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)**.**

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

$$\Longleftrightarrow$$

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

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.

$\square$

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

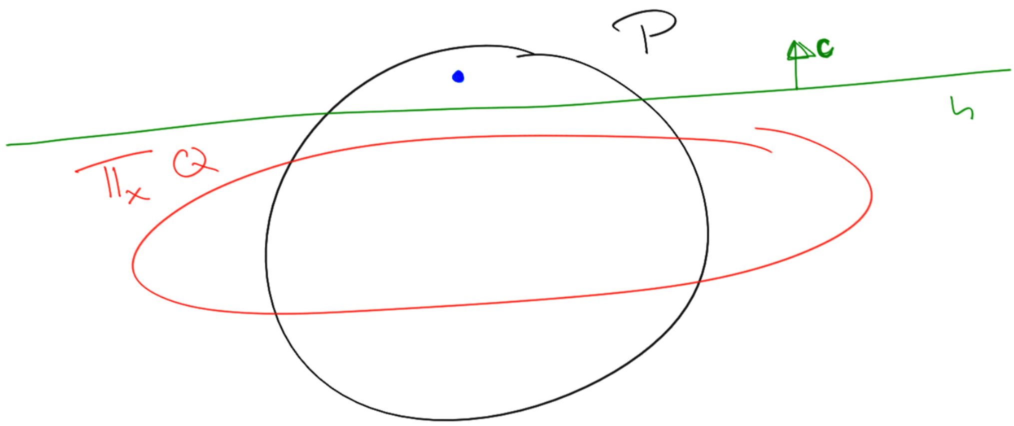

Figure 1: Difference in optimal value when optimizing along the direction perpendicular to the separating hyperplane.

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).

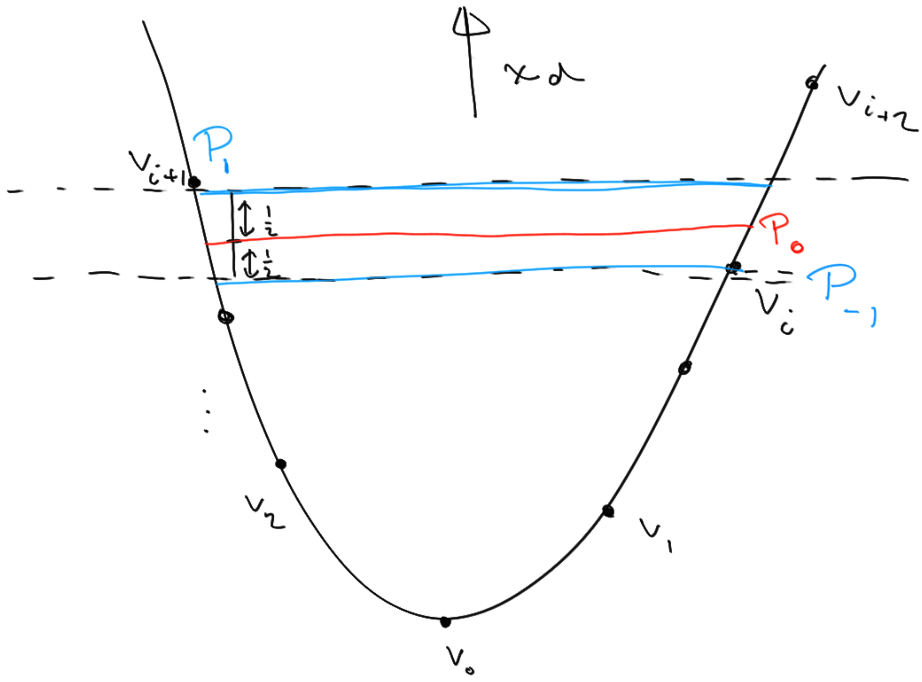

Figure 2: Polyhedron P oriented along the x_d dimension and cuts represented by polyhedra P_{-1}, P_0, P_1

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.

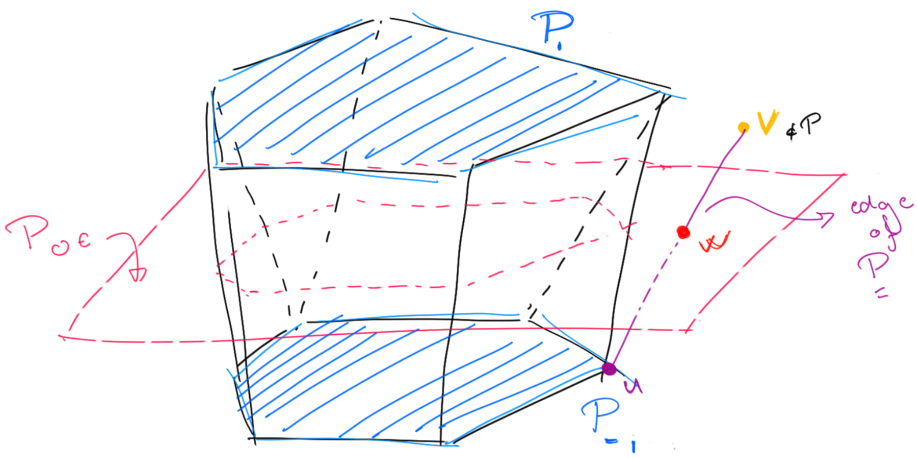

Figure 3: Situation in lemma

□

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.

□