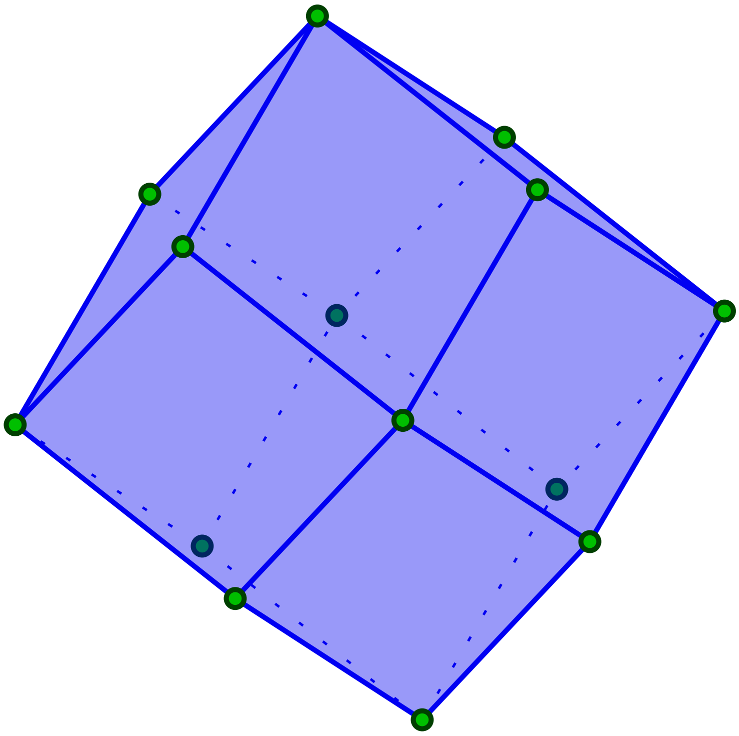
Polyhedral combinatorics

Tomáš Turek[1]



June 25, 2024

[1]Here are some of my notes taken from polyhedral combinatorics which was a part of course Mathematical programming and polyhedral combinatorics.Keep in mind there may be some mistakes. You may visit GitHub.

Chapter 1

Definitions

Reader may already know some basic definitions of polyhedrons and polytopes and also might be familiar with some basic theorems and characterization. But in the other case we will introduce some of these basics one more time. Also note that the main part is that we are considering somewhat basic linear program.

$$\max c^T x$$
$$Ax \leq b$$

Where we are considering a finite number of linear inequalities.

1.1 Polyhedra and Polytopes

The polyhedron created by such linear program is usually called $\mathcal{H}$-polyhedron. But we will formulate it more precisely.

Definition 1. *$\mathcal{H}$-polyhedron is prescribed as $\{x|Ax \leq b\}$ where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$.*

Definition 2 (Minkowski sum)**.** *Minkowski sum of two sets A, B denoted by $A +_{\mathsf{M}} B$ is $\{a + b|a \in A, b \in B\}$.*

Definition 3 (Combinations)**.** *Let V be a finite set, then by the following statements*

1. $x = \sum_{v_i \in V} \lambda_i v_i, \lambda_i \in \mathbb{R}$
2. $1 = \sum_{v_i \in V} \lambda_i$
3. $0 \leq \lambda_i$

we will define:

- *Linear combination* $\mathtt{lin}(V)$ *as 1.*
- *Affine combination* $\mathtt{aff}(V)$ *as 1. and 2.*
- *Conic combination* $\mathtt{cone}(V)$ *as 1. and 3.*
- *Convex combination* $\mathtt{conv}(V)$ *as 1., 2. and 3.*

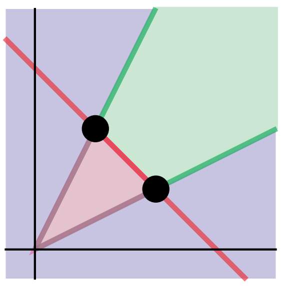


Figure 1.1: Example of combinations, where V are two points in $\mathbb{R}^2$, then we have their linear combination, affine combination, conic combination and convex combination.

Definition 4. *$\mathcal{V}$-polyhedron is defined as* $\texttt{conv}(V) +_{\mathsf{M}} \texttt{cone}(Y)$ *where* V, Y *are finite set of points.*

Definition 5. *Bounded-polyhedron is called **polytope**.*

This can be either visualized just by the definition or consider having a n-dimensional ball which is being cut by hyperplanes until no surface obtained by the ball itself persists.

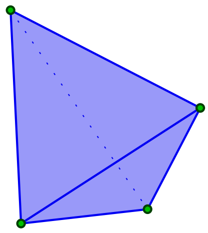
1.1.1 Examples of polytopes

Simplex

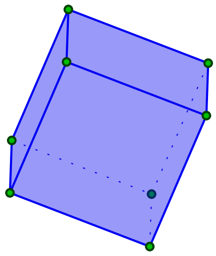
This is a well known polytope which can be prescribed as follows. k-simplex is a convex combination of $k+1$ affine independent vertices.

Cube

Cube is even more known than the simplex. Already here we can see that it can be prescribed as $\mathcal{H}$-polytope $\{x \in \mathbb{R}^k | 0 \leq x_i \leq 1\}$, but also as $\mathcal{V}$-polytope $\texttt{conv}(\{0,1\}^k)$. This is quite essential, because we will see that $\mathcal{H}$-polyhedra and $\mathcal{V}$-polyhedra are equal.



(a) 3 dimensional simplex

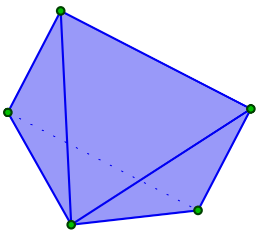


(b) 3 dimensional cube

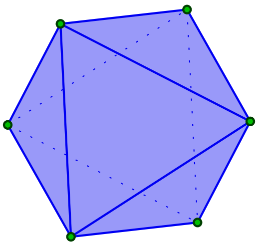
Pyramids and other creations

Also we will show us a simple way how to create new polytopes. That is imagine we have a polytope P and put it in a higher dimension, then by adding one point above the P and creating a convex hull of P and the point we obtain a so called pyramid. We may also denote it as $\texttt{pyr}(P)$. Similarly if we would take two points, where one is above and the second one is below the given P we get bipyramid or $\texttt{bipyr}(P)$.

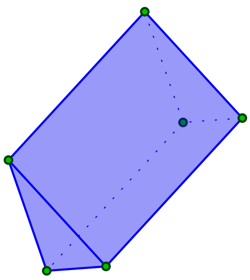
Last creation we will show us right now is if we would take a parallel copy of the polytope P, that is to some other parallel hyperplane and connect these two together. This way we obtain a prism.



(a) Pyramid



(b) Bipyramid



(c) Prism

Theorem 1 (Minkowski-Weyl)**.** *P is $\mathcal{H}$-polyhedron $\iff$ it is a $\mathcal{V}$-polyhedron.*

Sketch of the proof. "⇒" We will gradually make the polyhedron more non-general and then consider a simple case. So WLOG:

1. P is full-dimensional. Where dimension is defined as dimension of the smallest affine space containing it.
2. P is pointed, that is it does not contain a line. – If it contains a line we can split it by an orthogonal hyperplane, inductively use Minkowski-Weyl theorem and then extend Y by rays to both sides of the hyperplane. Use theorem 2.

3. $V = \emptyset$ – Use trick which is called **Homogenization** or **Homogenized cone** which is that $P : Ax \leq b$ create $P' : Ax - bz \leq 0$ and $z \geq 0$. So for $z = 1$ we have original P and then for all others z we have scaled copy of P. After this trick we use Minkowski-Weyl for this cone and create V by the points for which $z > 0$ and Y from points for which $z = 0$.

