Matroid Intersection, Pointer Chasing, and Young's Seminormal Representation of S_n

Nicholas J. A. Harvey*
Massachusetts Institute of Technology
nickh@mit.edu

Abstract

We consider the number of queries needed to solve the matroid intersection problem, a question raised by Welsh (1976). Given two matroids of rank r on n elements, it is known that $O(nr^{1.5})$ independence queries suffice. However, no non-trivial lower bounds are known for this problem.

We make the first progress on this question. We describe a family of instances of rank $r = n/2$ based on a pointer chasing problem, and prove that $(\log_2 3)\, n - o(n)$ queries are necessary to solve these instances. This gives a constant factor improvement over the trivial lower bound of n for matroids of this rank.

Our proof uses methods from communication complexity and group representation theory. We analyze the communication matrix by viewing it as an operator in the group algebra of the symmetric group and explicitly computing its spectrum.

1 Introduction

The matroid intersection problem — finding a maximum cardinality independent set in two given matroids — is a fundamental problem in combinatorial optimization. Research on this topic led to significant developments on integral polyhedra [5], submodular functions [6], and other areas. The combinatorial structure of matroids can be exploited algorithmically, leading to applications of matroids in many areas. This paper examines the computational efficiency of matroids, focusing on the query complexity of matroid intersection.

To give context for our work, some formal definitions are required (see also Cook et al. [2]). Let S be a *ground set* of size n, and let $\mathcal{I} \subseteq 2^S$ be a non-empty family of sets satisfying

- $A \subseteq B$ and $B \in \mathcal{I} \implies A \in \mathcal{I}$;
- $A \in \mathcal{I}$ and $B \in \mathcal{I}$ and $|A| < |B| \implies \exists b \in B \setminus A$ such that $A + b \in \mathcal{I}$.

The pair $(S, \mathcal{I})$ is called a *matroid* and the sets in $\mathcal{I}$ are called *independent sets*. As a motivating example, one may think of elements of S as vectors in a vector space, and the family $\mathcal{I}$ as all sets of linearly independent vectors. A *base* is a maximum cardinality independent set, and the *rank* of a matroid is the size of any base. The *rank function* $\rho : 2^S \to \mathbb{N}$ satisfies $\rho(A) = \max_{A \supseteq I \in \mathcal{I}} |I|$. Given two matroids $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$, the matroid intersection problem is to find a maximum cardinality set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$. A related (and equivalent) problem is to find a common base of the two matroids, if any.

Most of the algorithms developed for matroid intersection work in an *oracle model*. That is, the algorithms only access each matroid by performing a simple query: given $A \subseteq S$, is A in $\mathcal{I}_1$ (or in $\mathcal{I}_2$)? The procedure which answers the queries is called an *independence oracle*. Cunningham [3] developed a matroid intersection algorithm using only $O(nr^{1.5})$ independence oracle queries for matroids of rank r.

How many queries are necessary to solve the matroid intersection problem? To our knowledge, this question was first asked by Welsh[1] [21, p368] in 1976. As Cunningham's bound shows, the number of queries needed is linear in n whenever r is a constant. Thus Welsh's question is only interesting for matroids of large rank; the precise value of "large" is not so important since one can adjust the rank by padding arguments. Thus, to be definitive, we assume that $r \approx n/2$ throughout this paper.

We describe a new family of matroids based on a pointer chasing problem. Roughly speaking, $\mathbf{M}_1$ corresponds to a permutation π in the symmetric group $\mathcal{S}_n$ and $\mathbf{M}_2$ corresponds to a permutation $\sigma \in \mathcal{S}_n$. Both matroids have rank $n/2 + 1$. The two matroids have a common base iff the cycle structure of the composition $\sigma^{-1} \circ \pi$ satisfies a certain property. We prove that $(\log_2 3)\, n - o(n)$ queries are necessary to solve the matroid intersection problem for these instances. This result improves on the trivial lower bound of n that one can obtain for matroids of this rank by an easy adversary argument. We conjecture that actually $\omega(n)$ queries are necessary for our family of instances, although proving this seems to be difficult.

*Supported by a Natural Sciences and Engineering Research Council of Canada PGS Scholarship, by NSF contract CCF-0515221 and by ONR grant N00014-05-1-0148.

[1] To be precise, Welsh asked about the number of queries needed to solve the matroid partition problem, which is equivalent to matroid intersection, but was discovered earlier.

Our arguments are based on the communication complexity framework: the two given matroids are anthropomorphized into two computationally unbounded players, Alice and Bob, and one analyzes the number of bits that must be communicated between them to solve the matroid intersection problem. This yields a lower bound on the number of independence queries required by any algorithm.

A standard technique for proving lower bounds in this framework is based on the communication matrix C, which is the truth table of the function that Alice and Bob must compute. It is known that $\log_2 \operatorname{rank} C$ gives a lower bound on the number of bits which must be communicated between Alice and Bob. Since our instances are derived from the symmetric group, it is natural to use representation theory to analyze the matrix's rank. Section 4 does this by viewing the communication matrix as an operator in the group algebra. Surprisingly, we show that the matrix is diagonalizable (in Young's seminormal basis), its eigenvalues are all integers, and their precise values can be computed by considering properties of Young tableaux.

2 Communication Complexity

Our lower bound uses methods from the field of communication complexity [9, 10]. This section briefly describes the concepts that we will need.

