CSE 291: Communication Complexity, Winter 2019
Nondeterministic protocols

Shachar Lovett
January 15, 2019

1 Overview

Nondeterminism is studied in complexity theory, where one of the main open problems is whether P=NP. In this part, we define the analogous models in communication complexity and study their relation to the deterministic counterpart.

2 The model: nondeterministic protocols

Let \( f : X \times Y \to \{0, 1\} \) be a boolean function. A nondeterministic protocol for \( f \) is composed of two steps. First, an oracle which has access to both inputs \( x, y \) gives an advice \( a \) (a bit string) to the players. Then, the players, having access to this common advice string, as well as their own inputs, need to communicate and decided whether to accept (output 1) or reject (output 0). Let us denote by \( \Pi(x, y, a) \) the output of the protocol. This protocol computes \( f \) correctly if the following holds:

- If \( f(x, y) = 1 \) then \( \exists a, \Pi(x, y, a) = 1. \)
- If \( f(x, y) = 0 \) then \( \forall a, \Pi(x, y, a) = 0. \)

The cost of \( P \) is the sum of the maximal length of the advice \( a \) and the number of bits that the players communicate. The nondeterministic cost of \( f \), denoted \( N^1(f) \), is the minimal cost of a nondeterministic protocol computing \( f \).

A co-nondeterministic protocol for \( f \) (in analog with coNP) is a nondeterministic protocol for \( 1 - f \). The co-nondeterministic cost of \( f \) is defined analogously and denoted \( N^0(f) \).

2.1 Covers

There is a tight relation between nondeterministic protocols and covers.

**Definition 2.1.** Let \( f : X \times Y \to \{0, 1\} \). For \( z \in \{0, 1\} \), a \( z \)-cover of \( f \) is a collection of rectangles \( R_1, \ldots, R_N \), not necessarily disjoint, such that \( f^{-1}(z) = \cup R_i \). The minimal size of a \( z \)-cover for \( f \) is denote by \( C^z(f) = N \).
Nondeterministic protocols are in one to one correspondences with 1-covers, and co-nondeterministic protocols are in one to one correspondences with 0-covers.

Lemma 2.2. Let \( f : X \times Y \rightarrow \{0, 1\} \). Then \( N^z(f) = \log C^z(f) + O(1) \) for \( z \in \{0, 1\} \).

Proof. We prove for \( z = 1 \), the other case is analogous. First, assume a bound on \( C^1(f) \), namely a cover \( R_1, \ldots, R_N \) of \( f^{-1}(1) \) of size \( N \leq C^1(f) \). The advice \( a \) that the oracle gives is the rectangle \( R_i \) in the cover containing \( x, y \), if one exists. The players then verify that indeed \( x, y \in R \), which requires a communication of one bit from each. This gives a nondeterministic protocol of cost

\[ N^1(f) \leq \log C^1(f) + 2. \]

In the other direction, assume a nondeterministic protocol \( \Pi \) where the advice is \( a \in \{0, 1\}^k \), after which the players exchange \( \ell \) bits. The cost of this protocol is \( k + \ell \). We will define a \( 2^k + \ell \) cover for \( f^{-1}(1) \). To do that, for every choice of \( a \), and every choice for the transcript (bits that the players send) \( \tau \in \{0, 1\}^\ell \) after which the protocol accepts, define a rectangle \( R_{a, \tau} = A_{a, \tau} \times B_{a, \tau} \) as follows. \( A_{a, \tau} \) is the set of all \( x \in X \) for which Alice, on input \( x \) and given the advice \( a \), would follow the transcript \( \tau \) on her bits. \( B_{a, \tau} \) is defined similarly for Bob. Note that \( R_{a, \tau} \) is a 1-monochromatic rectangle for \( f \), and that as each \( x, y \) is contained in at least one such rectangle, this gives a 1-cover of size \( 2^k + \ell \). Thus we get

\[ C^1(f) \leq 2^{N^1(f)} \]

or equivalently, \( N^1(f) \geq \log C^1(f) \).

\[ \square \]

3 \( P = NP \cap coNP \) in the communication world

It is a big open problem in complexity whether \( P = NP \cap coNP \) for Turing machines. We will see that with the analogous definitions, this is true in the communication world. First, given a boolean function \( f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\} \), then:

- \( f \in P^{CC} \) if \( D(f) = \text{polylog}(n) \).
- \( f \in NP^{CC} \) if \( N^1(f) = \text{polylog}(n) \).
- \( f \in coNP^{CC} \) if \( N^0(f) = \text{polylog}(n) \).