4. P is a polytope. And with that we use claim 3.

"$\Leftarrow$" Set $P = \{x | x = \sum \lambda_i x_i, 1 = \sum \lambda_i, 0 \leq \lambda_i\}$ which is a $\mathcal{H}$-polytope. By also using Fourier Monskin split to positive, 0 and negative coefficients. ☐

Theorem 2. *P is a pointed $\iff$ it has an extreme point.*

Proof. If there is a line and we have extreme point we can shift the line so it goes through the extreme point. But now the line representing the optimization function is either parallel hence it is not an extreme point or not parallel which also implies it is not an extreme point. ☐

Claim 3. *Lets have polytope $P = \{x | Ax \leq b\}$ and V be the set of extreme point of P. Then $P = \texttt{conv}(V)$.*

Proof. "$P \supseteq \texttt{conv}(V)$" Is easy. So see "$P \subseteq \texttt{conv}(V)$". Suppose it is not true. Take any such x and find a hyperplane separating $\texttt{conv}(V)$ and x which can be done by Hyperplane separation lemma (that is choosing shortest segment and creating an orthogonal hyperplane between them). Then the optimum of the direction set by the norm of this hyperplane gives an extreme point, which is a contradiction. ☐

From the main theorem we may see that from mathematical perspective both $\mathcal{H}$-polyhedrons and $\mathcal{V}$-polyhedrons are the same. But for computer scientists it is pretty much the opposite. Consider solving an LP. Given linear inequalities it takes some time to solve it, but if we have all vertices we can just check every one of them if it is optimum. Also if we would like to see an intersection of two polytopes P, Q it is the opposite. That is we can just add all inequalities together and obtain their intersection. On the other hand for convex points it is known to be NP hard.

Fact. *For polytope $P \subseteq \mathbb{R}^d$ given by n inequalities it has $\leq n^{\lfloor d/2 \rfloor}$ vertices.*

1.2 Faces of polytopes (polyhedrons)

Definition 6. *Let P be a polyhedron. An inequality $\alpha^T x \leq \beta$ is **valid** for P if $P \cap \{x | \alpha^T x \leq \beta\} = P$.*

Definition 7. *Let P be a polyhedron and $\alpha^T x \leq \beta$ a valid inequality. Then $F = P \cap \{x | \alpha^T x = \beta\}$ is called a **face** of P.*

Keep in mind that there are two special cases that are usualy called *trivial* faces. Consider $\mathbf{0}^T x \leq 0$ and $\mathbf{0}^T x \leq 1$ which are valid and the first create a face P, whereas the second $\emptyset$. The other faces are called *non-trivial.*

Theorem 4. *Let P be a polytope $Ax \leq b$ then F is a face of P if and only if $F = \{x | A'x = b'\} \cap P$ for some subset of "original inequalities". Or sometimes called a subsystem.*

Proof. "$\Rightarrow$" Let F be a face of P. Then $\exists$ valid $c^T x \leq \delta$ such that $F = P \cap \{x | c^T x = \delta\}$. In the dual LP for $\max c^T x$ s.t. $Ax \leq b$ let y^* be optimum and let $I = \{i | y_i = 0\}$. Then $F \subseteq P \cap \{x | a_i^T x = b, i \in I\}$ can be seen from the fact about complementarity 1.2. But also the other inclusion holds thus $F = P \cap \{x | a_i^T x = b, i \in I\}$.

"$\Leftarrow$" Let $F = P \cap \{x | a_i^T x = b\}$. Then claim F is a face can be seen by setting $c := \sum_{i \in I} a_i$ and $\delta := \sum_{i \in I} b_i$. See that $c^T x \leq \delta$ is valid and it prescribe a face. ☐

Fact (Complementarity)**.** *For LP $\max c^T x$ s.t. $Ax \leq b$ and its dual $\min b^T y$ s.t. $A^T y = c, y \geq 0$. Let x^* and y^* be primal (dual) optimum solutions, then if $y_i^* > 0$ then $a_i^T x^* = b_i$.*

Proof. $c^T x = y^T Ax = y^T b$ therefore $y^T(Ax + b) = 0$ so component wise it must be 0 $\forall i$, hence either $y_i = 0$ or $(a_i x - b_i) = 0$. ☐

Also faces of polyhedron are also polyhdra as well. Face of a face is also a face and intersection of faces is a face. These are some properties which can be observed. Lastly we may look at special faces by their dimensions which are prescribed in table 1.1.

dimension	face
0	vertices
1	edges
$\vdots$	$\vdots$
$\dim(P) - 2$	ridges
$\dim(P) - 1$	facets

Table 1.1: Most important faces.

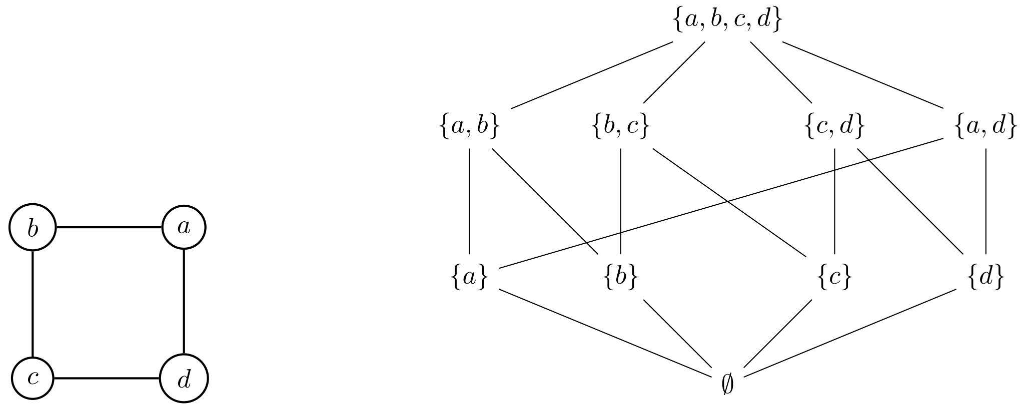


Figure 1.4: Example of face lattice for the given 2-dimensional cube.

1.2.1 Face lettice

Let $\mathcal{F}$ be the set of faces of P, then $(\mathcal{F}, \subseteq)$ is called the *face lattice*. The fact that it is called lattice is due to the properties it has. Moreover it has some other properties, which are sometimes called as a graded lattice (elements can be dividided by their grades). Also for all pairs it has its sublattice.

Example. See an easy example of 2-dimensional cube and its lattice on Fig. 1.4.

1.2.2 Polar duality

For a polytope P which is described as $Ax \leq \mathbf{1}$ and also by $\texttt{conv}(V)$ we have the polar dual polytope P^Δ prescribed as $Vx \leq \mathbf{1}$ which is same as $\texttt{conv}(A)$. Also its dual is the original P, i.e. $(P^\Delta)^\Delta = P$. Note that any $Ax \leq b$ can be changed to $Ax \leq \mathbf{1}$.

The polar duals have some interesting properties. For example a correspondence between vertice of P and facets of P^Δ, edges of P and ridges of P^Δ, ridges of P and edges od P^Δ and facets of P and vertices od P^Δ. Also face lattice of P^Δ is same as for P only "upside down". Lastly the polar duality can be even further generalized.

$$\begin{array}{ccc} Ax \leq \mathbf{1} & & Vx \leq \mathbf{1} \\ Bx \leq \mathbf{0} & \leftrightarrow & Yx \leq \mathbf{0} \\ \texttt{conv}(V) + \texttt{cone}(Y) & & \texttt{conv}(A \cup \{0\}) + \texttt{cone}(B) \end{array}$$

One of the interesting questions may be if we have two polytopes P, P' and we want to know if they are the same. But how they are same? Well there are mainly two ways how to define sameness. In one way by affine operations (which may include some transitions, rotations and scaling) or the other way is to define sameness in a combinatrial way. That is if lattices are equal.

