The Complexity of Matrix Completion

Nicholas J.A. Harvey* David R. Karger Sergey Yekhanin
MIT Computer Science and Artificial Intelligence Laboratory
{nickh,karger,yekhanin}@mit.edu

Abstract

Given a matrix whose entries are a mixture of numeric values and symbolic variables, the ***matrix completion problem*** is to assign values to the variables so as to maximize the resulting matrix rank. This problem has deep connections to computational complexity and numerous important algorithmic applications. Determining the complexity of this problem is a fundamental open question in computational complexity. Under different settings of parameters, the problem is variously in P, in RP, or NP-hard. We shed new light on this landscape by demonstrating a new region of NP-hard scenarios. As a special case, we obtain the first known hardness result for matrices in which each variable appears only twice.

Another particular scenario that we consider is the ***simultaneous matrix completion problem***, where one must simultaneously maximize the rank for several matrices that share variables. This problem has important applications in the field of network coding. Recent work has given a simple, greedy, deterministic algorithm for this problem, assuming that the algorithm works over a sufficiently large field. We show an *exact* threshold for the field size required to find a simultaneous completion efficiently. This result implies that, surprisingly, the simple greedy algorithm is optimal: finding a simultaneous completion over *any* smaller field is NP-hard.

1 Introduction

A ***mixed matrix*** is a matrix where each entry is either a number or an indeterminate. Such matrices have been intensively studied and have many deep combinatorial properties and close ties to matroid theory [21, 25, 7, 24]. Although initially studied as mathematical objects in differential geometry [26], their central importance in computer science was established by Edmonds [6] and Lovász [20].

Mixed matrices have important algorithmic uses for many problems: computing the cardinality of a maximum matching [20], constructing maximum matchings [22, 23], dynamically computing transitive closure [27], and constructing multicast network codes [13, 14, 17]. Algorithms that use mixed matrices typically begin by substituting numeric values for the indeterminates, since algebraic operations are more efficient on numeric matrices than on mixed matrices. Such an assignment of values to the indeterminates is called a ***completion***. One might imagine looking for completions that satisfy a variety of properties, such as full rank, positive definiteness, a desired spectrum, etc. A survey of various completion problems is given by Laurent [18]. In this paper, we will focus on finding completions with full rank. For brevity we will henceforth use the term "completion" to mean one with full rank.

In addition to its algorithmic applications, matrix completion has fundamental connections to computational complexity. In particular, matrix completion contains the polynomial identity testing problem (of formulae) [21] as a special case since any arithmetic formula can be recast as a determinant of a mixed matrix [29]. Polynomial identity testing is one of the most important problems in co-RP that is not known to be in P. This problem has received renewed attention recently since Kabanets and Impagliazzo [16] showed that derandomizing polynomial identity testing implies strong circuit lower bounds.

For the majority of this paper, we will restrict our attention to mixed matrices where the indeterminates are distinct and algebraically independent. (Most algorithmic applications of mixed matrices actually involve matrices of this type.) As an example of such a mixed matrix, consider $M = \left(\begin{smallmatrix} 1 & x \\ y & 1 \end{smallmatrix}\right)$. Assigning the values $x = 1$ and $y = 1$ does not give a (full rank) completion since the resulting matrix has rank 1. The assignment $x = 0$ and $y = 0$ is a completion since the resulting matrix has full rank.

The problem of finding completions has been considered for many years, dating back to the work of Edmonds [6] and Lovász [20]. In the latter paper, Lovász uses the Schwartz-Zippel lemma to show that simply choosing values at random from a sufficiently large field gives a completion with high probability. This work left open the question of finding an efficient deterministic al-

*Supported by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship.

Algorithm 1: A simple greedy algorithm for finding a simultaneous completion for a set $\mathcal{A}$ consisting of d mixed matrices. It can operate over any field of size strictly greater than d. Murota's algorithm for testing the rank of a mixed matrix is used as a subroutine.

Step 1: For each $A \in \mathcal{A}$, compute the rank r_A of A.

Step 2: For each indeterminate x

  Step 3: For i from 0 to d

    Step 4: Assign x the value i and compute the resulting rank r'_A of every matrix $A \in \mathcal{A}$.

    Step 5: If $r_A = r'_A$ for all $A \in \mathcal{A}$, skip to the next indeterminate in Step 2.

gorithm for constructing matrix completions. Considering now the case where each indeterminate appears only once in the matrix, this question was solved by the pioneering work of Geelen [7] and Murota [24]. Algorithms for completing other types of matrices are discussed in Section 4.

Geelen devised an elegant algorithm that blends ideas from combinatorics, matroid theory and linear algebra, but unfortunately requires $O(n^9)$ time. Some later improvements [9, 2] reduced the running time to $O(n^4)$. Geelen's algorithm also requires working over a field of size at least n. On the other hand, Murota [24, 25] showed that one can compute the rank of a mixed matrix in polynomial time via a reduction to matroid intersection. This work did not specifically consider the problem of constructing a completion or the requisite field size. Building on Murota's work, Harvey et al. [13] gave an algorithm to complete mixed matrices over *any* field. In this paper, we show that the algorithm of [13] is optimal[1] with respect to the number of occurrences of indeterminates: there exists a field (namely $\mathbb{F}_2$) over which it is NP-hard to find a completion for matrices in which indeterminates can occur twice.