The fact that \( P^{CC} = NP^{CC} \cap coNP^{CC} \) follows from the following general lemma.

Lemma 3.1. Let \( f : X \times Y \rightarrow \{0, 1\} \). Then \( D(f) = O(N^0(f)N^1(f)) \).

Proof. Let \( R_1, \ldots, R_N \) be a 0-cover of \( f \), and \( R'_1, \ldots, R'_{N'} \) be a 1-cover of \( f \), where \( N \leq 2^{N^0(f)} \) and \( N' \leq 2^{N^1(f)} \). The main observation is that \( R_i, R'_j \) are disjoint for all \( i \in [N], j \in [N'] \). Let \( R_i = A_i \times B_i \) and \( R'_j = A'_j \times B'_j \). Then either \( A_i, A'_j \) are disjoint, or \( B_i, B'_j \) are disjoint.
We now use this observation to design an efficient deterministic protocol. The goal of
the players is to find a rectangle \( R^* \in \{ R_1, \ldots, R_N, R'_1, \ldots, R'_{N'} \} \) which contains their inputs
\( x, y \). This will determine \( f(x, y) \). The players first guess \( z = f(x, y) \) and then try to verify
it. If they fail, the move to the other \( z \). We present the proof for the case of \( z = 0 \), the other
case is analogous.

So, assume \( f(x, y) = 0 \), and the goal of the players is to find a rectangle \( R^* = R_i \) which
contains \( (x, y) \). Denote \( R^* = A^* \times B^* \). For every rectangle \( R'_j \) we must have that either
\( A^*, A' \) or \( B^*, B' \) are disjoint. So one of the following must hold:

- Given knowledge that \( x \in A^* \), we can remove at least half of the rectangles \( R'_j = A'_j \times B'_j \)
since \( A^*, A' \) are disjoint; or
- Given knowledge that \( y \in B^* \), we can remove at least half of the rectangles \( R'_j = A'_j \times B'_j \)
since \( B^*, B' \) are disjoint.

We want to use this to iteratively shrink the size of the 1-cover by a factor of two in each
step. The problem is that the players don’t know \( R^* \). The man insight is that they don’t
need to!

The actual protocol proceeds as follows. In each step, each player checks the following:

- Alice checks if there exists \( R_i = A_i \times B_i \) such that (i) \( x \in A_i \) and (ii) \( A_i \) is disjoint
  from at least half of \( A'_j \) for \( j \in [N'] \).
- Bob checks if there exists \( R_i = A_i \times B_i \) such that (i) \( y \in B_i \) and (ii) \( B_i \) is disjoint from
  at least half of \( B'_j \) for \( j \in [N'] \).

By our previous argument, at least one of them must succeed. In this case they send the
index \( i \in [N] \), which requires \( N^0(f) \) bits. This shrinks the size of the 1-cover by a factor
of two, so they need to repeat this \( N^1(f) \) times. Together the overall communication is \( O(N^0(f)N^1(f)) \).

It is not enough to assume that just one of \( N^0(f), N^1(f) \) is small to deduce that \( D(f) \) is
small.

**Example 3.2.** Consider the set disjointness problem on \( n \) bits. We saw that \( D(DISJ) = \Omega(n) \). However \( DISJ \) has a 0-cover of size \( n \), given by \( R_i = \{(x, y) \in \{0, 1\}^n \times \{0, 1\}^n : x_i = y_i = 1\} \) for \( i \in [n] \). So \( N^0(DISJ) = O(\log n) \).

There are examples where \( D(f) \) is quadratic in \( N^0(f), N^1(f) \).

**Example 3.3.** Let \( n = m^2 \). Consider the inputs \( x, y \in \{0, 1\}^n \) as \( m \times m \) binary matrices.
Define \( f(x, y) = 1 \) if there exists \( i \in [m] \) such that the \( i \)-th row of \( x \) and \( i \)-th row of \( y \) are
the same. Then \( N^1(f) = O(m) \), \( N^0(f) = O(m \log m) \) and \( D(f) = \Omega(m^2) \).