2.1 Communication Problems A *communication problem* is specified by a function $f(X, Y)$, where X is Alice's input, Y is Bob's input, and the range is $\{0, 1\}$. A communication problem is solved by a *communication protocol*, in which Alice and Bob send messages to each other until one of them can decide the solution $f(X, Y)$. The player who has found the solution declares that the protocol has halted, and announces the solution.

The *deterministic communication complexity* of f is defined to be the minimum number of bits required by any deterministic communication protocol for f. This quantity is denoted $D(f)$.

Nondeterminism also plays an important role in communication complexity. This model involves a third party — a *prover* who knows both X and Y. In a *nondeterministic protocol* for f, the prover produces a certificate Z which is delivered to both Alice and Bob (Z is a function of both X and Y). Alice and Bob cannot communicate, other than receiving Z from the prover. If $f(X, Y) = 1$, then the certificate must suffice to convince Alice and Bob of this fact (Alice sees only X and Z, Bob sees only Y and Z). Otherwise, if $f(X, Y) = 0$, no certificate should be able to fool both Alice and Bob. The *nondeterministic communication complexity* is defined to be the minimum length of the certificate (in bits) in any nondeterministic protocol. We denote this quantity by $N^1(f)$.

A *co-nondeterministic protocol* is defined analogously, reversing the roles of TRUE and FALSE. The *co-nondeterministic complexity* is also defined analogously, and is denoted by $N^0(f)$. One can easily see that $N^0(f) \leq D(f)$ and $N^1(f) \leq D(f)$.

For any communication problem f, the *communication matrix* is a $\{0, 1\}$ matrix $C(f)$ whose rows are indexed by Alice's inputs X and whose columns are indexed by Bob's inputs Y. The entries of C are simply $C(f)_{X,Y} = f(X, Y)$. There is a connection between algebraic properties of the matrix $C(f)$ and the communication complexity of f, as shown in the following lemma.

FACT 1. (MEHLHORN AND SCHMIDT [12]) *Over the complex numbers $\mathbb{C}$, we have $D(f) \geq \log_2 \operatorname{rank} C(f)$.*

2.2 Communication Complexity of Matroid Intersection

Let us now consider the matroid intersection problem in the communication complexity framework.

DEFINITION. The communication problem MAT-$\cap$:

- *Alice's Input:* A matroid $\mathbf{M}_1 = (S, \mathcal{I}_1)$.
- *Bob's Input:* A matroid $\mathbf{M}_2 = (S, \mathcal{I}_2)$.
- *Output:* If $\mathbf{M}_1$ and $\mathbf{M}_2$ have a common base then $f(\mathbf{M}_1, \mathbf{M}_2) =$ TRUE. Otherwise, it is FALSE.

By standard arguments, any matroid intersection algorithm which uses independence oracle queries can be transformed into a communication protocol for MAT-$\cap$. The point is that both Alice and Bob can independently simulate the given algorithm, and they only need to communicate whenever an oracle query is made. Thus $D(\text{MAT-}\cap)$ gives a lower bound on the number of oracle queries made by any matroid intersection algorithm. The remainder of this paper focuses on analyzing the communication complexities of MAT-$\cap$.

We begin with some easy observations using matroids of rank one. Let $X \subseteq S$ be arbitrary, and let $\mathcal{B}(X) = \{ \{x\} : x \in X \}$. It is easy to verify that $\mathcal{B}(X)$ is the family of bases of a rank one matroid, denoted $\mathbf{M}(X)$. Given two sets $X, Y \subseteq S$, the two matroids $\mathbf{M}(X)$ and $\mathbf{M}(Y)$ have a common base iff $X \cap Y \neq \emptyset$. Thus, for this family of matroids, the MAT-$\cap$ problem is simply the complement of the well-known DISJOINTNESS problem. Thus, by classical results, we have $D(\text{MAT-}\cap) \geq n$ and $N^0(\text{MAT-}\cap) \geq n - o(n)$. The randomized communication complexity of MAT-$\cap$, which we will not define, is also $\Omega(n)$.

As it turns out, this argument gives a tight lower

bound on $N^0(\text{MAT-}\cap)$. To show this, we will use the following min-max relation of Edmonds [5].

FACT 2. (MATROID INTERSECTION THEOREM) *Let $\mathbf{M}_1 = (S, \mathcal{I}_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2)$ be given. Let ρ_1 and ρ_2 denote their rank functions, respectively. Then $\max_{I \in \mathcal{I}_1 \cap \mathcal{I}_2} |I| = \min_{A \subseteq S} \big(\rho_1(A) + \rho_2(S \setminus A) \big)$.*

LEMMA 3. *$N^1(\text{MAT-}\cap) \leq n$ and $N^0(\text{MAT-}\cap) \leq n + 2(\lfloor \log n \rfloor + 1)$.*

Proof. To convince Alice and Bob that their two matroids have a common base, it suffices to present them with that base B. Alice and Bob independently check that B is a base for their respective matroids. The set B can be represented using n bits, hence $N^1 \leq n$.

To convince Alice and Bob that their two matroids do not have a common base, we invoke the matroid intersection theorem. The prover computes a set $A \subseteq S$ which is a minimizing set in Fact 2. The co-nondeterministic certificate Z consists of the set A, and two integers z_1 and z_2. Alice and Bob both check that $z_1 + z_2 < r$, and individually check that $z_1 = \rho_1(A)$ and $z_2 = \rho_2(S \setminus A)$. If this holds then the two matroids cannot have a common base. The length of this certificate is at most $n + 2(\lfloor \log n \rfloor + 1)$. □

We remark that $N^1(\text{MAT-}\cap) = \Omega(n)$ can be proven. In fact, it can even be proven for the restricted class of matroid intersection instances that we define in the following section.

