor V. Fomin¹, Daniel Lokshtanov², S. M. Meesum³,
Saket Saurabh⁴, and Meirav Zehavi⁵

- 1 University of Bergen, Bergen, Norway
fomin@ii.uib.no
- 2 University of Bergen, Bergen, Norway
daniello@ii.uib.no
- 3 The Institute of Mathematical Sciences, Chennai, India
meesum@imsc.res.in
- 4 University of Bergen, Bergen, Norway; and
The Institute of Mathematical Sciences, Chennai, India
saket@imsc.res.in
- 5 University of Bergen, Bergen, Norway
meirav.zehavi@ii.uib.no

Abstract

The *rigidity* of a matrix A for a target rank r over a field \mathbb{F} is the minimum Hamming distance between A and a matrix of rank at most r . Rigidity is a classical concept in Computational Complexity Theory: constructions of rigid matrices are known to imply lower bounds of significant importance relating to arithmetic circuits. Yet, from the viewpoint of Parameterized Complexity, the study of central properties of matrices in general, and of the rigidity of a matrix in particular, has been neglected. In this paper, we conduct a comprehensive study of different aspects of the computation of the rigidity of *general matrices* in the framework of Parameterized Complexity. Naturally, given parameters r and k , the MATRIX RIGIDITY problem asks whether the rigidity of A for the target rank r is at most k . We show that in case $\mathbb{F} = \mathbb{R}$ or \mathbb{F} is any finite field, this problem is fixed-parameter tractable with respect to $k + r$. To this end, we present a dimension reduction procedure, which may be a valuable primitive in future studies of problems of this nature. We also employ central tools in Real Algebraic Geometry, which are not well known in Parameterized Complexity, as a black box. In particular, we view the output of our dimension reduction procedure as an algebraic variety. Our main results are complemented by a $W[1]$ -hardness result and a subexponential-time parameterized algorithm for a special case of MATRIX RIGIDITY, highlighting the different flavors of this problem.

1998 ACM Subject Classification G.2.2 Graph Algorithms, I.1.2 Analysis of Algorithms

Keywords and phrases Matrix Rigidity, Parameterized Complexity, Linear Algebra

Digital Object Identifier 10.4230/LIPIcs.STACS.2017.32

1 Introduction

The *rigidity of a matrix* is a classical concept in Computational Complexity Theory, which was introduced by Grigoriev [7, 8] in 1976 and by Valiant [22] in 1977. Constructions of rigid matrices are known, for instance, to imply lower bounds of significant importance relating to arithmetic circuits. Yet, from the viewpoint of Parameterized Complexity, the study of central properties of matrices in general, and of the rigidity of a matrix in particular, has been neglected. The few papers that do consider such properties are restricted to the very



© Fedor V. Fomin, Daniel Lokshtanov, S. M. Meesum, Saket Saurabh, and Meirav Zehavi;
licensed under Creative Commons License CC-BY

34th Symposium on Theoretical Aspects of Computer Science (STACS 2017).

Editors: Heribert Vollmer and Brigitte Vallée; Article No. 32; pp. 32:1–32:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

special case of adjacency matrices, and therefore they are primarily studied in Graph Theory rather than Matrix Theory [16, 17]. In this paper, we conduct a comprehensive study of different aspects of the computation of the rigidity of *general matrices* in the framework of Parameterized Complexity.

Formally, given a matrix A over a field \mathbb{F} , the rigidity of A , denoted by $\mathcal{R}_A^{\mathbb{F}}(r)$, is defined as the minimum Hamming distance between A and a matrix of rank at most r . In other words, $\mathcal{R}_A^{\mathbb{F}}(r)$ is the minimum number of entries in A that need to be edited in order to obtain a matrix of rank at most r . Naturally, given parameters r and k , the MATRIX RIGIDITY problem asks whether $\mathcal{R}_A^{\mathbb{F}}(r) \leq k$. The case when $\mathbb{F} = \mathbb{Q}$ or the edited entries must contain integers, it is not even known whether the problem is decidable [19]. We therefore focus on the cases where $\mathbb{F} = \mathbb{R}$ or \mathbb{F} is any finite field. Formally, we study the following forms of MATRIX RIGIDITY. Here, FF MATRIX RIGIDITY is not restricted to a specific finite field \mathbb{F}_p , but includes \mathbb{F}_p as part of the input.

REAL MATRIX RIGIDITY

Parameter: r, k

Input: A matrix A with each entry an integer, and two non-negative integers r, k .

Question: Is $\mathcal{R}_A^{\mathbb{R}}(r) \leq k$?

FF MATRIX RIGIDITY

Parameter: p, r, k

Input: A finite field \mathbb{F}_p of order p , a matrix A over \mathbb{F}_p , and two non-negative integers r, k .

Question: Is $\mathcal{R}_A^{\mathbb{F}_p}(r) \leq k$?

Valiant [22] presented the notion of the rigidity of a matrix as a means to prove lower bounds for linear algebraic circuits. He showed that the existence of an $n \times n$ matrix A with $\mathcal{R}_A^{\mathbb{R}}(en) \geq n^{1+\delta}$ would imply that the linear transformation defined by A cannot be computed by any arithmetic circuit having size $\mathcal{O}(n)$ and depth $\mathcal{O}(\log n)$ in which each gate of the circuit computes a linear combination of its inputs. Later, Razborov [18] (see [14]) established relations between lower bounds on rigidity of matrices over the reals or finite fields and strong separation results in Communication Complexity. Although many efforts have been made in this direction [6, 20, 12, 10] (this is not an exhaustive list), proofs of separation lower bounds (quadratic) for explicit families of matrices still remains elusive. For a recent survey on this topic we refer the reader to [13]. The formulation of the MATRIX RIGIDITY as stated in this paper was first considered by Mahajan and Sharma [15], and it was shown to be NP-Hard for any field by Deshpande [4]. In this paper, we study the concept of the rigidity of a matrix from a different perspective, given by the framework of Parameterized Complexity (see Section 2).