A similar lemma holds if we assume that just one of \( N^0(f), N^1(f) \) is small, but in addition
that the rank is small.

**Lemma 3.4.** Let \( f : X \times Y \to \{0, 1\} \). Assume that \( \text{rank}(M_f) = r \). Then for any \( z \in \{0, 1\} \)

it holds that \( D(f) = O(N^2(f) \log r) \).

The proof is very similar to the proof of Lemma 3.1 and will be left as homework.
**Partitions.** Let \( f : X \times Y \rightarrow \{0, 1\} \). The *partition number* of \( f \), denoted \( P(f) \), is the smallest \( N \) such that \( M_f \) can be partitioned to monochromatic rectangles. As deterministic protocols for \( f \) with cost \( c \) induces a partition of \( M_f \) into \( 2^c \) monochromatic rectangles, we get

\[
\log P(f) \leq D(f).
\]

In the other direction, we clearly have \( C^0(f), C^1(f) \leq P(f) \), which translates to \( N^0(f), N^1(f) \leq \log P(f) \), and so

\[
D(f) = O(N^0(f)N^1(f)) = O(\log^2 P(f)).
\]

For a long time, it was unknown whether the dependence between \( D(f) \) and \( \log P(f) \) should be linear, quadratic, or somewhat in between. Recently [2] showed that the dependence can be quadratic; concretely, there are boolean functions \( f \) for which \( D(f) = \tilde{\Theta}(\log^2 P(f)) \), where \( \tilde{\Theta} \) hides poly-logarithmic factors.

### 4 Lower bound techniques

Let \( f : X \times Y \rightarrow \{0, 1\} \). Our goal is to prove that \( N^1(f) \) is large. One option is to use Lemma 3.1 which gives

\[
N^1(f) \geq \Omega \left( \frac{D(f)}{N^0(f)} \right).
\]

Take for example the set-disjointness function on \( n \)-bit inputs. We already showed that \( D(\text{DISJ}) = \Omega(n) \). In addition, we saw that \( N^0(\text{DISJ}) = O(\log n) \). This gives

\[
N^1(\text{DISJ}) = \Omega(n/\log n).
\]

The following lemma allows to obtain a sharper bound of \( N^1(\text{DISJ}) = \Omega(n) \).

**Lemma 4.1.** Let \( f : X \times Y \rightarrow \{0, 1\} \). Assume that for some \( \alpha \leq \beta \) it holds that:

(i) Each 1-monochromatic rectangle \( R \) has size \( |R| \leq \alpha |X||Y| \).

(ii) The number of 1-inputs is \( |f^{-1}(1)| \geq \beta |X||Y| \).

Then \( N^1(f) \geq \log(\beta/\alpha) \).

**Proof.** Let \( R_1, \ldots, R_N \) be an optimal 1-cover for \( f \). Then \( |R_i| \leq \alpha |X||Y| \) and \( \sum |R_i| \geq |\bigcup R_i| \geq \beta |X||Y| \). Thus \( N \geq \beta/\alpha \).

The bound for set-disjointness follows from Lemma 4.1 using the following two claims.

**Claim 4.2.** Let \( R \) be a 1-monochromatic rectangle for set disjointness. Then \( |R| \leq 2^n \).

**Proof.** Assume that \( R = A \times B \). For each \( i \in [n] \), we must have that all sets \( x \in A \) do not contain \( i \), or that all sets \( y \in B \) do not contain \( i \). Thus we can partition \( [n] = I \cup J \) such that \( x \subset I \) for all \( x \in A \) and \( y \subset J \) for all \( y \in B \). Thus \( |R| = |A||B| \leq 2^{n+|I|+|J|} = 2^n \).
Claim 4.3. $\text{DISJ}^{-1}(1) = 3^n$.

Proof. Two inputs $x, y \in \{0,1\}^n$ are disjoint if for each $i \in [n]$, $(x_i, y_i) \in \{(0, 0), (0, 1), (1, 0)\}$. There are $3^n$ such choices.