3 Pointer Chasing Instances

One interesting category of communication problems is pointer chasing problems [1, 4, 15, 16, 17]. We now show that matroid intersection leads to an interesting pointer chasing problem.

The motivating example to keep in mind is the class of *almost 2-regular bipartite graphs*. Let G be a graph with a bipartition of the vertices into U and V. Each vertex in U (resp., in V) has degree 2, except for two distinguished vertices $u_1, u_2 \in U$ (resp., $v_1, v_2 \in V$), which have degree 1. (So $|U| = |V|$.) The connected components of G are two paths with endpoints in $\{u_1, u_2, v_1, v_2\}$, and possibly some cycles. It is easy to see that G has a perfect matching iff G does not contain a path from u_1 to u_2 (equiv., from v_1 to v_2).

Let us now reformulate this example slightly. Let $S = U \cup V$ where $|U| = |V| = N := n/2$. Let $\mathcal{P}$ be a partition of S into pairs, where each pair contains exactly one element of U and one element of V. We can write $\mathcal{P}$ as $\big\{ \{u_i, v_{\pi(i)}\} : i = 1, \ldots, N \big\}$, where $\pi : U \to V$ is a bijection. Now $\mathcal{P}$ can be used to define a matroid. Fix arbitrarily $1 \leq k \leq N$, and let $\mathcal{B}_k^\pi$ be the family of all B such that

$$|B \cap \{u_i, v_{\pi(i)}\}| = \begin{cases} 2 & (\text{if } i = k) \\ 1 & (\text{otherwise}). \end{cases}$$

One may verify that $\mathcal{B}_k^\pi$ is the family of bases of a matroid $\mathbf{M}_k^\pi$ (a partition matroid). Let $\mathcal{M}_k$ be the set of all such matroids (keeping k fixed, and letting π vary).

LEMMA 4. *Let $\mathbf{M}_1^\pi \in \mathcal{M}_1$ and $\mathbf{M}_2^\sigma \in \mathcal{M}_2$. Note that $\sigma^{-1} \circ \pi$ is a permutation on U. We claim that $\mathbf{M}_1^\pi$ and $\mathbf{M}_2^\sigma$ have a common base iff elements u_1 and u_2 are in the same cycle of $\sigma^{-1} \circ \pi$.*

The proof of this lemma mirrors the argument characterizing when almost 2-regular bipartite graphs have a perfect matching. Let us now interpret Lemma 4 in the communication complexity framework.

DEFINITION. The IN-SAME-CYCLE problem:

- *Alice's input:* A permutation $\pi \in \mathcal{S}_N$.
- *Bob's input:* A permutation $\sigma \in \mathcal{S}_N$.
- *Output:* If elements 1 and 2 are in the same cycle of $\sigma^{-1} \circ \pi$, then output TRUE. Otherwise FALSE.

Thus Lemma 4 has shown that IN-SAME-CYCLE reduces to MAT-$\cap$. Intuitively, Alice and Bob cannot decide the IN-SAME-CYCLE problem unless one of them has learned the entire cycle containing 1 and 2, which might have length $\Omega(N)$, so it is reasonable to believe that $\Omega(N \log N)$ bits of communication are required.

The remainder of this paper proves the following theorem.

THEOREM 5. *Let C denote the communication matrix for* IN-SAME-CYCLE. *Then* $\operatorname{rank} C$ *equals*

$$1 + \sum_{1 \leq i \leq N-1} \sum_{1 \leq j \leq \min\{i, N-i\}} \binom{N}{i,\, j,\, N-i-j}^2 \cdot \frac{j^2 (i-j+1)^2}{N (N-1) (N-i) (N-j+1)}.$$

COROLLARY 6. *$D(\text{IN-SAME-CYCLE}) \geq (\log_2 9)N - o(N)$. Consequently, the matroid intersection problem for matroids with rank $n/2 + 1$ and ground set size n requires at least $(\log_2 3)n - o(n)$.*

Proof. Note that $\binom{N}{N/3,\, N/3,\, N/3} = 3^{N - o(N)}$, and therefore $\operatorname{rank} C = 9^{N - o(N)}$. Fact 1 therefore implies the lower bound on $D(\text{IN-SAME-CYCLE})$. The lower bound for matroid intersection follows since the matroids in $\mathcal{M}_k$ have rank $n/2 + 1$ and ground set size n. □

4 Representation Theory and In-Same-Cycle

4.1 Preliminaries This section relies on the representation theory of the symmetric group. We give a brief introductory discussion here. More detailed expositions can be found in James-Kerber [8], Naĭmark [13], Sagan [19], and Vershik-Okounkov [20]. The exposition of Sagan is particularly lucid.

A representation h of $\mathcal{S}_N$ is a function $h : \mathcal{S}_N \to GL_m(\mathbb{C})$, where $GL_m(\mathbb{C})$ is the group of non-singular $m \times m$ matrices over the complex numbers. The function h need not be one-to-one, but must be a homomorphism: $h(\sigma)h(\tau) = h(\sigma \circ \tau)$ for all $\sigma, \tau \in \mathcal{S}_N$. The value m is called the *dimension* of the representation h.

