Algorithms – Spring '25

LP: Simplex

Recap

- Oral grading this week
  (No office hours for me today)
- HW 6 graded, HW7 coming soon
- Practice final: any questions on
  general topics/format?
- Wed: practice problems during
  class

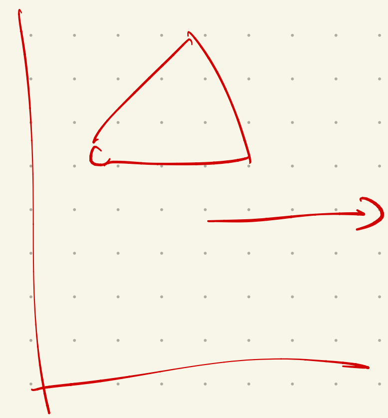
The algorithm: Simplex (Dantzig 1947)

Assumes canonical form:

$$\text{maximize } \sum_{j=1}^{d} c_j x_j$$

$$\text{subject to } \sum_{j=1}^{d} a_{ij} x_j \leq b_i \quad \text{for each } i = 1..n$$

$$x_j \geq 0 \quad \text{for each } j = 1..d$$



So:

- no min
- only $\leq$
- + ≥ 0 for all variables
- fast in practice, but exponential in worst case

$\boxed{\text{Klee-Minty, 1973: some feasible polytopes have } O(n^{\lfloor d/2 \rfloor}) \text{ vertices}}$

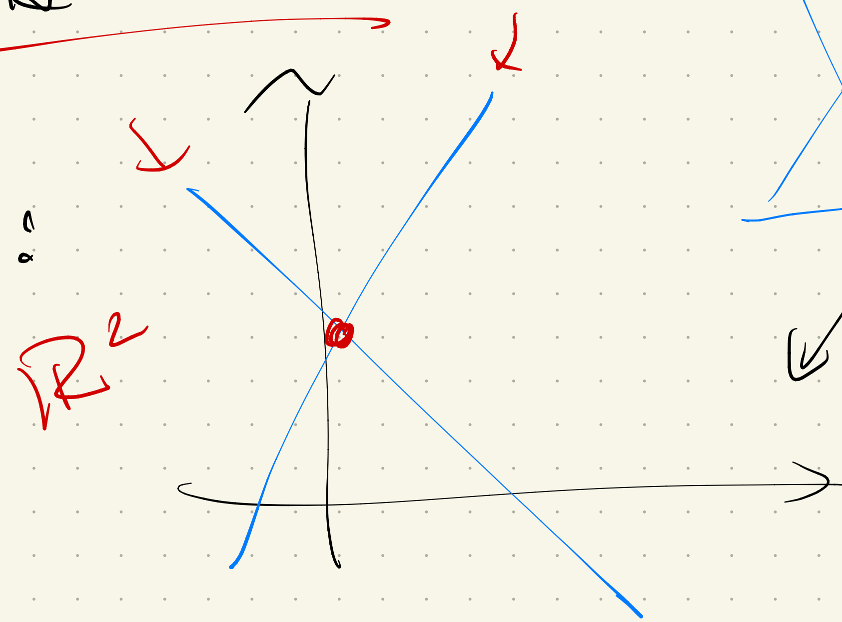
Algorithm (simplex):