Using Lemma 4.1 we obtain

\[ N^1(\text{DISJ}) \geq \log(3^n/2^n) = \Omega(n). \]

5 Application: Linear programming lower bounds

Linear programming is a powerful algorithmic technique. A polytope is a subset $P \subset \mathbb{R}^n$ given by

\[ P = \{ x \in \mathbb{R}^n : Ax \leq b \}, \]

where $A \in \mathbb{R}^{f \times n}, b \in \mathbb{R}^f$ and the condition $Ax \leq b$ is coordinate-wise. Linear programming solves the problem of maximizing $\langle x, c \rangle$ over $x \in P$, for some objective function $c \in \mathbb{R}^n$. The complexity of solving this problem is polynomial in the description complexity of the polytope, which is max($f, n$). The important parameter here is $f$, which can described in two equivalent ways:

1. $f$ is the number of linear inequalities needed to describe $P$.
2. $f$ is the number of facets (maximal dimension faces) of $P$.

Linear programming can possibly be used to solve hard optimization problem, if somehow the complexity of linear programming can be controlled.

Example 5.1 (MAX-CUT and the cut polytope). Consider MAX-CUT, which is an NP-hard problem. The input is a weighted graph $G = (V, E)$ with weights on the edges $w : E \to \mathbb{R}$. The goal is to find a set $S \subset V$ which maximizes the cut value

\[ \text{CUT}(G, S) = \sum_{i \in S, j \notin S} w(i, j). \]

The cut value of the graph is

\[ \text{CUT}(G) = \max_{S \subset V} \text{CUT}(G, S). \]

To formulate this as a linear programming problem, note that we may assume that the graph is the complete graph by having some edges with zero weight. Then the only input is the weight vector $w$. We think of $w$ as the direction in which we optimize the linear program. Next, we define the polytope, which in this case is known as the cut polytope. Its vertices correspond to all possible cuts. Concretely, it is $P_{\text{CUT}} \subset \mathbb{R}^{n^2}$ whose vertices are $x_S$ given by

\[ (x_S)_{i,j} = \begin{cases} 
1 & \text{if } i \in S, j \notin S \\
0 & \text{otherwise}
\end{cases} \]
Then it is simple to verify that
\[ \text{CUT}(G) = \max_{x \in \mathcal{PCUT}} \langle w, x \rangle. \]

Thus, if we could efficiently solve linear programming over the cut polytope, this would prove that \( P = NP \).

In the 1990s there have been several papers which claimed to solve NP-hard problems via complicated linear programs, that added auxiliary variables but claimed to solve the same optimization problem. It was hard to find the bug in them, but they seem to be flawed. Yannakakis [4] set out to rule out these approaches in a systematic manner. It turns out that all of this approaches can be formulated as optimizing over extended formulations of a basic polytope.

**Definition 5.2 (Extended formulation).** Let \( P \subset \mathbb{R}^n \) be a polytope. An extended formulation of it is a polytope \( Q \subset \mathbb{R}^{n+m} \) given by
\[
Q = \{(x, y) \in \mathbb{R}^{n+m} : Cx + Dy \leq d\},
\]
where \( C \in \mathbb{R}^{e \times n}, D \in \mathbb{R}^{e \times m}, d \in \mathbb{R}^e \), such that the projection of \( Q \) to the \( x \) coordinates gives \( P \). That is,
\[
P = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m, (x, y) \in Q\}.
\]
The extension complexity of \( P \) is the minimal \( e \), such that there exists an extended formulation \( Q \) of \( P \) with \( e \) facets (namely, described by \( e \) linear inequalities). We denote it by \( \text{ext}(P) = e \).

The main motivation for extended formulation is that linear programming over \( P \) can be solved using linear programming over \( Q \), namely
\[
\max_{x \in P} \langle x, c \rangle = \max_{(x, y) \in Q} \langle x, c \rangle.
\]
So, if say \( P \) has exponentially many facets, but \( \text{ext}(P) \) is polynomial, then we can still solve linear programming over \( P \) in polynomial time.

