

Lemma: $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $U =$ largest absolute value of entries in A and b . $P = \{x \mid Ax \geq b\}$

$$\varepsilon := \frac{1}{2^{(n+1)}} ((n+1)U)^{-(n+1)}$$

$$P_\varepsilon := \{x \mid Ax \geq b - \varepsilon \cdot \mathbb{1}\}$$

(i) P empty $\Rightarrow P_\varepsilon$ empty

(ii) $P \neq \emptyset \Rightarrow P_\varepsilon$ full-dimensional.

Proof: (i) P empty

$\Rightarrow \min c^T x$ infeasible
s.t. $Ax \geq b$

$\Rightarrow \max p^T b$ unbounded
s.t. $p^T A = c^T$
 $p \geq 0$

$\Rightarrow \exists p \geq 0 : p^T A = c^T, p^T b = 1$

By the last lemma, there is an extreme point

\hat{p} to the system

$$\begin{aligned} p^T A &= c^T \\ p^T b &= 1 \\ p &\geq 0 \end{aligned}$$

such that

$$\hat{p}_i \leq ((n+1) \cdot U)^{n+1} \quad \forall i$$

and at most $n+1$ components of \hat{p} are non-zero.

$$\Rightarrow \sum_{i=1}^m \hat{p}_i \leq (n+1) \cdot ((n+1) \cdot U)^{n+1}$$

$$\Rightarrow \hat{p}^T \cdot (\delta - \varepsilon \cdot \mathbb{1}) = 1 - \varepsilon \cdot \sum_{i=1}^n \hat{p}_i$$

$$\geq 1 - \frac{1}{2} = \frac{1}{2} > 0$$

$$\Rightarrow \max p^T \cdot (\delta - \varepsilon \cdot \mathbb{1}) \text{ unbounded}$$

$$\text{s.t. } p^T A = 0^T$$

$$p \geq 0$$

$$\Rightarrow \min 0^T x \text{ infeasible.}$$

$$\text{s.t. } Ax \geq \delta - \varepsilon \cdot \mathbb{1}$$

$$(ii) \text{ Let } x \in \mathbb{R}^n : Ax \geq \delta \quad (x \in P)$$

$$\Rightarrow A \cdot y \geq \delta - \varepsilon \cdot \mathbb{1} \quad \forall y : |y_j - x_j| \leq \frac{\varepsilon}{n \cdot U} \quad \forall j$$

$\Rightarrow P_\varepsilon$ contains small cube of positive volume. \square

Lemma: If $P = \{x \mid Ax \geq \delta\}$ is full-dimensional and bounded with U defined as above then

$$\text{Vol}(P) > n^{-n} \cdot (n \cdot U)^{-n^2 \cdot (n+1)}$$

Sketch of proof: P has $n+1$ extreme

points v^0, \dots, v^n such that

$$\text{Vol}(\text{conv}(v^0, \dots, v^n)) > n^{-n} \cdot (n \cdot U)^{-n^2 \cdot (n+1)}$$

\square

Theorem: The number of iterations of the ellipsoid method can be bounded by $O(n^6 \log(n \cdot U))$.

Major problem: Bound the number of arithmetic operations in an iteration of the ellipsoid method. How to take square roots? \rightarrow only finite precision possible.

Theorem: Using only $O(n^3 \log U)$ binary digits of precision, the ellipsoid method still correctly decides whether P is empty in $O(n^6 \cdot \log(n \cdot U))$ iterations. In particular, linear programming feasibility can be decided in polynomial time.

