Optimization problem with linear inequalities -
variables $x_1, \ldots, x_d$ in $d$-dimensions.

\[
\begin{align*}
\text{max} & \quad c_1 x_1 + c_2 x_2 + \cdots + c_d x_d \\
\text{st.} & \quad a_{11} x_1 + a_{12} x_2 + \cdots + a_{1d} x_d \leq b_1 \\
& \quad \vdots \\
& \quad a_{n1} x_1 + \cdots + a_{nd} x_d \leq b_n
\end{align*}
\]

i.e.

\[
\text{max } cx \\
A x \leq b
\]

$c$ $1 \times d$ vector
$x$ $d \times 1$ vector
$A$ $n \times d$ matrix
$b$ $n \times 1$

in 2D:

Each constraint $a_i x_1 + a_2 x_2 \leq b_i$ is a half-space.

The feasible region is the intersection of all half-spaces.

Optimal solution may not be unique.

So long as the feasible region is non-empty and bounded, there is an optimal solution at a vertex. Pick $d$ of inequalities and set to equality.

This gives a stupid algorithm - try all $\binom{n}{d}$ vertices.

- Which are feasible?
- Which gives max. obj. value.

Applications:

Planning menus. Have $n$ nutrients. Need amount $b_i$ of nutrient $i$. Have $d$ foods, food $j$ has cost $c_j$ and amount $a_{ij}$ of nutrient $i$.

\[
\begin{align*}
\min & \quad cx \\
A x & \geq b \\
& \quad x_j - \text{cost food } j
\end{align*}
\]
History 40's, 50's
Dantzig - simplex method '40's
- spurred development of computers
- geometrically, it walks from one vertex of feasible region to adjacent one

Ineq: rules - which

ineq. to remove
& which one to add.

- for almost all simplex pivot rules we know examples where it takes exponential time.

OPEN - is there some pivot rule that gives poly time?
related to Hirsch conjecture: diameter of convex polyhedron

1957: Disproved in 2011 Santos
$d = 43 \quad n = 86$

* inequalities

but there could still be polynomial (even linear) bound.

Simplex method very good in practice.

Poly time alg. for linear programming:
'80 - Kachian - ellipsoid method.
'84 - Karmarkar - interior point method.

Operate on bit representations of numbers.

OPEN: alg. that uses #arithmetic operations poly in n and d.

recent: - Spielman & Teng 2001, smoothed analysis
- why the simplex method is good.
- Disser & Skutella 2018 simplex method is NP-mighty.
- 70's & 80's - linear prog. in small dim.
  \[ d = 2, \quad d = 3 \]

Applications:
- find best line fitting pts
- de Berg et al. - whether a cast can be removed from a mold - 3D linear programming.

Megiddo '83 \( O(n) \) for fixed \( d \)
actually \( O(2^d \cdot n) \)

today's topic: Seidel's Randomized Incremental LP Alg.
idea: add half-planes one by one, updating opt. set \( u \) (vertex)

Add \( h_i \)

2 cases:
1. \( u \in h_i \) - no update.

2. \( u \not\in h_i \) - new opt. will lie on \( l_i \) - line of \( h_i \)

So solve 1-dimensional LP problem along line \( l_i \):

1D LP
\[
\begin{align*}
\max \quad & x \\
\text{s.t.} \quad & x \geq 2 \\
& x \leq 5 \\
& -1 \leq x
\end{align*}
\]

find lowest upper bound on \( x \): \( O(i) \)
Algorithm $\text{LP}_2(H)$

$H = \{ h_1, \ldots, h_n \}$

1. take random ordering $h_1, h_2, \ldots, h_n$

2. $v \leftarrow \text{opt. corner of large square containing all feasible sols}$

3. for $i = 1 \ldots n$ add $h_i$ line of $h_i$

4. if $v \notin h_i$ then

5. $v \leftarrow \text{LP}_1(\{h_1, h_2, \ldots, h_{i-1}, h_i\} \cap \lambda_i)$

$\text{LP} - \text{solve in } O(i)$

Worst case: $O(n^2)$

Can this worst case happen? Yes: exercise.

Expected run-time. Intuition: line 5 is not executed too often use Backwards Analysis.

After adding $h_i$ suppose opt. is vertex $v'$ at intersection of $h', h''$

$\overline{\overline{h'}} \cap \overline{\overline{h''}}$ — we have $i$ lines

halfplane $h_i$ is equally likely to be any one of them.

We did work for $h_i$ (i.e. executed line 5 of Alg.) only if $h_i = h'$ or $h''$.

$\text{Prob } \{ h_i = h' \text{ or } h_i = h'' \} = \frac{2}{i}$ because

Expected total work in line 5 $\sum_{i=1}^{n} \frac{2}{i} O(i) = O(n)$

In higher dimensions

$\frac{2}{i}$ becomes $\frac{d}{i}$ because it takes $d$ hyperplanes to specify a vertex.

$T_d(n) = T_d(n-1) + \frac{d}{n} O(T_{d-1}(n))$ so $\text{the answer is } T_d(n) = O(d!n)$
Same approach can be used to find smallest enclosing disc.

given points \( P = \{ p_1, \ldots, p_n \} \) in \( \mathbb{R}^d \)
find smallest radius disc enclosing all points

- a facility location problem — use center of disc

Not linear programming
(it is quadratic programming).

Megiddo's approach does work. \( \mathcal{O}(n^2 \log n) \) algo.

Not covered in class:

Randomized Incremental approach:

solve for \( n-1 \) points

add new point

if \( p \) inside current disc — OK
what if \( p \) is outside?

Fact: updated disc goes through \( p \).

Get "smaller" problem: given points \( P + \) special point \( p \)
find smallest disc \( D \supseteq P \) with
\( p \) on boundary.

Solve this problem (from scratch) using randomized incremental approach

- to add new point \( q \):
- if \( q \) inside current disc — OK
What if \( q \) is outside?

Fact: updated disc goes through \( p \) and \( q \)

Get easier problem: given \( P \) and special points \( p, q, \) find smallest disc \( D \supseteq P \) with \( p, q \) on boundary

... same idea once more ...

Problem: given \( P, p, q, r \), find \( D \supseteq P \) with \( p, q, r \) on boundary

Now \( D \) is unique! (3 points determine a circle)

This leads to algorithm with expected run time \( O(n) \).