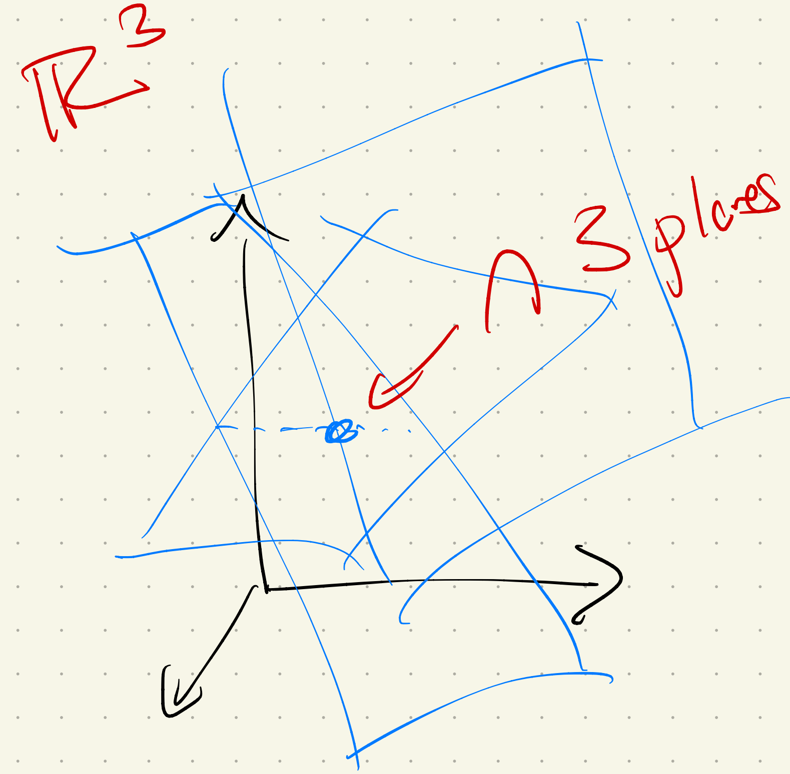
Take any vertex v in feasible region
while some neighbor v' is better
$v \leftarrow v'$

Details lacking!

Step 1: find a vertex

take d hyperplanes!





Any d hyperplanes give a vertex:
Loop through m-d others!
Either feasible for each
Or Not
↳ use this new hyperplane

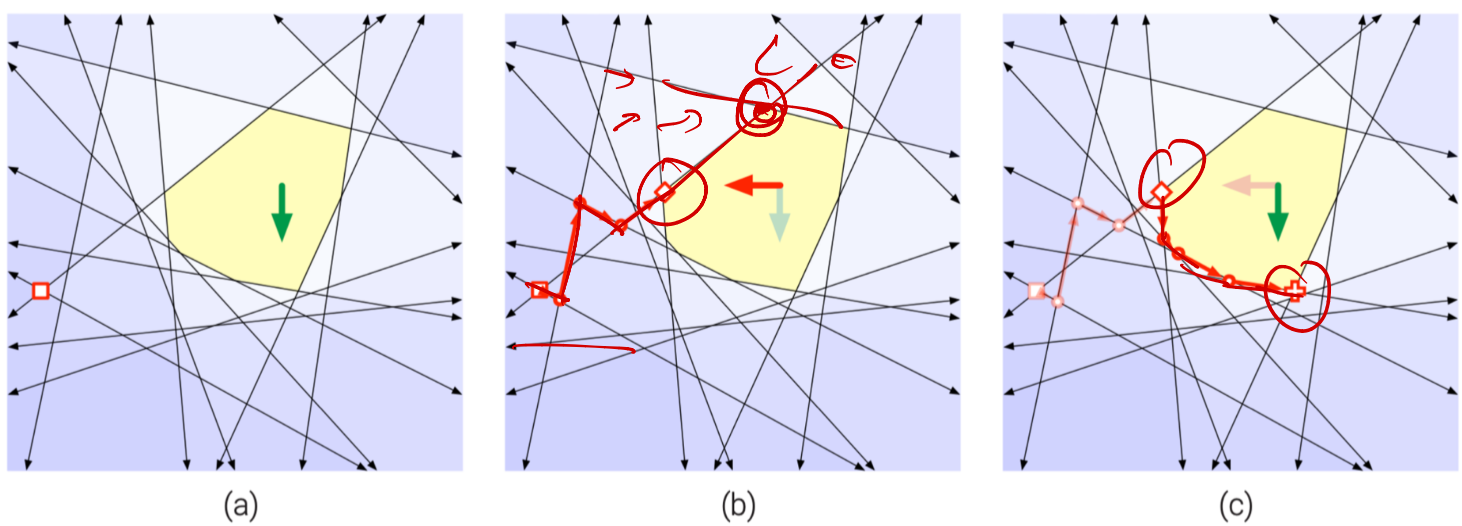
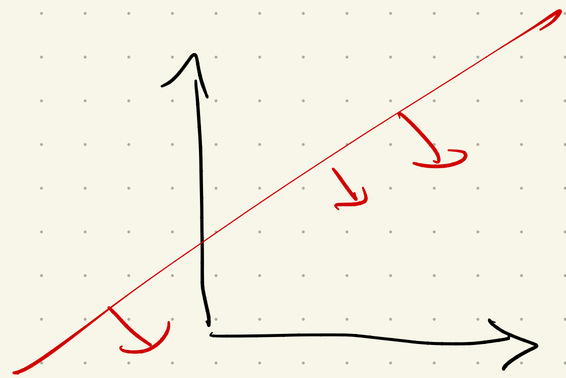