1.3 1-skeleton of polytope

Definition 8. *1-skeleton of polytope is a graph $G = (V, E)$ such that $V = F_0$, which are 0-dimensional faces (vertices) and $E = F_1$, which are 1-dimensional faces (edges). This can be generalized to k-skeleton of polytope by setting $V = F_{k-1}$ and $E = F_k$.*

Theorem 5 (Steinitz)**.** *Planar 3-connected graphs are exactly 1-skeletons of 3-dimensional polytopes.*

There are also present some conjectures. First is that this is not true only for 3-connected planar graphs but generally for d-connected graphs and d-dimensional polytopes. Another conjecture is that 1-skeletons are somewhat nice expanders.

Lecture 10

Extended Formulation

Suppose we are given a description of a $\mathcal{H}$-polyhedron given as an In many cases, the number of equations that define the polyhedron plays a role, for example when optimizing over it (using some optimization method). Therefore, for a given polyhedron it is desireable to find as small a description as possible.

We might ask ourselves: is it possible (by some clever trick on our part) to reduce the number of equations necessary, perhaps by increasing the number of dimensions (number of variables)?

This leads to the notion of *extended formulation.*

Definition 1 (Extended Formulation)**.** *Let $P \subseteq \mathbb{R}^d$ be a polytope. Then polytope $Q \subseteq \mathbb{R}^{d+r}$ is called an* extended formulation *of P if*

$$P = \{x \in \mathbb{R}^d \mid (\exists y \in \mathbb{R}^r)\ (x, y) \in Q\}.$$

In other words, if we project Q to the original variables we get the original polytope P.

Denote $\prod_x Q$ the projection of Q to variables/coordinates x.

The motivation by optimization problems is justified by the following theorem.

Theorem 1 (Extended Formulation Preserves Optima)**.**

$$Q \text{ is an extended formulation of } P$$

$$\Longleftrightarrow$$

$$\begin{matrix} \max c^T x \\ x \in P \end{matrix} \equiv \begin{matrix} \max c^T x \\ (x, y) \in Q \end{matrix}$$

Proof. We prove two implications.

($\Rightarrow$) Immediate as we optimize over the same space.

($\Leftarrow$) For contradiction, suppose that $P \neq \prod_x Q$. Then, there exists a point x *not common* to both P and $\prod_x Q$. Without loss of generality, let $x \in P \setminus \pi_x Q$. Take the separating hyperplane h defined by a vector c separating x from $\pi_x Q$. Optimizing in the direction of c, we get different answers. See Figure 1.

□

Observation: The difference between the number of inequalities of P and those of its extended formulation can be drastic. As an example, consider an n-gon which can be

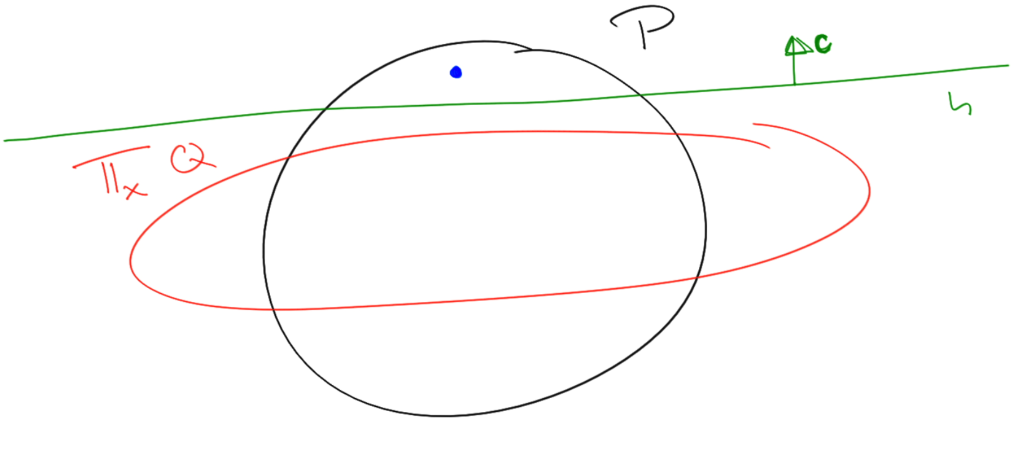


Figure 1: Difference in optimal value when optimizing along the direction perpendicular to the separating hyperplane.

defined using n inequalities. Its extended formulation is a $\log(n)$-cube which can be described using just $2\log(n)$-inequalities.

Firstly, observe that some problems regarding $\mathcal{V}$- and $\mathcal{H}$-polytopes are easy. If we are given an LP oracle then easy problems for $\mathcal{V}$-polytopes are the problems

- computing a non-redundant representation of $\text{conv}(P \cup Q)$

  Simply take all vertices and remove the redundant ones. Notice that deciding whether a vertex is redundant is easy.

  Suppose $v_1, v_2, \ldots, v_n$ are points. Fix one index $j \in [n]$. Then we can check the redundancy by checking the feasibility of the following system.

$$\begin{aligned} \sum_{i \neq j} \lambda_i v_i &= v_j \\ \sum_{i \neq j} \lambda_i &= 1 \\ \lambda_j &= 0 \\ \lambda &\geq 0 \end{aligned}$$

- computing a non-redundant representation of $P + Q$ (analogously)

and for $\mathcal{H}$-polytopes

- computing a non-redundant representation of $P \cap Q$

  To check redundancy of hyperplane optimize in direction of the hyperplane and compare the output after removing it.

A natural step is to ask what is the complexity of the complementary problems, i.e. $\text{conv}(P \cup Q)$ and $P + Q$ for $\mathcal{H}$-polytopes, and $P \cap Q$ for $\mathcal{V}$-polytopes.

Question: *How difficult is it to check that a polytope is an extended formulation of other polytope?*

Problem 1 (Extended Formulation)**.** *Let P and Q be $\mathcal{H}$-polytopes. Check*

$$P \stackrel{?}{=} \prod_x Q.$$

We will sketch that deciding Extended Formulation is NP-hard.

One might wonder, whether we could just enumerate all the vertices of the given polyhedra reduce the problem to an easy one.

Problem 2 (Vertex Enumerate)**.** *Given $\mathcal{H}$-polyhedron P and set of points V, is $vert(P) \stackrel{?}{=} V$?*

However, it was shown by Khachiyan that it is coNP-Hard to enumerate all vertices of a polyhedron given by its facets.

We will reduce the problem to the following one

Problem 3 (Minkowski Verify)**.** *Given $\mathcal{H}$-polyhedra P_1, P_2, S. Is $S = P_1 + P_2$?*

We claim the following

Theorem 2. *Problem Minkowski Verify is NP-hard.*

We now show, that by proving this we also prove the hardness of the extended formulation problem.

Claim 1. *Extended Formulation is NP-hard if Minkowski Verify is NP-hard.*

Proof. Suppose

$$\begin{aligned} P_1 &\equiv A_1 x \leq b_1, \\ P_2 &\equiv A_2 x \leq b_2. \end{aligned}$$

Notice that the Minkowski sum of P_1, P_2 can be expressed as a projection of a suitable polyhedron as

$$P_1 + P_2 = \prod_z \left(\underbrace{\left\{ \begin{array}{c} A_1 x \leq b_1 \\ A_2 y \leq b_2 \\ z = x + y \end{array} \right\}}_{Q} \right).$$

Then by checking $S = \prod_z Q$ we solve the original problem. $\square$

Proof of Theorem 2.