The *regular representation* R is of particular importance. In the standard basis, R may be defined as follows. $R(\tau)$ is a matrix whose rows and columns are indexed by $\mathcal{S}_N$, and whose entries are given by

$$R(\tau)_{\pi,\sigma} \;=\; \begin{cases} 1 & \text{if } \pi = \sigma \circ \tau \\ 0 & \text{otherwise.} \end{cases}$$

Another way of describing the regular representation is as follows. Consider the set of all formal linear combinations over $\mathbb{C}$ of elements of $\mathcal{S}_N$; a typical element is $\sum_{\pi \in \mathcal{S}_N} \alpha_\pi \pi$. This set is called the *group algebra* of $\mathcal{S}_N$: it is a vector space over $\mathbb{C}$, with multiplication defined by

$$\Big(\sum_{\pi \in \mathcal{S}_N} \alpha_\pi \pi\Big) \cdot \Big(\sum_{\pi \in \mathcal{S}_N} \beta_\pi \pi\Big) = \sum_{\pi \in \mathcal{S}_N} \big(\textstyle\sum_{\sigma \in \mathcal{S}_N} \alpha_\sigma \beta_{\sigma^{-1} \circ \pi}\big)\pi.$$

The matrix $R(\tau)$ can be viewed as the permutation matrix expressing how multiplication on the right by τ permutes the standard basis of the group algebra.

Let λ be a *partition* of N, i.e., λ is a non-increasing sequence $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ of positive integers whose sum is N. This is denoted $\lambda \vdash N$. A *Ferrers diagram of shape* λ is a two-dimensional array of boxes such that the rows are left-aligned, and the i^{th} row contains λ_i boxes. For example, Figure 1 (a) shows the Ferrers diagram of the partition $(5, 4, 4, 2, 1, 1) \vdash 17$. We will use the terms "partition" and "shape" interchangeably.

Let $\lambda \vdash N$. A *Young tableau* of shape λ is a bijective assignment of the integers in $\{1, \ldots, N\}$ to the boxes of the Ferrers diagram for λ. A *standard Young tableau*, or *SYT*, is one such that the values in each row increase from left to right, and the values in each column increase from top to bottom. Figure 1 (b) shows an example.

Let $\lambda \vdash N$. Let v be a box in the Ferrers diagram of λ. The *hook* of box v, denoted h_v, is the set of boxes in the same row as v but to its right or in the same column as v but beneath it (including v itself). This is illustrated in Figure 1 (c).

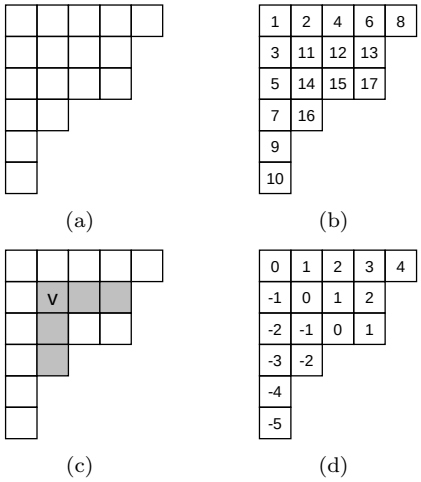


Figure 1: (a) A Ferrers diagram. (b) A standard Young tableau. (c) A box v and its hook h_v. (d) The "content" of all boxes in this Ferrers diagram.

Fact 7. (Hook Length Formula) *The number of SYT of shape λ is denoted f_λ, and has value $f_\lambda = \frac{N!}{\prod_v |h_v|}$, where the product is over all boxes v in the Ferrers diagram for λ.*

There are several canonical representations of $\mathcal{S}_N$ known as *irreducible representations*. (Henceforth, we use the shorthand *irrep.*) The irreps are indexed (in a canonical way) by the partitions of N. Let Y_λ denote the irrep corresponding to partition λ. For any $\pi \in \mathcal{S}_N$, the notation $Y_\lambda(\pi)$ denotes the matrix associated with π by this irrep. For any set $S \subseteq \mathcal{S}_N$, let $Y_\lambda(S) = \sum_{\pi \in S} Y_\lambda(\pi)$.

Two representations h and h' are said to be equivalent if one can be obtained from the other simply by changing the basis, i.e., if $\exists B$ such that for all π, $h(\pi) = B\, h'(\pi)\, B^{-1}$. When discussing irreps, we will also need to specify the particular basis that is used. Throughout this paper, the basis of interest will be Young's seminormal basis. The definition of this basis is not crucial for us, but we will need some of its properties. One important property is: in the vector space operated upon by the irrep Y_λ, the basis elements correspond bijectively (in a canonical way) to the various SYT of shape λ. Thus the dimension of Y_λ is f_λ.

Fact 8. *The regular representation R can be expressed as a direct sum of irreps. Specifically, there exists a change of basis matrix B such that, for each $\tau \in \mathcal{S}_N$, $BR(\tau)B^{-1}$ is a block-diagonal matrix where each block is a copy of an irrep $Y_\lambda(\tau)$; additionally, each irrep $Y_\lambda(\tau)$ occurs exactly f_λ times in this direct sum.*

For $1 \le j \le N$, the j^{th} *Jucys-Murphy element* is the member of the group algebra defined by $J_j = \sum_{1 \le i < j} (i,j)$. Here, (i,j) denotes a transposition in $\mathcal{S}_N$. For convenience, we may also view J_j as a subset of $\mathcal{S}_N$: the set of $j-1$ transpositions which appear with non-zero coefficient in J_j.

