Linear Programming

Constraints in $\mathbb{R}^d$

point in $(x_1, \dots, x_d) \in \mathbb{R}^d$

express a constraint as $a_1x_1 + a_2x_2 + \dots + a_dx_d \le b$

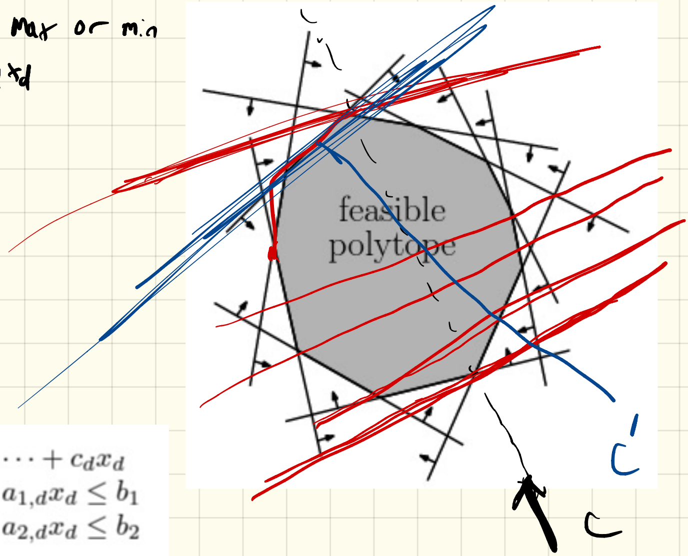
Feasible region: all points in $\mathbb{R}^d$ satisfying all constraints

Objective function:

linear function to max or min

$c_1x_1 + c_2x_2 + \dots + c_dx_d$

$c = \begin{bmatrix} c_1 \\ \vdots \\ c_d \end{bmatrix}$ $x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$



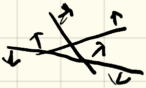
Maximize: $c_1x_1 + c_2x_2 + \dots + c_dx_d$

Subject to:
$$a_{1,1}x_1 + \dots + a_{1,d}x_d \le b_1$$
$$a_{2,1}x_1 + \dots + a_{2,d}x_d \le b_2$$
$$\vdots$$
$$a_{n,1}x_1 + \dots + a_{n,d}x_d \le b_n,$$

Maximize c^Tx w/ $c \in \mathbb{R}^d$

Subject to $Ax \le b$ — A is an $n \times d$ real valued matrix

3 outcomes:

- Feasible: opt vertex (and general pos) is unique vertex of feasible polytope
- Infeasible: no point satisfys contraints 
- Unbounded:  ⇒ no finite op soln exists

construct polytope:

$\Omega(n^{\lfloor \frac{d}{2} \rfloor})$ faces $\Rightarrow$ constructing all of feasible polytope is bad idea

Solve an LP for const dim
- # of x_i's is const but n can be large

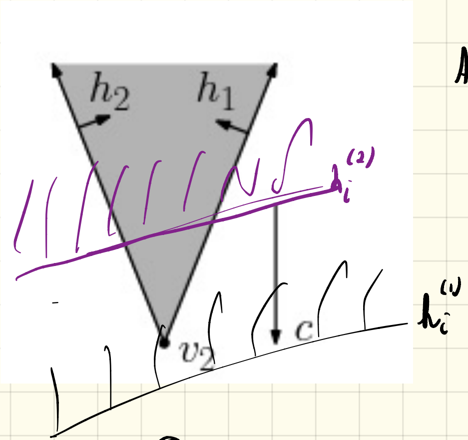
... assume that the 1st d halfspaces provide an initial feasible point $v_d = \bigcap_{j=1}^{d} h_j$

add halfspaces $h_{d+1}, h_{d+2}, \dots$

Add h_i:

case 1: v_{i-1} is in halfspace of h_i
$\Rightarrow v_i = v_{i-1}$

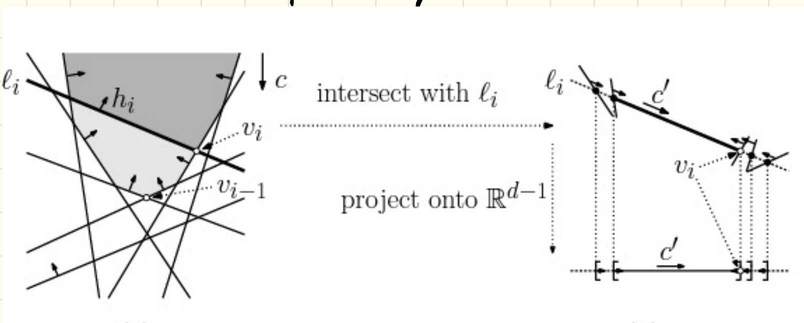
case 2: v_i violates constraint h_i



Case I

Lemma (*): if insert h_i and LP is still feasible and vertex changes
$\Rightarrow v_i$ is on boundry w/ h_i
(pf by contradiction)

Lemma * $\Rightarrow$ we can try recursing in dim



(a) (b)

let ℓ_i hyperplane bounding h_i $\quad h_i: a_1^{(i)} x_1 + \cdots + a_d^{(i)} x_d \le b^{(i)}$

let c' be the projection of c onto ℓ_i $\quad \ell_i: a_1^{(i)} x_1 + \cdots + a_d^{(i)} x_d = b$

intersect $\{h_1, \ldots, h_{i-1}\}$ w/ ℓ_i

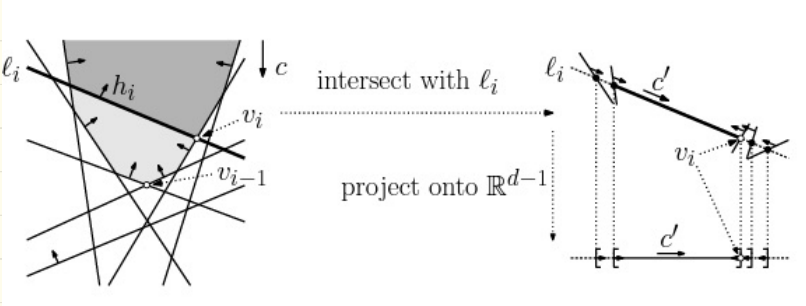
produce $(i-1)$ $(d-1)$-dim halfspaces all lie on ℓ_i

project onto $\mathbb{R}^{d-1}$

$\Rightarrow$ we solve LP in $d-1$ dim w/ $i-1$ constraints

recursively solve LP to get a point on ℓ_i

lift back to $\mathbb{R}^d$



(a) (b)

Recursion termination

* LP in 1 dim
  - obj function is a direction (meaning +/−)
  - constraints are rays

How long to compute? $O(i)$-time!

Worst Case analysis:

- n-d halfspace intersections
- consider step i
  - v_{i-1} is in feasible region $O(d)$ - time ✓
  - v_{i-1} violates $h_i \rightarrow d-1$ dim LP w/ $i-1$ constraints
    $O(d(i-1))$ time to intersect each constraint w/ ℓ_i
    $O(d)$ time to project c onto ℓ_i

Lets write as recurrence:

Let $T_d(n)$ be the time to solve for n constraints in dim d

$\Rightarrow T_d(i) = O(di + T_{d-1}(i-1))$

ignor consts write as:

$$T_d(n) = \sum_{i=d+1}^{n} \max(d, di + T_{d-1}(i-1))$$

w/ d const

$$* \begin{cases} T_d(n) = \sum_{i=d+1}^{n} (i + T_{d-1}(i-1)) \\ \text{base case for recurrence is } T_1(n) = n \end{cases}$$

interval intersection

Recurren $*$ $T_d(n) = O(n^d)$ = :(