8.4 The ellipsoid method for optimization

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

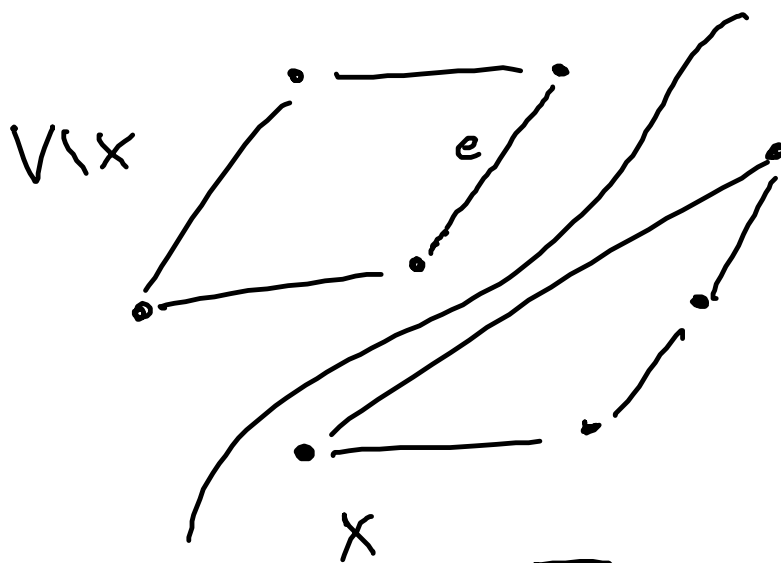
$$\begin{aligned} \max \quad & p^T b \\ \text{s.t.} \quad & p^T A \leq c^T \\ & p \geq 0 \end{aligned}$$

Solve the primal and the dual LP by finding a point (\bar{x}, \bar{p}) in the polyhedron given by:

$$\begin{aligned} c^T x &= p^T b \\ Ax &\geq b \\ p^T A &\leq c^T \\ x, p &\geq 0 \end{aligned}$$

Theorem: Linear programs can be solved in polynomial time.

8.5 Optimization is equivalent to separation



$$\begin{aligned} \min \sum_{e \in E} c_e \cdot x_e \\ \text{s.t. } \sum_{e \in S(v)} x_e &= 2 \\ &\forall v \in V \\ x_e &\in \{0, 1\} \end{aligned}$$

$$\sum_{e \in S(x)} x_e \geq 2 \quad \forall \phi \neq x \notin V$$

Notice that the number of iterations of the ellipsoid method only depends on the dimension n and on U but not on

the number of constraints m . Thus there is hope to solve linear programs with exponentially many constraints in polynomial time. These LPs are given implicitly.

Describe polyhedron $P = \{x \mid Ax \geq b\}$ by specifying n and an integer vector h of "primary data" of dimension $O(n^k)$ for some constant k . Let $U_0 := \max_i |h_i|$.

There is a mapping which, given n and h , defines an integer matrix A with n columns and an integer vector b .

Let $U := \max \{|a_{ij}|, |b_i| \mid i = \dots, j = \dots\}$.

We assume that

$$\log U \leq C \cdot n^l \cdot \log^l U_0$$

for constants C and l .

The number of iterations of the ellipsoid method is

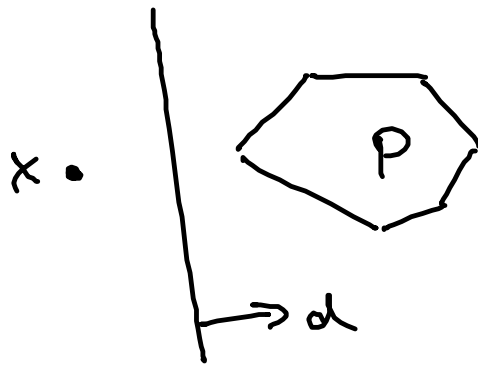
$$O(n^6 \log(n \cdot U)) = O(n^6 \cdot \log n + n^{6+l} \log^l U_0)$$

and thus polynomial in the size of the primary data.

It remains to analyze a single iteration of the ellipsoid method.

Def: Given polyhedron $P \subseteq \mathbb{R}^n$ and $x \in \mathbb{R}^n$, the separation problem is to

- (i) either decide that $x \in P$, or
- (ii) find a vector $d \in \mathbb{R}^n$ such that $d^T x < d^T y \quad \forall y \in P$.



Theorem: If we can solve the separation problem for a family of polyhedra in time polynomial in n and $\log U$, then we can also solve LPs over those polyhedra in time polynomial in n and $\log U$. The converse is also true under some technical assumptions.

Optimization is as hard/easy
as separation.