A useful generalization of the matrix completion problem is the ***simultaneous matrix completion*** problem. Suppose one is given a set of mixed matrices, each of which contains a mixture of numbers and indeterminates. Each particular indeterminate can only appear once per matrix but may appear in several matrices. The objective is to find values for these indeterminates such that all resulting matrices simultaneously have full rank. Simultaneous matrix completions were used in drawing a connection to multicast network coding [17, 13]. Lovász's randomized approach from [20] trivially extends to handle this problem by simply working over a larger field. The work of Harvey et al. [13] also gives efficient deterministic algorithms for finding simultaneous completions. In fact, Murota's rank computation algorithm immediately yields a simple, but less efficient, algorithm for finding simultaneous completions. Pseudocode for this algorithm is shown in Algorithm 1. This algorithm can compute a simultaneous completion for a set of d mixed matrices over any field $\mathbb{F}_q$ where $q > d$. Specializing to the case where $\mathcal{A}$ is a singleton set, the algorithm can compute a completion for a single mixed matrix over *any* field.

Algorithm 1's restriction on the field size may seem somewhat excessive, since it requires that the field size exceed the number of matrices. Surprisingly, although Algorithm 1 is so trivial, its restriction on the field size cannot be removed. We show that the hardness of finding a simultaneous completion crosses a sharp threshold when the field size equals the number of matrices. Our main theorem is as follows.

Theorem 1. *Let q be a prime power and let $\mathcal{A}$ be a set of d mixed matrices over $\mathbb{F}_q$.*

- *If $q > d$ then a simultaneous completion necessarily exists and one can be computed in polynomial time.*
- *If $q \le d$ then a simultaneous completion may not exist. Furthermore, deciding whether a simultaneous completion exists is NP-complete, even if all numbers in $\mathcal{A}$ are either 0 or 1.*

The first claim of this theorem is implied by Algorithm 1, which appeared in [13]. The second claim shows that the field size required by this algorithm is optimal.

Our main theorem generates several interesting corollaries, which we elaborate on in Section 5. The first corollary is a new hardness result for deciding non-singularity of a single mixed matrix where the same indeterminate can appear in multiple entries. The best previously-known result is as follows.

Theorem 2. (Buss et al. [4]) *Let M be a matrix over any field $\mathbb{F}_q$ where indeterminates can appear any number of times. The problem of deciding whether M is non-singular is NP-complete.*

[1]The claims of optimality in this paper are all contingent on the assumption that P≠NP.

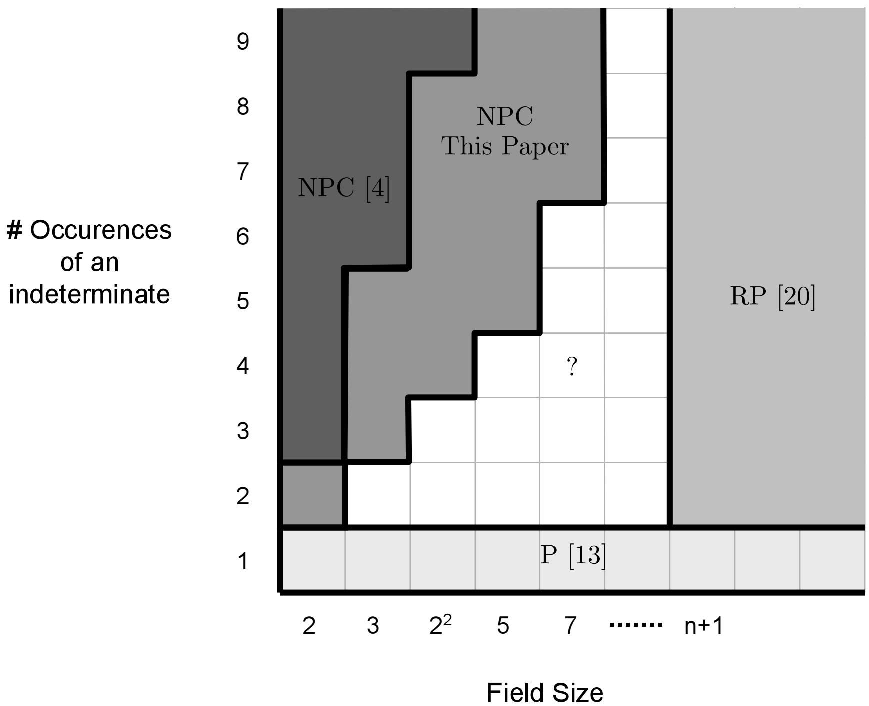

Figure 1: This chart shows the computational complexity of matrix completion as a function of the field size, and the number of occurrences of a variable. The label NPC denotes that the associated completion problems are NP-complete. Here, n denotes the size of the matrix, and it is assumed to have a fixed value throughout the chart. Without loss of generality, assume that $n+1$ is a prime power.

