RIMS Kôkyûroku Bessatsu
B23 (2010), 81–105

Query Lower Bounds for Matroid Intersection

By

Nicholas J. A. Harvey *

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. Unfortunately, very little is known about lower bounds for this problem. This paper describes three lower bounds which, to our knowledge, are the best known: $2n-2$ queries are needed for rank 1 matroids, n queries are needed for rank $n-1$ matroids, and $(\log_2 3)\, n - o(n)$ queries are needed for matroids of rank $n/2$. The first two results are elementary, and the last uses methods from communication complexity and group representation theory.

§ 1. Introduction

Matroids are objects of fundamental importance in combinatorial optimization. We assume some basic familiarity with matroids; a brief summary is given in Section 2. One of the most important optimization problems relating to matroids is the *matroid intersection problem*. This paper considers the number of queries needed to solve matroid intersection in the independence oracle model. To be more specific, we consider the decision version of the problem: do two given matroids have a common base?

Let us review the known upper bounds. The best result is due to Cunningham [3]. He gives a matroid intersection algorithm using only $O(nr^{1.5})$ independence oracle queries for matroids of rank r. It would be truly remarkable if one could show that this is optimal. (For example, it might suggest that the Hopcroft-Karp algorithm [5] for bipartite matching is "morally" optimal.) Unfortunately, we are very far from being able to show anything like that: even a super-linear lower bound is not presently known.

How could one prove a super-linear lower bound on the number of queries needed to solve matroid intersection? This would require that $r = \omega(1)$, since Cunningham's

Received September 10, 2008. Revised July 27, 2010.
2000 Mathematics Subject Classification(s): 05B35, 68Q25
*Department of Combinatorics and Optimization, University of Waterloo, Canada.

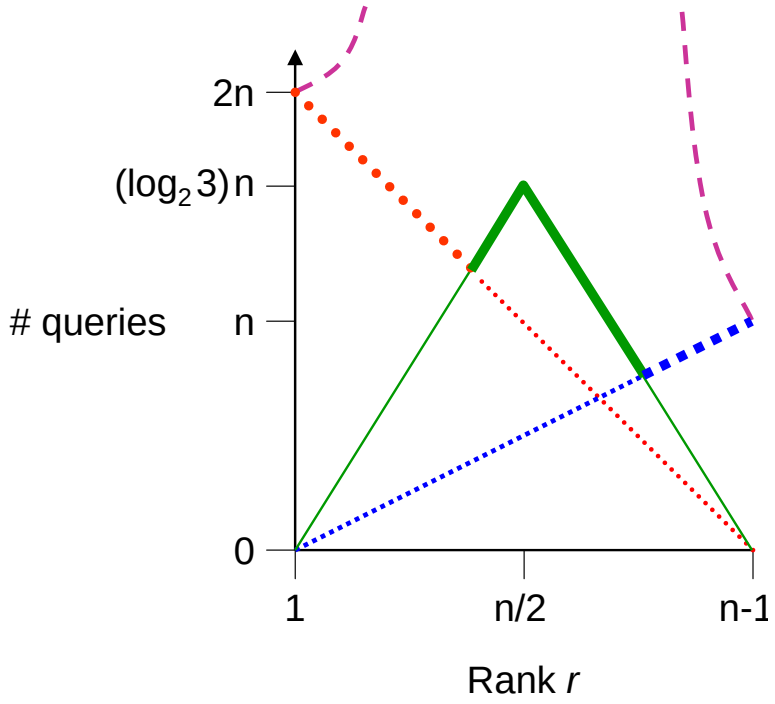


Figure 1. This chart reflects our knowledge concerning the number of independence oracle queries needed to solve matroid intersection for matroids with ground set size n and rank r. The purple, dashed lines (which are not to scale) correspond to Cunningham's upper bound of $O(nr^{1.5})$ queries, and a "dual" algorithm which is more efficient for matroids of large rank. The remaining lines correspond to lower bounds, proven in the following sections: Section 3.1 (red, round dots), Section 3.2 (blue, square dots), and Section 4 (green, solid). The best lower bound, corresponding to the upper envelope of these lines, is indicated with thick lines.

algorithm implies a bound of $O(n)$ for any constant r. One can use dual matroids to show that $n - r = \omega(1)$ is also necessary to obtain a super-linear lower bound. So the rank cannot be too large or too small. Since one can adjust the rank by padding arguments (for example, see Section 3.3 below), it suffices to prove a super-linear lower bound for $r = n/2$.

This paper describes three lower bounds on the number of queries needed, as illustrated in Figure 1. Two of these are elementary: we show in Section 3 that $2n - 2$ queries are needed for matroids of rank 1, and n queries are needed for matroids of rank $n - 1$. In Section 4, we use more involved techniques to show that $(\log_2 3) \cdot n - o(n)$ queries are necessary when $r = n/2$. The latter result is, to our knowledge, the only non-trivial progress on Welsh's question from 1976, which we paraphrase as: what is a lower bound on the number of oracle queries needed to solve matroid intersection?

§ 2. Preliminaries

We now give a brief introduction to matroids. For further discussion, we refer the reader to standard references [11, 19].

Matroids. A pair $\mathbf{M} = (V, \mathcal{I})$ is called a matroid if V is a finite set and $\mathcal{I} \subseteq 2^V$ is a non-empty family such that

- if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$, and
- if $I, J \in \mathcal{I}$ and $|J| < |I|$, then there exists an $i \in I \setminus J$ such that $J + i \in \mathcal{I}$.

The sets in $\mathcal{I}$ are called *independent* and those not in $\mathcal{I}$ are called *dependent*. A maximal independent set is called a *base* of $\mathbf{M}$. All bases have the same size, which is called the *rank* of the matroid. The *rank function* of the matroid is the function $r : 2^V \to \mathbb{N}_+$ defined by

$$r(S) := \max \{\, |I| \,:\, I \subseteq S,\, I \in \mathcal{I} \,\}.$$

It is well-known that r satisfies the following properties.

- *Normalization:* $r(\emptyset) = 0$.
- *Non-negativity:* $r(S) \geq 0$ for all $S \subseteq V$.
- *Monotonicity:* $r(S) \leq r(T)$ whenever $S \subseteq T \subseteq V$.
- *Submodularity:* $r(A) + r(B) \geq r(A \cup B) + r(A \cap B)$ for all $A, B \subseteq V$.

