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

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

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

(i) P empty $\Rightarrow$ P_ε empty

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

Proof: (i) P empty

$$\Rightarrow \begin{array}{ll} \min & 0^T \cdot x \\ \text{s.t.} & Ax \geq b \end{array} \quad \text{infeasible}$$

$$\Rightarrow \begin{array}{ll} \max & p^T \cdot b \\ \text{s.t.} & p^T A = 0^T \\ & p \geq 0 \end{array} \quad \text{unbounded}$$

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

By the last lemma, there is an extreme point $\hat{p}$ to the system

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

such that

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

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

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

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

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

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

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

$$p \geq 0$$

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

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

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

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

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

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

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

<u>Sketch of proof:</u> P has $n+1$ extreme points $v^0, \dots, v^n$ such that

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

$\square$

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? $\to$ 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 $O(n^6 \cdot \log(n U))$ iterations. In particular, linear programming feasibility can be decided in polynomial time.

8.4 The ellipsoid method for optimization

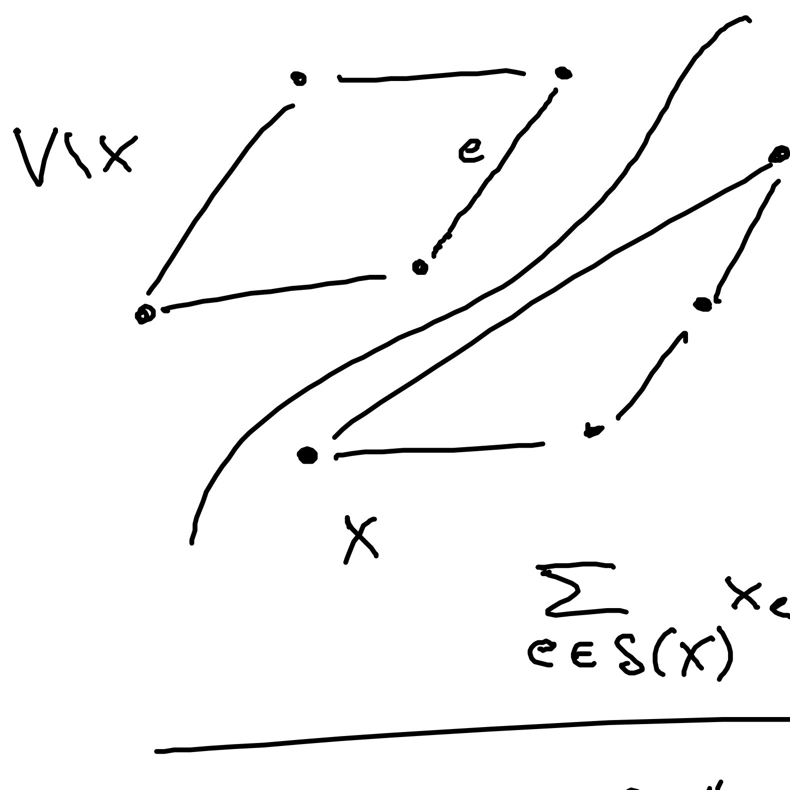
$$\begin{array}{llll} \min & c^T x & \qquad \max & p^T b \\ \text{s.t.} & Ax \geq b & \qquad \text{s.t.} & p^T A \leq c^T \\ & x \geq 0 & & p \geq 0 \end{array}$$

Solve the primal and the dual LP by finding a point $\binom{x}{p}$ in the polyhedron given by:

$$c^T x = p^T b$$
$$A x \geq b$$
$$p^T A \leq c^T$$
$$x, p \geq 0$$

Theorem: Linear programs can be solved in polynomial time.

8.5 Optimization is equivalent to separation



$$\min \sum_{e \in E} c_e \cdot x_e$$
$$\text{s.t. } \sum_{e \in \delta(v)} x_e = 2 \quad \forall v \in V$$
$$x_e \in \{0, 1\}$$

$$\sum_{e \in \delta(X)} x_e \geq 2 \quad \forall \emptyset \neq X \subsetneq V$$

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. Those LPs are given implicitely.

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

$$\log U \leq C \cdot n^{\ell} \cdot \log^{\ell} U_0$$

for constants C and ℓ.

The number of iterations of the ellipsoid method is

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

and thus polynomial in the size of the primary data.

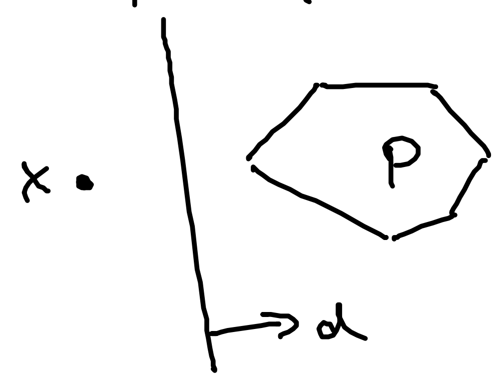
It remains to analyze a single iteration of the ellipsoid method.

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

(i) either decide that $x \in P$, or

(ii) find a vector $d \in \mathbb{R}^n$ such that

$$d^T x < d^T y \quad \forall y \in P.$$



<u>Theorem</u>: 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.