For a Ferrers diagram of shape λ, the *content* of the box (a,b) (i.e., in row a and column b) is the integer $b-a$. This is illustrated in Figure 1 (d); note that the content values are constant on each negative-sloping diagonal. For any standard Young tableau t and $1 \le j \le N$, define $\mathrm{cont}(t,j)$ to be the content of the box occupied by element j in tableau t.

FACT 9. *$Y_\lambda(J_j)$ is a diagonal matrix and the diagonal entries are $Y_\lambda(J_j)_{t,t} = \mathrm{cont}(t,j)$, where t is a tableau of shape λ.*

4.2 The In-Same-Cycle Problem In this section, we compute the rank of the communication matrix for the IN-SAME-CYCLE problem, and thereby prove Theorem 5. Surprisingly, we will show that this matrix is diagonalizable, and that the values of those diagonal entries (i.e., the spectrum) are integers that can be precisely computed.

Overview. Our argument proceeds as follows.

- The matrix C can be written as a sum of matrices in the regular representation.
- There exists a change-of-basis matrix which block-diagonalizes the matrices of the regular representation (i.e., decomposes them into irreps). Thus C can also be block-diagonalized.
- The blocks of C can be expressed as a polynomial in the matrices corresponding to the Jucys-Murphy elements. Thus each block is actually a diagonal matrix (if the change-of-basis matrix is chosen properly).
- The diagonal entries of each block (i.e., eigenvalues of C) are given by a polynomial in the content values, so they can be explicitly computed. The rank of C is simply the number of non-zero eigenvalues, so a closed form for the rank can be given.

Let $\pi \in \mathcal{S}_N$ be the permutation corresponding to Alice's input and let $\sigma \in \mathcal{S}_N$ correspond to Bob's input. Define $\mathcal{K}_N$, or simply $\mathcal{K}$, to be

$$\mathcal{K}_N = \{\, \tau \in \mathcal{S}_N \,:\, 1 \text{ and } 2 \text{ are in the same cycle of } \tau \,\}.$$

Recall the definition of the communication matrix C: the entry $C_{\pi,\sigma}$ is 1 if $\sigma^{-1} \circ \pi \in \mathcal{K}$, and 0 otherwise. This leads to the following easy lemma.

LEMMA 10. *$C = \sum_{\tau \in \mathcal{K}} R(\tau)$, where $R(\tau)$ denotes a matrix of the regular representation.*

Now let B be the change-of-basis matrix which decomposes the regular representation into irreps, as mentioned in Fact 8. We will analyze the rank of C by considering the contribution from each irrep. We have

$$\begin{aligned} \operatorname{rank} C &= \operatorname{rank} B\,C\,B^{-1} \\ &= \operatorname{rank}\Big(\sum_{\tau \in \mathcal{K}} B\,R(\tau)\,B^{-1}\Big) \\ &= \sum_{\lambda \vdash N} f_\lambda \cdot \operatorname{rank} Y_\lambda(\mathcal{K}), \end{aligned} \tag{4.1}$$

where the third equality follows from Fact 8.

The following lemma gives the reason that the communication matrix for IN-SAME-CYCLE can be analyzed so precisely. It gives a direct connection between the IN-SAME-CYCLE problem and the Jucys-Murphy elements.

LEMMA 11. ("INSERTION SORT LEMMA")
$\sum_{\pi \in \mathcal{K}} \pi \;=\; J_2 \cdot \prod_{j=3}^{N} (1 + J_j)$.

Proof sketch. First, we show $\sum_{\pi \in \mathcal{S}_N} \pi = \prod_{j=2}^{N} (1+J_j)$. The argument is inductive: any permutation $\pi \in \mathcal{S}_N$ can be expressed as the product of a permutation $\pi' \in \mathcal{S}_{N-1}$ and some transposition (i,N) which places element N next to its neighbours in π. This argument is analogous to the behaviour of the Insertion Sort algorithm.

A similar argument proves the lemma. The difference lies in the factor of J_2 rather than $1 + J_2$. This ensures that element 2 will always be in the same cycle as element 1. □

We remark that Lemma 11 also shows that $|\mathcal{K}| = |\mathcal{S}_N|/2$. In other words, for any π,

$$\Pr_\sigma\,[\,\text{IN-SAME-CYCLE}(\pi,\sigma) = 1\,] \;=\; 1/2,$$

which is an easy but interesting fact.

Lemma 11 shows that the sum $\sum_{\pi \in \mathcal{K}} \pi$ can be expressed as a polynomial in the Jucys-Murphy elements. In other words, for every $\lambda \vdash N$, the matrix $Y_\lambda(\mathcal{K})$ can be expressed as a polynomial in the matrices $\{\, Y_\lambda(J_j) \,:\, 2 \le j \le N \,\}$. It follows directly from Fact 9 that $Y_\lambda(\mathcal{K})$ is diagonal. Moreover, for every SYT t of shape λ, the corresponding diagonal entry of $Y_\lambda(\mathcal{K})$ satisfies the expression

$$Y_\lambda(\mathcal{K})_{t,t} \;=\; Y_\lambda(J_2)_{t,t} \cdot \prod_{j=3}^{N} \big(1 + Y_\lambda(J_j)_{t,t}\big). \tag{4.2}$$

As mentioned above, the blocks of $B\,C\,B^{-1}$ are all of the form $Y_\lambda(\mathcal{K})$. Thus $B\,C\,B^{-1}$ is actually diagonal, and (4.2) completely determines the spectrum of C, using Fact 9.