**Example 5.3 (Permutahedron).** The permutahedron \( P \subset \mathbb{R}^n \) is the convex hull of all vectors \((\pi(1), \ldots, \pi(n))\), where \( \pi \in S_n \) ranges over all permutations of \( \{1, \ldots, n\} \). It is known that \( P \) has exponentially many facets (\( 2^n - 2 \) to be exact), and in addition it has \( n! \) vertices.

However, the extension complexity of \( P \) is small. Let \( Q \in \mathbb{R}^{n^2} \) denote the convex hull of all doubly-stochastic matrices. That is, if we index the coordinates of \( y \in Q \) as \((y_{i,j})_{i,j \in [n]}\) then
\[
Q = \left\{ y \in \mathbb{R}^{n^2} : y_{i,j} \geq 0, \sum_i y_{i,j} = 1 \forall j, \sum_j y_{i,j} = 1 \forall i \right\}.
\]
It is well known that the vertices of \( Q \) are the permutation matrices, which are in a one-to-one correspondence with the vertices of \( P \). Concretely, let \( \pi \in S_n \) be a permutation. The corresponding vertex of \( P \) is \( x^\pi = (\pi(1), \ldots, \pi(n)) \) and of \( Q \) is \( y^\pi \) given by \( y^\pi_{i,j} = 1_{\pi(i)=j} \).
We need a linear map \( L : \mathbb{R}^{n^2} \to \mathbb{R}^n \) which maps each \( y^\pi \) to \( x^\pi \). By linearity, it will satisfy \( L(Q) = P \). For example, one can take

\[
L(y) = \left( \sum_j jy_{1,j}, \ldots, \sum_j jy_{n,j} \right)
\]

We thus obtain that \( \text{ext}(P) = O(n^2) \).

### 5.1 Slack matrix

Yannakakis proved that the extension complexity of \( P \) is captured by its slack matrix.

**Definition 5.4 (Slack matrix).** Let \( P \subset \mathbb{R}^n \) be a polytope. Assume that \( P \) has \( v \) vertices \( x_1, \ldots, x_v \in \mathbb{R}^n \), and that \( P \) is described using \( f \) inequalities (facets) given by \( \langle a_i, x \rangle \leq b_i \) for \( i \in [f] \). The slack matrix of \( P \), denoted \( S(P) \), is a \( f \times v \) matrix with non-negative entries, given by

\[
S(P)_{i,j} = b_i - \langle a_i, x_j \rangle.
\]

Geometrically, each (non-redundant) inequality \( \langle a_i, x \rangle \leq b_i \) corresponds to a facet of \( P \). If \( v_j \) lies on this facet then \( S(P)_{i,j} = 0 \). Otherwise, \( S(P)_{i,j} > 0 \) is the distance between \( v_j \) and the hyperplane \( \{ x : \langle a_i, x \rangle = b \} \) supporting the facet.

**Definition 5.5 (Non-negative rank).** Let \( A \in \mathbb{R}^{n \times m} \) be a matrix with non-negative entries. The non-negative rank of \( A \) is the minimal \( r \), such that \( A \) can be factored as \( A = ST \), where \( S \in \mathbb{R}^{n \times r}_+, T \in \mathbb{R}^{r \times m}_+ \) have non-negative entries. Equivalently, it is the minimal \( r \) such there exist vectors \( s_1, \ldots, s_n \in \mathbb{R}^r_+, t_1, \ldots, t_m \in \mathbb{R}^r_+ \), such that

\[
A_{i,j} = \langle s_i, t_j \rangle \quad \forall i, j.
\]

We denote the non-negative rank of \( A \) by \( \text{rank}_+(A) = r \).

Yannakakis proved an amazing result: the extension complexity of \( P \) equals the non-negative rank of the slack matrix of \( P \).

**Theorem 5.6.** Let \( P \) be a polytope. Then \( \text{ext}(P) = \text{rank}_+(S(P)) \).

**Proof.** We will prove here the direction \( \text{rank}_+(S(P)) \leq \text{ext}(P) \), which is what we need to prove lower bounds on \( \text{ext}(P) \).