We prove that if we have an algorithm for deciding Minkowski Verify for two arbitrary polytopes, then we can invoke the oracle polynomial number of times and decide for some set of vertices V and an $\mathcal{H}$-polytope P, whether $V = vert(P)$. The hardness then comes from the hardnes of Problem 2.

WLOG assume for polyhedron $P \subseteq \mathbb{R}^d$

- P has a "up" direction and suppose this is along the x_d axis (see Figure 2 below)
- all vertices are at a different height

Now, consider the vertices v_i of V in the order of their x_d-coordinate (in the order of increasing height). Now, consider some v_i and v_{i+1} and define three polytopes in the following way:

$$\begin{aligned} P_{-1} &= P \cap \{x_d = e_d^T v_i\} \\ P_1 &= P \cap \{x_d = e_d^T v_{i+1}\} \\ P_0 &= P \cap \left\{x_d = \frac{e_d^T v_i + e_d^T v_{i+1}}{2}\right\} \end{aligned}$$

where the dot product $e_d^T v_i$ is just the x_d-coordinate of v. See Figure 2 for illustration.

The crucial observation here is that P_0 is actually equal to $\frac{1}{2} P_{-1} + P_1$ (this is called the Cayley trick).

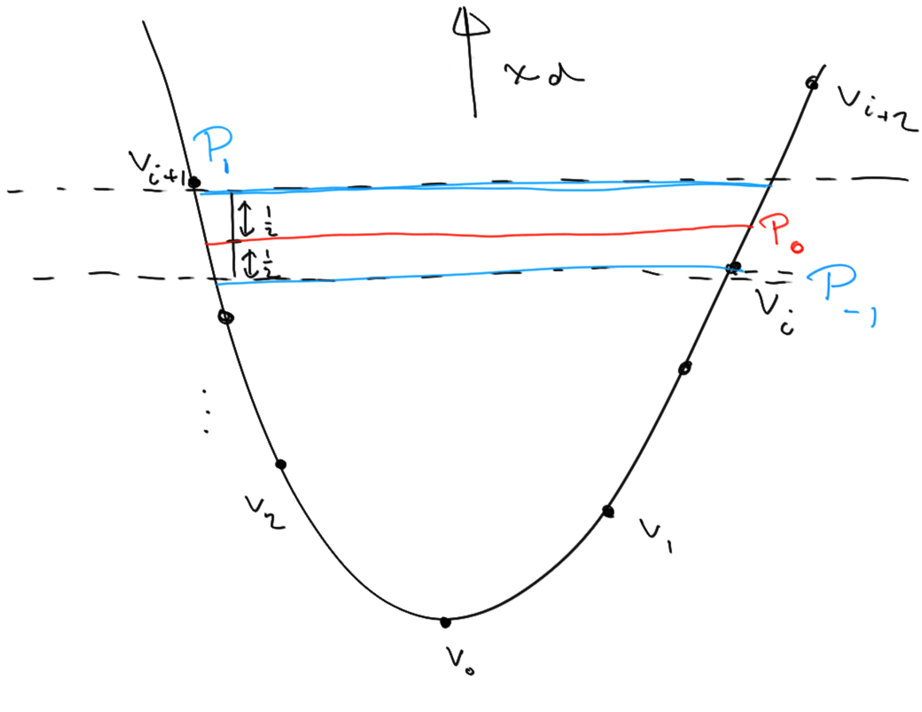


Figure 2: Polyhedron P oriented along the x_d dimension and cuts represented by polyhedra P_{-1}, P_0, P_1

To finish the proof we need to prove the following lemma (this is just copied from the paper, follow the proof along the Figure 3 and it is quite straightforward actually)

Lemma 1. $2P_0 \neq P_{-1} + P_1$ *if and only if there exists some* $v \in vert(P)$ *that is* not *in* V *and* $e_d^T v_i < e_d^T v < e_d^T v_{i+1}$

Proof. We prove the non-trivial direction only. Suppose some vertex $v \in vert(P)$ is not in V and $e_d^T v_i < e_d^T v < e_d^T v_{i+1}$ for some i. WLOG we can assume that v lies above the hyperplane containing P_0. If so, there is an $u \in vert(P_{-1})$ such that $\overrightarrow{uv}$ lies on some edge of P. Clearly, $\overrightarrow{uv}$ intersects P_0, say at w. We claim that $2w \notin P_{-1} + P_1$.

Assume for the sake of contradiction that $2w \in P_{-1} + P_1$. Then there are $x \in P_{-1}$ and $y \in P_1$ such that $2w = x + y$. Since, any point on an edge of a polytope can be *uniquely* represented as the convex combination of the vertices defining the edge, it follows that $x = u$ and y is a vertex of P_1. This implies that v is a convex combination of x, y as well and hence, v *can not* be a vertex of P, a contradiction.

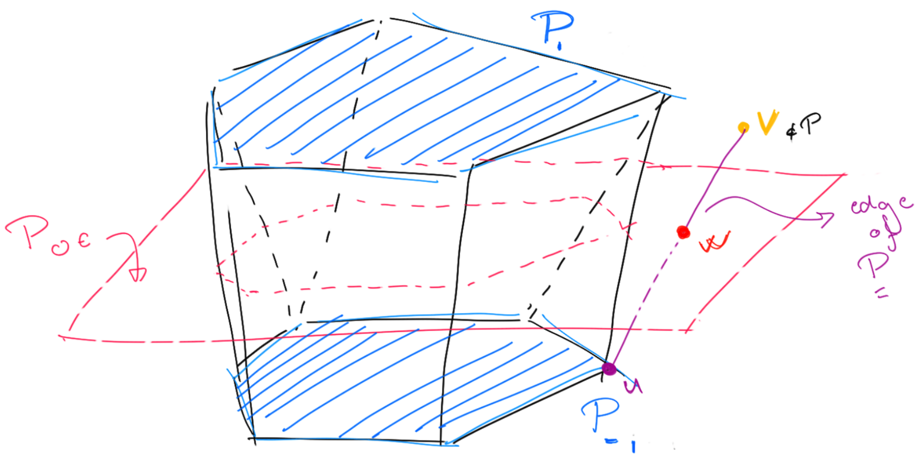


Figure 3: Situation in lemma

□

Now, this lemma gives us a way to use the Minkowski Verify problem to decide whether some vertex between v_i and v_{i+1} is missing. And thus if we can decide Minkowski Verify in poly time, we can also decide Vertex Enumerate, a contradiction.

□

Polyhedral combinatorics – Lecture 11

Filip Kastl

June 8, 2024

version 2 – June 8, 2024

1 Non-negative rank (propositions from the tutorial)

Definition (Non-negative rank). *The* non-negative rank $rk_+(M)$ *for* $M_{m\times n} \geq 0$ *is defined as*

$$rk_+(M) = \min\{r \mid \exists T_{m\times r} \geq 0, U_{r\times n} \geq 0 \ s.t.\ M = TU\}$$

Proposition. $rk_+(M)$ *is equal to the minimum number of rank-1 non-negative matrices that sum to* M.

Observation.

- *The non-negative rank of a matrix is at least as large as its rank*
- *The non-negative rank of a matrix is at most as large as the minimum number of rows and columns of the matrix.*
- *The non-negative rank of a matrix is equal to the non-negative rank of its transpose.*

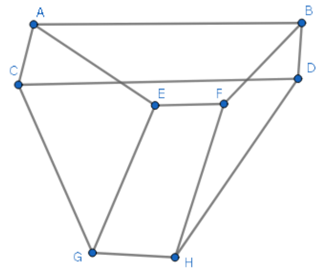
Proposition. *The non-negative rank of the product of two matrices* A *and* B *is at most as large as the minimum of the non-negative rank of* A *and the non-negative rank of* B.

