Recap

- HW due
- Practice final here
- Final : Monday Dec 16 at 8am
- Practice session: Friday (in 1 week)

9am: 4
10am: 1
11 am: 3
Noon: > 5
1pm: > 5

Q: Is everything an LP?

No!

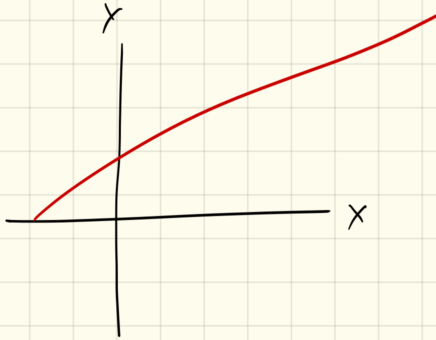
Some things are quadratic!

- Least squares
- Minimize area of some volume/surface

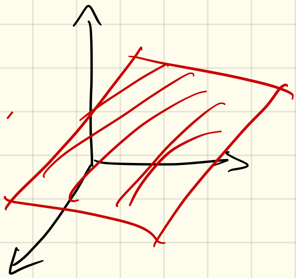
LP w/ d variables:

Each LP equality or inequality describes a hyperplane in $\mathbb{R}^d$.

2d: $ax+by \quad c$



3d: $ax+by+cz \le d$

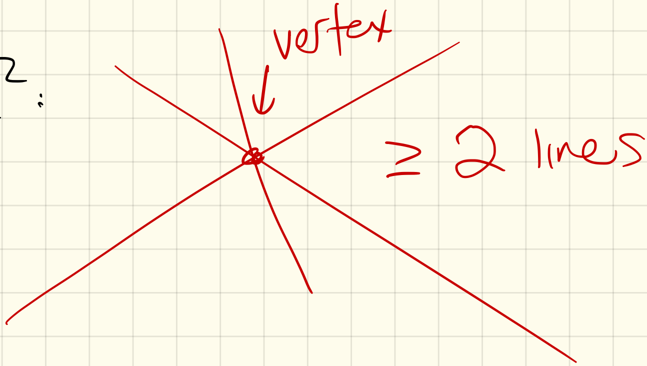


$\mathbb{R}^d$: $a_1x_1 + \cdots + c_d x_d \le c$ (d-1) dim subspace

Vertices:

These happen when $\geq d$ hyperplanes meet in $\mathbb{R}^d$.

In $\mathbb{R}^2$:



$\mathbb{R}^d$: d hyperplanes (here - eqns)

In $\mathbb{R}^3$:

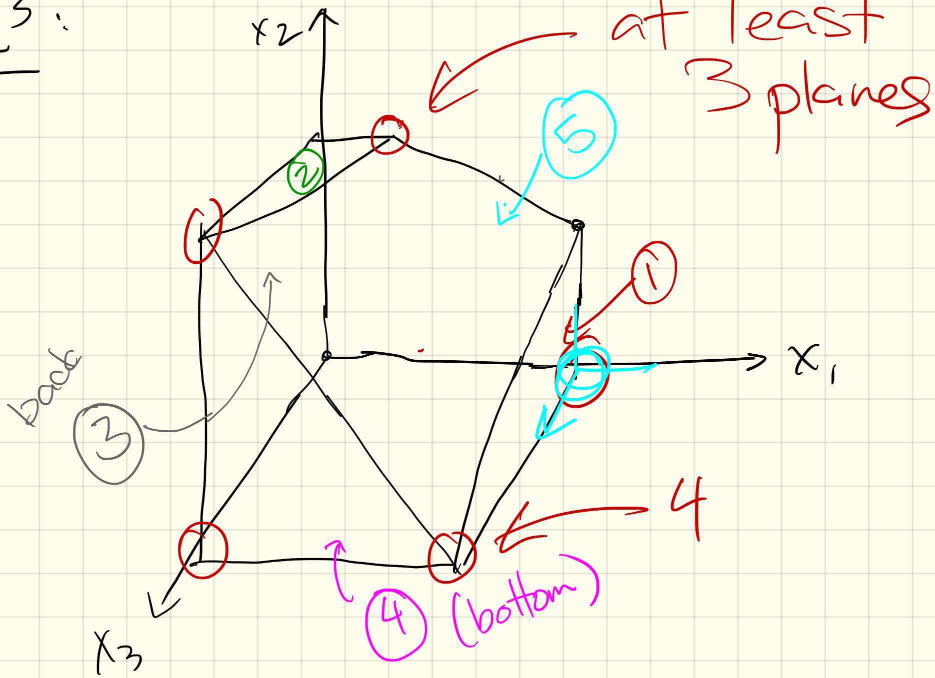
Maximize $x_1 + 6x_2 + 13x_3$
s.t.