Let \( P \subset \mathbb{R}^n \) be a polytope described by \( P = \{ x \in \mathbb{R}^n : Ax \leq b \} \), where \( A \in \mathbb{R}^{f \times n} \), \( b \in \mathbb{R}^f \). Let \( Q \) be an extended formulation of \( P \), given by \( Q = \{ (x, y) \in \mathbb{R}^{n+m} : Cx + Dy \leq d \} \), where \( C \in \mathbb{R}^{e \times n}, D \in \mathbb{R}^{e \times m}, d \in \mathbb{R}^e \). Let \( S = S(P) \) be the slack matrix of \( P \). We will prove that \( \text{rank}_+(S) \leq e \).

What does it mean that \( P \) is a projection of \( Q \)? we need that each inequality of \( P \) given by \( \langle a_i, x \rangle \leq b_i \), can be obtained from the inequalities of \( Q \). One way that this could happen is the following. Assume that there exists a non-negative vector \( s_i \in \mathbb{R}^e_+ \) such that

\[
 s_i^T C = a_i, \quad s_i^T D = 0, \quad s_i^T d = b_i.
\]
If this happens then
\[ Cx + Dy \leq d \Rightarrow s^T_i (Cx + Dy \leq d) \equiv \langle a_i, x \rangle \leq b_i. \]

Farkas lemma, which is a fundamental lemma in linear programming, tells us that this is the only way that this could happen. So, as we know that \( P \) is a projection of \( Q \), we obtain that there exist vectors \( s_1, \ldots, s_f \in \mathbb{R}_e^+ \) which generate all the inequalities of \( P \).

Next, let \( x_1, \ldots, x_v \in \mathbb{R}^n \) be the vertices of \( P \). As \( P \) is a projection of \( Q \), there exist \( y_1, \ldots, y_v \in \mathbb{R}^m \) such that \( (x_i, y_i) \in Q \) for all \( i \) (in fact, we can take these to be vertices of \( Q \), but this is not necessary). So we have:
\[ S_{i,j} = b_i - \langle a_i, x_j \rangle = s_i^T (d - Cx_j) = s_i^T (d - Cx_j - Dy_j). \]

Define \( t_j = d - Cx_j - Dy_j \), where \( t_j \in \mathbb{R}_e^+ \) since \( (x_j, y_j) \in Q \). We thus obtain that
\[ S_{i,j} = \langle s_i, t_j \rangle \]
where \( s_i, t_j \in \mathbb{R}_e^+ \). This implies that \( \text{rank}_+(S) \leq e \), as claimed.

\[ \Box \]

### 5.2 Connections to nondeterministic complexity

Remember that our goal is to prove that for some hard polytopes (such as the cut polytope) linear programming requires exponential time. Formally, we want to prove that their extension complexity is exponential. Theorem 5.6 shows that it is sufficient to prove that their slack matrix has exponential non-negative rank. In this section, we show how communication complexity can help in proving this.

**Definition 5.7 (Sign matrix).** Let \( S \in \mathbb{R}^{n \times m}_+ \) be a matrix with non-negative entries. Its sign matrix \( \text{sign}(S) \in \{0, 1\}^{n \times m} \) is a sign matrix, with the same dimensions, defined as
\[
\text{sign}(S)_{i,j} = \begin{cases} 
0 & \text{if } S_{i,j} = 0 \\
1 & \text{if } S_{i,j} > 0 
\end{cases}
\]

Note that if \( M \) is a sign matrix then we can consider it as a two-party boolean function, \( M = M_f \) for \( f : X \times Y \rightarrow \{0, 1\} \), where \( X \) are the rows of \( M \) and \( Y \) the columns of \( M \). Thus we extend the communication complexity definitions to sign matrices. Nondeterministic communication complexity will be specifically important.

**Lemma 5.8.** Let \( S \) be a non-negative matrix, and let \( M = \text{sign}(S) \) be its sign matrix. Then
\[ \text{rank}_+(S) \geq C^1(M), \]
where to recall \( C^1(f) \) is the 1-covering number of \( M \).

**Proof.** Assume \( \text{rank}_+(S) = r \). We can factor \( S_{i,j} = \langle u_i, v_j \rangle \) where \( u_i, v_j \in \mathbb{R}_e^+ \). We will show that \( C^1(M) \leq r \). Define \( r \) rectangles as follows: \( R_k = A_k \times B_k \), where \( A_k = \{x : u_{i,k} > 0\} \) and \( B_k = \{y : v_{j,y} > 0\} \). Note that \( S_{x,y} > 0 \) iff \( (x, y) \in R_k \) for some \( k \). \( \Box \)
This suggests the following strategy: show that the sign matrix of the slack matrix of the target polytope contains as a sub-matrix a problem for which we have exponential bounds on its 1-covering number. For example, as we showed it for set-disjointness, so this could be a good target problem to try and embed.