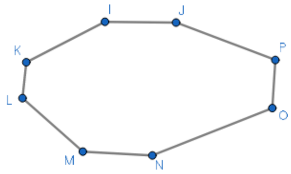
Proposition. *The non-negative rank of the sum of two matrices* A *and* B *is at most as large as the sum of the non-negative rank of* A *and the non-negative rank of* B.

2 Yannakakis theorem

Say we have a problem of size n. We search for a polyhedron representing the problem and its description using linear equalities and inequalities. We might end up with exponentially many inequalities [1] We might search for other descriptions of the polyhedron in $\mathbb{R}^d$ with polynomially many inequalities. If that doesn't work out, another approach is to search for polyhedrons of higher dimension whose projection to $\mathbb{R}^d$ is equivalent to the original polyhedron.

[1] I'd like to stress that we only count the number of inequalities. The equalities don't bother us since we can use linear algebra tools to efficiently solve for the affine space they represent and move into that space. Then we are only left with inequalities.





Definition (Extended formulation). *Let* $P \subseteq \mathbb{R}^d, P = \{x| \dots\}$ *and* $Q \subseteq \mathbb{R}^{d+r}, Q = \{(x,y)| \dots\}$ *be polytopes.*

Q is an extended formulation of P if $P = \Pi_x(Q) := \{x|\exists y : (x,y) \in Q\}$

Definition (Extension complexity). *The* extension complexity $xc(P) =$ *min number of inequalities describing any extended formulation of P.*

Definition (Slack matrix). *Let* $P = \{x|A_{m\times d}x \leq b\} = conv(V_{n\times d})$.

The non-negative slack matrix $S(P)$ *is an* $m \times n$ *matrix s.t.*

$$S_{ij} = b_i - a_i^T v_j$$

The following theorem is important because it enables us to go from bounds on the non-negative rank to bounds on xc.

Theorem (Yannakakis). *Let P be a polytope. Then* $xc(P) = rk_+(S(P))$.

Proof.

reference: Mihalis Yannakakis: Expressing Combinatorial Optimization Problems by Linear Programs, Theorem 3

"$\leq$"

Let's show that for a given slack matrix $S(P)$ of non-negative rank r we can construct an extended formulation Q of the polytope P. Let $P = \{x|A_{m\times n}x \leq b\}$ (WLOG there are no equalities). Let $S(P) = T_{m\times r}U_{r\times n}$ be a non-negative factorization of the slack matrix $S(P)$. We define the extended formulation as follows

$$Q := \{(x,y)|Ax + Ty = b, y \geq 0\}$$

So Q only has r inequalities ($y \geq 0$).

Now let's show that Q is really an extended formulation of P – show that $\Pi_x(Q) = P$.

- "$\subseteq$" Let $x, y \in Q$. By definition of Q it holds that $Ax + Ty = b$. Because T is non-negative it holds that $Ty \geq 0$. That means that $Ax \leq b$ which by definition means that $x \in P$.
- "$\supseteq$" Let's show that for each vertex v_i of P exists y s.t. $(v_i, y) \in Q$. This will suffice since points of P are convex combinations of vertices and those will then translate to convex combinations of points of Q.

  Let's use $y := U_{*i}$. For each i, (v_i, U_{*i}) is a feasible solution of $Ax + Ty = b, y \geq 0$ because $TU_{*i} = S_{*i}$, which is exactly the slack of vertex v_i.

"$\geq$"

Let $P = \{x | Ax \leq b\} = conv(V)$ and let $Q = \{(x, y) | Ex + Fy \leq g\}$ be its extended formulation with r inequalities. Note that $\forall i : a_i^T x \leq b_i$ is a valid inequality for Q. For each of these inequalities it should be possible to express them as non-negative linear combinations of the inequalities of Q [2]. Let's denote the coefficients of these linear combinations as $\lambda_1^i, ..., \lambda_r^i, \lambda_k^i \geq 0$.

$$(a_i, 0) = \sum_k \lambda_k^i (E_{k*}, F_{k*})$$

$$b_i = \sum_k \lambda_k^i g_k$$

Notice that for a vertex v of P there is a point (v, u) of Q s.t. the inequality $a_i x \leq b_i$ has the same slack w.r.t. v and (v, u). In the lecture we assumed that (v, u) is a vertex of Q. Let me give an explanation of why we can assume this. We can observe that (v, u) lies on a facet – v is an extreme point of P w.r.t. some direction and optimizing along this direction gives us a facet of Q. All of the points of the facet will have the form (v, y) for some y. This facet will contain some vertices of Q. Let (v, u) be one of these vertices. Let's now continue with the proof.

Slack of the inequality of P $a_i^T x \leq b_i$ with respect to vertex v_j of P is the same as the slack of the inequality $a_i^T x \leq b_i$ that we constructed from inequalities of Q w.r.t. a vertex (v_j, u_j) of Q and that can be expressed as the slack of $\sum_{k=1}^r \lambda_k^i \cdot (E_i x + F_i y \leq g_i)$ w.r.t. (v_j, u_j). Since when you combine inequalities you also combine slack, we finaly get this: $\sum_{k=1}^r \lambda_k^i \cdot$ (slack of $E_i x + F_i y \leq g_i$ w.r.t. (v_j, u_j)).

Now let's define the non-negative matrices that form the rank r factorization of $S(P)$. One of them will be Λ whose rows are the coefficient vectors λ^i. The second will be submatrix S of the slack matrix of Q consisting of those columns of Q corresponding to vertices (v_j, u_j). Both matrices are non-negative, have the right number of rows and columns and due to what we have shown in the previous paragraph, $\Lambda S = S(P)$. □

In particular, this means that all possible slack matrices of P will have the same non-negative rank.

3 Communication complexity

We now sidestep into the theory of communication complexity. We will show that a specific type of a communication complexity problem can be used to bound the non-negative rank of matrix. That

[2] This intuitively makes sense to me but I wouldn't be able to prove it. In the lecture this was handwaved as a consequence of duality.

will in turn be useful for us when we try to bound the extension complexity of some combinatorial problems.

Communication complexity scenario. Let $M_{m\times n}$ be a non-negative matrix. There are two parties: Alice and Bob. Alice gets a row index i and Bob gets a column index j. They communicate and then one of them outputs a number X_{ij}. Their task is to match M_{ij}.

We count the number of bits exchanged. For a given matrix M the *communication complexity* of a communication protocol is the maximum number of bits exchanged over all i, j. The *communication complexity* of the matrix M ($cc(M)$) is the minimum number of bits exchanged over all possible communication protocols.

We will concern ourselves with a variant of this problem where the communication protocol can make decisions base on chance, X_{ij} is a random variable and the goal is $\mathbb{E}[X_{ij}] = M_{ij}\ \forall i, j$. We will denote the communication complexity of this problem as $cc_+(M)$.

Definition (Communication protocol (formally))**.** *A* communication protocol *is a binary tree with internal nodes labeled Alice/Bob. The leaves represent output. The left downwards edge represents sending the other party a 0 bit, the right downwards edge represents sending the other party a 1 bit.*

For the probabilistic version of the problem each internal node labeled Alice has a probability $p(i) \in [0, 1]$ *of sending the 0 bit dependent on the row index* i *asociated with it and each internal node labeled Bob has a probability* $p(j) \in [0, 1]$ *of sending the 0 bit dependent on the column index* j *asociated with it.*

Theorem. $\log(rk_+(M)) \sim cc_+(M)$

We leave this theorem without a proof for now. The proof will be presented in the next lecture. Instead, we present an example of usage of this theorem.