Figure I.2. Primal simplex with dual initialization: (a) Choose any basis. (b) Rotate the objective to make the basis locally optimal, and pivot "up" to a feasible basis. (c) Pivot down to the optimum basis for the original objective.

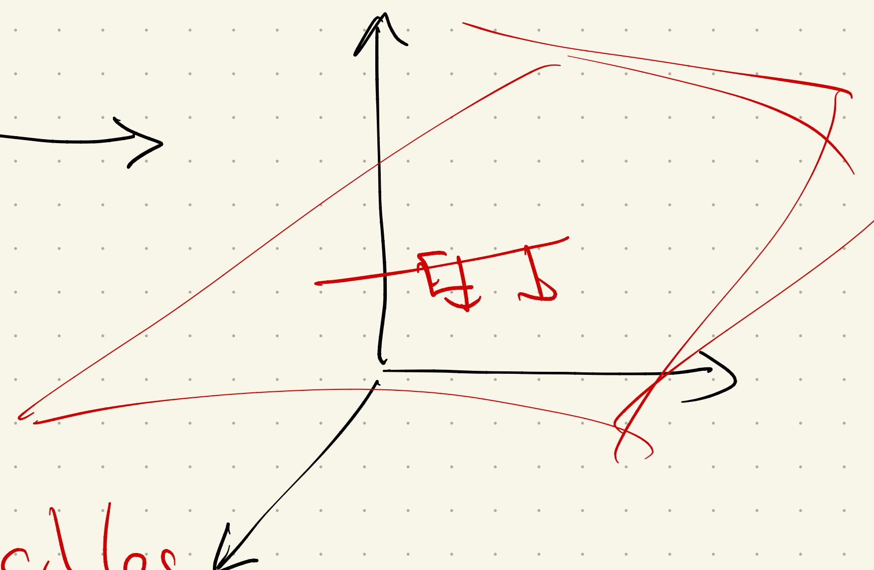
Let's go over that more carefully:

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

2d: $ax+by \leq c$



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



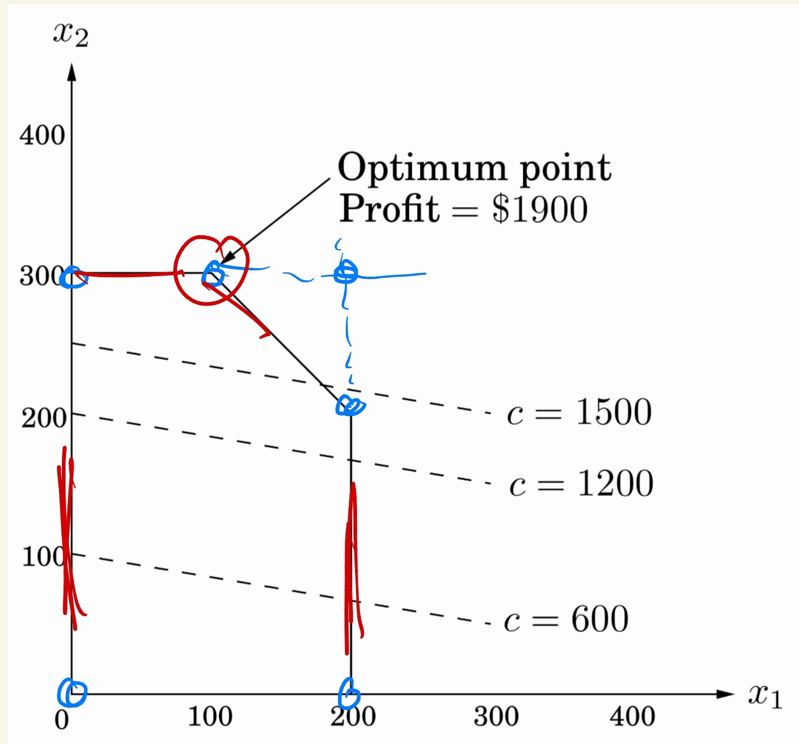
d variables

$\mathbb{R}^d$: $a_1x_1 + a_2x_2 + \cdots + a_dx_d \leq c$

Vertices:

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

In $\mathbb{R}^2$:



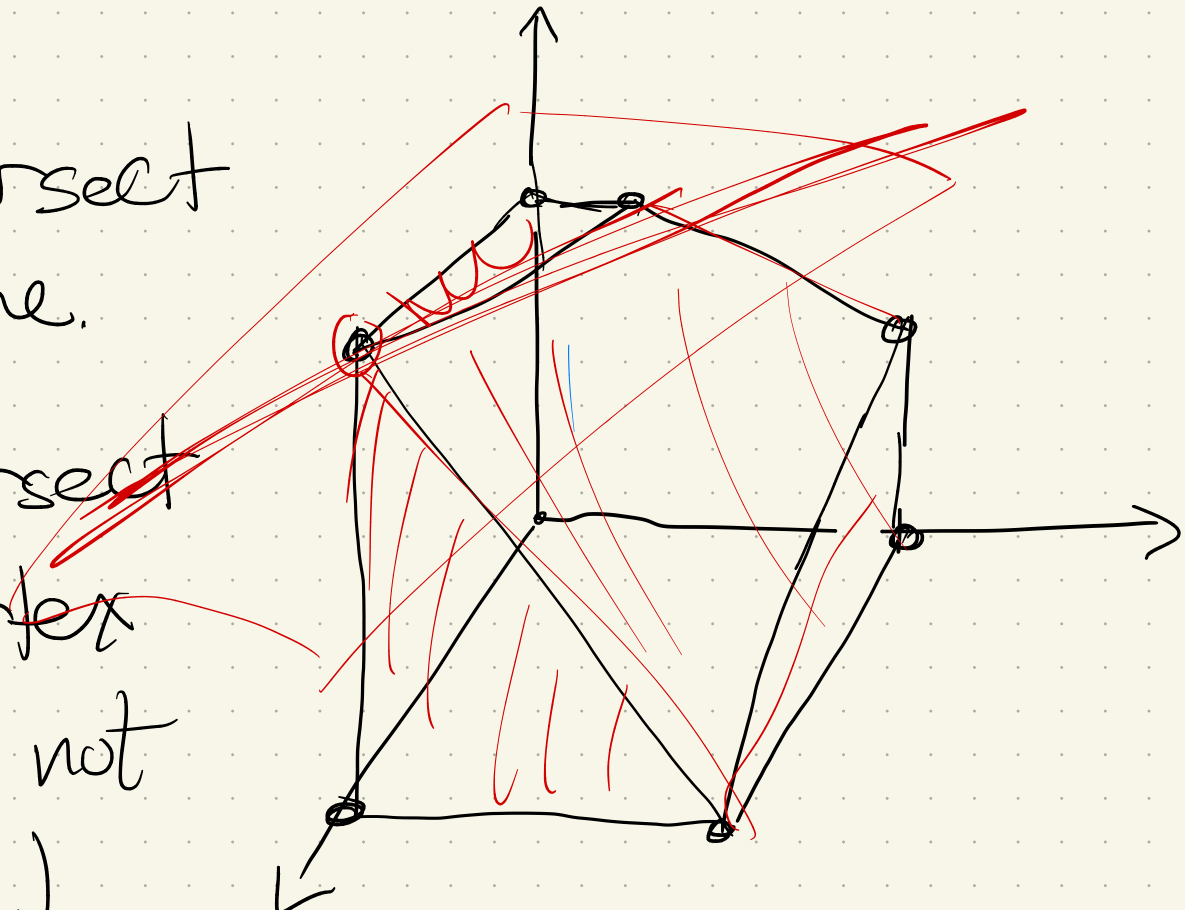
eqn 1
eqn 2
solve for point

Note:
Not every pair will meet!

In $\mathbb{R}^3$

Any 2 intersect
in a line.

Any 3 intersect
in a vertex
(assuming not
parallel)

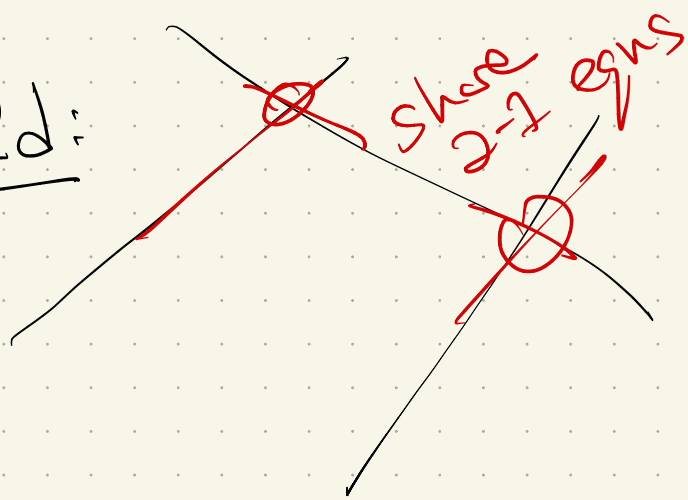


In $\mathbb{R}^d$: d equations
d variables } $\Rightarrow$ one point

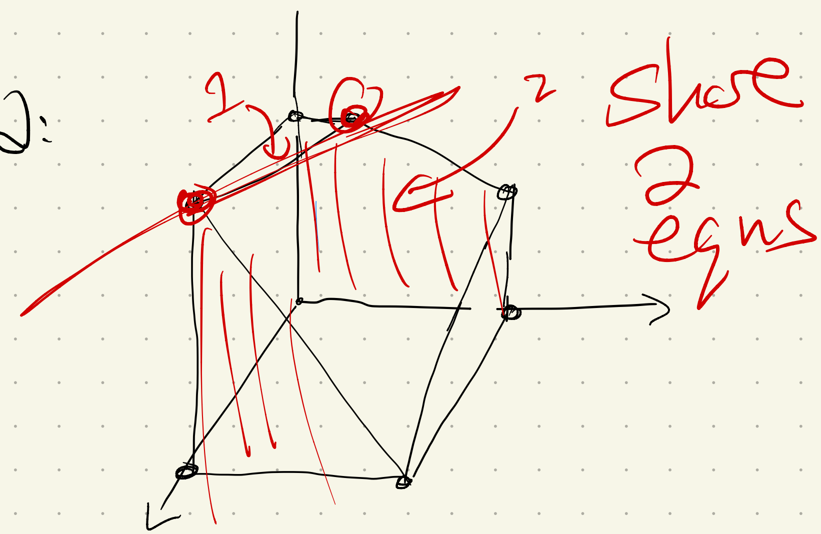
Dfn: Pick a subset of inequalities
If there is a unique point that satisfies all with equality, & it is feasible
↳ this is a vertex of the solution