The work of Buss et al. unfortunately did not explicitly describe the regions of field sizes and occurrences of indeterminates for which their NP-completeness result is valid. However, the number of occurrences of indeterminates is a key parameter for any hardness result since the current deterministic algorithms depend crucially on this parameter. Figure 1 shows a pictorial summary of the state-of-the-art for completion of mixed matrices, as we understand it. Algorithms for various other special cases of matrix completion are discussed in Section 4.

This paper demonstrates a new region of NP-hard scenarios for matrix completion by proving the following refinement of Theorem 2. To the best of our knowledge, this corollary gives the first known hardness result for matrices where indeterminates can occur at most twice.

Corollary 3. *Let M be a mixed matrix over any field $\mathbb{F}_q$ where indeterminates can appear several times. The problem of deciding whether M is non-singular is NP-complete, even if we require that each indeterminate appear in M at most q times, and all numbers are either* 0 *or* 1*.*

Another important matrix completion problem is completion of ***mixed skew-symmetric matrices*** [10, 11, 25]; we discuss this further in Section 4. This problem has deep connections to (non-bipartite) matchings in graphs and matroids. As another corollary of our main theorem, we obtain the following result.

Corollary 4. *Let M be a skew-symmetric mixed matrix over $\mathbb{F}_2$ where each indeterminate appears at most twice (four times, if we include symmetric partners). The problem of deciding whether M is non-singular is NP-complete.*

2 Simultaneous Completion when $q < d$

Define Sim-Completion(q, d) to be the problem of deciding whether a given collection of d matrices over $\mathbb{F}_q$ has a simultaneous completion, where q is a prime power. Note that fixing q and d does not determine the size of the input, because the matrices can be arbitrarily large.

The proof of the main theorem requires introducing a sequence of increasingly intricate gadgets. For clarity of exposition, we present these gadgets in a sequence of lemmas, each of which proves hardness for a certain restricted case of the main theorem. We begin by proving hardness for large sets of matrices over $\mathbb{F}_2$.

LEMMA 5. *For arbitrarily large collections of mixed matrices, finding a completion over $\mathbb{F}_2$ is NP-complete.*

Proof. First we remark that a completion can be checked in polynomial time. Simply perform Gaussian elimination on the completed matrices and verify that they have full rank. The Gaussian elimination process runs in polynomial time because the matrices no longer contain any indeterminates, only numerical values.

We now show hardness via a reduction from CIRCUIT-SAT: given an acyclic boolean circuit of NAND gates with a single output, determine if the inputs can be set such that the output becomes 1. Suppose we are given a circuit ϕ with n gates. We will construct a collection $\mathcal{A}$ of $n+1$ matrices over $\mathbb{F}_2$ such that a completion for $\mathcal{A}$ corresponds to a satisfying assignment for ϕ and vice-versa.

The key step is the construction of a matrix gadget that behaves like a NAND gate. Consider a completion of the mixed matrix

$$N(a,b,c) \;:=\; \begin{pmatrix} 1 & a \\ b & c \end{pmatrix}.$$

Recall that a completion must cause the matrix to have full rank and therefore must satisfy $0 \neq \det N(a,b,c) = c - ab$. The only non-zero element of $\mathbb{F}_2$ is 1, hence $c = ab+1$. Since multiplication over $\mathbb{F}_2$ corresponds to the boolean AND operation and adding 1 corresponds to boolean NOT, we have $c = a$ NAND b.

It is now straightforward to construct a collection of matrices for which a simultaneous completion corresponds to a satisfying assignment for ϕ. For every NAND gate in ϕ with inputs a and b and output c, add an instance of $N(a,b,c)$ to $\mathcal{A}$. We also add to $\mathcal{A}$ the matrix $(\,y\,)$ where y is the output of the circuit, reflecting the desire that the circuit output the value 1. It is straightforward to verify that a completion for $\mathcal{A}$ is a satisfying assignment for ϕ and vice-versa. $\square$

We now strengthen the previous lemma by showing hardness for smaller sets of mixed matrices.

LEMMA 6. *For any $d \geq 3$, the problem* SIM-COMPLETION$(2,d)$ *is NP-complete.*

Proof. We will construct a collection $\mathcal{A}$ of d matrices over $\mathbb{F}_2$ such that a completion for $\mathcal{A}$ corresponds to a satisfying assignment for ϕ and vice-versa. It suffices to consider the case that $d = 3$ because otherwise we may simply pad the set $\mathcal{A}$ with additional matrices. To reduce the number of matrices used in Lemma 5, we might imagine combining the numerous NAND matrices into a single block-diagonal matrix A. For example, we might have

$$A \;=\; \begin{pmatrix} N(a,b,c) & & & \\ & N(b,c,d) & & \\ & & N(a,d,e) & \\ & & & \ddots \end{pmatrix}.$$

Unfortunately such a matrix is not a valid input for the simultaneous matrix completion problem since indeterminates appear multiple times in A. To circumvent this restriction, we construct a gadget that allows us to create duplicate copies of variables. Consider the following pair of matrices.

$$B \;:=\; \begin{pmatrix} N(1,x^{(1)},y^{(2)}) & & \\ & N(1,x^{(2)},y^{(3)}) & \\ & & \ddots \end{pmatrix}$$

$$C \;:=\; \begin{pmatrix} N(1,y^{(2)},x^{(2)}) & & \\ & N(1,y^{(3)},x^{(3)}) & \\ & & \ddots \end{pmatrix}$$

The matrix B enforces that $y^{(i+1)} = 1$ NAND $x^{(i)} =$ NOT $x^{(i)}$ for all $i \geq 1$. The matrix C similarly enforces that $x^{(i)} =$ NOT $y^{(i)}$ for all $i \geq 2$. Thus B and C together enforce that $x^{(i)} = x^{(1)}$ for all $i \geq 1$. We extend the definition of B and C to generate n duplicates of all variables in ϕ (including the inputs and intermediate wires).