We adopt the following notational shorthand. For any set S and element x, we let $S + x$ denote $S \cup \{x\}$ and $S - x$ denote $S \setminus \{x\}$. It is known that the submodularity property is equivalent to

$$r(A + i) - r(A) \;\geq\; r(B + i) - r(B) \qquad \forall A \subseteq B \subseteq V \text{ and } i \notin B.$$

Associated with any matroid $\mathbf{M} = (V, \mathcal{I})$ is a unique *dual matroid*. It is defined as follows. Let $\mathcal{B}$ be the base family of $\mathbf{M}$, i.e., $\mathcal{B}$ consists of the maximal sets in $\mathcal{I}$. Define

$$\begin{aligned} \mathcal{B}^* &= \{\, V \setminus B \,:\, B \in \mathcal{B} \,\} \\ \mathcal{I}^* &= \{\, I \,:\, \exists B \in \mathcal{B}^* \text{ with } B \supseteq I \,\}. \end{aligned}$$

Then the dual matroid is $\mathbf{M}^* = (V, \mathcal{I}^*)$ and its base family is $\mathcal{B}^*$.

Optimization. Matroids are very useful objects in combinatorial optimization, and there are algorithms for efficiently solving several optimization problems relating to matroids. However, to make this precise, one must be careful to define the computational model for such algorithms. The main issue is that, on a ground set V with $|V| = n$, the number of matroids is doubly-exponential in n, and so the number of bits needed to represent a typical matroid is exponential in n. It is undesirable to use such a huge representation of matroids, so it is more common to assume an *oracle model*. An algorithm in the *independence oracle model* is given access to an oracle which, given $S \subseteq V$, can determine whether $S \in \mathcal{I}$. An algorithm in the *rank oracle model* is given access to an oracle which, given $S \subseteq V$, can compute the rank $r(S)$.

One of the most important optimization problems relating to matroids is the *matroid intersection problem*. Given two matroids $\mathbf{M}_1 = (V, \mathcal{I}_1)$ and $\mathbf{M}_2 = (V, \mathcal{I}_2)$, the problem is

$$\max \{\, |I| \,:\, I \in \mathcal{I}_1 \cap \mathcal{I}_2 \,\}.$$

In computational complexity, it is often more convenient to consider decision problems, rather than optimization problems. We will consider the following decision form of matroid intersection. We are given two matroids $\mathbf{M}_1 = (V, \mathcal{I}_1)$ and $\mathbf{M}_2 = (V, \mathcal{I}_2)$, whose respective base families are $\mathcal{B}_1$ and $\mathcal{B}_2$. The problem is to decide whether $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$.

§ 3. Elementary lower bounds

§ 3.1. Adversary argument for rank-1 matroids

We begin with some easy observations using matroids of rank one. Let S be a finite ground set with $|S| = n$. Let $\emptyset \neq X \subseteq S$ 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, which we denote $\mathbf{M}(X)$. Let $\mathcal{M} = \{\, \mathbf{M}(X) \,:\, \emptyset \neq X \subseteq S \,\}$. Given two sets $S_0, S_1 \subseteq S$, the two matroids $\mathbf{M}(S_0)$ and $\mathbf{M}(S_1)$ have a common base iff $S_0 \cap S_1 \neq \emptyset$.

We show the following simple theorem.

Theorem 3.1. *Any deterministic algorithm that performs fewer than $2n - 2$ queries cannot solve the matroid intersection problem when given two matroids in $\mathcal{M}$.*

We will prove this theorem in a rather pedantic manner, since the following section requires a similar proof for a slightly less obvious result. Let us first introduce some terminology. Let $Y_i \subseteq S$ be the set of "yes" elements y for which we have decided $\{y\} \in \mathcal{B}(S_i)$. Similarly, let $N_i \subseteq S$ be the set of "no" elements y for which we have decided $\{y\} \notin \mathcal{B}(S_i)$. Let us define the following predicates concerning the adversary's responses to the queries.

Consistent $\forall\, i \in \{0,1\}\,,\ Y_i \cap N_i = \emptyset$

No-Extensible $Y_0 \cap Y_1 = \emptyset$

Yes-Extensible $N_0 \cup N_1 \neq S$

Intuitively, the responses are Consistent if they are valid responses corresponding to some matroid. They are No-Extensible if there exist matroids $\mathbf{M}(S_0)$ and $\mathbf{M}(S_1)$ that do not have a common base and are consistent with the query responses given so far. Yes-Extensible is analogous.

Proof. If $n = 1$ there is nothing to prove, so assume $n \geq 2$. To prove the theorem, we will describe an adversary which replies to the queries of the algorithm and ensures that the responses are Consistent, No-Extensible and Yes-Extensible. The adversary initially adds distinct elements to Y_0 and Y_1, thereby ensuring that $|Y_0| = |Y_1| = 1$ and hence the two matroids do not have rank 0. Let q denote the number of queries performed so far. The adversary maintains two additional properties:

Property 1 $|Y_0 \cup Y_1| + |N_0 \cup N_1| \leq q + 2$

Property 2 $N_i \subseteq Y_{1-i}$

The adversary behaves roughly as follows. The first time a singleton set $\{a\}$ is queried, it returns Yes. Whenever $\{a\}$ is subsequently queried in the other matroid, it returns No. A more formal description is given in the following pseudocode.

Algorithm 1 Adversarial responses to the independence oracle queries. The adversary decides whether $A \in \mathcal{I}_i$.

```
QUERY(i, A)
    If |A| = 0, return Yes
    If |A| > 1, return No
    Let a be the unique element in A
    If a ∈ Y_{1−i}, add a to N_i and return No
    Add a to Y_i and return Yes
```

Let us check the correctness of this adversary. First of all, the empty set is independent in every matroid so if $|A| = 0$ then the adversary must return Yes. The adversary is behaving as a rank one matroid, so every independent set has size at most one. So if $|A| > 1$ then the adversary must return No.

So let us suppose that $A = \{a\}$ and $a \in Y_{1-i}$. The No-Extensible property implies $a \notin Y_i$. So adding a to N_i does not violate the Consistent property. Both Y_0 and Y_1 are unchanged so the No-Extensible property is preserved. The algorithm adds a only to N_i so property 1 is preserved. Since $a \in Y_{1-i}$, property 2 is preserved. We now claim that

the Yes-Extensible property is maintained, so long as $q < 2n - 2$. Combining property 1 and 2, we get

$$2 \cdot |N_0 \cup N_1| \;\leq\; |Y_0 \cup Y_1| + |N_0 \cup N_1| \;\leq\; q + 2$$

and hence

$$|N_0 \cup N_1| \leq (q+2)/2 < n.$$

Thus $N_0 \cup N_1 \neq S$, so the responses are Yes-Extensible.