We remark that Meesum et al. [16] and Meesum and Saurabh [17] studied the following problems, which are related to MATRIX RIGIDITY but are simpler than it as they are restricted to graphs. Given a graph $G = (V, E)$ and two non-negative integers r, k , r -RANK VERTEX DELETION (r -RANK EDGE DELETION) asks whether one can delete at most k vertices (resp. edges) from G so that the rank of its adjacency matrix would be at most r , while r -RANK EDGE EDITING asks whether one can edit k edges in G so that the rank of its adjacency matrix would be at most r .¹ For undirected graphs, Meesum et al. [16] proved that these problems are NP-Hard even if r is fixed, but can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(k \log r)})$. They also showed that r -RANK EDGE DELETION and r -RANK EDGE EDITING can be solved

¹ Editing an edge $\{u, v\}$ means that if $\{u, v\} \in E$ then $\{u, v\}$ is deleted, and otherwise it is added.

in time $\mathcal{O}^*(2^{\mathcal{O}(f(r)\sqrt{k}\log k)})$. Meesum and Saurabh [17] obtained similar results for directed graphs.

Our Contribution. In this paper, we establish that both REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY are FPT with respect to $r+k$. Specifically, we obtain the following results.

► **Theorem 1.** REAL MATRIX RIGIDITY can be solved in time $\mathcal{O}^*(2^{\mathcal{O}((r+k)\cdot\log(r\cdot k))})$.

► **Theorem 2.** FF MATRIX RIGIDITY can be solved in time $\mathcal{O}^*(f(r, k))$ for a function f that depends only on r and k .

Observe that the dependency of the running times on the dimension of the input matrix is polynomial, and in the case of FF MATRIX RIGIDITY, the dependency of the running time on p is also polynomial. In the case of REAL MATRIX RIGIDITY, the dependency of the running time on the maximum length (in binary) of any entry in *both* input and output matrices is polynomial. In this context, recall that in case $\mathbb{F} = \mathbb{Q}$ or the edited entries must contain integers, it is not even known whether MATRIX RIGIDITY is decidable [19]. We also show that,

► **Theorem 3** (\star^2). FF MATRIX RIGIDITY is solvable in time $\mathcal{O}^*(2^{\mathcal{O}(f(r,p)\sqrt{k}\log k)})$ for some function f that depends only on r and p .

Here, the dependency of the running time on k is subexponential, but the dependency of the running time on p is unsatisfactory in case p is not fixed. This algorithm adapts ideas from the papers [16, 17].

To obtain our main results, we first present a dimension reduction procedure, which we believe to be a valuable primitive in future studies of problems of this nature. Our procedure is simple to describe and given an instance of MATRIX RIGIDITY, it outputs (in polynomial time) an equivalent instance where the matrix contains at most $\mathcal{O}((r \cdot k)^2)$ entries. Furthermore, the set of entries of the output matrix is a subset of the set of entries of the input matrix. We believe this procedure to be of interest independent of our main results as it establishes that FF MATRIX RIGIDITY admits a polynomial kernel with respect to $r+k+p$. The simplicity of our procedure also stems from its modularity—it handles rows and columns in separate phases. On a high-level, this procedure is defined as follows. For $k+1$ steps, it repeatedly selects a set of maximum size consisting of rows that are linearly independent, where if the size of this set exceeds $r+1$, it is replaced by a subset of size exactly $r+1$. Each such set of rows is removed from the input matrix, and then it is inserted into the output matrix. At the end of this greedy process, rows that remain in the input matrix are simply discarded. The correctness of our procedure relies on two key insights: (i) if the input instance contains more than $k+1$ pairwise-disjoint sets of rows that are linearly independent, and each of these sets is of size at least $r+1$, then the input instance is a NO-instance; (ii) by the pigeonhole principle, any row discarded from the input matrix belongs to the span of at least one set of rows that cannot be edited. Having an intermediate matrix with a small number of rows, the procedure applies the exact same process to the input that is the transpose of this intermediate matrix, thus overall obtaining a matrix with a small number of entries.

Armed with our dimension reduction procedure, we tackle REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY by employing central tools in Algebraic Geometry, which are not well

² Due to space constraints, proofs marked by \star are omitted.

known in Parameterized Complexity, as a black box. For this purpose, we first recall that the rank of a matrix is at most r if and only if the determinant of all of its $(r + 1) \times (r + 1)$ submatrices is 0. Since at this point we can assume that we have a matrix containing only $\mathcal{O}((r \cdot k)^2)$ entries at hand, we may “guess” *which* entries should be edited. Yet, it is not clear *how* these entries should be edited. However, with the above observation in mind, we are able to proceed by viewing our current problem in terms of an algebraic variety (such a formulation was also used in the context of complexity analysis in [19]). In particular, this viewpoint gives rise to the applicability of firmly established tools that determine the feasibility of a system of polynomials [1, 9].

Our main results are complemented by a $W[1]$ -hardness result and a subexponential-time parameterized algorithm for a special case of MATRIX RIGIDITY, which overall present the different flavors of this problem and the techniques relevant to its study. We show that both REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY are $W[1]$ -hard with respect to the parameter k . (The papers [16, 17] already imply that both of these problems are para-NP-Hard with respect to the parameter r .) Our reduction is inspired by studies in Parameterized Complexity that involve the ODD SET problem [5], and consists of four reductions, one of which builds upon the recent work of Bonnet et al. [2].

The complexity of our reduction stems from the fact that unlike previous studies of this nature, we establish the $W[1]$ -hardness of our problem of interest over *any* finite field and over the field of reals rather than only over a specific finite field. Thus, we first need to define a special case of ODD SET, which we call PARTITIONED ODD SET, and observe that its $W[1]$ -hardness follows from the proof of the $W[1]$ -hardness of ODD SET that is given in [5]. The correctness of our reductions crucially relies on the implications of the properties of this special case. Our first reduction translates PARTITIONED ODD SET to a problem involving matrices rather than sets, which we call PARTITIONED ODD MATRIX. Then, to be able to discuss any finite field as well as the field of reals, we introduce new variants of PARTITIONED ODD MATRIX and the NEAREST CODEWORD problem, called \mathbb{F} -ODD MATRIX and \mathbb{F} -NEAREST CODEWORD, respectively. The application of our second reduction results in an instance of \mathbb{F} -ODD MATRIX. Then, the application of our third reduction, which builds upon [2], results in an instance of \mathbb{F} -NEAREST CODEWORD. Finally, we devise a reduction whose application results in an instance of MATRIX RIGIDITY. Here, we make explicit use of the fact that the rank of the target matrix can be large. The overall structure of the reduction may be relevant to studies of other problems where the field is not fixed.