4 The spanning tree polytope

Let $G = (V, E)$ be a graph. We call the polytope $P_{ST}(G) = \{\chi^{E'} \in \{0, 1\}^{|E|} | E'$ is a spanning tree of $G\}$ the *spanning tree polytope* of G. Here is a description of the polytope using linear inequalities:

$$\left\{ x \;\middle|\; \begin{array}{cc} \sum_{e\in E} x_e = n-1 & \\ x_e \geq 0 & \forall e \in E \\ \sum_{e\in E[U]} \leq |U|-1 & \forall U \subseteq V \end{array} \right\}$$

The third system of inequalities basically says that each subset of vertices should induce a forest in the spanning tree E'.

How does the slack matrix of this polytope look like? It has one column for each possible spanning tree of G. The rows of the slack matrix corresponding to the first two systems of inequalities are trivial. For the first system we get all zeros. For the second system we get a row for each edge e where positions represent whether e is contained in the spanning tree corresponding to the column.

The part of the matrix corresponding to the third set of inequalities is more interesting. Rows correspond to subsets $U \subseteq V$. Here is a formula for elements of this part of the matrix:

$$S(U, T) = \left(|U| - 1 - \sum_{e\in E[U]} [e \in T] \right) - 1$$

This can be interpreted as (# components in $T[U]$) $- 1$.

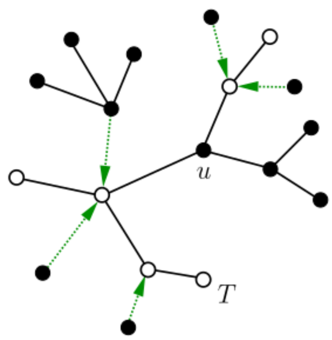
4.1 Communication protocol

reference: "basics" paper, section 5.2

Let's bound the extension complexity of the spanning tree polytope by constructing a communication protocol for its slack matrix.

Let $G = (V, E)$ be a graph. $P_{ST}(G)$ is its spanning tree polytope. In terms of the corresponding communication problem, Alice has a proper nonempty set $U \subsetneq V$ and Bob a spanning tree T. Together, they wish to compute $S(U, T)$.

Alice sends the name of some (arbitralily chosen) vertex u of U. Then Bob picks an edge e of T uniformly at random and sends to Alice the endpoints v and w of e as an ordered pair of vertices (v, w), where the order is chosen in such a way that w is on the unique path from v to u in the tree. That is, she makes sure that the directed edge (v, w) "points" towards the root u. Then Alice checks that $v \in U$ and $w \notin U$, in which case she outputs $n - 1$; otherwise she outputs 0.



The resulting randomized protocol is clearly of complexity $\log |V| + \log |E| + O(1)$. Moreover, it computes the slack matrix in expectation because for each connected component of $T[U]$ distinct from that which contains u, there is exactly one directed edge (v, w) that will lead Alice to output a non-zero value. Since she outputs $(n - 1)$ in this case, the expected value of the protocol on pair (U, T) is $(n - 1) \cdot (k - 1)/(n - 1) = k - 1$, where k is the number of connected components of $T[U]$. Therefore we obtain the following result.

Proposition. *For every graph G with n vertices and m edges, $xc(P_{ST}(G)) \in O(mn)$*

Polyhedral combinatorics – Lecture 12

Cyril Kotecký

14 June, 2024

version 1 - June 16, 2024

Definition: ***Randomized protocol***

- A, B finite sets assigned to Alice and Bob respectively

A **randomized protocol** is a rooted binary tree such that:

1. Each node of the tree has type A or B
2. Each node v of type A has functions $p_{0,v}, p_{1,v} : A \to [0,1]$ such that $p_{0,v}(i) + p_{1,v}(i) \leq 1$.
3. Each node v of type B has functions $q_{0,v}, q_{1,v} : B \to [0,1]$ such that $q_{0,v}(j) + q_{1,v}(j) \leq 1$.
4. Each leaf v of type A has a nonnegative vector Λ_v of size $|A|$.
5. Each leaf v of type B has a nonnegative vector Λ_v of size $|B|$.

- Nowhere is it mentioned that vertices of type A have vertices of type B as children and vice versa, but this seems implicit from the context.

An **execution** of the protocol on input $(i, j) \in A \times B$ is a random path from the root.

- It descends to the left child of an internal node v with probability $\begin{cases} p_{0,v}(i) & v \text{ is of type } A \\ q_{0,v}(j) & v \text{ is of type } B \end{cases}$ and to its right child with probability $\begin{cases} p_{1,v}(i) & v \text{ is of type } A \\ q_{1,v}(j) & v \text{ is of type } B \end{cases}$.
- The execution stops at v with probability $\begin{cases} 1 - p_{0,v}(i) - p_{1,v}(j) & v \text{ is of type } A \\ 1 - q_{0,v}(i) - q_{1,v}(j) & v \text{ is of type } B \end{cases}$.
- If an execution stops at a node v, the **value** of the execution is $\begin{cases} 0 & v \text{ is an internal node} \\ \Lambda_v(i) & v \text{ is a leaf of type } A \\ \Lambda_v(j) & v \text{ is a leaf of type } B \end{cases}$.
- For a fixed input $(i, j) \in A \times B$, the value of the execution is a random variable.
- Transitioning from a node A to the left or right child corresponds to Alice sending 0 or 1 respectively, and symmetrically for B.

The **communication complexity** of the protocol is the maximum number of bits exchanged over all (i, j), or equivalently, the height of the tree.

Problem: ***Computing a matrix in expectation***

- For a given matrix M and a protocol outputting X, the goal is to get $\mathbb{E}[X_{i,j}] = M_{i,j}$ for all entries.
- The **communication complexity** $cc_+(M)$ of the protocol is the maximum number of bits exchanged.

Theorem:

$$\lceil \log \mathrm{rk}_+(M) \rceil = \mathrm{cc}_+(M)$$

Proof.

$\leq$
- Assume we have a protocol computing X with $\mathbb{E}[X_{xy}] = M_{xy}$ with complexity c.
- Each node v of the protocol has a corresponding traversal probability matrix $P_v \in \mathbb{R}_+^{A \times B}$ with $\forall (x, y) \in A \times B : \; P_v(x, y) = P[\text{execution on input } (x, y) \text{ goes through } v]$. Let $v_1, ..., v_k$ be the nodes of type A on the unique path P from the root r to the parent of v. Let $w_1, ..., w_l$ be the nodes of type Y on this path. Then

$$P_v(x,y) = \prod_{i\in[k]} p_{\alpha_i,v_i}(x) \prod_{j\in[l]} q_{\beta_j,w_j}(y),$$

where $\alpha_i = \begin{cases} 1 & \text{path goes into right subtree of } v_i \\ 0 & \text{path goes into left subtree of } v_i \end{cases}$ and similarly for β_j. Hence P_v is a rank one matrix of the form $a_v b_v$ for a_v column vector of size $|A|$ and b_v row vector of size $|B|$.

  – Let L_X, L_Y be the sets of leaves of types A and B respectively. Let Λ_v denote the vector of values at a leaf $v \in L_X \cup L_Y$. Since the protocol computes $\mathbb{E}[X_{xy}] = M_{xy}$, we have

$$M(x,y) = \sum_{v\in L_X} \Lambda_v(x) P_v(x,y) + \sum_{w\in L_y} P_w(x,y)\Lambda_w(y).$$

  Thus

$$M = \sum_{v\in L_X} (\Lambda_v \circ a_v) b_v + \sum_{w\in L_Y} a_w (b_w \circ \Lambda_w)$$

  where $\circ$ denotes the hadamard (element-wise) product. Hence we can express M as a sum of at most $|L_X \cup L_Y| \leq 2^c$ nonnegative rank one matrices and therefore $rank_+(M) \leq 2^c$.

