CS314 - Simplex Algorithm

Note Title 11/22/2013

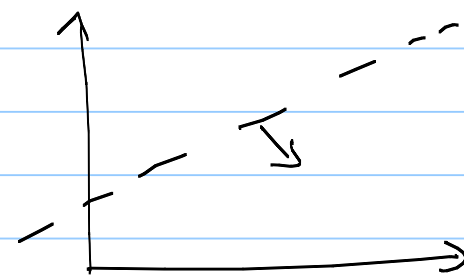
Announcements

- Next HW is up
  (oral or written)
  due Tuesday after break

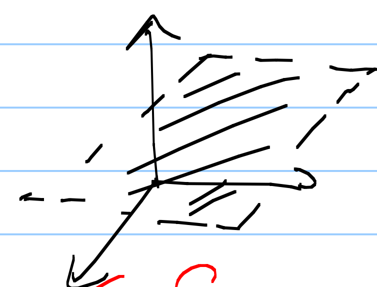
Higher dimensions

Each LP equality or inequality describes a hyper plane.

2D: $ax + by \le c$



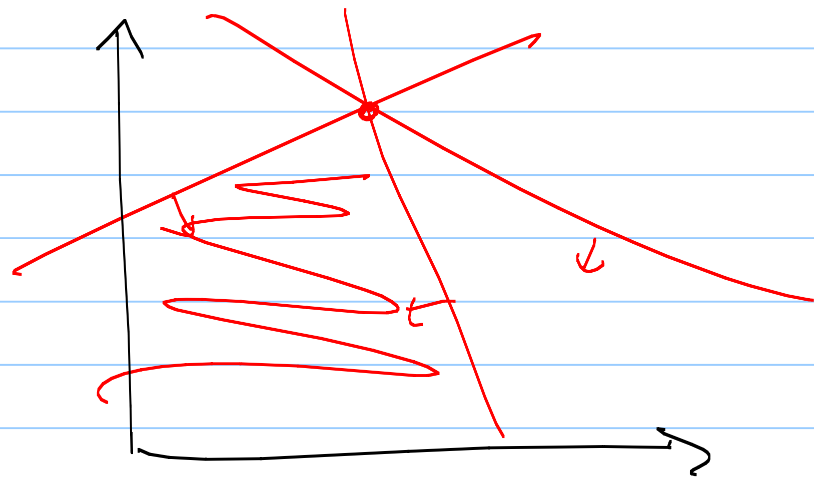
3-D: $ax + by + cz \le d$



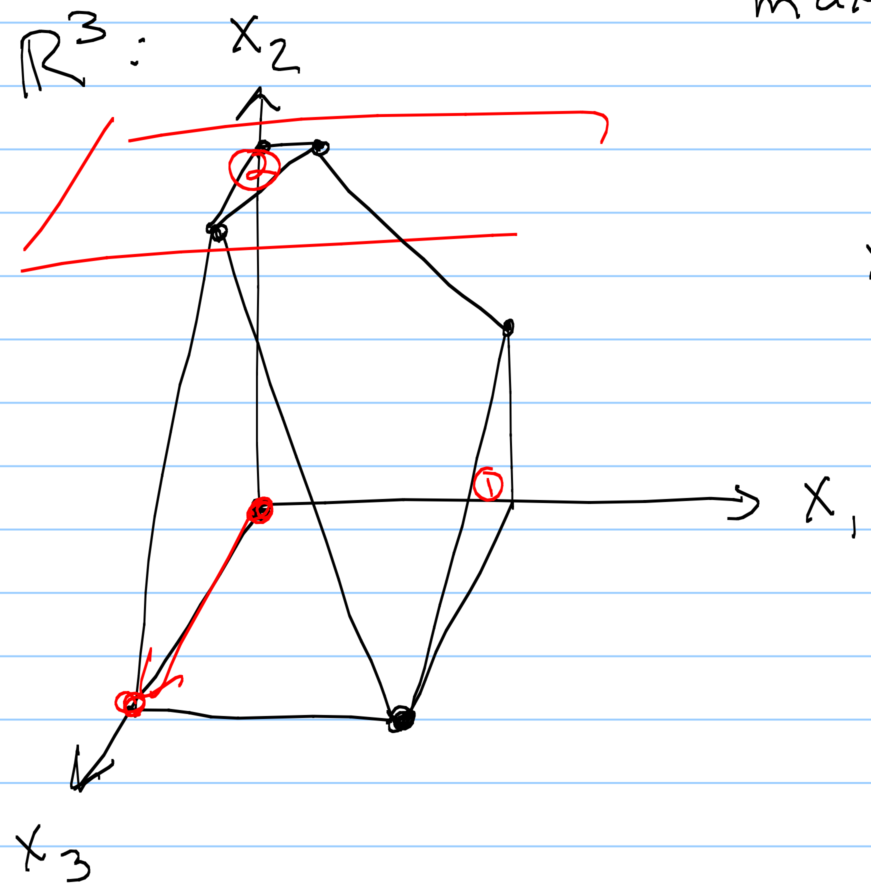
$\mathbb{R}^d$: $c_1x_1 + c_2x_2 + \cdots + c_dx_d \le c$