2 Preliminaries

Due to space constraints, some standard notations and definitions have been omitted. The notation $[n]$ is used to denote the set of integers $\{1, \dots, n\}$. Given a matrix A of dimension $m \times n$, the i -th row of A is denoted by A_i , and the j -th column of A is denoted by A^j . The *rank* of a matrix is the cardinality of a maximum-sized collection of linearly independent columns (or rows), and is denoted by $\text{rank}(A)$. We call a matrix \tilde{A} a *jumbled matrix* of A if one can perform a series of row and column exchanges on \tilde{A} to obtain the matrix A . Similarly, we call a matrix \tilde{A} a *jumbled submatrix* of A if there exists a submatrix of A which is a *jumbled matrix* of \tilde{A} . A *mixed matrix* is a matrix having either an indeterminate or a value at each entry. We will be dealing with mixed matrices where the values belong to a finite field or \mathbb{R} . We use I_n to denote the *identity matrix* of size $n \times n$.

Let x_1, \dots, x_n be variables. Then, a *monomial* is defined as a product $\prod_{i=1}^n x_i^{a_i}$ for non-negative integers a_1, \dots, a_n . The degree of a variable x_i in a monomial $\prod_{i=1}^n x_i^{a_i}$ is

defined to be the number a_i , for $i \in [n]$. The degree of a *monomial* is defined as the sum of degrees of each variable occurring in it. A polynomial over a field \mathbb{F} consists of a sum of monomials with coefficients from the field \mathbb{F} . The *total degree* of a polynomial is the degree of a monomial having maximum degree. Given a system of polynomial equations $\mathcal{P} = \{P_1 = 0, P_2 = 0, \dots, P_m = 0\}$ over a field \mathbb{F} , we say that \mathcal{P} is *feasible* over \mathbb{F} if there exists an assignment of values from the field \mathbb{F} to the variables in \mathcal{P} which satisfies every polynomial contained in \mathcal{P} .

Parameterized Complexity. Each problem instance is associated with a parameter k , and we say that a problem is *fixed parameter tractable* (FPT) if any instance (I, k) of the problem can be solved in time $\tau(k)|I|^{\mathcal{O}(1)}$, where τ is an arbitrary function of k . Throughout this paper, we use the standard notation \mathcal{O}^* to hide factors polynomial in $|I|$.³ On the one hand, to prove that a problem is FPT, it is possible to give an explicit algorithm of the required form, called a *parameterized algorithm*, which solves it. On the other hand, to show that a problem is unlikely to be FPT, it is possible to use parameterized reductions analogous to those employed in Classical Complexity. Here, the concept of W[1]-hardness replaces the one of NP-hardness, and we need not only construct an equivalent instance in FPT time, but also ensure that the size of the parameter in the new instance depends only on the size of the parameter in the original instance. For our purposes, it is sufficient to note that if there exists such a reduction transforming a problem known to be W[1]-hard to another problem Π , then the problem Π is W[1]-hard as well.

A central notion in Parameterized Complexity is the one of *kernelization*. Formally, a parameterized problem Π is said to admit a *polynomial kernel* if there is a polynomial-time algorithm, called a *kernelization algorithm*, that given any instance of Π , outputs an equivalent instance of Π whose size is bounded by $\tau(k)$, where τ is a function polynomial in k and independent of $|I|$. We say that the reduced instance is a $p(k)$ -kernel for Π . Roughly speaking, a kernelization algorithm can be viewed as an efficient preprocessing procedure that satisfies a well defined restriction with respect to the size of its output. For more information about Parameterized Complexity in general and Kernelization in particular, we refer the reader to monographs such as [5, 3].

3 Dimension Reduction Procedure

In this section we show how to compress an input instance of MATRIX RIGIDITY to an equivalent instance in which the matrix has at most $\mathcal{O}(r^2 \cdot k^2)$ entries. This is a crucial step in obtaining our FPT algorithms for REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY. In particular, this step will imply that FF MATRIX RIGIDITY admits a polynomial kernel with respect to $r + k + p$.

Our algorithm is based on the following intuition. Suppose that A is a matrix of rank ℓ . If we could obtain a sequence B_1, \dots, B_k of pairwise disjoint sets of columns of A where each set forms a column basis of A , then the answer to the question “can we reduce the rank of A to a number $r < \ell$ by editing at most k entries in A ” would have been completely determined by the answer to the same question where the editing operations are restricted to the submatrix of A formed by columns in the sets B_1, \dots, B_k . The same conclusion is also true in the case where each B_i is not necessarily a basis, but simply a set of $r + 1$ linearly independent columns. Keeping this intuition in mind, we turn to examine an approach where

³ That is, $\mathcal{O}^*(\tau(k)) = \tau(k) \cdot |I|^{\mathcal{O}(1)}$.

we greedily select and remove (one-by-one) $k + 1$ pairwise disjoint sets of linearly independent columns. In each iteration, we attempt to select a set whose size is exactly $r + 1$, where if it is not possible, we select a set of maximum size.

Now, let us move to the formal part of our arguments. Note that the relation “is a *jumbled matrix* of” as defined in Section 2 is an equivalence relation. We need the following simple observations which follow from the definition of the rank of a matrix.

► **Observation 4.** *Let $A \in \mathbb{F}^{m \times n}$ be a matrix of rank equal to r . To make the rank of A at most $r - 1$, one needs to change at least one entry in A .*

► **Observation 5.** *For a matrix A , let \tilde{A} be a jumbled matrix of A . Then, the instances (A, r, k) , (A^T, r, k) , (\tilde{A}, r, k) and (\tilde{A}^T, r, k) are equivalent instances of MATRIX RIGIDITY.*

► **Observation 6.** *For a matrix A , let \tilde{A} be its jumbled submatrix. Then $\text{rank}(\tilde{A}) \leq \text{rank}(A)$.*

Using Observation 6, we have the following.