$\geq$
  – Denote $r := \mathrm{rk}_+(M)$ and let $A \in \mathbb{R}_+^{m\times r}$, $B \in \mathbb{R}_+^{r\times n}$ be such that $M = AB$. WLOG we can assume that the maximum row sum of A is 1, as we can rescale B appropriately.
  – Alice knows a row index i and Bob knows a column index j. Together, they want to compute $\mathbb{E}[X_{ij}] = M_{ij}$ by exchanging as few bits as possible.
  1. Alice selects a column index $k \in [r]$ according to the probabilities in row i of A. She sends this index to Bob.
  2. Bob outputs entry of B_{kj}.
  – With probability $1 - \delta_i$, where $\delta_i := \sum_k A_{ik} \leq 1$, Alice does not send any index to Bob and the computation stops with implicit output zero.
  – This randomized protocol computes X with $\mathbb{E}[X_{ij}] = M_{ij}$, since for the input (i,j), the expected value is $\sum_{k\in[r]} A_{ik}B_{kj} = M_{ij}$. Moreover, the complexity of the protocol is precisely $\lceil \log r \rceil$.

□

Corollary:

- $P \neq \emptyset$ polytope that is not a point
- S its associated slack matrix

$$\lceil \log \mathrm{xc}(P) \rceil = \mathrm{cc}_+(S)$$

Proof.

- Follows from the previous theorem and Yannakis' theorem.

□

1 Perfect matching polytope

Definition: ***Perfect matching polytope***

$$P_{PM} := \mathrm{conv}\{\chi^{E'} \mid E' \subseteq E \text{ perfect matching}\}$$

$$= \{x \in \mathbb{R}^{|E|} \mid \forall v \in V: \sum_{v\in e} x_e = 1,\ x_e \geq 0,\ \forall U \in V,\ |U| \text{ odd: } \sum_{e\in\delta(U)} x_e \geq 1\}$$

$$\text{or } \{x \in \mathbb{R}^{|E|} \mid \forall v \in V: \sum_{v\in e} x_e = 1,\ x_e \geq 0\} \text{ for bipartite graphs.}$$

Note: ***Perfect matching polytope***

- Complexity is exponential.
- Looking at a vertex and its neighbors, we have a polyhedral cone called the **vertex figure**.
- Vertex figures of perfect matching polytope have small extension complexity.

Lecture notes Mathematical Programming

Kateřina Vokálová

21. 5. 2024

1 Cut and correlation polytopes

Definition 1.1. A *polytope* $P \subseteq \mathbb{R}^n$ is the convex hull of a finite set of points in $\mathbb{R}^n$. It can also be viewed as a bounded set defined by a finite number of linear constraints (halfspaces in $\mathbb{R}$).

$$P = \text{conv}(\{v_1, v_2, \dots, v_k\})$$

$$P = \{x \in \mathbb{R}^n | Ax \leq b \text{ for } A \in \mathbb{R}^{r \times n}, b \in \mathbb{R}^r\}$$

Remark. The number of vertices defining a polytope may be exponential in the number of halfspaces. Consider the hypercube H_d in $\mathbb{R}^d$ which needs d inequalities $0 \leq x_i \leq 1$, but 2^d vertices ($H_d = \text{conv}(\{0,1\}^d)$).

Definition 1.2. For graph $G = (V, E)$ and cut $E' \subseteq E$ define its *incidence vector* $\chi^{E'}$ of size $|E|$ as

$$\chi_e^{E'} = \begin{cases} 1 & e \in E' \\ 0 & e \notin E' \end{cases}$$

and define the *cut polytope* as

$$\text{CUT}(G) := \text{conv}\{\chi^{E'} | \ E' \subseteq E \text{ is an edge cut}\}.$$

If G is the complete graph K_n, we simply denote $\text{CUT}(K_n)$ by CUT_n.

Definition 1.3. We define the *correlation polytope* as

$$\text{CORR}(n) = \text{conv}\{bb^T | \ b \in \{0,1\}^n\}$$

The polytope lies in $\mathbb{R}^{n^2}$. The feasible point of this polytope is a matrix $x \in \mathbb{R}^{n \times n}$.

2 Extension complexity and rectangle covering bound

Definition 2.1. The *extension complexity* of a polytope P, denoted by $\text{xc}(P)$, is the smallest number of facets of polytope $Q \subseteq \mathbb{R}^m$ such that P is a projection of Q.

Alternatively, the *extension complexity*, $\text{xc}(P)$, is the minimum number of inequalities required to describe Q, even when allowed to use auxiliary variables or extended formulations.

Definition 2.2. Let P be a polytope as defined in Definition 1.1. Then $S \in \mathbb{R}^{r \times k}$ defined as $S_{ij} := b_i - A_i v_j$ is the *slack matrix of* P w.r.t. $Ax \leq b$ and V.

Definition 2.3. We define the support matrix $\mathrm{supp}(S)$ for the slack matrix S as

$$\mathrm{supp}(S)_{i,j} = \begin{cases} 1 & S_{i,j} \neq 0 \\ 0 & S_{i,j} = 0 \end{cases}$$

Definition 2.4. A *rectangle* is the cartesian product of a set of row indices and a set of column indices. The *rectangle covering bound* is the minimum number of rectangles needed to cover all the 1-entries of $\mathrm{supp}(S)$.

Definition 2.5. *Monochromatic 1-rectangle* is a rectangle, which has all elements ones.

Theorem 2.1 (Yannakakis)**.** *Let M be any matrix with nonnegative real entries and $\mathrm{supp}(M)$ its support matrix. Then $\mathrm{rk}_+(M) \geq$ rectangle covering bound for $\mathrm{supp}(M)$.*

Theorem 2.2 (Yannakakis from the Lecture 11)**.** *For polytope P and its slack matrix S it holds*

$$\mathrm{xc}(P) = \mathrm{rk}_+(S).$$

3 $\mathbf{CORR}_{n-1} \approxeq \mathbf{CUT}_n$

Lemma 3.1. *For all $a \in \{0,1\}^n$, the inequality*

$$\left\langle 2\mathrm{diag}(a) - aa^T, x \right\rangle \leq 1$$

is valid for $x \in \mathrm{CORR}_n$.

Proof. We can show the inequality is satisfied for vertices $x = bb^T$ and by convexity, it is satisfied for every point of CORR_n. The inequality can be rewritten as

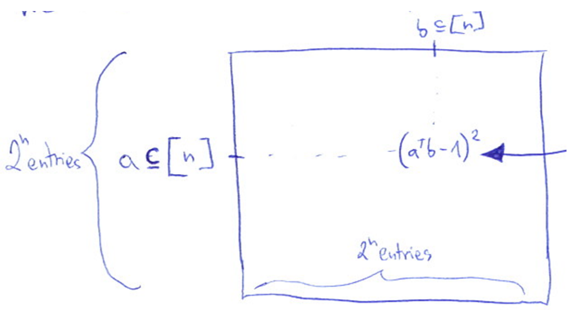
$$(1 - a^T b)^2 \geq 0$$

which trivially holds. □

Definition 3.1. The slack matrix of the CORR_n is $2^n \times 2^n$ matrix $M^* = M^*(n)$

$$M^*_{a,b} = (a^T b - 1)^2.$$

Each row a represents a subset of $[n]$ (can be viewed as n-bit strings), the same applies to the column b.