Vertices occur when $\geq d$ hyperplanes meet in $\mathbb{R}^d$:

In $\mathbb{R}^2$:



In $\mathbb{R}^3$:



maximize $x_1 + 6x_2 + 13x_3$

$x_1 \leq 200$ 1

$x_2 \leq 300$ 2

$x_1 + x_2 + x_3 \leq 400$

$x_2 + 3x_3 \leq 600$

$$\begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$$

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

In general:
- each vertex is specified by d inequalities (in $\mathbb{R}^d$)

Neighbors: a pair of vertices which share $d-1$ inequalities

Simplex

In each stage, two tasks:

1. Check if current vertex is optimal (+ halt, if so).
2. Find what vertex to move to next.

Both are easy at the origin (see next slide).

And if not at the origin, transform the coordinate system to move it to the origin!

$c = (c_1, \ldots, c_d)$

LP: $\max\ c^T x = c_1 x_1 + c_2 x_2 + \cdots + c_d x_d = 0$

s.t.

$$A\vec{x} \le b$$
$$x_i \ge 0$$

Note:

since $x \in \mathbb{R}^d$,

$x = (x_1, x_2, \ldots, x_d)$

Now consider origin, so $(x_1, x_2, \ldots, x_d) = (0, \ldots, 0)$

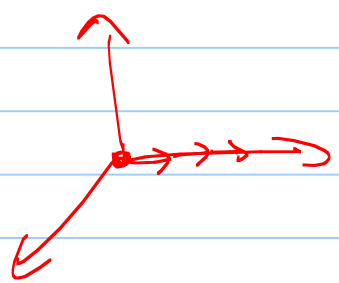
- Certainly a vertex! (which d eqns?)
- Optimal only if: all $c_i < 0$

Conversely, if any $c_i > 0$, we can increase the objective function $c^T \cdot x$ by raising x_i.

So: increase an x_i where $c_i > 0$!

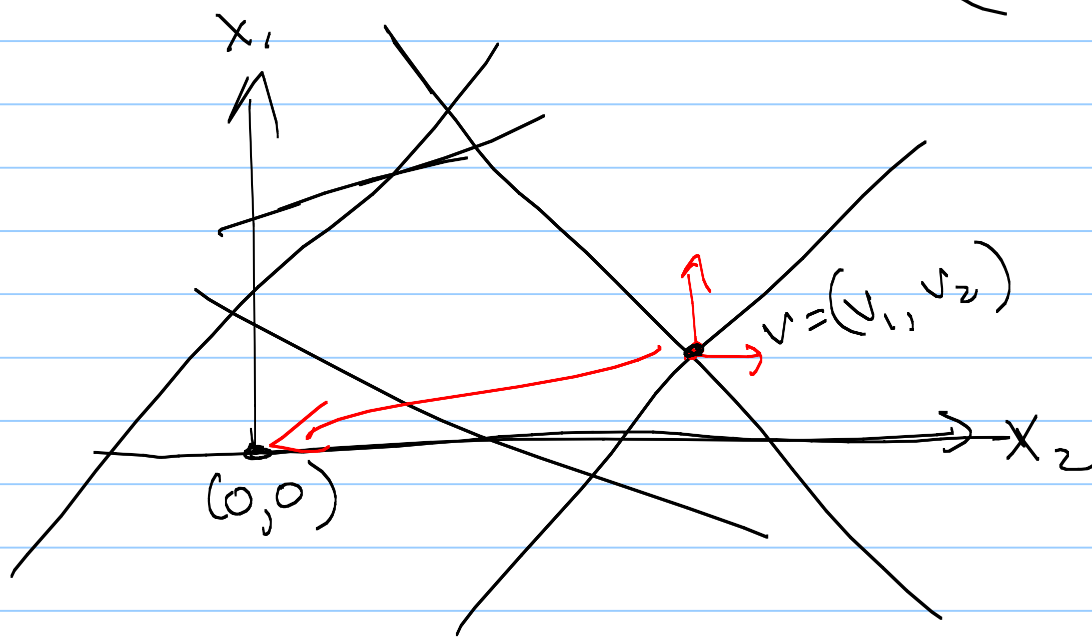
How much?

until we hit a constraining inequality



Now: what if not at the origin?

Transform the LP! (Shift the coordinate system.)



Some details

- Can't always have the origin be feasible, so need to find a first feasible point (which we then reset as our "origin").