5.3 The correlation polytope

We will prove the lower bound on a polytope related to the cut polytope - the correlation polytope. Our proof follows [1] and its exposition in [3]. The correlation polytope is a polytope $P_{COR} \subset \mathbb{R}^{n^2}$ whose vertices are $x_S$ for $S \subseteq [n]$, given by

$$(x_S)_{i,j} = \begin{cases} 1 & \text{if } i, j \in S \\ 0 & \text{otherwise} \end{cases}$$

The proof for the cut polytope (or other polytopes considered in the literature, such as the TSP polytope) is similar but more technical, so we avoid it here. We will prove the following theorem.

**Theorem 5.9.** The extension complexity of $P_{COR}$ is at least $(3/2)^n$.

The faces of the correlation polytope are well understood. We will need the following lemma.

**Lemma 5.10.** For every $S \subseteq [n]$, there exists a face $f_S$ of $P_{COR}$ that satisfies the following. For every $R \subseteq [n]$, $x_R \in f_S$ iff $|S \cap R| = 1$.

**Proof.** Fix $S \subseteq [n]$. Consider for moment $z = 1_R$ the indicator vector of $R$. It satisfies the quadratic inequality:

$$\left( \sum_{i \in S} z_i - 1 \right)^2 \geq 0,$$

with equality iff $|S \cap R| = 1$. Expanding the expression gives

$$\sum_{i,j \in S} z_i z_j - 2 \sum_{i \in S} z_i + 1 \geq 0.$$ 

We will linearize this quadratic equation in the $n^2$ variables $x_{i,j} = z_i z_j$. Note that every vertex $x_R$ of $P_{COR}$ indeed satisfies $(x_R)_{i,j} = (z_R)_i (z_R)_j$, where $z_R = 1_R$ is a 0/1 vector. Such vectors also satisfy $(z_R)_i (z_R)_j^2 = (x_R)_{i,i}$. Thus every vertex $x = x_R$ of $P_{COR}$ satisfies the linear inequality

$$\sum_{i,j \in S} x_{i,j} - 2 \sum_{i \in S} x_{i,i} + 1 \geq 0.$$ 

As this holds for all vertices, it holds for all $x \in P_{COR}$. In addition, equality is obtained iff $|S \cap R| = 1$, so $x_R \in f_S$ iff $|S \cap R| = 1$. 

\[ \Box \]
We obtain that the correlation polytope (that is, the sign rank of the slack matrix of the correlation polytope) contains as a sub-matrix \( M_g \), where \( g \) is the two-party \( n \)-bit function given by:

\[
g(x, y) = \begin{cases} 
0 & \text{if } |x \cap y| = 1 \\
1 & \text{if } |x \cap y| \neq 1
\end{cases}
\]

So, we need to prove lower bounds on the nondeterministic complexity of this \( g \).

### 5.4 Unique disjointness

Instead of studying this (somewhat ad-hoc) problem \( g \), we consider instead the unique disjointness problem: this is a promise version of the set-disjointness problem, but where we only care about inputs \( x, y \subseteq [n] \) where we are promised that \( |x \cap y| \in \{0, 1\} \). Namely,

\[
\text{UDISJ}(x, y) = \begin{cases} 
1 & \text{if } |x \cap y| = 0 \\
0 & \text{if } |x \cap y| = 1 \\
\ast & \text{if } |x \cap y| \geq 2
\end{cases}
\]

Note that UDISJ is also a promise problem of \( g \), and so it suffices to prove lower bounds on the nondeterministic complexity of UDISJ. To conclude the proof, we will prove that \( N^1(\text{UDISJ}) = \Omega(n) \), which will show that the extension complexity of \( P_{\text{COR}} \) is \( 2^{\Omega(n)} \).