$x_1 \le 200$ ①
$x_2 \le 300$ ②
$x_1 + x_2 + x_3 \le 400$
$x_2 + 3x_3 \le 600$

and

$x_1 \ge 0$ ③
$x_2 \ge 0$ ④
$x_3 \ge 0$ ⑤

$\mathbb{R}^3$:



Dfn: Pick a subset of inequalities.
If there is a unique point that satisfies all with equality, & it is feasible

↳ this is a <u>vertex</u> of the solution.

In general: Each vertex is specified by exactly d equations (in $\mathbb{R}^d$)

(Again, think 2 + 3d examples)

Neighbors:
Any vertices that share d-1 inequalities

Simplex algorithm:

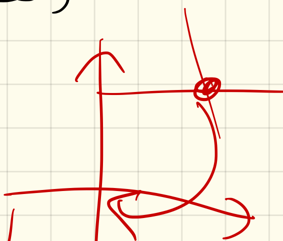
In each stage, 2 tasks:

① Check if current vertex is optimal

② If not, choose a nbr vertex that improves the result

Both are easy at the origin (next slide).

If not at $\vec{0}$:



~~rescale~~: translate slide so vertex is at 0.

$(v_1, v_2, v_3, \ldots v_d)$:

subtract $v_1, \ldots v_d$ from my eqns.

$(x_1 - v_1)$

LP: max $c^T x$

s.t. $A\vec{x} \le \vec{b}$

$x_i \geq 0 \quad \forall i$ ← d eqns

Note: $\vec{x} \in \mathbb{R}^d$, so

$x = (x_1, \dots, x_d)$

Start w/ origin, so

our $\vec{x} = \vec{0}$

d eqns give 1 vertex

It is always a vertex!

(Why?)

optimal only if:

max $c_1 x_1 + c_2 x_2 + \cdots + c_d x_d$

all c_i's are neg.

Conversely:

If any $c_i > 0$, we can increase the obj. function $c^T \vec{x}$.

How? just move that x_i above 0

So: pick one & increase!

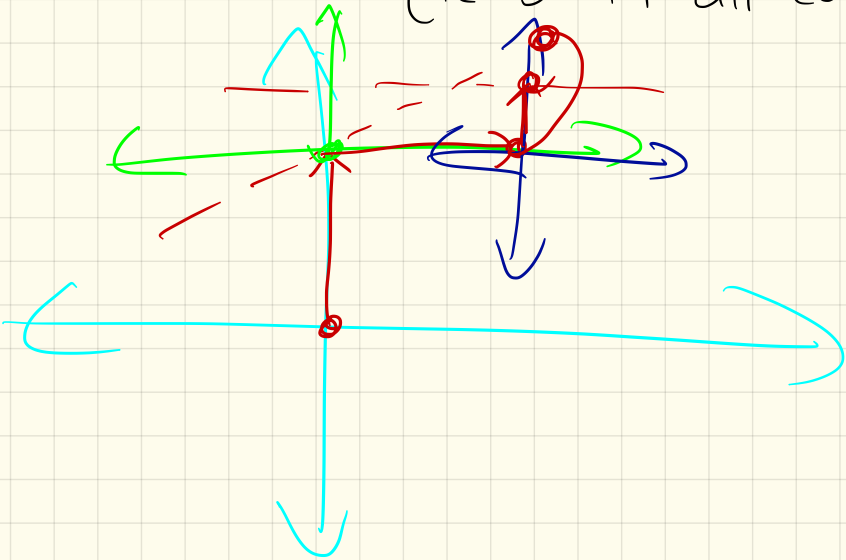
How much?