Similar arguments establish correctness for the case $a \notin Y_{1-i}$. Since the adversary's responses are both No-Extensible and Yes-Extensible, the algorithm cannot have decided whether the two matroids have a common base. □

The lower bound presented above is essentially tight.

Proposition 3.2. *There exists a deterministic algorithm using only $2n$ queries that decides the matroid intersection problem for matroids in $\mathcal{M}$.*

Proof. For every $s \in S$, decide whether $\{s\} \in \mathcal{B}(S_1)$ and $\{s\} \in \mathcal{B}(S_2)$. This takes $2n$ queries, and the algorithm completely learns the set S_1 and S_2. Deciding whether they are disjoint is now trivial. □

§ 3.2. Adversary argument for large-rank matroids

For any $\emptyset \neq X \subseteq S$, let $\mathcal{B}^*(X) = \{\, S - x : x \in X \,\}$, let $\mathbf{M}^*(X) = (S, \mathcal{B}^*(X))$, and let $\mathcal{M}^* = \{\, \mathbf{M}^*(X) : \emptyset \neq X \subseteq S \,\}$. (Here $\mathbf{M}^*(X)$ is the dual matroid for $\mathbf{M}(X)$.) The matroids in $\mathcal{M}^*$ all have rank $n - 1$. As above, $\mathbf{M}^*(S_0)$ and $\mathbf{M}^*(S_1)$ have a common base iff $S_0 \cap S_1 \neq \emptyset$. These matroids satisfy the following useful property.

Proposition 3.3. *Let $Z \subseteq S$. Then $S \setminus Z$ is an independent set in $\mathbf{M}^*(X)$ iff $X \cap Z \neq \emptyset$.*

Proof. Suppose that $z \in X \cap Z$, so $S - z \in \mathcal{B}^*(X)$. Then $S \setminus Z$ is independent, since $S - Z \subseteq S - z$. Conversely, suppose that $S \setminus Z$ is independent. Then there exists some set $S - z \in \mathcal{B}^*(X)$ with $S \setminus Z \subseteq S - z$. Thus $z \in X$ and $z \in Z$, as required. □

Theorem 3.4. *Let $n = |S| \geq 2$. Any deterministic algorithm that performs fewer than n queries cannot solve the matroid intersection problem when given two matroids in $\mathcal{M}^*$.*

As above, let $Y_i \subseteq S$ be the set of elements y for which we have decided that $S - y \in \mathcal{B}^*(S_i)$. And let $N_i \subseteq S$ be the set of elements y for which we have decided that $S - y \notin \mathcal{B}^*(S_i)$. The predicates are again:

Consistent $\forall\, i \in \{0,1\}\,,\ Y_i \cap N_i = \emptyset$

No-Extensible $Y_0 \cap Y_1 = \emptyset$

Yes-Extensible $N_0 \cup N_1 \neq S$

Proof. Let $q < n$ be the number of queries performed so far. The adversary also maintains two properties:

Property 1 $|Y_0 \cup Y_1| \leq q$

Property 2 $N_i \subseteq Y_{1-i}$

The adversary's behavior is described in the following pseudocode.

Algorithm 2 Adversarial responses to the independence oracle queries.

```
Query(i, A): Decide if S \ A ∈ I_i
    If A ∩ Y_i ≠ ∅, return Yes
    If A ⊈ Y_{1−i}
        Pick a ∈ A \ Y_{1−i}, and add a to Y_i
        Return Yes
    Set N_i ← N_i ∪ A
    Return No
```

Let us check that the stated properties are maintained by this algorithm.

Case 1: $A \cap Y_i \neq \emptyset$. Then, by Proposition 3.3, $S \setminus A \in \mathcal{I}_i$ as required. The sets Y_j and N_j are not affected, so all properties are maintained.

Case 2: $A \cap Y_i = \emptyset$ and $A \not\subseteq Y_{1-i}$. In this case, we add a to Y_i. We have $a \notin Y_{1-i}$ so the responses are No-Extensible. Furthermore, $a \notin N_i$ by property 2, and thus the responses are Consistent. $|Y_0 \cup Y_i|$ increases by at most 1 so Property 1 holds. Property 2 and the Yes-Extendibility are trivial.

Case 3: $A \cap Y_i = \emptyset$ and $A \subseteq Y_{1-i}$. In this case, we add A to N_i. It is easy to verify that Consistency, No-Extendibility, Property 1 and Property 2 are all maintained. Let us consider Yes-Extendibility. By Properties 1 and 2,

$$|N_0 \cup N_1| \ \leq\ |Y_0 \cup Y_1| \ \leq\ q.$$

So if $q < n$ then the responses are Yes-Extensible.

Since the responses are both No-Extensible and Yes-Extensible, the algorithm cannot have decided whether the two matroids have a common base. □

The lower bound presented above is essentially tight.

Proposition 3.5. *There exists a deterministic algorithm using only $n+1$ queries that decides the matroid intersection problem for matroids in $\mathcal{M}^*$.*

Proof. For every $s \in S$, decide whether $S - s \in \mathcal{B}^*(S_1)$. In this way, the algorithm completely learns the set S_1. It must decide whether $S_0 \cap S_1 = \emptyset$. By Proposition 3.3, this can be decided by testing whether $S \setminus S_1 \in \mathcal{I}(S_0)$. □

§ 3.3. A padding argument

We now build on the previous two sections and give a lower bound for matroids of any rank via a padding argument.

First we start by padding the matroids from Section 3.1. For any $r \geq 1$, let P be an arbitrary set such that $|P| = r-1$ and $S \cap P = \emptyset$. Let $m = |S|$ and $n = |S \cup P| = m+r-1$. For any $\emptyset \neq X \subseteq S$, we define the matroid $\mathbf{M}_r(X)$ as follows: it has ground set $S \cup P$ and base family $\mathcal{B}_r(X) = \{ P + x : x \in X \}$. (In matroid terminology, $\mathbf{M}_r(X)$ is obtained from $\mathbf{M}(X)$ by adding the elements in P as coloops.) This family of matroids is denoted $\mathcal{M}_r = \{ \mathbf{M}_r(X) : \emptyset \neq X \subseteq S \}$. Clearly $\mathbf{M}_r(X)$ and $\mathbf{M}_r(Y)$ have a common base if and only if $\mathbf{M}(X)$ and $\mathbf{M}(Y)$ do. Thus, the number of queries needed to solve matroid intersection for matroids in $\mathcal{M}_r$ is at least $2m - 2 = 2(n - r)$, by Theorem 3.1.