► **Observation 7.** *If \tilde{A} is a jumbled submatrix of A and (\tilde{A}, r, k) is a NO-instance of MATRIX RIGIDITY, then (A, r, k) is also a NO-instance of MATRIX RIGIDITY.*

As stated before, our procedure greedily selects a set of columns of A of appropriate dimension iteratively. A detailed description of the procedure, called COLUMN-REDUCTION, can be found in Figure 1. We will now explain the ideas necessary to understand this procedure, which is the heart of this section. The input to COLUMN-REDUCTION consists of a matrix A over any field, given along with non-negative integers k and r . It outputs a matrix \tilde{A} whose number of columns is bounded by a function of k and r such that the instances (A, r, k) and (\tilde{A}, r, k) are equivalent instances of MATRIX RIGIDITY. The computation of a column basis and linearly independent vectors are done in the field \mathbb{F} over which the matrix A is provided.

The procedure employs several variables. The variable i is used as an index variable whose initial value is 0, and it is incremented by 1 at a time. The case when the value of i exceeds k we will show that we are dealing with a NO-instance, otherwise the value depends on a particular input matrix A and is at most k . The variables M_0, M_1, \dots are submatrices of the input matrix A , satisfying the property that M_i is a submatrix of M_{i-1} with $M_0 = A$. In the first loop of COLUMN-REDUCTION (line 3), if the matrix M_i has rank at least $r + 1$ then the variable L_i stores a set of $r + 1$ linearly independent columns in the matrix M_i . Additionally, M_i can be obtained by appending the columns in L_i to the matrix M_{i+1} . The variable $i_{\leq r}$ is set to the value of i where the rank of M_i falls below $r + 1$ —after its initialization the value of $i_{\leq r}$ is not changed. In the second half of the procedure, similar to the set of variables L_i , we define a set of variables B_i which store a column basis of the matrix M_i (line 6). Recall that in this half of the procedure $i \geq i_{\leq r}$, and therefore each matrix M_i is of rank at most r . Additionally, M_i can be obtained by appending the columns in B_i to the matrix M_{i+1} , for $i \geq i_{\leq r}$. Finally, the matrix \mathcal{L} is constructed using all the columns in each matrix L_i , and the matrix \mathcal{B} is constructed using all the columns in each matrix B_i for appropriate values of i . By Observation 4, we have to edit at least $i_{\leq r}$ entries of \mathcal{L} to make its rank at most r .

In the procedure COLUMN-REDUCTION, a YES-instance of appropriate size can be obtained by taking the matrix $Z = [0]$ (of rank 0), which contains 0 as its only entry. Clearly, (Z, r, k) is a YES-instance of MATRIX RIGIDITY irrespective of the values of r and k . On the other hand, the instance (I_{r+k+1}, r, k) is a NO-instance of MATRIX RIGIDITY. Therefore, the matrix I_{r+k+1} can be used in place of a NO-instance of appropriate size. We need Z and

Algorithm: COLUMN-REDUCTION

Input: A matrix A over some field \mathbb{F} , and two non-negative integers r, k .

Output: A matrix having $\mathcal{O}(r \cdot k)$ columns.

1. If $\text{rank}(A) \leq r$ then return a YES-instance of appropriate size and exit.
2. Initialize $M_0 = A$ and $i = 0$.
3. While $\text{rank}(M_i) \geq r + 1$:
 - a. Let L_i be a set of columns of M_i which is linearly independent in \mathbb{F} and whose size is $r + 1$.
 - b. Let M_{i+1} be the matrix obtained by deleting the columns in L_i from M_i .
 - c. Increment i by 1.
4. If $i > k$ then return a NO-instance of appropriate size and exit.
 // The matrix A has more than k pairwise-disjoint blocks of the form L_j for $j \leq i$, each having $r + 1$ linearly independent columns. By Observation 4, each block L_i requires at least 1 edit, hence, by Observation 7, (A, r, k) is a NO-instance of MATRIX RIGIDITY.
5. Let $i_{\leq r} = i$ store the index where the rank of M_i falls below $r + 1$.
6. While $i \leq k$:
 - a. Let B_i be a column basis of M_i .
 - b. Obtain M_{i+1} by deleting the columns in B_i from M_i .
 - c. If M_{i+1} is empty (in other words, $B_i = M_i$) then return A .
 - d. Increment i by 1.
7. Let \mathcal{L} be a matrix formed by the columns in each L_i for $i \in \{0, \dots, i_{\leq r} - 1\}$.
8. Let \mathcal{B} be a matrix formed by the columns in each B_i for $i \in \{i_{\leq r}, \dots, k\}$.
9. Return the matrix formed by the columns in $\mathcal{L} \cup \mathcal{B}$.
 //Note that M_{k+1} is non-empty if output occurs here.

■ **Figure 1** The column reduction procedure.

I_{r+k+1} to satisfy the constraint that a kernel is an instance of the same problem as the input instance (even though, if the output is given by either line 1 or 4, we have actually solved the input instance (A, r, k) of MATRIX RIGIDITY in polynomial time). Using the procedure COLUMN-REDUCTION, it is straightforward to reduce the number of rows too. The details of this procedure are given in Figure 2.

► **Lemma 8.** *Let A be a matrix over some field \mathbb{F} , and let r and k be two non-negative integers. Given an instance (A, r, k) , the procedure MATRIX-REDUCTION runs in time polynomial in input size and returns a matrix \tilde{A} satisfying the following properties:*

1. \tilde{A} has at most $\mathcal{O}(r^2 \cdot k^2)$ entries.
2. If the output is produced by lines 6c and 9 of COLUMN-REDUCTION (when called by MATRIX-REDUCTION), then \tilde{A} is a jumbled submatrix of A .
3. (A, r, k) is a YES-instance of MATRIX RIGIDITY if and only if (\tilde{A}, r, k) is a YES-instance.

Proof. The steps of procedure COLUMN-REDUCTION are all computable in polynomial time, and therefore MATRIX-REDUCTION runs in polynomial time. We now prove the desired properties one by one. Let the matrix \tilde{N} denote the output of COLUMN-REDUCTION on the input instance (N, r, k) .

Proof of 1. We first bound the size of the output of COLUMN-REDUCTION. The output of this procedure can occur at lines 1, 4, 6c and 9. If the output happens at line 1, it has 1 column

Algorithm: MATRIX-REDUCTION

Input: A matrix A over some field \mathbb{F} , and two non-negative integers r, k .

Output: A matrix having $\mathcal{O}(r \cdot k) \times \mathcal{O}(r \cdot k)$ entries.