Construct the matrix A' by modifying A so that the first appearance of a variable x uses $x^{(1)}$, the second appearance uses $x^{(2)}$, etc. Let $\mathcal{A} = \{A', B, C\}$. It is easy to verify that a simultaneous matrix completion for $\mathcal{A}$ determines a satisfying assignment for ϕ and vice-versa. $\square$

The preceding lemma applies to small sets of matrices, but only over $\mathbb{F}_2$. We now generalize to arbitrary finite fields.

LEMMA 7. *For any prime power q and any $d > q$, the problem* SIM-COMPLETION(q,d) *is NP-complete, even if all numbers in the matrices are 0 or 1.*

Proof. As before, we will construct a collection $\mathcal{A}$ of matrices over $\mathbb{F}_q$ such that a completion for $\mathcal{A}$ corresponds to a satisfying assignment for ϕ and vice-versa. As before, it suffices to consider the case that $d = q+1$.

When operating over a field $\mathbb{F}_q$ where $q > 2$, Lemma 6 fails because the matrix $N(a,b,c)$ only behaves like a NAND gate when its entries are either 0 or

1. To extend that lemma to arbitrary fields, we must introduce a gadget which ensures that the indeterminates corresponding to signals in the circuit must take values in $\{0,1\}$.

Consider an arbitrary indeterminate x. We will create $q-2$ auxiliary variables $\hat{x}_2, \ldots, \hat{x}_{q-1}$. Relabeling x to be $\hat{x}_1$, we will enforce that all variables $\hat{x}_1, \ldots, \hat{x}_{q-1}$ take distinct values. Furthermore, we will enforce that each $\hat{x}_i \notin \{0,1\}$ for $i \geq 2$. These conditions clearly imply that $x \in \{0,1\}$.

To enforce these desired conditions, we will construct various 2×2 matrix blocks. After describing these constituent blocks, we show that they can be assembled into a small number of matrices where each variable appears at most once.

First, consider the conditions $\hat{x}_i \notin \{0,1\}$ (for $1 < i < q$). These conditions can be enforced with the matrices $N(0,0,\hat{x}_i) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & \hat{x}_i \end{smallmatrix}\right)$ and $N(1,1,\hat{x}_i) = \left(\begin{smallmatrix} 1 & 1 \\ 1 & \hat{x}_i \end{smallmatrix}\right)$. Next, consider the conditions $\hat{x}_i \neq \hat{x}_j$ ($1 \leq i < j < q$). These conditions can be enforced with the matrices $N(1,\hat{x}_i,\hat{x}_j) = \left(\begin{smallmatrix} 1 & 1 \\ \hat{x}_i & \hat{x}_j \end{smallmatrix}\right)$.

To analyze the number of matrices required, we define a graph G which captures our desired conditions. The vertices and edges of G are as follows.

$$\begin{aligned} V(G) &:= \{\, 0, 1, \hat{x}_1, \ldots, \hat{x}_{q-1} \,\} \\ E(G) &:= \{\, \{\hat{x}_i, \hat{x}_j\} \,:\, 1 \leq i < j < q \,\} \\ &\quad \cup \{\, \{0, \hat{x}_i\} \,:\, 1 < i < q \,\} \\ &\quad \cup \{\, \{1, \hat{x}_i\} \,:\, 1 < i < q \,\} \end{aligned}$$

Each edge corresponds to a 2×2 matrix defined above. Our objective is to partition the edges of G into sets such that each $\hat{x}_i$ is incident on at most one edge per part of the partition. (This objective corresponds to assembling the 2×2 blocks into matrices such that each indeterminate appears at most once per matrix.) We call such a partition a ***pseudo-coloring***, since it is an edge-coloring where the usual coloring constraints are relaxed at vertices 0 and 1. We now show that a pseudo-coloring of G with q colors exists.

Suppose first that q is odd. Let G' be the induced subgraph obtained by deleting vertices 0 and 1 from G. Then G' is the complete graph on $q-1$ vertices. We now use the well-known[2] result that the edge-chromatic number of K_ℓ is $\ell - 1$ if ℓ is even. Furthermore, such a coloring can be constructed in polynomial time. Thus we obtain a coloring of G' with $q-2$ colors. We extend this to a pseudo-coloring of G by assigning the edges incident on 0 a new color, and the edges incident on 1 another new color. The case when q is even is handled by applying a similar argument to the subgraph obtained by deleting the vertex 1. This subgraph is a complete graph of even order minus a single edge, and hence can be colored with $q-1$ colors.