Since each vertex is specified by d equations, we call any two that share $(d-1)$ of them neighbors:

2d:



3d:



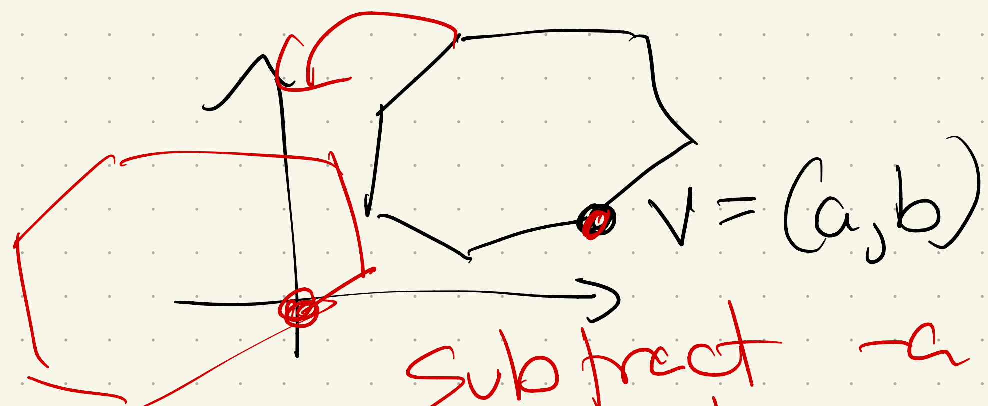
Simplex algorithm:

In each stage, 2 tasks:

1. Check if current vertex is optimal
2. If not choose a neighbor that gives a better score under the objective function

Both are easy if $v = \vec{0}$ (the origin)
(see next slide)

If not at $\vec{0}$:
translate



Subtract $-a$ & $-b$ from all eqns

LP: $\max \ c^T x$

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

$\boxed{x_i \ge 0} \ \forall i$

Note: $\vec{x} \in \mathbb{R}^d$, so $x = (x_1, \dots, x_d)$

Now, $\vec{0}$ is always a vertex — why?

→ those d eqns intersect at $\vec{0}$

Optimal only if all c_i's are negative

Conversely:

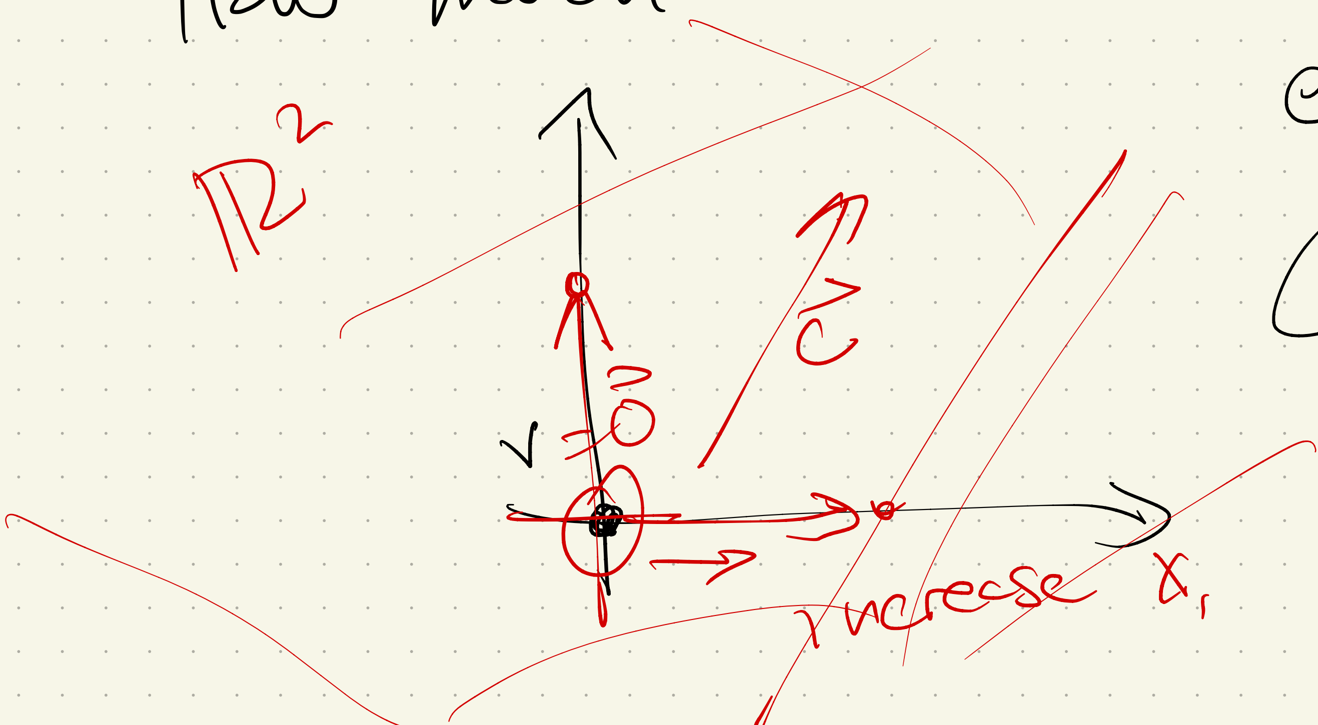
obj fcn is not

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

How?

So: pick one & increase!

How much? until stuck on some eqn



either c_1 or c_2 is positive

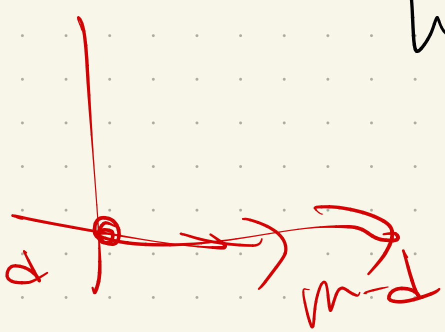
↳ when do we get stuck?

Runtime:

Consider a vertex $\vec{v} \in \mathbb{R}^d$ with m inequalities & ~~n~~ d variables

How many possible neighbors?

each removes 1 eqn & replaces with another



$$\longrightarrow \quad \le \boxed{d \cdot (m-d)}$$

← other eqns

Checking if it is a neighbor & is feasible: basically dot product & Guassian elimination.

So: each iteration is

$$\leq \underbrace{d \cdot (m-d)}\cdot \underbrace{[\text{time for G.E}]}_{O(n^{2.3\ldots}) = O(n^\omega)}$$

Can improve slightly:
- just need one $c_i > 0$ + rescaling to $\bar{0}$ is easy.

So can get to $O(m \cdot n)$

per nbr we find.

How many times do we need to find a nbr?

How many iterations?

$\left\{\begin{array}{l} m+d \text{ inequalities} \\ \text{any } d \text{ give a vertex} \end{array}\right\} \Longrightarrow \binom{m+d}{d} \rightsquigarrow O((m+d)^d)$

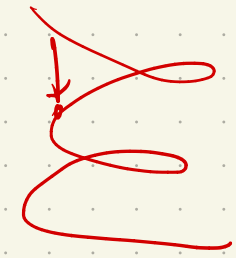
So exponential in worst case.

(Remember, for a while people thought this might be NP-Hard)

Klee + Minty gave examples in the ~~50's~~ 70's that actually take exponential time.

Can we avoid this by choosing the "best" neighbor in our update?

No ideal way!