Theorem 3.2. *For a slack matrix M^* of a correlation polytope CORR_n it holds that every 1-monochromatic rectangle cover of $\mathrm{supp}(M^*)$ has size $2^{\Omega(n)}$.*

Corollary 1.

$$\text{rk}_+(M^*) \geq 2^{\Omega(n)}$$

Corollary 2.

$$\text{xc}(\text{CORR}_n) = 2^{\Omega(n)}$$

We have somewhat an equivalence between polytopes CUT_n and CORR_n formulated in the following theorem:

Theorem 3.3. *Let $M(n)$ denote the slack matrix of CUT_n, extended with a suitably chosen set of 2^n redundant inequalities. Then $M^*(n-1)$ occurs as a submatrix of $M(n)$ and hence*

$$\text{xc}(\text{CUT}_n) = 2^{\Omega(n)}.$$

The theorem can be also formulated as:

Lemma 3.4. *For every $n \geq 1$, the polytopes CUT_{n+1} and CORR_n are affine-equivalent.*

4 Lower bound on $\text{xc}(\text{CORR}_n)$

Definition 4.1. Define a *rectangle of the slack matrix M^**

$$D(n) = \{(a,b), a \subseteq [n], b \subseteq [n] |\ a \cap b = \emptyset\}.$$

There are only positions where a and b are different, so $a^T b = 0$ and $(a^T b - 1)^2 = (0-1)^2 = 1$. It is a monochromatic 1-rectangle.

Proposition 4.1.

$$|D(n)| = 3^n$$

Proof. We have n vertices and the vertex can have 3 states. Either it is in a, or it is in b, or it is neither in a nor b. □

Proposition 4.2 (Decomposition of D(n) into rectangles)**.** *Let us have the non-negative rank of matrix M^* $\text{rk}_+(M^*) = k$. We can find $T \in \mathbb{R}^{2^n \times k}$ and $U \in \mathbb{R}^{k \times 2^n}$ such that $M^* = TU$. We will furtherly decompose M^* as*

$$M^* = T_1 U^1 + T_2 U^2 + \ldots + T_k U^k$$

with $T_i \in \mathbb{R}^{2^n \times 1}$ and $U_i \in \mathbb{R}^{1 \times 2^n}$.

For support matrices, it holds that

$$supp(M^*) = \bigcup_{i=1}^{k} supp(T_i U^i) = \bigcup_{i=1}^{k} supp(T_i) \times supp(U^i)$$

Moreover, if the all the matrices T_i, U^i are nonzero, we can define a rectangle $R_i = supp(T_i) \times supp(U^i)$ for each $T_i U^i$. These rectangles together cover the whole matrix $D(n)$

$$D(n) \subseteq \bigcup_{i=1}^{k} R_i.$$

Definition 4.2. A set $R \subseteq D(n)$ is *valid* if $\forall (a,b),(a',b') \in R : |a \cap b'| \neq 1$. For a valid R define two sets R_1, R_2 as following

$$R_1 := \{(a,b) \in R \text{ such that } n \in a, n \notin b\} \cup \{(a,b) \in R \text{ such that } (a \cup \{n\}, b) \notin R, n \notin b\}$$

$$R_2 := \{(a,b) \in R \text{ such that } n \notin a, n \in b\} \cup \{(a,b) \in R \text{ such that } (a, b \cup \{n\}) \notin R, n \notin a\}.$$

Lemma 4.3.

$$(a,b) \in R \Rightarrow (a,b) \in R_1 \cup R_2$$

Proof. Assume $a \cap b = \emptyset$ and we will split the proof into 3 cases: $n \in a$, $n \notin a, n \in b$, $n \notin a$ and $n \notin b$.

At first, $n \in a$ implies $n \notin b$, so $(a,b) \in R_1$.

Secondly, $n \notin a, n \in b$ implies $(a,b) \in R_2$.

Thirdly, we will split the case $n \notin a, n \notin b$ into subcases.

- $(a \cup \{n\}, b) \notin R \Rightarrow (a,b) \in R_1$
- $(a, b \cup \{n\}) \notin R \Rightarrow (a,b) \in R_2$
- $(a \cup \{n\}, b) \in R$ and $(a, b \cup \{n\}) \in R \Rightarrow$ we have item at row $a \cup \{n\}$ and column b and another item at row a and column $b \cup \{n\}$, so we must have the item at row $a \cup \{n\}$ and column $b \cup \{n\}$ since it is rectangle. This implies $(a \cup \{n\}, b \cup \{n\}) \in R$ which is contradiction with the assumption that they are disjoint.

□

Lemma 4.4. *Let Q be a polyhedron having f facets such that* CORR_n *is an affine image of Q. Then there exists a covering of $D(n)$ of size f.*

Proof. By Definition 2.1 of the extension complexity and Theorem 2.2,

$$f = \mathrm{xc}(\mathrm{CORR}_n) = \mathrm{rk}_+(M^*).$$

In Proposition 4.2, we have found a rectangle covering of size $\mathrm{rk}_+(M^*)$, so it is a rectangle covering of size f. □

Theorem 4.5.

$$\mathrm{xc}(\mathrm{CORR}_n) \geq \left(\frac{3}{2}\right)^n$$

Proof. Let $\rho(n)$ be the largest cardinality of any valid subset of $D(n)$. Any covering of $D(n)$ must have size $\geq \frac{|D(n)|}{\rho(n)} = \frac{3^n}{\rho(n)}$. It remains to prove that $\rho(n) \leq 2^n$, which we will show by proving that $\rho(n) \leq 2\rho(n-1)$ holds for all $n \geq 1$. We will prove this by induction on n.

Let R be a valid subset with sets R_1, R_2. Define the function $f : R \to D(n-1)$ as

$$f((a,b)) := (a \setminus \{n\}, b \setminus \{n\})$$

$$f(R_1) = \{(a \setminus \{n\}, b \setminus \{n\}) | (a,b) \in R_1\}$$
$$f(R_2) = \{(a \setminus \{n\}, b \setminus \{n\}) | (a,b) \in R_2\}$$

The two parts of each R_i are disjoint when n is subtracted, so it holds that $|R_i| = |f(R_i)|$.

Then by Lemma 4.3 and using induction hypothesis, we have

$$|R| \leq |R_1| + |R_2| = |f(R_1)| + |f(R_2)| \leq \rho(n-1) + \rho(n-1) = 2\rho(n-1)$$

□

Corollary 3.

$$\mathrm{xc}(\mathrm{CUT}_n) \geq \left(\frac{3}{2}\right)^n$$

5 Under construction

Let P be polytope (by default let it be 0/1-polytope so all vertices lie on $\{0,1\}^d$), S is slack matrix of P.

Definition 5.1. Let C be a cut. Define $1,-1$ encoding C' for each edge e:

$$C'_e := \begin{cases} -1 & e \in C \\ +1 & e \notin C \end{cases}$$

Proposition 5.1. *The* $1,-1$ *encoding can be obtained from* $0,1$ *encoding by linear transformations.*

Proof. $0,1 \rightsquigarrow -\frac{1}{2}, \frac{1}{2} \rightsquigarrow -1, 1 \rightsquigarrow 1, -1$ □

Definition 5.2. Denote by $U \subseteq V$ one part of vertices of the graph using the cut C. Then define χ_U as following

$$(\chi_U)_v = \begin{cases} +1 & v \in U \\ -1 & v \notin U \end{cases}$$

Then we have $C' = \chi_U \chi_U^T$

Have the graph K_n with vertices numbered from 1 to n (so the numbers are from set $[n]$) and $x_{ij} := x_i x_j$ corresponding to the edge between vertices i and j.

$$(w^T x - 1)^2 = \left(\sum_{i \in [n]} w_i x_i - 1\right)^2 \geq 0$$

Have the quadratic polynomial with variable vector x

$$\left(w^T x - 1\right)^2 = \left(\sum_{i \in [n]} w_i x_i - 1\right)^2 = \left(\sum_i w_i x_i\right)^2 - 2\sum_{i \in [n]} w_i x_i + 1 = \sum_{i,j} w_i w_j x_i x_j - 2\sum_i w_i x_i + 1 =$$

$$= \sum_{i,j} w_i w_j x_{ij} - 2\sum_i w_i x_i + 1$$