The pseudo-colorings of G constructed above uses q colors. Given this pseudo-coloring, we construct q block-diagonal matrices $M_{x,1}, \ldots, M_{x,q}$ as follows. The matrix $M_{x,i}$ is associated with the i^{th} color class. For each edge with color i, the 2×2 block matrix associated with this edge is added as a block on the diagonal of $M_{x,i}$.

We now observe that only $q-2$ color classes of our pseudo-coloring contain edges incident on the vertex $\hat{x}_1$ since it has degree $q-2$. Thus we may assume that matrices $M_{x,q-1}$ and $M_{x,q}$ do not contain the variable $\hat{x}_1$, which is the alternative label of our original variable x.

Now, we are ready to describe the set $\mathcal{A}$ corresponding to the circuit ϕ. We first construct the matrices A', B, and C as defined in Lemma 6. Next, we construct new matrices $S_1, \ldots, S_{q-2}$. For each indeterminate x in ϕ and $1 \leq i \leq q-2$, the matrix $M_{x,i}$ is added as a new diagonal block in S_i. (This construction is also performed for the duplicate copies of variables $x^{(j)}$ defined in Lemma 6.) Since the matrices $M_{x,q-1}$ and M_{q-2} do not contain the variable x, they can be added to A' and B without violating the requirement that each variable appear at most once per matrix. We set $\mathcal{A} := \{A', B, C, S_1, \ldots, S_{q-2}\}$. Our preceding discussion shows that a satisfying for ϕ yields a simultaneous completion for $\mathcal{A}$ and vice-versa. $\square$

3 Simultaneous Completion when $q = d$

As a special case of Lemma 7, we see that deciding whether three matrices over $\mathbb{F}_2$ have a simultaneous completion is NP-complete. What about two matrices over $\mathbb{F}_2$? It is often the case that the difference between 2 and 3 often means the difference between polynomial-time decidability and NP-hardness, the classic example being 2SAT vs. 3SAT. A more relevant example is obtained by restricting the number of appearances of a variable in a CNF formula: 3SAT is NP-hard even if each variable appears at most three times, but SAT is polynomial-time decidable if each variable appears at most twice [28]. (A simple proof of this latter fact may be obtained by eliminating variables that appear negated twice or unnegated twice, considering the LP relaxation, and observing that the constraint matrix is totally unimodular.)

One might conjecture that this 2-or-3 phenomenon extends to simultaneous matrix completions, since the number of appearances of a variable in a set of matrices

[2] See, for example, Bondy and Murty [3, §6.2]. For completeness, we give a proof in Appendix A.

seems related to the number of appearances of a variable in a CNF formula. Can we show analogously that completing two matrices over $\mathbb{F}_2$ is easy? Somewhat surprisingly, we will show in that the opposite result is true: SIM-COMPLETION$(2,2)$ is NP-complete.

As a first step in this direction, we show that Algorithm 1 can fail when given two matrices and forced to operate over $\mathbb{F}_2$. Define the following two matrices.

$$T_1 := \begin{pmatrix} x & 0 & 1 \\ y & 1 & 0 \\ z & 1 & 1 \end{pmatrix} \quad \text{and} \quad T_2 := \begin{pmatrix} x & 0 & 1 \\ y & 1 & 0 \\ 0 & z & 1 \end{pmatrix}$$

These matrices have completions when the following equations are satisfied.

$$\begin{aligned} &\det T_1 \;=\; x+y+z \;=\; 1 \\ &\implies (x,y,z) \in \{(1,0,0),(0,1,0),(0,0,1),(1,1,1)\} \\ &\det T_2 \;=\; x+yz \;=\; 1 \\ &\implies (x,y,z) \in \{(1,0,0),(1,0,1),(1,1,0),(0,1,1)\} \end{aligned}$$

The only simultaneous completion for these two matrices is $(x,y,z)=(1,0,0)$. Consider now the operation of Algorithm 1 on these matrices. It begins by arbitrarily setting the indeterminate x to 0. The resulting matrices are both still non-singular because their determinants are $\det T_1 = y+z$ and $\det T_2 = yz$ respectively. Next, the indeterminate y is considered. Setting y to 0 makes B singular so y is set to 1. The resulting determinants are $\det T_1 = 1+z$ and $\det T_2 = z$. However, there is no choice of z that ensures both determinants are non-zero, and hence the algorithm fails.

The preceding example shows that our simple greedy algorithm fails to find a completion for two matrices over $\mathbb{F}_2$. We will show that this is unavoidable: SIM-COMPLETION(q,q) is NP-complete for every prime power $q \geq 2$. The key to proving this result involves the following interesting matrix. For $n \geq 1$, we define the matrix

$$R_n(x_1,\ldots,x_n) \;:=\; \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 & 0 \\ x_1 & 1 & 1 & \ldots & 1 & 1 \\ 0 & x_2 & 1 & \ldots & 1 & 1 \\ 0 & 0 & x_3 & \ldots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & x_n & 1 \end{pmatrix}.$$

We remark that R_n differs from other matrices used in this paper since it is in Hessenberg form, not in the form of a block-diagonal matrix with 2×2 blocks.