1. Let $C_A = \text{COLUMN-REDUCTION}(A, r, k)$.
2. Let $R_A = \text{COLUMN-REDUCTION}(C_A^T, r, k)$.
3. Return R_A^T .

■ **Figure 2** The dimension reduction procedure.

by construction. Similarly, if the output happens at line 4, it has $r + k + 1 \leq (r + 1) \cdot (k + 1)$ columns by construction. If the output occurs at line 6c or line 9, then the number of columns in \tilde{N} is at most $(k + 1) \cdot (r + 1)$ as it is constructed using columns of at most $i \leq k$ matrices, $L_0, \dots, L_{i \leq r-1}, B_{i \leq r}, \dots, B_i$, each having at most $r + 1$ columns.

The procedure MATRIX-REDUCTION first obtains a matrix C_A with the aforementioned number of columns by running COLUMN-REDUCTION. Then, it runs COLUMN-REDUCTION again on the transpose of C_A to get its rows bounded. Thus, the dimensions of the output matrix are as claimed. ◀

Proof of 2. The relation “is a *jumbled submatrix* of” is a transitive relation, therefore it suffices to show that the procedure COLUMN-REDUCTION outputs a *jumbled submatrix* of A . If the output happens at lines 6c and 9, then the columns in the output matrix are a subset of the columns in the input matrix. Therefore, in the first line of procedure MATRIX-REDUCTION C_A is a *jumbled submatrix* of A . Similarly, R_A is a *jumbled submatrix* of C_A^T . Finally note that for matrices X and Y , X is a *jumbled submatrix* of Y if and only if X^T is a *jumbled submatrix* of Y^T . Hence, the output matrix R_A^T is a *jumbled submatrix* of A . ◀

Proof of 3: We first show that the procedure COLUMN-REDUCTION produces an equivalent instance of MATRIX RIGIDITY. In the forward direction, suppose that (N, r, k) is a YES-instance of MATRIX RIGIDITY. If the output occurs at line 1, it is a YES-instance by construction. The output cannot occur at line 4 as (N, r, k) is a YES-instance. At lines 6c and 9, by property 2, COLUMN-REDUCTION outputs a *jumbled submatrix* \tilde{N} of the input matrix N . Let S denote a solution of instance (N, r, k) of MATRIX RIGIDITY. The set S consists of replaced entries and their indices. We denote the edited matrix obtained from N by N_S . We construct \tilde{S} , a solution of (\tilde{N}, r, k) , by remembering the new positions of columns and rows of N in the matrix \tilde{N} , and then performing the same permutations on the indices of entries in S . As $\tilde{N}_{\tilde{S}}$ is a *jumbled submatrix* of N_S , by Observation 6, $\text{rank}(\tilde{N}_{\tilde{S}}) \leq \text{rank}(N_S) \leq r$, hence (\tilde{N}, r, k) is a YES-instance of MATRIX RIGIDITY. ◀

In the backward direction, suppose (\tilde{N}, r, k) is a YES-instance of MATRIX RIGIDITY. If the output of \tilde{N} occurs at lines 1 or 4, then we actually know the solution to the instance (N, r, k) of MATRIX RIGIDITY (as explained in the pseudocode). If the output occurs at line 6c, then the output \tilde{N} of COLUMN-REDUCTION is a *jumbled matrix* of N and the result holds by Observation 5. Now we are left with the case when the output occurs at line 9. Let \tilde{S} be any solution to the instance (\tilde{N}, r, k) of MATRIX RIGIDITY. The matrix edited using a solution \tilde{S} is denoted by $\tilde{N}_{\tilde{S}}$. Notice that the matrix \tilde{N} consists of two submatrices \mathcal{L} and \mathcal{B} . As \mathcal{L} consists of $i_{\leq r}$ blocks having rank $r + 1$, by Observation 4, we need to edit at least $i_{\leq r}$ entries in \mathcal{L} . So, we can afford to make at most $k - i_{\leq r}$ edits in the matrix \mathcal{B} . As \mathcal{B} consists of $k + 1 - i_{\leq r}$ blocks, by pigeonhole principle there exists at least one block in \mathcal{B} , say B_t ,

which is not subject to any edit by the solution \tilde{S} . The block B_t has rank at most r and spans the matrix M_{k+1} (refer to pseudocode), by construction. Construct the matrix N'_S by concatenating the columns of $\tilde{N}_{\tilde{S}}$ and M_{k+1} ; it has rank at most r as the columns of the matrix M_{k+1} are in the span of the columns of $\tilde{N}_{\tilde{S}}$. Similarly, construct another matrix N' by concatenating the columns of \tilde{N} and M_{k+1} . As observed in the first half of this proof, we can easily construct a solution S for the matrix N using N'_S , as N' is a *jumbled matrix* of N . Thus, $\text{rank}(N_S) = \text{rank}(N'_S) = \text{rank}(\tilde{N}_{\tilde{S}}) \leq r$, proving that the instance (N, r, k) is a YES-instance of MATRIX RIGIDITY.

To complete the proof, observe that in the procedure MATRIX-REDUCTION, the instances (A, r, k) and (C_A, r, k) are equivalent by the argument above. By Observation 5, (C_A, r, k) and (C_A^T, r, k) are equivalent. As R_A is the output of COLUMN-REDUCTION, (R_A, r, k) is equivalent to (C_A, r, k) . Finally, by Observation 5, again (R_A, r, k) and (R_A^T, r, k) are equivalent. ◀

If the matrix A is over a fixed finite field \mathbb{F} , we obtain a kernel as well.

► **Theorem 9.** *Given an instance (A, r, k) of FF MATRIX RIGIDITY over the field \mathbb{F}_p , the procedure MATRIX-REDUCTION outputs an $\mathcal{O}(r^2 \cdot k^2 \cdot \log p)$ -kernel.*

Proof. The number of entries in the output matrix of MATRIX-REDUCTION is bounded by $\mathcal{O}(r^2 \cdot k^2)$, and the bit length of each entry is at most $\lceil \log_2 p \rceil$. ◀

In case the field \mathbb{F} is infinite—for example, if \mathbb{F} is either \mathbb{Q} or \mathbb{R} —the procedure is not guaranteed to produce a kernel as the bit lengths of matrix entries may not be bounded by a function of r and k .

4 Fixed-Parameter Tractability with Respect to $k + r$

This section describes an algorithm for MATRIX RIGIDITY. The formulation it presents was also used in the context of complexity analysis in [19].