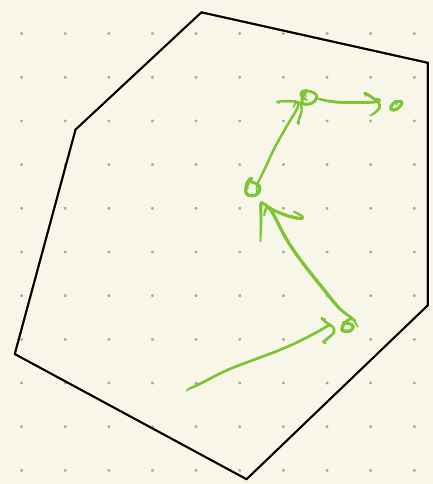
Many proposed, but for almost every one, there is some input polyhedron that needs an exponential number of pivots.

Ellipsoid algorithm, Khachiyan 1979

- (weakly) polynomial time
  - ⤷ dependent of precision
- high level idea: compute smaller & smaller ellipses which contain solution

Interior point Methods, Karmarkar 1984

- Move through polytope's interior:



- Still weakly polynomial
  - but practical

More recent:

- Matrix Multiply time (in 2019!)

RESEARCH-ARTICLE

Solving linear programs in the current matrix multiplication time

Authors: Michael B. Cohen, Yin Tat Lee, Zhao Song | Authors Info & Claims

STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing • Pages 938 - 942
https://doi.org/10.1145/3313276.3316303

Published: 23 June 2019 Publication History Check for updates

Abstract

This paper shows how to solve linear programs of the form $\min_{Ax=b,x\geq 0} c^\top x$ with n variables in time $O^*((n^\omega + n^{2.5-\alpha/2} + n^{2+1/6}) \log(n/\delta))$ where ω is the exponent of matrix multiplication, α is the dual exponent of matrix multiplication, and δ is the relative accuracy. For the current value of $\boxed{\omega \sim 2.37}$ and $\alpha \sim 0.31$, our algorithm takes $O^*(n^\omega \log(n/\delta))$ time. When $\omega = 2$, our algorithm takes $O^*(n^{2+1/6} \log(n/\delta))$ time.

Our algorithm utilizes several new concepts that we believe may be of independent interest:

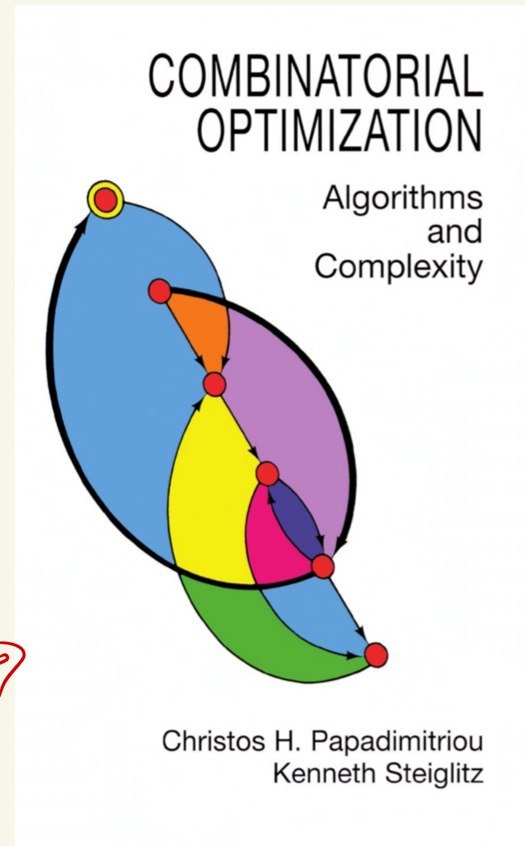
(1) We define a stochastic central path method.

(2) We show how to maintain a projection matrix

$$\sqrt{W} A^\top (A W A^\top)^{-1} A \sqrt{W}$$

in sub-quadratic time under ℓ_2 multiplicative changes in the diagonal matrix W.

This is a large + active area of study:



Now:

5 minutes to do CIFs, if not already done!

Spring 2025 CIF Blue

(CSE 40113-01) Design/Analysis of Algorithms

42	Invited
1	Started
10	Responded
0	Opted out

thanks!

Evaluation ends on: **2025-05-04**

23% Response Rate

Changes allowed until 2025-05-04