LEMMA 8. *The determinant of R_n (over any field) is*

$$\det R_n \;=\; \prod_{i=1}^{n}(1-x_i) \;-\; \prod_{i=1}^{n}(-x_i). \tag{3.1}$$

Proof. To prove this result, we generalize the matrix R_n slightly setting the top-right entry to be the indeterminate v rather than 0. We claim that, for this modified matrix,

$$\det R_n \;=\; \prod_{i=1}^{n}(1-x_i) \;+\; (v-1)\cdot\prod_{i=1}^{n}(-x_i).$$

The proof is by induction. For $n=1$, we have $\det R_1 = \det\left(\begin{smallmatrix} 1 & v \\ x_1 & 1 \end{smallmatrix}\right) = 1 - vx_1$, as required. For $n>1$, we compute the determinant by performing row-expansion on the last row.

$$\begin{aligned} \det R_n \;&=\; \det\begin{pmatrix} 1 & \ldots & 1 & 1 & v \\ x_1 & \ldots & 1 & 1 & 1 \\ 0 & \ddots & \vdots & \vdots & \vdots \\ 0 & \ldots & x_{n-1} & 1 & 1 \\ 0 & \ldots & 0 & x_n & 1 \end{pmatrix} \\ &=\; (-x_n)\cdot\det\begin{pmatrix} 1 & \ldots & 1 & v \\ x_1 & \ldots & 1 & 1 \\ 0 & \ddots & \vdots & \vdots \\ 0 & \ldots & x_{n-1} & 1 \end{pmatrix} \\ &\quad+\; 1\cdot\det\begin{pmatrix} 1 & \ldots & 1 & 1 \\ x_1 & \ldots & 1 & 1 \\ 0 & \ddots & \vdots & \vdots \\ 0 & \ldots & x_{n-1} & 1 \end{pmatrix} \\ &=\; (-x_n)\cdot\left(\prod_{i=1}^{n-1}(1-x_i) \;+\; (v-1)\cdot\prod_{i=1}^{n-1}(-x_i)\right) \\ &\quad+\; \left(\prod_{i=1}^{n-1}(1-x_i)\right) \\ &=\; (1-x_n)\cdot\prod_{i=1}^{n-1}(1-x_i) \\ &\quad+\; (-x_n)\cdot(v-1)\cdot\prod_{i=1}^{n-1}(-x_i) \end{aligned}$$

Setting $v=0$ completes the proof. $\square$

LEMMA 9. *For any prime power q, the problem SIM-COMPLETION(q,q) is NP-complete.*

Proof. This theorem is an adaption of Lemma 7, where the replication gadget is based on the matrix R_n rather than the matrices B and C. Suppose we are given ϕ, an instance of CIRCUIT-SAT. We construct the matrices A' and S_i $(1 \leq i \leq q-2)$ as in Lemma 7. Recall that A' is a block-diagonal matrix with a block for each NAND gate, where each appearance of a variable uses a distinct replicated copy.

Next we will use the matrix R_n. Lemma 8 shows that $\det R_n$ is the arithmetization of the boolean formula $\bigwedge_{i=1}^{n} \bar{x}_i \ \vee \ \bigwedge_{i=1}^{n} x_i$. Thus the matrix R_n satisfies the following curious property: any completion for R_n where $x_1, \ldots, x_n \in \{0, 1\}$ satisfies $x_1 = \cdots = x_n$. This is precisely the property needed to replicate a variable n times. Thus, for every variable (including the inputs and intermediate wires) $a, b, c, \ldots$ in ϕ, we add a copy of R_n to the following matrix.

$$F \ := \ \begin{pmatrix} R_n(a^{(1)}, \ldots, a^{(n)}) & & \\ & R_n(b^{(1)}, \ldots, b^{(n)}) & \\ & & \ddots \end{pmatrix}$$

Let $\mathcal{A} = \{A', F, S_1, \ldots, S_{q-2}\}$. For each indeterminate x, we add the matrices $M_{x,1}$ as blocks in A' and matrices $M_{x,2}$ as blocks in F, as in Lemma 7. Our construction implies that a completion for $\mathcal{A}$ determines a satisfying assignment for ϕ and vice-versa. Hence SIM-COMPLETION(q, q) is NP-complete. $\square$

This completes the proof of the main theorem. Several hardness results that are consequences of this theorem are given in Section 5. Before describing these consequences, the following section discusses certain algorithms that contrast with our hardness results.

4 Algorithms for Matrix Completion

The first efficient algorithm for matrix completion is Lovász's randomized algorithm [20], mentioned in Section 1. This algorithm is oblivious to the number of occurrences of an indeterminate in the matrix. When executed over a field of size at least cn for any $c > 1$, it succeeds with probability at least $1 - c^{-1}$. The time required is $O(n^{\omega})$, assuming that one wishes to verify the rank of the resulting matrix, where $\omega < 2.376$ gives the running time of matrix multiplication [1, 5]. One may boost the probability of success by running n independent iterations of the algorithm. Thus the algorithm can be made succeed with constant probability over a field of size $n + 1$, as indicated in Figure 1.