Turns out — this can be done via a different LP!

How??

Standard form: $\min\ c^T x$
s.t. $Ax = b$ } m equations
$x \geq 0$

- Create m artificial z_i's – one per eqn.
- Add z_i to i^{th} eqn.
- Let obj. fcn be $z_1 + z_2 + \cdots + z_m$

goal → minimize

Starting vertex here is $z = (b_1, b_2, \ldots, b_m)$
(& all else = 0).

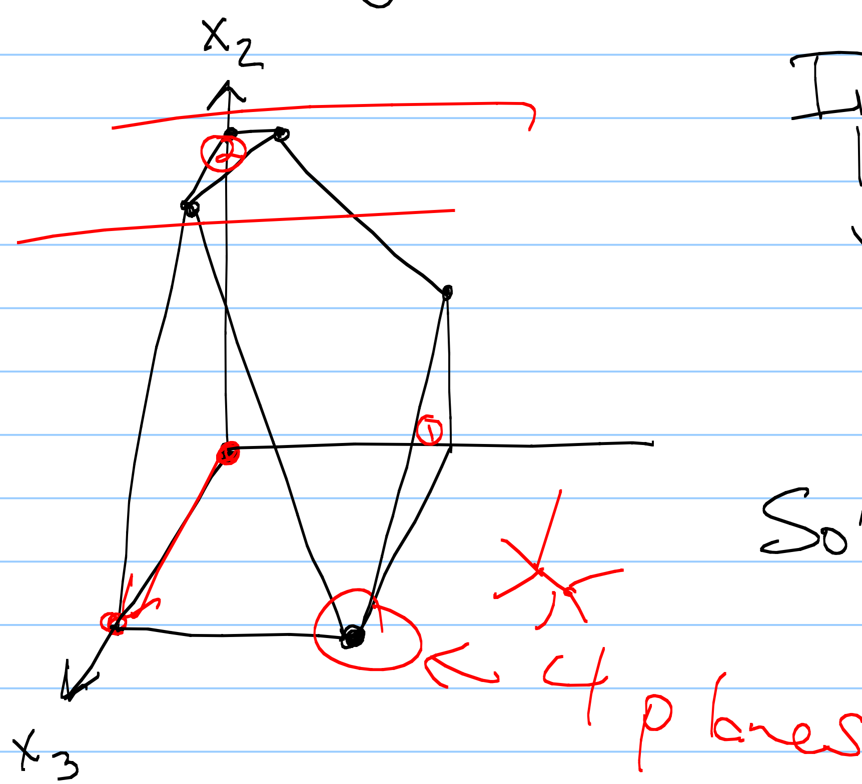
Then run simplex!

Then 2 cases:

1. If $z_1 + \cdots + z_m = 0$ in opt, then opt vertex returned by this LP is feasible for original.

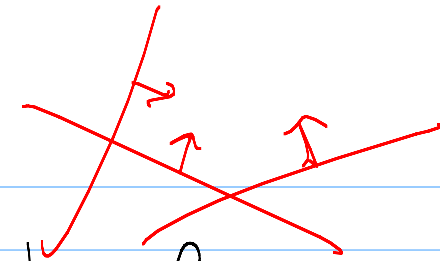
2. If $\sum z_i > 0$, then simplex decided minimizing z_i's can't give 0 — needed some > 0. This means original is actually in<u>feasible</u>!

Degeneracy: Can have $\geq d$ hyperplanes make a vertex!



In our test, will look like the vertex is the same as its neighbors!

Solution: Perturb the b_i's (slightly).



Unboundedness

If when exploring the nbhd of a vertex, taking out an inequality & adding another can give <u>underdetermined</u> system – an ∞ of possible solutions.

If so, simplex halts & complains.

Run time:

Consider a vertex $u \in \mathbb{R}^n$, with m inequalities.

$(x_1, \dots, x_n)$

At most $n \cdot m$ neighbors:
choose an inequality to drop & a new one to add.

$n \cdot (m-n)$

Checking a neighbor:
Each:
Operation w/ matrices & dot products.
Gaussian elimination: $O(n^3)$

Ick! Each iteration is $O(m \cdot n^4)$ time, to check all possible neighbors.

But can improve:

- recall just need a $c_i > 0$
- and rescaling to "new" origin is easier.

$\Rightarrow$ Can do in $O(mn)$ per iteration.

How many iterations?

$m+n$ inequalities

any n form a vertex

$\Rightarrow \leq \binom{m+n}{n}$ vertices. $\approx m^n$

exponential!!

Take away: worst case, $m^n \cdot mn$

(horrible!)

Klee-Minty

Note:

There are examples where simplex takes exponential time!
(Klee in '50's)

Polynomial time options

- ellipsoid algorithm (Khachiyan '79)
- interior point method (Karmarkar in '80's)