Now we consider the matroids from Section 3.2. Let r satisfy $0 < r < n$. Let P and S be disjoint sets with $|P| = n - r - 1$ and $|S| = r + 1$, so $|S \cup P| = n$. For any $\emptyset \neq X \subseteq S$, we define the matroid $\mathbf{M}^*_r(X)$ as follows: it has ground set $S \cup P$ and base family $\mathcal{B}^*_r(X) = \{ S - x : x \in X \}$. (In matroid terminology, the matroid $\mathbf{M}^*_r(X)$ is obtained from $\mathbf{M}^*(X)$ by adding the elements in P as loops.) This family of matroids is denoted $\mathcal{M}^*_r = \{ \mathbf{M}^*_r(X) : \emptyset \neq X \subseteq S \}$. Clearly $\mathbf{M}^*_r(X)$ and $\mathbf{M}^*_r(Y)$ have a common base if and only if $\mathbf{M}^*(X)$ and $\mathbf{M}^*(Y)$ do. Thus, the number of queries needed to solve matroid intersection for matroids in $\mathcal{M}_r$ is at least $r + 1$, by Theorem 3.4.

We summarize this discussion with the following theorem.

Theorem 3.6. *The number of independence oracle queries needed by any deterministic algorithm that solves matroid intersection for matroids with ground set size $n \geq 2$ and rank $0 < r < n$ is at least* $\max\{2(n - r), r + 1\}$.

§ 4. An algebraic lower bound

This section improves on Theorem 3.6 by showing an improved lower bound for matroids of rank close to $n/2$. Formally, we show the following theorem.

Theorem 4.1. *The number of independence oracle queries needed by any deterministic algorithm that solves matroid intersection for matroids with even ground set size n and rank $n/2 + 1$ is at least* $(\log_2 3)\, n - o(n)$.

Thus by combining Theorem 3.6 and Theorem 4.1 and using padding arguments, we obtain the following result, which justifies Figure 1.

Corollary 4.2. *The number of independence oracle queries needed by any deterministic algorithm that solves matroid intersection is lower bounded as follows. Suppose the algorithm is given two matroids with ground set size $n \geq 2$ and rank $0 < r < n$, with $\tilde{r} = \min\{r, n-r\}$. Then the lower bound is* $\max\{\, 2(n-r),\ r+1,\ (\log_2 9)\tilde{r} - o(\tilde{r}) \,\}$.

Proof. We consider the third term. Let $\mathcal{M}$ be the family of matroids for which the lower bound of Theorem 4.1 is proven, where we choose their ground set to be S, with $|S| = 2r-2$. Add $n-2r+2$ loops to the matroids in $\mathcal{M}$; the resulting matroids have ground set size n and rank $|S|/2+1 = r$. Then we have $\tilde{r} \geq r-2$ and, by Theorem 4.1, the lower bound on the required number of queries is

$$(\log_2 3)(2r-2) - o(r) \;=\; (\log_2 9)\tilde{r} - o(\tilde{r}).$$

If we had added $n-2r+2$ coloops instead of loops, the resulting matroids would have ground set size n and rank $\big(|S|/2+1\big) + \big(n-2r+2\big) = n-r+2$. Then we have $\tilde{r} = r-2$ and the lower bound is again

$$(\log_2 3)(2r-2) - o(r) \;=\; (\log_2 9)\tilde{r} - o(\tilde{r}).$$

This completes the proof. □

The remainder of this section describes the proof of Theorem 4.1. A high-level overview is as follows. We describe a family of matroids that correspond to 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. The difficulty of deciding this property is analyzed using the *communication complexity* framework, which we introduce next. Roughly speaking, 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 4.5 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.

§ 4.1. Communication complexity

Our lower bound uses methods from the field of communication complexity. The basics of this field are covered in the survey of Lovász [9], and further details can be found in the book of Kushilevitz and Nisan [8]. This section briefly describes the concepts that we will need.

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 total bit-length of the messages sent 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 single 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 1 and 0. The *co-nondeterministic complexity* is also defined analogously, and is denoted by $N^0(f)$.

Fact 4.3. $N^0(f) \leq D(f)$ and $N^1(f) \leq D(f)$.

Proof. See [8, §2.1]. Consider any deterministic communication protocol for f. Since the prover has both Alice's and Bob's inputs, it can produce a certificate containing the sequence of messages that would have been exchanged by this protocol on the given inputs. Alice and Bob can therefore use this certificate to simulate execution of the protocol, without exchanging any messages. Therefore this certificate acts both as a nondeterministic and co-nondeterministic proof. □

Fact 4.4. For any communication problem f, we have $D(f) = O(N^0(f) \cdot N^1(f))$.

Proof. See [8, p20] or [9, p244]. □

For any communication problem f, the *communication matrix* is a matrix $C(f)$, or simply C, whose entries are in $\{0, 1\}$, whose rows are indexed by Alice's inputs X and whose columns are indexed by Bob's inputs Y. The entries of C are $C(f)_{X,Y} = f(X, Y)$. 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 4.5 (Mehlhorn and Schmidt [10])**.** Over any field (including the complex numbers), we have $D(f) \geq \log_2 \operatorname{rank} C(f)$.

Proof. See [8, p13]. □

§ 4.2. Communication complexity of matroid intersection

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

Definition 4.6. The communication problem MatInt:

- *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 $\textsc{MatInt}(\mathbf{M}_1, \mathbf{M}_2) = 1$. Otherwise, it is 0.

Fact 4.7. $D(\textsc{MatInt})$ gives a lower bound on the number of oracle queries made by any deterministic matroid intersection algorithm.

Proof. See [8, Lemma 9.2]. The proof is a simulation argument: any deterministic matroid intersection algorithm which uses q independence oracle queries can be transformed into a deterministic communication protocol for MatInt that uses q bits of communication. Both Alice and Bob can independently simulate the given algorithm, and they only need to communicate whenever an oracle query is made, so the number of bits of communication is exactly q. □

The remainder of this section focuses on analyzing the communication complexities of MatInt. Some easy observations can be made using matroids of rank one, as defined

in Section 3.1. Recall that for two matroids $\mathbf{M}(X), \mathbf{M}(Y) \in \mathcal{M}$, they have a common base iff $X \cap Y \neq \emptyset$. Thus, for the family $\mathcal{M}$, the MATINT problem is simply the complement of the well-known *disjointness* problem (denoted DISJ) [8]. It is known that $D(\text{DISJ}) \geq n$ and $N^1(\text{DISJ}) \geq n - o(n)$. Although we will not discuss randomized complexity in any detail, it is also known [16] that the randomized communication complexity of DISJ is $\Omega(n)$, and consequently the same is true of MATINT.