The complexity of matrix completion depends not only on the number of occurrences of an indeterminate, but also on the rank of the matrix induced by the entries occupied by each indeterminate (i.e., set entries containing the indeterminate to 1 and all other entries to 0, and consider the rank of the resulting matrix). If all numbers in the matrix are 0 and, for each indeterminate, the positions that it occupies form a matrix with rank 1, or a skew-symmetric matrix with rank 2, Lovász [21] showed that deciding whether the matrix is singular is in NP ∩ co-NP. Gurvits [12] studied the related notion of mixed discriminants of rank 2 positive semi-definite matrices.

Geelen developed a conceptually simple algorithm find finding completions in matrices where each indeterminate appears at most once [7]. This work was extended to handle mixed skew-symmetric matrices whose entries are either indeterminates or zero, and each indeterminate appears at most once (twice, if we include symmetric partners) [8]. Berdan [2] developed more efficient implementations of these algorithms. Geelen, Iwata and Murota [11], and later Geelen and Iwata [10], give quite technical algorithms for computing the rank of mixed skew-symmetric matrices whose entries can contain arbitrary numbers.

5 Consequences of Main Theorem

In this section, we describe the consequence of our main theorem which were briefly mentioned in the introduction.

5.1 Completion of a Single Mixed Matrix

To formally discuss completion of a single matrix, we must distinguish two notions of rank. The distinction was not important up to this point because the notions are equivalent for mixed matrices where each variable appears only once. The ***term-rank*** of A is the largest value r such that there exists an $r \times r$ submatrix whose determinant is a non-zero polynomial. The ***generic-rank*** of A over $\mathbb{F}_q$ is the largest value r such that there exists a completion of A in $\mathbb{F}_q$ and an $r \times r$ submatrix whose determinant *evaluates* to a non-zero value under this completion. Our terminology here follows that of Murota [25] and Lovász [21]. The term ***max-rank*** has also used to refer to the generic-rank in other work [4]. The term-rank and generic-rank are not in general the same. In particular, the term-rank does not depend on the field under consideration, but the generic-rank does. As an example, note that $\det \left(\begin{smallmatrix} x & x \\ 1 & x \end{smallmatrix}\right) = x(x-1)$ evaluates to zero at every point in $\mathbb{F}_2$. Thus this matrix has term-rank 2 but generic-rank 1 over $\mathbb{F}_2$.

Both the term-rank and the generic-rank are interesting from a computational complexity perspective, as both convey insight into the intrinsic difficulty of matrix completion and polynomial identity testing problems. We focus on the generic-rank, which has been studied previously [4], although to a lesser extent than the term-rank. Our corollary is as follows.

COROLLARY 10. *Let M be a mixed matrix over any field $\mathbb{F}_q$ where variables can appear several times. The problem of deciding whether M is non-singular (with respect to generic-rank) is NP-complete, even if we require that each variable appear in M at most q times, and all numbers are either* 0 *or* 1.

Proof. The proof is a simple reduction. Suppose we are given a set of matrices $\mathcal{A} = \{A_1, \ldots, A_q\}$ for which we must find a completion over $\mathbb{F}_q$. Each indeterminate can appear at most once in any particular matrix A_i. Define M to be the block-diagonal matrix with the matrices in $\mathcal{A}$ on its diagonal, and note that M satisfies the hypotheses of this corollary. Note that a completion for M implies a simultaneous completion for $\mathcal{A}$. Hence, deciding whether M is non-singular with respect to its generic rank implies deciding whether there is a simultaneous completion for $\mathcal{A}$. $\square$

For the sake of comparison, we provide a brief description of the proof of Buss et al. They reduce from 3SAT by arithmetizing the CNF formula. It is well-known that 3SAT is hard only when each variable appears at least three times. The resulting formula is raised to the power $q-1$ to ensure that its value is either 0 or 1. They then apply Valiant's construction [29] to this formula, obtaining a (small) matrix whose determinant equals that formula. The matrix that results from their reduction is hard only when there are at least $3q-3$ occurrences of some variable. In contrast, Corollary 10 shows hardness even when all variables occur at most q times. Thus our result is strictly stronger, even in the case that $q = 2$.

5.2 Completion of a Skew-Symmetric Mixed Matrix

Consider now the class of ***skew-symmetric*** mixed matrices (i.e., matrices M where $M = -M^{\mathsf{T}}$). Geelen, Iwata and Murota [11, 25, 10] give an efficient deterministic algorithm for computing the rank of a skew-symmetric matrix where each indeterminate appears exactly once (twice, if we include its symmetric partner). This implies a greedy algorithm to compute a completion over $\mathbb{F}_2$, along the lines of Algorithm 1.

However, if we allow variables to occur more than once, the problem becomes hard again. The following is a consequence of Corollary 10.