Using Lemma 8, we can reduce any instance (A, r, k) to an equivalent instance (A', r, k) such that the matrix A' is a *jumbled submatrix* of A and the number of entries in A' is $\mathcal{O}(r^2 \cdot k^2)$. Once we have such a matrix A' , it is useful to examine an alternative definition of the rank of a matrix, which is given in terms of the determinant of its square submatrices. Specifically, we will rely on the following lemma.

► **Lemma 10** (Chapter 7, [21]). *A matrix A over \mathbb{R} has rank at most r if and only if all the $(r + 1) \times (r + 1)$ submatrices of A have determinant 0.*

The correctness of our algorithm MATRIG-ALG for MATRIX RIGIDITY, which is described in Figure 3, follows in a straightforward fashion using Lemma 10. This algorithm for MATRIX RIGIDITY crucially relies on a procedure which can decide the feasibility of a system of polynomials over a given field. This procedure shall be the object of discussion in the rest of the section.

Observe that each polynomial in \mathcal{P} , as defined in the algorithm MATRIG-ALG, has at most k unknowns and its *total degree* is at most k . The size of \mathcal{P} is of order $(r \cdot k)^{\mathcal{O}(r)}$. In the case where the underlying field is \mathbb{R} , suppose the longest length entry has bit length L . The coefficients of polynomials in \mathcal{P} are obtained by computing the determinant of matrices which have size at most $r \times r$. By *Hadamard's inequality* [11], for a matrix M of size $r \times r$ the $\det(M) \leq \prod_{i \in [r]} \|M_i\|_2$. If the bit length of entries in M is at most L then the coefficients of polynomials in \mathcal{P} can be shown to be bounded by $L' = r \cdot L + \mathcal{O}(r \cdot \log r)$.

Algorithm: MATRIG-ALG
Input: A matrix A over a field \mathbb{F} , and two non-negative numbers r, k .
Output: Can one edit at most k entries of A to obtain a matrix of rank at most r ?

1. Let $A' = \text{MATRIX-REDUCTION}(A, r, k)$.
2. For each set E of k entries in A' :
 - a. Replace each entry of A' indexed by an element in E by a distinct indeterminate to obtain a *mixed matrix* A'_E .
 - b. Let \mathcal{P} be the set of equations obtained by setting the determinant of each $(r + 1) \times (r + 1)$ submatrix of A'_E to 0.
 - c. If \mathcal{P} is feasible over \mathbb{F} then return YES and exit.
3. Return NO and exit.

■ **Figure 3** Description of the algorithm for MATRIX RIGIDITY.

We use the following proposition, given in [1] (Proposition 13.19), to check the feasibility of the system of polynomials \mathcal{P} when it is defined over \mathbb{R} .

► **Proposition 11.** *Given a set \mathcal{P} of ℓ polynomials of degree d in k variables with coefficients in C , we can decide with complexity $\ell \cdot d^{\mathcal{O}(k)}$ in D (where D is the ring generated by the real and imaginary parts of the coefficients of the polynomials in \mathcal{P}) whether $\text{Zer}(\mathcal{P}, C^k)$ is empty. Moreover, if $D = \mathbb{Z}$ and the bit-sizes of the coefficients of the polynomials are bounded by τ , then bit-sizes of the integers appearing in the intermediate computations and the output are bounded by $(\tau + \log \ell) \cdot d^{\mathcal{O}(k)}$.*

Applying the proposition above on the system of equations \mathcal{P} , we get the following.

► **Theorem 12.** *Suppose we are given a matrix A over \mathbb{R} such that the bit length of each of its entries is bounded by L , and let r and k be two non-negative integers. Then, the instance (A, r, k) of REAL MATRIX RIGIDITY can be solved in time $\mathcal{O}^*(2^{\mathcal{O}((r+k) \cdot \log(r \cdot k))})$. Here, the bit lengths of integers appearing in intermediate computations and the output are bounded by $\mathcal{O}(r \cdot (\log(k \cdot r) + L)) \cdot 2^{\mathcal{O}(k \cdot \log k)}$.*

Proof. The algorithm MATRIG-ALG generates $\mathcal{O}((r \cdot k)^{2k})$ many systems of equations. Each system of equations has $\ell = (r \cdot k)^{\mathcal{O}(r)}$ many equations, where the degree $d = k$ and there are k variables. Using Proposition 11, we get the required running time and the bit lengths.

Notice that a system of equations \mathcal{P} is feasible if and only if the chosen entries of the matrix can be edited to reduce the rank. Since we exhaustively try all possible entries that can be edited, the correctness of MATRIG-ALG follows. ◀

In the case where the underlying field \mathbb{F}_p is finite, the coefficients of the polynomials are elements of \mathbb{F}_p and hence have bounded bit lengths. The feasibility of \mathcal{P} over a finite field can be decided using the following known algorithm which also gives us an algorithm for FF MATRIX RIGIDITY.

► **Proposition 13** ([9]). *There is a deterministic algorithm which, given an input consisting of a finite field \mathbb{F}_p and system of polynomials $f_1, \dots, f_\ell \in \mathbb{F}_p[x_1, \dots, x_k]$ of total degree bounded by d , decides the feasibility of the system in time $d^{k^{\mathcal{O}(k)}} \cdot (\ell \cdot \log p)^{\mathcal{O}(1)}$.*

Similar to the proof of Theorem 12, we obtain the following.

► **Theorem 14.** *The problem FF MATRIX RIGIDITY, where the input matrix A is an $m \times n$ matrix over a field \mathbb{F}_p , can be solved in time $f(r, k)(\log p + m + n)^{\mathcal{O}(1)}$ for some function f .*

This algorithm for FF MATRIX RIGIDITY has the advantage that it runs in time which is *polynomial in the logarithm of the order of the field*, even though the dependence on k is exponential.

5 W[1]-Hardness with Respect to k

In this section, we first reduce (in two steps) a special case of ODD SET to a problem that has a formulation easier to use in our context. The latter problem is reduced to a variant of NEAREST CODEWORD, which, in its turn, is reduced to REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY.