In the remainder of this section, we will analyze (4.2) more closely. Our main goal is to determine when its value is non-zero. This holds whenever $Y_\lambda(J_2)_{t,t} \neq 0$ and $Y_\lambda(J_j)_{t,t} \neq -1$ for all $j \geq 3$. By Fact 9, $Y_\lambda(J_2)_{t,t} = 0$ only when 2 lies on the main diagonal of t, which is impossible in any SYT. Similarly, $Y_\lambda(J_j)_{t,t} = -1$ only when j lies on the first subdiagonal. So we have the following fact, which is crucial to the analysis.

For an SYT t, $Y_\lambda(\mathcal{K})_{t,t} \neq 0 \iff$
in tableau t, all values $j \geq 3$ avoid the first subdiagonal.

Let us now consider three cases.

Case 1: $\lambda_3 > 1$. Fix an arbitrary SYT t of shape λ. The box in position $(3,2)$ (row 3, column 2) of t contains some value $j \geq 6$. Since this box is on the first subdiagonal, we have $Y_\lambda(\mathcal{K})_{t,t} = 0$.

Case 2: $\lambda_2 = 0$, i.e., $\lambda = (N)$. There is a unique SYT of shape λ, in which every box $(1,j)$ contains j. Thus $Y_\lambda(J_j) = j - 1$ for all j, so (4.2) shows that the unique entry of $Y_\lambda(\mathcal{K})$ has value $N!/2$.

Case 3: $\lambda_2 \geq 1$ and $\lambda_3 \leq 1$. In the Ferrers diagram of shape λ, only the box $(2,1)$ is on the first subdiagonal. Consider now an SYT t of shape λ. If the box $(2,1)$ contains $j \geq 3$ then $Y_\lambda(\mathcal{K})_{t,t} = 0$.

On the other hand, if the box $(2,1)$ contains the value 2 then all values $j \geq 3$ avoid the first subdiagonal, implying that $Y_\lambda(\mathcal{K})_{t,t} \neq 0$. In fact, the precise value of $Y_\lambda(\mathcal{K})_{t,t}$ can be determined. Since the value 2 is in box $(2,1)$ we have $Y_\lambda(J_2)_{t,t} = -1$. The multiset $\{\, Y_\lambda(J_j)_{t,t} : j \geq 3 \,\}$ is simply the multiset of all content values in boxes excluding $(1,1)$ and $(2,1)$. Let B denote this set of $N-2$ boxes. Then

$$\begin{aligned} &Y_\lambda(\mathcal{K})_{t,t} \\ &\quad = Y_\lambda(J_2)_{t,t} \cdot \prod_{j=3}^{N} \big(1 + Y_\lambda(J_j)_{t,t}\big) \\ &\quad = -\prod_{(a,b)\in B} (1 + b - a) \\ &\quad = \lambda_1! \cdot (\lambda_2 - 1)! \cdot (N - \lambda_1 - \lambda_2)! \cdot (-1)^{N-\lambda_1-\lambda_2+1} \end{aligned}$$

We have now computed the entire spectrum of C. The remaining task is to compute the rank (i.e., enumerate the number of non-zero eigenvalues). As argued above, any shape λ with $\lambda_3 > 1$ contributes zero to the rank, and the shape $\lambda = (N)$ contributes exactly 1. It remains to consider shapes with $\lambda_2 \geq 1$ and $\lambda_3 \leq 1$. As argued above, the number of non-zero diagonal entries in a block corresponding to shape λ equals the number of SYT in which box $(2,1)$ contains the value 2; let us denote this quantity by g_λ. Furthermore, there are precisely f_λ copies of the block corresponding to shape λ (cf. Fact 8). Thus,

$$\operatorname{rank} C \;=\; 1 + \sum_{\substack{\lambda \text{ s.t.} \\ \lambda_2 \geq 1 \text{ and } \lambda_3 \leq 1}} f_\lambda \cdot g_\lambda. \tag{4.3}$$

LEMMA 12. *Let $\lambda \vdash N$ satisfy $\lambda_2 \geq 1$ and $\lambda_3 \leq 1$. Then*

$$f_\lambda = \binom{N}{\lambda_1,\, \lambda_2,\, N-\lambda_1-\lambda_2} \cdot \frac{\lambda_2\,(\lambda_1 - \lambda_2 + 1)}{(N-\lambda_1)\,(N-\lambda_2+1)}.$$

$$g_\lambda = \binom{N}{\lambda_1,\, \lambda_2,\, N-\lambda_1-\lambda_2} \cdot \frac{\lambda_2\,(\lambda_1 - \lambda_2 + 1)}{N(N-1)}.$$

The proof of Lemma 12 is given in Appendix A. Substituting into (4.3) yields

$$1 + \sum_{1 \leq \lambda_1 \leq N-1} \; \sum_{1 \leq \lambda_2 \leq \min\{\lambda_1, N-\lambda_1\}} \binom{N}{\lambda_1,\, \lambda_2,\, N-\lambda_1-\lambda_2}^2 \cdot \frac{\lambda_2^2\,(\lambda_1 - \lambda_2 + 1)^2}{N(N-1)(N-\lambda_1)(N-\lambda_2+1)}.$$

This concludes the proof of Theorem 5.