Corollary 11. *Let M be a skew-symmetric mixed matrix over $\mathbb{F}_2$ where each indeterminate appears at most twice (four times, if we include symmetric partners). The problem of deciding whether M is non-singular (with respect to generic-rank) is NP-complete.*

Proof. Simply observe that a mixed matrix M is non-singular iff $\begin{pmatrix} 0 & M \\ -M^{\mathsf{T}} & 0 \end{pmatrix}$ is non-singular. $\square$

6 Open Questions

Our interest in matrix completions originated with the application to network coding [17, 14, 13]. The field size over which matrices can be completed is of critical interest in this application because it directly relates to the block length used for data transmission. Codes with short block length are desirable since they require less storage and computation overhead at the network routers. It was previously known that finding network codes over fields of *constant size* is NP-hard [19]. However, the best known algorithms [15, 13] require fields of linear size. It would be interesting to see if our results on hardness of matrix completion can be refined to show hardness of network coding over small fields.

Acknowledgments

The authors would like to thank Swastik Kopparty, Kazuo Murota, David Woodruff, and the anonymous reviewers for many helpful comments and suggestions.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] M. Berdan. A matrix rank problem. Master's thesis, University of Waterloo, Dec. 2003.

[3] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Elsevier Science, 1976.

[4] J. F. Buss, G. S. Frandsen, and J. O. Shallit. The computational complexity of some problems in linear algebra. *Journal of Computer and System Sciences*, 58(3):572–596, 1999.

[5] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[6] J. Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, 71B:241–245, 1967.

[7] J. F. Geelen. Maximum rank matrix completion. *Linear Algebra and its Applications*, 288:211–217, 1999.

[8] J. F. Geelen. An algebraic matching algorithm. *Combinatorica*, 20(1):61–70, 2000.

[9] J. F. Geelen. Matching theory. Lecture notes from the Euler Institute for Discrete Mathematics and its Applications, 2001.

[10] J. F. Geelen and S. Iwata. Matroid matching via mixed skew-symmetric matrices. *Combinatorica*, 25(2):187–215, 2005.

[11] J. F. Geelen, S. Iwata, and K. Murota. The linear delta-matroid parity problem. *Journal of Combinatorial Theory, Series B*, 88(2):377–398, 2003.

[12] L. Gurvits. On the complexity of mixed discriminants and related problems. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 447–458, Aug. 2005.

[13] N. J. A. Harvey, D. R. Karger, and K. Murota. Deterministic network coding by matrix completion. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 05)*, pages 489–498, 2005.

[14] T. Ho, D. R. Karger, M. Médard, and R. Koetter. Network coding from a network flow perspective. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2003.

[15] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, June 2005.

[16] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computation (STOC)*, pages 355–364, June 2003.

[17] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, 2003.

[18] M. Laurent. Matrix completion problems. In C. Floudas and P. Pardalos, editors, *The Encyclopedia of Optimization*, volume III, pages 221–229. Kluwer, 2001.

[19] A. R. Lehman and E. Lehman. Complexity classification of network information flow problems. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 04)*, pages 135–143, Jan. 2004.

[20] L. Lovász. On determinants, matchings and random algorithms. In L. Budach, editor, *Fundamentals of Computation Theory, FCT '79*, pages 565–574. Akademie-Verlag, Berlin, 1979.

[21] L. Lovász. Singular spaces of matrices and their applications in combinatorics. *Bol. Soc. Braz. Mat*, 20:87–99, 1989.

[22] M. Mucha and P. Sankowski. Maximum matchings via Gaussian elimination. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 248–255, Oct. 2004.

[23] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[24] K. Murota. Mixed matrices – irreducibility and decomposition. In R. A. Brualdi, S. Friedland, and V. Klee, editors, *Combinatorial and Graph-Theoretic Problems in Linear Algebra*, volume 50 of *IMA Volumes in Mathematics and its Applications*, pages 39–71. Springer-Verlag, 1993.

[25] K. Murota. *Matrices and Matroids for Systems Analysis*. Springer-Verlag, 2000.

[26] T. G. Room. *The Geometry of Determinantal Loci*. Cambridge University Press, 1938.

[27] P. Sankowski. Dynamic transitive closure via dynamic matrix inverse. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 509–517, Oct. 2004.

[28] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984.

[29] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computation (STOC)*, pages 249–261, Apr. 1979.

A The edge-chromatic number of K_ℓ

The following result is well-known, and can be found in Bondy and Murty [3], for example. This proof was communicated to us by Swastik Kopparty.

LEMMA 12. *Suppose that ℓ is even. Then the edge-chromatic number of K_ℓ is $\ell - 1$.*

Proof. The key to coloring K_ℓ is to find a coloring of $K_{\ell-1}$ with appropriate algebraic structure. Consider the complete graph whose vertices are the elements of the ring $\mathbb{Z}_{\ell-1}$. In this graph, we assign colors that are also elements of $\mathbb{Z}_{\ell-1}$ as follows: the edge $\{v, w\}$ receives the color $v + w$. Now observe that at each vertex v, the color $2v$ is unused. Furthermore, since $\ell - 1$ is odd, the map $v \mapsto 2v$ is a bijection. This implies that the colors missing at the vertices are all distinct. We now add a new vertex x and connect it to all existing vertices. Each new edge $\{v, x\}$ is assigned the missing color $2v$. This yields a coloring of K_ℓ with $\ell - 1$ colors. $\square$