Thus we have shown that $D(\text{MATINT}) \geq n$ and $N^0(\text{MATINT}) \geq n - o(n)$. In Section 5, we will also show that $N^1(\text{MATINT}) = \Omega(n)$. As it turns out, these lower bounds for N^0 and N^1 are essentially tight. To show this, we will use Edmonds' matroid intersection theorem.

Fact 4.8 (Matroid Intersection Theorem)**.** Let $\mathbf{M}_1 = (S, \mathcal{I}_1, r_1)$ and $\mathbf{M}_2 = (S, \mathcal{I}_2, r_2)$ be given. Then

$$\max_{I \in \mathcal{I}_1 \cap \mathcal{I}_2} |I| \;=\; \min_{A \subseteq S} \big(r_1(A) + r_2(S \setminus A) \big).$$

Lemma 4.9. *$N^1(\text{MATINT}) \leq n$ and $N^0(\text{MATINT}) \leq n + \lfloor \log n \rfloor + 1$.*

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 $N^1(\text{MATINT}) \leq n$.

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 4.8. The co-nondeterministic certificate Z consists of the set A and an integer z. Alice checks that $z = r_1(A)$. Bob checks that $z + r_2(S \setminus A) < r$. If this holds then the two matroids cannot have a common base. The length of this certificate is at most $n + \lfloor \log n \rfloor + 1$. □

Lemma 4.9 is an unfortunate obstacle in our quest to prove a super-linear lower bound on $D(\text{MATINT})$. The fact that both the nondeterministic and co-nondeterministic communication complexities are $O(n)$ makes our task more difficult, for two reasons. First, we must use techniques that can separate the deterministic complexity from the nondeterministic complexities: we need a super-linear lower bound for $D(\text{MATINT})$ which does not imply that either $N^0(\text{MATINT})$ or $N^1(\text{MATINT})$ is super-linear (since this is false!). Second, the nondeterministic and co-nondeterministic communication complexities provably constrain the quality of any lower bound on the deterministic complexity, as shown in Fact 4.4. Thus, the communication complexity technique cannot prove a super-quadratic lower bound for the matroid intersection problem; at least, not in the present formulation.

§ 4.3. The In-Same-Cycle problem

One interesting category of communication problems is pointer chasing problems [2, 4, 12, 13, 15]. 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. One can argue 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). The main idea of the argument is that odd-length paths have a perfect matching whereas even-length paths do not.

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 $\left\{ \left\{u_i, v_{\pi(i)}\right\} : i = 1, \ldots, N \right\}$, 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

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

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

Lemma 4.10. *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; we omit a formal argument. Let us now interpret Lemma 4.10 in the communication complexity framework.

Definition 4.11. The In-Same-Cycle, or ISC, 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 $\mathrm{ISC}(\pi, \sigma) = 1$. Otherwise it is 0.

We will show hardness for MATINT by analyzing ISC. First, Lemma 4.10 shows that ISC reduces to MATINT. Next, we will argue that ISC is a "hard" problem. Intuitively, it seems that Alice and Bob cannot decide the ISC 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 section proves the following theorem.

Theorem 4.12. *Let C denote the communication matrix for* ISC*. Then* $\operatorname{rank} C$ *equals*

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

Corollary 4.13. *$D(\mathrm{ISC}) \ge (\log_2 9)N - o(N)$. Consequently, any deterministic algorithm solving the matroid intersection problem for matroids with rank $n/2+1$ and ground set size n must use at least $(\log_2 3)n - o(n)$ queries.*

Proof. Stirling's approximation shows that

$$e\left(\frac{n}{e}\right)^n \;<\; n! \;<\; e\,n\left(\frac{n}{e}\right)^n.$$

Thus,

$$\binom{N}{N/3,\, N/3,\, N/3} \;=\; \frac{N!}{\big((N/3)!\big)^3} \;\ge\; \frac{e\,(N/e)^N}{\big(e\,(N/3)\,(N/3e)^{N/3}\big)^3} \;=\; 3^{N-o(N)}.$$

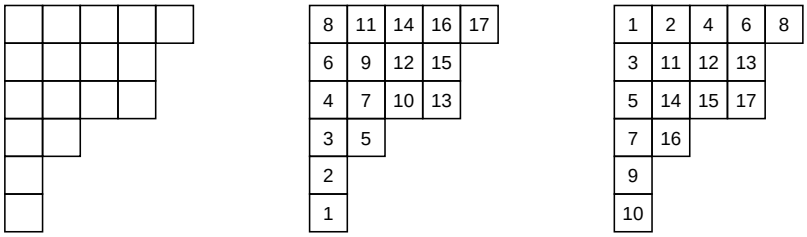
In Theorem 4.12, considering just the term $i = j = N/3$ shows that $\operatorname{rank} C \ge 9^{N-o(N)}$. Fact 4.5 therefore implies the lower bound on $D(\mathrm{ISC})$. The lower bound for matroid intersection follows since the matroids in $\mathcal{M}_k$ have rank $n/2+1 = N+1$ and ground set size n. □

This corollary establishes Theorem 4.1.

§ 4.4. Group theory

The proof of Theorem 4.12 relies on several notions from the theory of the symmetric group. We review the necessary notions in this section. We recommend James-Kerber [6] and Sagan [18] for a more detailed exposition of this material.

Let $\mathcal{S}_N$ be the group of all permutations on $[N]$, i.e., bijections from $[N]$ to $[N]$ under the operation of function composition. Let $\pi \in \mathcal{S}_N$. The *cycle type* of π is a sequence of integers in which the number of occurrences of k equals the number of distinct cycles of length k in π. Without loss of generality, we may assume that this



(a) (b) (c)

Figure 2. (a) The Ferrers diagram for the partition $(5, 4, 4, 2, 1, 1) \vdash 17$. (b) A Young tableau. (c) A standard Young tableau.

sequence is in non-increasing order. Thus, the cycle type of π is a *partition* of N, which is defined as a non-increasing sequence $\lambda = (\lambda_1, \ldots, \lambda_\ell)$ of positive integers such that $N = \sum_{i=1}^{\ell} \lambda_i$. The value ℓ is called the *length* of λ, and it is also denoted $\ell(\lambda)$. The notation $\lambda \vdash N$ denotes that the sequence λ is a partition of N.