5 Discussion

Adversary Arguments. The trivial way to prove lower bounds on matroid intersection is using rank-one matroids, as in Section 2.2. However, instead of considering communication complexity, one can instead use adversary (i.e., evasiveness) arguments to show that the algorithm must make n queries to *each* matroid, giving a lower bound of $2n$. By padding the ground set with coloops, one obtains a lower bound of $2(n-r+1)$ queries for matroids of any rank $0 < r < n$.

In comparison, our result yields a lower bound of $3.16 \cdot \min\{r, n-r\}$ for matroids of rank r, by padding the ground set with either loops or coloops. Thus our lower bound is stronger for sufficiently large r.

Queries vs Communication. This suggests that one may obtain better lower bounds by directly considering query complexity rather than communication complexity. Indeed, it is conceivable that matroid intersection requires $\Omega(nr^{1.5})$ queries but $D(\text{MAT-}\cap) = O(n)$. However, directly analyzing the query complexity seems quite difficult as the independence oracle queries are very powerful compared to, say, the simple edge queries in the work of Rivest and Vuillemin [18].

One definitive statement concerning $D(\text{MAT-}\cap)$ is an upper bound of $O(n^2)$. This follows from the general result that $D(f) \leq (N^0(f)+1)(N^1(f)+1)$ (see [10]). Thus communication complexity will not suffice to prove a $\Omega(nr^{1.5})$ lower bound; at least, not in the present formulation.

In-Same-Cycle. In this paper, we have analyzed the In-Same-Cycle problem, using a rank argument to lower bound D(In-Same-Cycle). We conjecture that the rank lower bound is weak for this problem, and that actually D(In-Same-Cycle) $= \omega(n)$ holds. This seems difficult to prove, due to the paucity of techniques for proving gaps between the deterministic and non-deterministic complexities.

We were able to show an $\Omega(n \log n)$ lower bound on the *one-round* communication complexity (where only Alice talks to Bob). This bound holds even for randomized protocols. Also, one can show that N^0(In-Same-Cycle) $= \Omega(n)$ and N^1(In-Same-Cycle) $= \Omega(n)$.

Submodular Function Minimization. A problem that generalizes matroid intersection is that of minimizing a *submodular function*. The connection between these two problems stems from Fact 2, since $g(A) := \rho_1(A) + \rho_2(S \setminus A)$ is a submodular function. It is known that $O(n^5)$ queries suffice to minimize a submodular function [14], and it has been an outstanding open question to prove a lower bound better than n (see [7], [11, p387]). One can show that $D(\text{Mat-}\cap)/\log r$ gives a lower bound on the number of queries needed to minimize a submodular function; this seems like a promising direction for further progress.

Raz and Spieker. Our proof in Section 4 is inspired by the work of Raz and Spieker [17], who used representation theory to analyze a similar pointer chasing problem. Define $\mathcal{L}$ to be the set of all permutations in $\mathcal{S}_N$ whose cycle structure consists of a single (Hamiltonian) cycle. Raz and Spieker analyze the communication complexity of deciding whether $\sigma^{-1} \circ \pi \in \mathcal{L}$, where Alice has $\pi \in \mathcal{S}_N$ and Bob has $\sigma \in \mathcal{S}_N$. Their analysis is somewhat easier than ours because $\mathcal{L}$ is a conjugacy class of $\mathcal{S}_N$ and the communication matrix is in the center of the commutant algebra of $\mathcal{S}_N$. An immediate consequence is that the communication matrix is diagonalizable.

Interestingly, their result can easily be recovered using our framework of Jucys-Murphy elements. We observe that an analog of Lemma 11 holds for their problem: $\prod_{j=2}^{N} J_j = \sum_{\pi \in \mathcal{L}} \pi$. Thus, for any $\lambda \vdash N$,

$$Y_\lambda(\mathcal{L})_{t,t} = \prod_{j=2}^{N} Y_\lambda(J_j)_{t,t}, \quad \text{for all SYT } t \text{ of shape } \lambda.$$

Thus $Y_\lambda(\mathcal{L})_{t,t} \neq 0$ iff in tableau t, every value $j \geq 2$ avoids the main diagonal. This clearly holds iff $\lambda_2 \leq 1$. Furthermore, the precise value of $Y_\lambda(\mathcal{L})_{t,t}$ can be determined using content values, as we have done in Section 4.

We remark that the work of Raz and Spieker has different motivations than our work. They compute the rank of the communication matrix in order to show that the rank lower bound can be much smaller than the non-deterministic complexity (by a $\log \log$ factor). In our case, the non-deterministic complexities are both known to be $n + o(n)$, but we show that the rank lower bound is strictly larger than the non-deterministic complexities.

Acknowledgements

Considerable thanks are due to Paul Beame, Bill Cunningham, Jim Geelen, Michel Goemans, Gordon James, Laci Lovasz, Sasha Postnikov, Ran Raz, Mike Saks, David Woodruff and Sergey Yekhanin for much discussion, technical ideas, and encouragement.

References

[1] A. Chakrabarti. Lower bounds for multi-player pointer jumping. In *Proceedings of the 23rd IEEE Conference on Computational Complexity (CCC)*, pages 33–45, 2007.

