CSCI 3100

LP: Simplex
Today:

- HW due
- Next HW up
- Oral grading signup on Monday
LP w/ d variables:

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

2d: $ax + by \leq c$

- $-\frac{a}{b}$ slope
- $\frac{c}{b}$ y-intercept

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

$\mathbb{R}^d_+ : c_1 x_1 + \ldots + c_d x_d \leq C$
Vertices:
These happen when \( \geq d \) hyperplanes meet in \( \mathbb{R}^d \).

In \( \mathbb{R}^2 \):

\[
d = 2
\]

2 lines meet at
a point

\[\begin{array}{c}
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\quad \\
\end{array}\]
In $\mathbb{R}^3$:

Maximize $x_1 + 6x_2 + 13x_3$

s.t.

- $x_1 \leq 200$ \(\textcolor{red}{1}\)
- $x_2 \leq 300$ \(\textcolor{green}{2}\)
- $x_1 + x_2 + x_3 \leq 400$ \(\textcolor{blue}{3}\)
- $x_2 + 3x_3 \leq 600$

and

- $x_1 \geq 0$ \(\textcolor{blue}{4}\)
- $x_2 \geq 0$ \(\textcolor{red}{5}\)
- $x_3 \geq 0$

The solution is found at the vertex where three planes meet.
**Dfn:** Pick a subset of inequalities. If there is a unique point that satisfies all with equality, then it is feasible. So this is a vertex 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 $\bar{0}$:

\[ d(x, y) \text{ in } 2d \]

Subtract x's or translate to $\bar{0}$
LP: $\max \ c^T x = c_1 x_1 + \cdots + c_d x_d$

subject to $A x \leq b, \quad a_1 x_1 + x_2 x_1 + \cdots + x_d x_1 \leq b,$

$x_i \geq 0 \quad \forall i$

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

$x = (x_1, \ldots, x_d)^T$

Start w/ origin, so

our $x = \underline{0} \Rightarrow x_1 = 0$

It is always a vertex!

(Why?) all $x_i \geq 0 \Rightarrow$

optimal only if:

all $c_i$'s are negative
Conversely:
If any $c_i > 0$, we can increase the obj. function $C^T x$.

How? Increase $x_i$.

So: pick one & increase!
How much?

Until we hit another constraint:
calculate $x_i$'s intercepts.
Now: What if not at origin?

Transform LP!

(i.e. shift all coords)

Subtract these from all inequalities
Some details

- Origin isn't always feasible, so must find a starting feasible point.

  (reset to be \( \hat{0} \))

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

- Degeneracy: Can have \( > d \) hyperplanes at a 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
Runtime:
Consider a vertex \( u \in \mathbb{R}^n \), with \( m \) inequalities.
At most \( n \cdot m \) nbrs:
choose one to drop:
one to add:
\( \leq n(m-n) \) (could be smaller)

Checking for nbr:
Each is a dot product/matrix operation.
Gaussian elimination: \( O(n^3) \) (basically)

\( \Rightarrow \) Each iteration:
\( O(mn^4) \)
Can improve slightly:
- just need one $c_i > 0$
  + rescaling to $\frac{1}{\tilde{c}}$ is easy.

$\Rightarrow$ Can improve to $O(mn)$ per iteration.
How many iterations?
- $m + n$ inequalities
- any $n$ give a vertex

$$\Rightarrow \binom{m+n}{n} = O((m+n)^n)$$

Klee-Minty examples that are actually this slow.
(in 50's)
Alternatives

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

Polyomial

But:

In practice, simplex does 'better'!