Let $\mathcal{C}(\lambda) \subseteq \mathcal{S}_N$ be the set of all permutations with cycle type $\lambda \vdash N$. The set $\mathcal{C}(\lambda)$ is a conjugacy class of $\mathcal{S}_N$. Moreover, every conjugacy class of $\mathcal{S}_N$ is obtained in this way, so the number of conjugacy classes of $\mathcal{S}_N$ equals the number of partitions of N. Thus, the non-isomorphic irreducible matrix representations (henceforth, *irreps*) of $\mathcal{S}_N$ can be indexed by the partitions of N. The irreps of $\mathcal{S}_N$ will be denoted ρ_λ where $\lambda \vdash N$.

A *Ferrers diagram* of $\lambda \vdash N$ is a left-aligned array of boxes in the plane for which the i^{th} row contains λ_i boxes. An example is shown in Figure 2 (a). A *Young tableau* of shape λ is a bijective assignment of the integers in $[N]$ to the boxes of the Ferrers diagram for λ. An example is shown in Figure 2 (b). A *standard Young tableau*, or *SYT*, is one in which

- for each row, the values in the boxes increase from left to right, and
- for each column, the values in the boxes increase from top to bottom.

An example is shown in Figure 2 (c).

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 3.

Fact 4.14 (Hook Length Formula)**.** The number of SYT of shape λ is denoted f_λ, and has value

$$f_\lambda \;=\; \frac{N!}{\prod_v |h_v|},$$

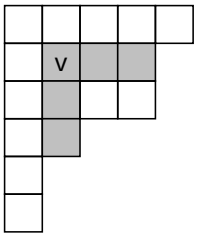


Figure 3. A box v and its hook h_v.

0	1	2	3	4
-1	0	1	2	
-2	-1	0	1	
-3	-2			
-4				
-5				

Figure 4. The "content" of all boxes in this Ferrers diagram.

where the product is over all boxes in the Ferrers diagram for λ.

Fact 4.15. The dimension of irrep ρ_λ equals f_λ, the number of SYT of shape λ. Thus Fact 4.14 provides a formula for the dimension of ρ_λ.

There exist several canonical ways of defining the irrep associated to partition λ, since a change of basis produces an isomorphic representation. In this paper, we will fix *Young's seminormal basis* [6] as the specific basis in which each irrep is presented. The formal definition of this basis is not crucial for us, but we will need some of its properties.

First, we introduce some notation. Let Y_λ denote the irrep corresponding to partition λ in Young's seminormal basis. 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 $Y_\lambda(S) = \sum_{\pi \in S} Y_\lambda(\pi)$.

For $1 \leq j \leq N$, the j^{th} *Jucys-Murphy element* is the member of the group algebra defined by $J_j = \sum_{1 \leq i < j} (i,j)$. (Here, (i,j) denotes a transposition in $\mathcal{S}_N$.) For convenience, we may also view J_j as a subset of $\mathcal{S}_N$, namely 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., the box in row a and column b) is the integer $b-a$. This is illustrated in Figure 4; note that the content values are constant on each negative-sloping diagonal. For any standard Young tableau t and $1 \leq j \leq N$, define $\text{cont}(t,j)$ to be the content of the box occupied by element j in tableau t.

Fact 4.16. $Y_\lambda(J_j)$ 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.

A proof of this fact can be found in the book of James and Kerber [6].

§ 4.5. Analysis of In-Same-Cycle

In this section, we describe a method for computing the rank of the communication matrix for the ISC problem. This proves Theorem 4.12 (up to some omitted details). It turns out that the communication matrix is diagonalizable, and that the values of those diagonal entries (i.e., the spectrum) are integers that can be precisely computed.

Overview of Proof. Our argument proceeds as follows.

- **Step 1.** The matrix C can be written as a sum of matrices in the regular representation.
- **Step 2.** 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.
- **Step 3.** 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).
- **Step 4.** 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 expression for the rank can be given.

Step 1. 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

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

Note that $\mathcal{K}$ is closed under taking inverses: $\pi \in \mathcal{K} \implies \pi^{-1} \in \mathcal{K}$. Recall the definition of the communication matrix C:

$$C_{\pi,\sigma} = \begin{cases} 1 & \text{if } \sigma^{-1} \circ \pi \in \mathcal{K}, \\ 0 & \text{otherwise.} \end{cases}$$

This leads to the following easy lemma.

Lemma 4.17. *$C = \sum_{\tau \in \mathcal{K}} R(\tau)$, where $R(\tau)$ denotes a matrix of the regular representation.*

Proof. Let $\rho = \sigma^{-1} \circ \pi$, implying that $\pi = \sigma \circ \rho$. Clearly ρ is the unique permutation with this property. Thus $R(\tau)_{\pi,\sigma} = 1$ iff $\tau = \rho$. Thus, the entry in row π and column σ of $\sum_{\tau \in \mathcal{K}} R(\tau)$ is 1 if $\rho \in \mathcal{K}$ and 0 otherwise. This matches the definition of C. □

Step 2. By Maschke's theorem, there exists a change-of-basis matrix B which decomposes the regular representation into irreps. (We can choose B such that each irrep in the decomposition is a matrix representation in Young's seminormal basis.) We will analyze the rank of C by considering the contribution from each irrep. We have

$$\text{(4.1)} \quad \operatorname{rank} C \;=\; \operatorname{rank} B\,C\,B^{-1} \;=\; \operatorname{rank}\Big(\sum_{\tau \in \mathcal{K}} B\,R(\tau)\,B^{-1}\Big) \;=\; \sum_{\lambda \vdash N} f_\lambda \cdot \operatorname{rank} Y_\lambda(\mathcal{K}),$$

where the second equality follows from Lemma 4.17. To see the third equality, recall that each $B\,R(\tau)\,B^{-1}$ is decomposed into blocks of the form $Y_\lambda(\tau)$, so each block of BCB^{-1} is of the form $Y_\lambda(\mathcal{K})$. Furthermore, each irrep λ appears f_λ times.

Step 3. The following lemma gives the reason that the communication matrix for ISC can be analyzed so precisely. It gives a direct connection between the ISC problem and the Jucys-Murphy elements.

Lemma 4.18. *$\sum_{\pi \in \mathcal{K}} \pi \;=\; J_2 \cdot \prod_{j=3}^{N} (1 + J_j)$, where 1 denotes the identity permutation.*

Proof. The proof is by induction on N, the trivial case being $N = 2$. So let $N > 2$. For any $\pi \in \mathcal{K}_{N-1}$ and transposition (i, N), we have $\pi \circ (i, N) \in \mathcal{K}_N$. Conversely, for any $\pi \in \mathcal{K}_N$, there is a unique way to obtain π as a product of $\pi' \in \mathcal{K}_{N-1}$ and a transposition (i, N), by taking $i = \pi^{-1}(N)$ and $\pi' = \pi \circ (i, N)$ (restricted to $\mathcal{S}_{N-1}$). □

Here is a simple, but interesting, corollary of this lemma.