Our goal is to prove that if \( R_1, \ldots, R_N \) are a 1-cover of \( f \) then \( N = 2^{\Omega(n)} \). As \( f \) is a promise problem, lets spell this out explicitly. The rectangles need to satisfy:

(i) If \( f(x, y) = 1 \) then \( \exists R_i, (x, y) \in R_i \).

(ii) If \( f(x, y) = 0 \) then \( \forall R_i, (x, y) \notin R_i \).

That is, if we take \( S = \bigcup R_i \), then \( f^{-1}(1) \subseteq S \) and \( S \cap f^{-1}(0) = \emptyset \). We do not care about \( S \cap f^{-1}(\ast) \).

The proof strategy will be similar to how we proved that \( N^1(\text{DISJ}) = \Omega(n) \): we will show that there are many 1-entries in \( f \), and that each rectangle can only cover few of them, so many rectangles are needed. We already saw that \( |f^{-1}(1)| = 3^n \). The result then follows from the following lemma, which is a strengthening of Claim 4.2.

**Lemma 5.11.** Let \( R \) be a rectangle such that \( R \cap f^{-1}(0) = \emptyset \). Then \( |R \cap f^{-1}(1)| \leq 2^n \).

Before proving the lemma, note that it implies Theorem 5.9:

\[
\text{ext}(P_{\text{COR}}) \geq C^1(g) \geq C^1(\text{UDISJ}) \geq (3/2)^n.
\]

**Proof of Lemma 5.11.** Let \( R = A \times B \), where we assume that there is no \( x \in A, y \in B \) with \( |x \cap y| = 1 \). Our goal is to upper bound the number of \( x \in A, y \in B \) with \( |x \cap y| = 0 \). The proof will be by induction on \( n \).

Denote \( S = R \cap f^{-1}(1) \). Let \( (x, y) \in S \), where we identify \( x, y \in \{0, 1\}^n \). We denote \( x = x'a, y = y'b \) with \( x', y' \in \{0, 1\}^{n-1} \) and \( a, b \in \{0, 1\} \). There are two important observations:
1. It cannot be that \(a = b = 1\).

2. It cannot be that \((x'0, y'1)\) and \((x'1, y'0)\) are both in \(R\). Because in this case, also \((x'1, y'1) \in R\); but \(|x'1 \cap y'1| = |x' \cap y'| + 1 = 1\), contradicting our assumption of \(R\).

Define two sets \(S_1, S_2 \subset \{0, 1\}^{n-1} \times \{0, 1\}^{n-1}\) as follows:

\[
S_1 = \{(x', y') : (x'0, y'0) \in S \text{ or } (x'0, y'1) \in S\}
\]

and

\[
S_2 = \{(x', y') : (x'0, y'0) \in S \text{ or } (x'1, y'0) \in S\}.
\]

We claim that \(|S_1| + |S_2| \geq |S|\). This is since each \((x'0, y'1) \in S\) and each \((x'1, y'0)\) in \(S\) is counted once. If \((x'0, y'0) \in S\), then as we observed it cannot be that both \((x'0, y'1), (x'1, y'0)\) are in \(S\). Thus, it is counted (at least) once.

Next, Let \(R_i\) be the minimal rectangle containing \(S_i\) for \(i = 1, 2\). We will show that each \(R_i\) satisfies the induction hypothesis for \(n-1\), and hence \(|S_i| \leq |R_i| \leq 2^n-1\), which concludes the proof as \(|S| \leq |S_1| + |S_2| \leq 2^n\).

To conclude the proof, we show this for \(R_1\), where the proof for \(R_2\) is analogous. Assume \(R_1\) violates the induction hypothesis. Namely, there exists \((x', y'') \in R_1\) such that \(|x' \cap y''| = 1\).

However, the fact that \((x', y'') \in R_1\) means that there must be \((x', y'), (x'', y'') \in S_1\) for some \(y'', x'\). By the definition of \(S_1\), this means that \((x'0, y'b'), (x''0, y''b'') \in S \subset R\) for some \(b', b'' \in \{0, 1\}\). But then also \((x'0, y''b'') \in R\). This is however a contradiction to our assumption as

\[|x'0 \cap y''b''| = |x' \cap y''| = 1.\]

\(\Box\)

References