► **Theorem 15.** *REAL MATRIX RIGIDITY and FF MATRIX RIGIDITY for any choice of a finite field \mathbb{F}_p are W[1]-hard with respect to k .*

Proof. Let \mathbb{F} denote the field, which is either \mathbb{R} or some finite field \mathbb{F}_p , over which we define MATRIX RIGIDITY. First, we observe that the reduction from MULTICOLORED CLIQUE given in the book [3] (Theorem 13.31) to show that ODD SET is W[1]-hard actually shows that the following special case of ODD SET is W[1]-hard. That is, the constructed instances have the form specified in the special case.

PARTITIONED ODD SET

Parameter: k

Input: A family \mathcal{F} of sets over a universe U , a non-negative integer k , a partition (U_1, \dots, U_k) of U such that for every $i \in [k]$, $U_i \in \mathcal{F}$, and for every $F \in \mathcal{F}$, there exist $i, j \in [k]$ for which $F \subseteq U_i \cup U_j$.

Question: Is there a subset $S \subseteq U$ of size at most k such that the intersection of S with every set in \mathcal{F} has size 1?

The arguments below will crucially rely on the fact that we restrict ourselves to this special case. Given a vector v , we let $\bar{I}_0(v)$ denote the indices of the entries of v that do not contain 0. Now, we reformulate PARTITIONED ODD SET in the language of matrices as follows.

PARTITIONED ODD MATRIX

Parameter: k

Input: A $t \times r$ binary matrix L over \mathbb{R} , a non-negative integer k , a partition (U_1, \dots, U_k) of $[r]$ such that for every $i \in [k]$, there exists $j \in [t]$ for which $U_i = \bar{I}_0(L_j)$, and for every $i \in [t]$, there exist $j, \ell \in [k]$ for which $\bar{I}_0(L_i) \subseteq U_j \cup U_\ell$.

Question: Is there an r -dimensional binary vector x such that $|\bar{I}_0(x)| \leq k$ and $Lx = 1$?

Given an instance $(\mathcal{F}, U, (U_1, \dots, U_k), k)$ of PARTITIONED ODD SET, it is straightforward to obtain (in polynomial time) an equivalent instance $(L_{t \times r}, (U'_1, \dots, U'_k), k')$ of PARTITIONED ODD MATRIX as follows. First, we let $t = |\mathcal{F}|$ and $r = |U|$. We assume w.l.o.g. that $U = [r]$. Now, we associate a row L_i with each set $F \in \mathcal{F}$ by letting L_i contain 1 at each entry whose index belongs to F and 0 at each of the remaining entries. That is, $\bar{I}_0(L_i) = F$. Finally, we let $(U'_1, \dots, U'_k) = (U_1, \dots, U_k)$ and $k' = k$. It is easy to see that $S \subseteq U$ is a solution to $(\mathcal{F}, U, (U_1, \dots, U_k), k)$ if and only if the binary vector $x_{r \times 1}$ such that $\bar{I}_0(x) = S$ is a solution to $(L_{t \times r}, (U'_1, \dots, U'_k), k)$, and therefore the instances are equivalent.

We now incorporate the input field \mathbb{F} .

ℱ-ODD MATRIX**Parameter:** k **Input:** A $t \times r$ binary matrix L over \mathbb{F} and a non-negative integer k .**Question:** Is there an r -dimensional vector x over \mathbb{F} such that $|\bar{I}_0(x)| \leq k$ and $Lx = 1$?

We reduce PARTITIONED ODD MATRIX to \mathbb{F} -ODD MATRIX as follows. Given an instance $(L_{t \times r}, (U_1, \dots, U_k), k)$ of PARTITIONED ODD MATRIX, we simply output $(L_{t \times r}, k)$ as the equivalent instance of \mathbb{F} -ODD MATRIX. In one direction, let x be a solution to $(L_{t \times r}, (U_1, \dots, U_k), k)$. Recall that L is a binary matrix. Thus, since x is a binary vector satisfying $Lx = 1$ over \mathbb{R} , it must also satisfy $Lx = 1$ over \mathbb{F} . Since $|\bar{I}_0(x)| \leq k$, we get that x is a solution to $(L_{t \times r}, k)$. In the second direction, let x be a solution to $(L_{t \times r}, k)$. Since $|\bar{I}_0(x)| \leq k$ and for every $s \in [k]$, there exists $s' \in [t]$ for which $U_s = \bar{I}_0(L_{s'})$, it must hold that for every $s \in [k]$, $|\bar{I}_0(x) \cap U_{s'}| = 1$. Thus, since for every $s \in [k]$, $L_{s'}x = 1$ over \mathbb{F} , it holds that x is a binary vector. It remains to show that $Lx = 1$ over \mathbb{R} . Consider some index $i \in [t]$. Then, there exist $j, \ell \in [k]$ such that $\bar{I}_0(L_i) \subseteq U_j \cup U_\ell$. Therefore, since for every $s \in [k]$, $|\bar{I}_0(x) \cap U_s| = 1$, and since both L and x are binary, it is only possible that $L_i x = 1$ over \mathbb{F} if $|\bar{I}_0(x) \cap \bar{I}_0(L_i)| = 1$, which allows us to conclude that $L_i x = 1$ also over \mathbb{R} .

In what follows, calculations are performed over \mathbb{F} . Next, we reduce \mathbb{F} -ODD MATRIX to the following variant of the NEAREST CODEWORD problem. This specific reduction is inspired by a reduction from NEAREST CODEWORD to ODD SET of Bonnet et al. [2].

ℱ-NEAREST CODEWORD**Parameter:** k **Input:** An $m \times n$ matrix M , an m -dimensional vector b over \mathbb{F} , and a non-negative integer k .**Question:** Is there an n -dimensional vector y over \mathbb{F} such that the Hamming distance between My and b is at most k ?