Corollary 4.19. *$|\mathcal{K}_N| = |\mathcal{S}_N|/2$. In other words, for any π,*

$$\Pr_\sigma\left[\,\mathrm{ISC}(\pi, \sigma) = 1\,\right] \;=\; 1/2,$$

where σ is chosen uniformly from $\mathcal{S}_N$.

Proof. Viewing the Jucys-Murphy elements as sets, we have $|J_i| = i - 1$. Since the permutations arising in the product $J_2 \cdot \prod_{j=3}^{N} (1 + J_j)$ are distinct, we have $|\mathcal{K}_N| = 1 \cdot \prod_{j=3}^{N} j = N!/2$. □

Lemma 4.18 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 $\{ Y_\lambda(J_j) : 2 \leq j \leq N \}$. It follows directly from Fact 4.16 that $Y_\lambda(\mathcal{K})$ is diagonal. Furthermore, we can determine the diagonal entries explicitly. 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 Eq. (4.2) completely determines the spectrum of C, since the values $Y_\lambda(J_j)_{t,t}$ are known (see Fact 4.16).

Step 4. In the remainder of this section, we will analyze Eq. (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 4.16, $Y_\lambda(J_2)_{t,t} = 0$ 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,

$$Y_\lambda(\mathcal{K})_{t,t} \neq 0 \iff \text{in tableau } t\text{, all values } j \geq 3 \text{ avoid the first subdiagonal.} \tag{4.3}$$

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, Eq. (4.3) shows that $Y_\lambda(\mathcal{K})_{t,t} = 0$.

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 Eq. (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 $Y_\lambda(\mathcal{K})_{t,t} = 0$ by Eq. (4.3).

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

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

We have now computed the entire spectrum of C. The remaining task is to compute the rank (i.e., count 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 $Y_\lambda(\mathcal{K})$ 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 (by Fact 4.15). Thus,

$$\operatorname{rank} C = 1 + \sum_{\substack{\lambda \text{ s.t.} \\ \lambda_2 \geq 1 \text{ and } \lambda_3 \leq 1}} f_\lambda \cdot g_\lambda. \tag{4.4}$$

The value of this expression is obtained by the following lemma.

Lemma 4.20. *Let $\lambda \vdash N$ satisfy $\lambda_2 \geq 1$ and $\lambda_3 \leq 1$. Then*

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

Substituting into Eq. (4.4) 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 4.12.

§ 5. Paving matroids

In this section, we introduce "one-alternation" matroid intersection algorithms, which first $\mathbf{M}_1$, then query $\mathbf{M}_2$, but do not again query $\mathbf{M}_1$. We show that any such algorithm requires $2^{n-o(n)}$ queries to solve matroid intersection. This implies another linear lower bound for ordinary matroid intersection algorithms.

Our arguments are based on the use of paving matroids, which we now introduce. To do so, we first describe another operation on matroids which we call *base-removal*.

Lemma 5.1. *Let $\mathbf{M} = (S, \mathcal{B})$ be a matroid. Let $B \in \mathcal{B}$ be a base such that, for all $A \subseteq S$ with $|A| = |B|$ and $|A \oplus B| = 2$, we have $A \in \mathcal{B}$. Then $(S, \mathcal{B} - B)$ is also a matroid.*

Proof. Let r be the rank function of $\mathbf{M}$. Define the function $\tilde{r} : S \to \mathbb{N}$ as follows.

$$\tilde{r}(A) \;=\; \begin{cases} r(A) - 1 & \text{if } A = B \\ r(A) & \text{otherwise} \end{cases}$$

We now claim that $\tilde{r}$ is the rank function of the matroid $(S, \mathcal{B} - B)$. To show this, it suffices to show that it is submodular, i.e., satisfies

$$\tilde{r}(A) + \tilde{r}(B) \;\geq\; \tilde{r}(A \cup B) + \tilde{r}(A \cap B) \qquad \forall\ A, B \subseteq S.$$

It is known [22] [19, Theorem 44.1] that this is equivalent to

$$\tilde{r}(A + a) + \tilde{r}(A + b) \;\geq\; \tilde{r}(A \cup \{a, b\}) + \tilde{r}(A) \qquad \forall A \subseteq S \;\;\&\;\; \forall a, b \in S \setminus A.$$

Since $\tilde{r}$ differs from r only in that $\tilde{r}(B) = r(B) - 1$, it suffices to verify whether

$$r(B) + r(B - j + i) \;\stackrel{?}{\geq}\; r(B + i) + r(B - j) \qquad \forall j \in B, i \in S \setminus B.$$

We have

$$\tilde{r}(B) + \tilde{r}(B - j + i) \;=\; (|B| - 1) + |B|,$$

by definition of $\tilde{r}$ and since $|B \oplus (B - j + i)| = 2$. Also,

$$\tilde{r}(B + i) + \tilde{r}(B - j) = |B| + (|B| - 1)$$

since B is a base. Thus the desired inequality is satisfied (with equality). □

Now let S be a ground set of cardinality n, where n is even. Let $\mathbf{M} = (S, \mathcal{B})$ be the uniform matroid of rank $n/2$. Let $\mathcal{C}^* \subseteq 2^S$ be a code of minimum distance 4 for which all codewords have weight $n/2$. That is, $\mathcal{C}^* \subset \mathcal{B}$, and for all $A, B \in \mathcal{C}$ we have $|A \oplus B| \geq 4$. A greedy code construction shows that we may take

$$|\mathcal{C}^*| \;\geq\; \binom{n}{n/2}/n^2 \;=\; 2^{n - o(n)}.$$

For any subcode $\mathcal{C} \subseteq \mathcal{C}^*$, we obtain a new matroid by applying base-removal $\mathbf{M}$ at every set $C \in \mathcal{C}$. Formally, we define $\mathbf{P}_{\mathcal{C}} = (S, \mathcal{B} \setminus \mathcal{C})$. Lemma 5.1 shows that $\mathbf{P}_{\mathcal{C}}$ is indeed a matroid. Such matroids are a type of *paving matroids* [7] [21, §16.6].