Until I get stuck (on some inequality) from $Ax \geq b$

Now: What if not at origin?

Transform LP!

(ie shift all coords)

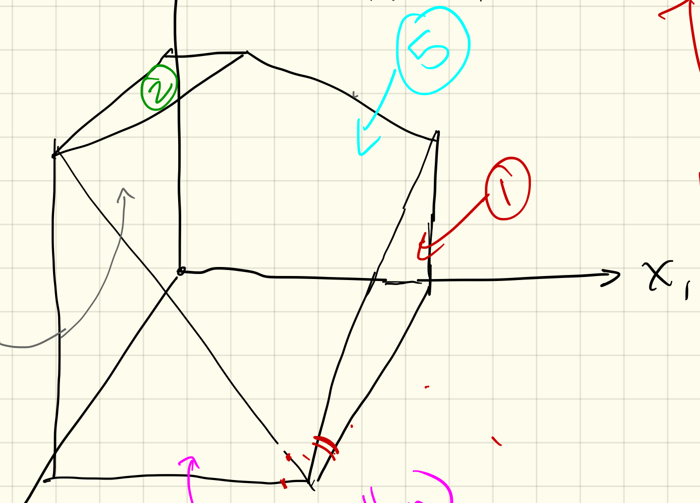


Some details

- Origin isn't always feasible,
  ↳ so must find a starting feasible point.
  (& reset to be $\vec{0}$)

  Turns out, this is a (simpler) LP!
  (see notes)

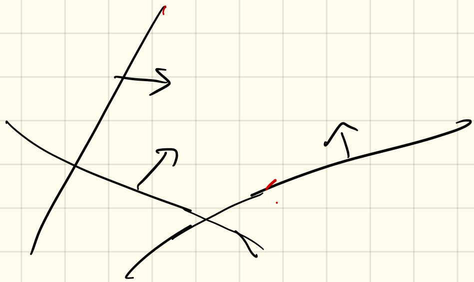
- Degeneracy: Can have $>d$ hyperplanes at a vertex!



doesn't matter: & still give me vertex

- Unboundedness:

Can have unbounded situation:



Detection:

When exploring for next vertex, swapping out an equality for another will not give a bound.

↳ Simplex stops + complains

$\max\ c_1x_1 + \dots + c_dx_d$
s.t. $A\vec{x} \geq \vec{b}$

Runtime:

Consider a vertex $u \in \mathbb{R}^{\cancel{n}\,d}$,

with m inequalities.

At most $d \cdot (m-d)$ nbrs:

choose one to drop & any d one to add:

make a vertex. drop one of yours, + pick a new one

Checking for nbr:

Each is a dot product/ matrix operation.

Gaussian elimination: $O(\cancel{n^3})$ m^3 or d^3

(basically)

$\Rightarrow$ Each iteration:

$$d \cdot (m) \cdot m^3$$

Can improve slightly:

- just need one $c_i > 0$
+ rescaling to $\vec{0}$ is easy.

$\Rightarrow$ Can improve to $O(mn)$ per iteration.

How many iterations?

- $m+d$ inequalities
- any d give a vertex

$$\Rightarrow \binom{m+d}{d} \approx (m+d)^d$$

Ick! Klee-Minty give examples ~~that~~ are actually this slow.
(in 50's)

Alternatives

- Ellipsoid algorithm
  (Khachiyan '79)
- Interior point method
  (Karmarkar in '80's)

Runtime: slow

But:

Simplex works better

Now:

Evals!

Take out an internet device + do them!