Given an instance $(L_{t \times r}, k)$ of \mathbb{F} -ODD MATRIX, we construct an instance $(M_{m \times n}, b, k')$ of \mathbb{F} -NEAREST CODEWORD as follows. First, let $k' = k$. Now, let M be an $m \times n$ matrix, where $m = r$ and $n = r - \text{rank}(L)$, such that the rows of L form a basis for the subspace orthogonal to the column space of M . Then, an r -dimensional vector v over \mathbb{F} satisfies $Lv = 0$ if and only if v belongs to the column space of M (i.e., there is an n -dimensional vector y over \mathbb{F} such that $My = v$). Finally, let b be an r -dimensional vector such that $Lb = -1$. If no such vector exists, then there is no r -dimensional vector over \mathbb{F} such that $Lv = 1$, which in particular implies that $(L_{t \times r}, k)$ is a NO-instance, and thus we can return a trivial NO-instance of \mathbb{F} -NEAREST CODEWORD. Therefore, next assume that b exists. To prove that the reduction is correct, first let x be a solution to $(L_{t \times r}, k)$. Then, $Lx = 1$, and since $Lb = -1$, we have that $L(x + b) = Lx + Lb = Lx - 1 = 0$. Therefore, by the choice of M , there exists an n -dimensional vector y over \mathbb{F} such that $My = (x + b)$. Since $|\bar{I}_0(x)| \leq k$, we have that the Hamming distance between My and b is at most k , which implies that y is a solution to $(M_{m \times n}, b, k')$. In the other direction, let y be a solution to $(M_{m \times n}, b, k')$. Then, since the Hamming distance between My and b is at most k , there exists an m -dimensional vector x such that $|\bar{I}_0(x)| \leq k$ and $My = x + b$. Therefore, by the choice of M , $L(x + b) = 0$. Since $Lb = -1$, we get that $Lx = 1$, which implies that x is a solution to $(L_{t \times r}, k)$.

Finally, we reduce \mathbb{F} -NEAREST CODEWORD to MATRIX RIGIDITY over \mathbb{F} . For this purpose, let $(M_{m \times n}, b, k)$ be an instance of \mathbb{F} -NEAREST CODEWORD. We assume w.l.o.g. that the columns of M are linearly independent. We construct an equivalent instance $(A_{\hat{a} \times \hat{b}}, r, \hat{k})$ of MATRIX RIGIDITY over \mathbb{F} as follows. First, let $\hat{k} = k$. Now, let $\hat{a} = m$, $r = n$ and $\hat{b} = (k + 1)n + 1$. For each $i \in [k + 1]$ and $j \in [n]$, we define $A^{(k+1)(i-1)+j} = M^j$. Finally, we

define $A^{(k+1)m+1} = b$. On the one hand, let y be a solution to $(M_{m \times n}, b, k)$. Then, there are at most k entries that should be changed in b to obtain an m -dimensional vector b' over \mathbb{F} such that $My = b'$. In the matrix A , replace the last column b by b' . Denote the resulting matrix by A' . Then, the last column of A' is a linear combination of its other columns (by the construction of A and since $My = b'$), and among the other columns of A' , there are only m distinct columns. Therefore, $\text{rank}(A') = n$, which implies that $(A_{\widehat{a \times b}}, r, \widehat{k})$ is a YES-instance. In the other direction, suppose that $(A_{\widehat{a \times b}}, r, \widehat{k})$ is a YES-instance. Then, it is possible to change at most k entries in A and obtain a matrix A' such that $\text{rank}(A') = n$. Since besides the last column of A , each column of A is repeated $k + 1$ times (i.e., more times than the number of changes), and there are n such distinct rows, it must be that the last column of A' is a linear combination of the distinct columns of A excluding the last column of A . By the construction of A , we get that there exists an n -dimensional vector y over \mathbb{F} such that $My = b'$, where b' is the last column of A' . Since the Hamming distance between b and b' is at most k , we have that y is a solution to $(M_{m \times n}, b, k)$. ◀

References

- 1 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- 2 E. Bonnet, L. Egri, and D. Marx. Fixed-parameter approximability of boolean MinCSPs. In *ESA*, pages 18:1–18:18, 2016.
- 3 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshantov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, 2015.
- 4 Amit Jayant Deshpande. *Sampling-based algorithms for dimension reduction*. PhD thesis, Massachusetts Institute of Technology, 2007.
- 5 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 6 Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993.
- 7 D. Grigoriev. Using the notions of separability and independence for proving the lower bounds on the circuit complexity (in russian). *Notes of the Leningrad branch of the Steklov Mathematical Institute, Nauka*, 1976.
- 8 D. Grigoriev. Using the notions of separability and independence for proving the lower bounds on the circuit complexity. *Journal of Soviet Math.*, 14(5):1450–1456, 1980.
- 9 Neeraj Kayal. Solvability of a system of bivariate polynomial equations over a finite field. In *ICALP*, pages 551–562, 2005.
- 10 Abhinav Kumar, Satyanarayana V. Lokam, Vijay M. Patankar, and Jayalal M. N. Sarma. Using elimination theory to construct rigid matrices. *Computational Complexity*, 23(4):531–563, 2013.
- 11 Kenneth Lange. Hadamard’s determinant inequality. *The American Mathematical Monthly*, 121(3):258–259, 2014.
- 12 Satyanarayana V. Lokam. On the rigidity of Vandermonde matrices. *Theoretical Computer Science*, 237(1–2):477–483, 2000.
- 13 Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4:1–155, January 2009.
- 14 S. V. Lokam. Spectral methods for matrix rigidity with applications to size-depth tradeoffs and communication complexity. In *FOCS*, pages 6–15, 1995.
- 15 Meena Mahajan and Jayalal M. N. Sarma. On the complexity of matrix rank and rigidity. In *CSR*, pages 269–280, 2007.

32:14 Matrix Rigidity from the Viewpoint of Parameterized Complexity

- 16 S. M. Meesum, Pranabendu Misra, and Saket Saurabh. Reducing rank of the adjacency matrix by graph modification. In *COCOON*, pages 361–373, 2015.
- 17 S. M. Meesum and S. Saurabh. Rank reduction of directed graphs by vertex and edge deletions. In *LATIN*, pages 619–633, 2016.
- 18 A. A. Razborov. On rigid matrices. *Manuscript in russian*, 1989.
- 19 Jayalal M. N. Sarma. *Complexity Theoretic Aspects of Rank, Rigidity and Circuit Evaluation*. PhD thesis, The Institute of Mathematical Sciences, CIT Campus, Taramani, Chennai, 2009.
- 20 M. A. Shokrollahi, D. Spielman, and V. Stemann. A remark on matrix rigidity. *Information Processing Letters*, 64(6):283–285, 1997.
- 21 L. E. Sigler. *Algebra*. Undergraduate Texts in Mathematics. Springer-Verlag, 1976.
- 22 L. G. Valiant. Graph-theoretic arguments in low-level complexity. In *MFCS*, pages 162–176, 1977.