Suppose that Alice is given a matroid $\mathbf{P}_{\mathcal{C}}$ where $\mathcal{C} \subseteq \mathcal{C}^*$ and Bob is given a matroid $\mathbf{M}_B = (S, \{B\})$ where $B \in \mathcal{C}^*$. It is clear that $\mathbf{P}_{\mathcal{C}}$ and $\mathbf{M}_B$ have a common base iff

$B \notin \mathcal{C}$. This shows a connection to the INDEX problem in communication complexity, in which Alice is given a vector $x \in \{0,1\}^m$ and Bob is given an index $i \in [m]$. Their task is to compute the value x_i. The INDEX problem reduces to matroid intersection in the following way. First, we identify $\mathcal{C}^*$ with $[m]$. Alice, given x, constructs the corresponding subset $\mathcal{C} \subseteq \mathcal{C}^*$, and the matroid $\mathbf{P}_{\mathcal{C}}$. Bob, given i, Bob constructs the corresponding set $B \in \mathcal{C}^*$ and $\mathbf{M}_B$. We have $x_i = 1$ precisely when $\mathbf{P}_{\mathcal{C}}$ and $\mathbf{M}_B$ have a common base.

This reduction implies a few results. The first result relates to *one-round communication protocols*, in which Alice can send messages to Bob, but Bob cannot reply to Alice. These protocols correspond to "one-alternation algorithms" for matroid intersection: algorithms which first make some number of queries to $\mathbf{M}_1$, then make some number of queries to $\mathbf{M}_2$, then halt without querying $\mathbf{M}_1$ again.

Lemma 5.2. *Any (randomized or deterministic) one-alternation matroid intersection algorithm must perform $2^{n-o(n)}$ queries to $\mathbf{M}_1$.*

Proof. It is known [8] that any randomized or deterministic one-round protocol for the INDEX problem must use $\Theta(m)$ bits of communication. It follows that the communication complexity of any one-round protocol for MATINT is $\Theta(|\mathcal{C}^*|) = 2^{n-o(n)}$. The desired result then follows by a simulation argument similar to the one in Fact 4.7. □

Lemma 5.2 yields yet another linear lower bound on the number of queries needed by any matroid intersection algorithm, even randomized algorithms. This result is a consequence of the following fact.

Fact 5.3. The deterministic (multi-round) communication complexity of any function f is at least the logarithm (base 2) of the deterministic one-round communication complexity. This also holds for randomized protocols.

Proof. See Kushilevitz and Nisan [8, p49]. □

Finally, it holds that $N^0(\text{INDEX}) \geq \log m$ and $N^1(\text{INDEX}) \geq \log m$. (This follows via a trivial reduction from the EQ and NEQ functions on $\log m$ bits; these functions are defined and analyzed in Kushilevitz and Nisan [8].) Our reduction therefore shows that $N^0(\text{MATINT}) \geq n - o(n)$ and $N^1(\text{MATINT}) \geq n - o(n)$.

§ 6. Discussion

Queries vs Communication. Can one prove better lower bounds by directly analyzing query complexity rather than resorting to communication complexity? It is conceivable that matroid intersection requires $\Omega(nr^{1.5})$ queries but $D(\text{MATINT}) = O(n)$.

In Section 3, we presented an analysis of the query complexity for very elementary matroids. Extending such an analysis to general matroids seems quite difficult as the independence oracle queries are very powerful compared to queries that have been successfully analyzed in other work, e.g., Rivest and Vuillemin [17].

In-Same-Cycle. Section 4 analyzed the ISC problem, using a rank argument to lower bound $D(\text{ISC})$. We conjecture that the rank lower bound is weak for this problem, and that actually $D(\text{ISC}) = \omega(n)$ 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 communication complexity of (randomized or deterministic) *one-round* communication protocols for this problem. We have also shown that $N^0(\text{ISC}) = \Omega(n)$ and $N^1(\text{ISC}) = \Omega(n)$.

The definition of ISC involved a partition $\mathcal{P}$ of a ground set $S = U \cup V$ into pairs such that each pair has exactly one element of U and one of V. This "bipartite restriction" of $\mathcal{P}$ allows us to draw a connection to permutations, and consequently to the representation theory of $\mathcal{S}_n$. However, from a matroid perspective, the assumption is unnecessary. We could have defined the ISC problem simply to involve a partition $\mathcal{P}$ of the ground set S into pairs, without respecting any bipartition. This definition does not result in a connection to $\mathcal{S}_n$, but rather to the *Brauer algebra* [1, 14], whose representation theory is also well-studied. However, we have shown that the rank of the resulting communication matrix is only $2^{O(n)}$.

Are there other families of matroids for which matroid intersection reduces to a permutation problem that can be analyzed by similar techniques? Could this lead to stronger lower bounds? We were unable to find other interesting families of matroids which give a clean connection to Jucys-Murphy elements, as in Lemma 4.18. However, we did find a different approach to analyzing the ISC problem, using *characters* rather than directly computing the spectrum. We precisely computed the number of non-zero characters using tools from the theory of symmetric functions [20]. It is possible that this approach may be less brittle than the approach using Jucys-Murphy elements, and might allow a broader class of problems to be analyzed.

Bounded-Alternation Matroid Intersection Algorithms. In Section 5, we defined the notion of one-alternation algorithms for matroid intersection, and proved that such algorithms must perform $2^{n-o(n)}$ queries. The definition generalizes in the natural way to algorithms with only k alternations. Can one prove a query lower bound for k-alternation matroid intersection algorithms? Is it true that $2^{\Omega(n)}$ queries are required for any constant k?

References

[1] Hélène Barcelo and Arun Ram. Combinatorial representation theory. In Louis J. Billera, Anders Björner, Curtis Greene, Rodica Simion, and Richard P. Stanley, editors, *New perspectives in algebraic combinatorics*, volume 38 of *Mathematical Sciences Research Institute Publications*, pages 23–90. Cambridge University Press, 1999.

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

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

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

[5] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[6] Gordon James and Adalbert Kerber. *The Representation Theory of the Symmetric Group*. Addison-Wesley, 1981.

[7] Donald E. Knuth. The asymptotic number of geometries. *Journal of Combinatorial Theory, Series A*, 16:398–400, 1974.

[8] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

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

[10] Kurt Mehlhorn and Erik 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.

[11] James G. Oxley. *Matroid Theory*. Oxford University Press, 1992.

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

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

[14] Arun Ram. Characters of Brauer's centralizer algebras. *Pacific Journal of Mathematics*, 169(1):173–200, 1995.

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

[16] Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

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

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

[19] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2003.

[20] Richard P. Stanley. *Enumerative Combinatorics*, volume 2. Cambridge University Press, 1999.

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

[22] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57:509–533, 1935.