[2] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1997.

[3] W. H. Cunningham. Improved bounds for matroid partition and intersection algorithms. *SIAM Journal on Computing*, 15(4):948–957, Nov. 1986.

[4] C. Damm, S. Jukna, and J. Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Computational Complexity*, 7(2):109–127, 1998.

[5] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In R. Guy, H. Hanani, N. Sauer, and J. Schönheim, editors, *Combinatorial Structures and Their Applications*, pages 69–87. Gordon and Breach, 1970.

[6] S. Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, second edition, 2005.

[7] S. Iwata. Submodular function minimization. *Mathematical Programming*, 2007. To appear: DOI 10.1007/s10107-006-0084-2.

[8] G. James and A. Kerber. *The Representation Theory of the Symmetric Group*. Addison-Wesley, 1981.

[9] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[10] L. Lovász. Communication complexity: A survey. In B. H. Korte, editor, *Paths, Flows, and VLSI Layout*, pages 235–265. Springer Verlag, 1990.

[11] S. T. McCormick. Submodular function minimization. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Science*, pages 321–391. North-Holland, 2005.

[12] K. Mehlhorn and E. Schmidt. Las vegas is better than determinism in vlsi and distributed computing. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)*, pages 330–337, 1982.

[13] M. A. Naĭmark. *Theory of Group Representations*. Springer-Verlag, 1982.

[14] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. In *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 240–251, 2007.

[15] C. H. Papadimitriou and M. Sipser. Computational complexity. *Journal of Computer and System Sciences*, 28(2):260–269, 1984.

[16] S. Ponzio, J. Radhakrishnan, and S. Venkatesh. The communication complexity of pointer chasing. *Journal of Compter and System Sciences*, 62(2):323–355, 2001.

[17] R. Raz and B. Spieker. On the "log rank"-conjecture in communication complexity. *Combinatorica*, 15(4):567–588, 1995.

[18] R. L. Rivest and J. Vuillemin. On recognizing graph properties from adjacency matrices. *Theoretical Computer Science*, 3(3):371–384, 1976.

[19] B. E. Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Springer, second edition, 2001.

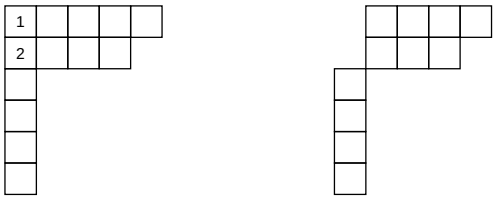
[20] A. Vershik and A. Okounkov. A new approach to the representation theory of the symmetric groups, 2. Zapiski Seminarod POMI (In Russian) v.307, 2004. arXiv:math.RT/0503040.

[21] D. J. A. Welsh. *Matroid Theory*, volume 8 of *London Mathematical Society Monographs*. Academic Press, 1976.

A Proof of Lemma 12

Let $\lambda \vdash n$ and $\mu \vdash n$ be such that $\mu_i \leq \lambda_i$ for all i. Consider the set of boxes that are contained in the Ferrers diagram of λ but not of μ. This set is called a *skew shape*, and is denoted $\lambda \setminus \mu$. The definition of a standard Young tableau generalizes to skew shapes in the obvious way.

We seek to understand g_λ, the number of SYT of shape λ in which the value 2 is in box $(2,1)$. (Note that the box $(1,1)$ contains the value 1 in any SYT.) Equivalently, g_λ equals the number of SYT of skew shape $\lambda \setminus \mu$ where μ is the partition $(1,1)$. This is illustrated in Figure 2.



(a) Shape λ (b) Shape $\lambda \setminus \mu$

Figure 2: The SYT of shape λ in which box $(2,1)$ contains element 2 correspond to SYT of shape $\lambda \setminus \mu$.

The SYT of shape $\lambda \setminus \mu$ are easily enumerated. First, one chooses the elements from $\{3, \ldots, n\}$ which will occupy the first two rows. (The remaining elements will occupy the vertical bar, i.e., the rows other than the first two.) There are $\binom{n-2}{\lambda_1+\lambda_2-2}$ ways to choose these elements. If the final arrangement is to be an SYT, then there is a single way to arrange the remaining elements in the vertical bar, i.e., increasing downwards.

It remains to enumerate the number of SYT on the first two rows. It follows from the Hook Length Formula (Fact 7) that the number of SYT of shape (a,b) is $\binom{a+b}{a} - \binom{a+b}{a+1}$.

Thus a simple manipulation shows that

$$\begin{aligned} g_\lambda &= \binom{n-2}{\lambda_1+\lambda_2-2} \cdot \left(\binom{\lambda_1+\lambda_2-2}{\lambda_1-1} - \binom{\lambda_1+\lambda_2-2}{\lambda_1} \right) \\ &= \binom{n}{\lambda_1,\, \lambda_2,\, n-\lambda_1-\lambda_2} \cdot \frac{\lambda_2\,(\lambda_1-\lambda_2+1)}{n(n-1)}. \end{aligned}$$

A similar application of the Hook Length Formula shows that

$$f_\lambda = \binom{n}{\lambda_1,\, \lambda_2,\, n-\lambda_1-\lambda_2} \cdot \frac{\lambda_2\,(\lambda_1-\lambda_2+1)}{(n-\lambda_1)\,(n-\lambda_2+1)}.$$