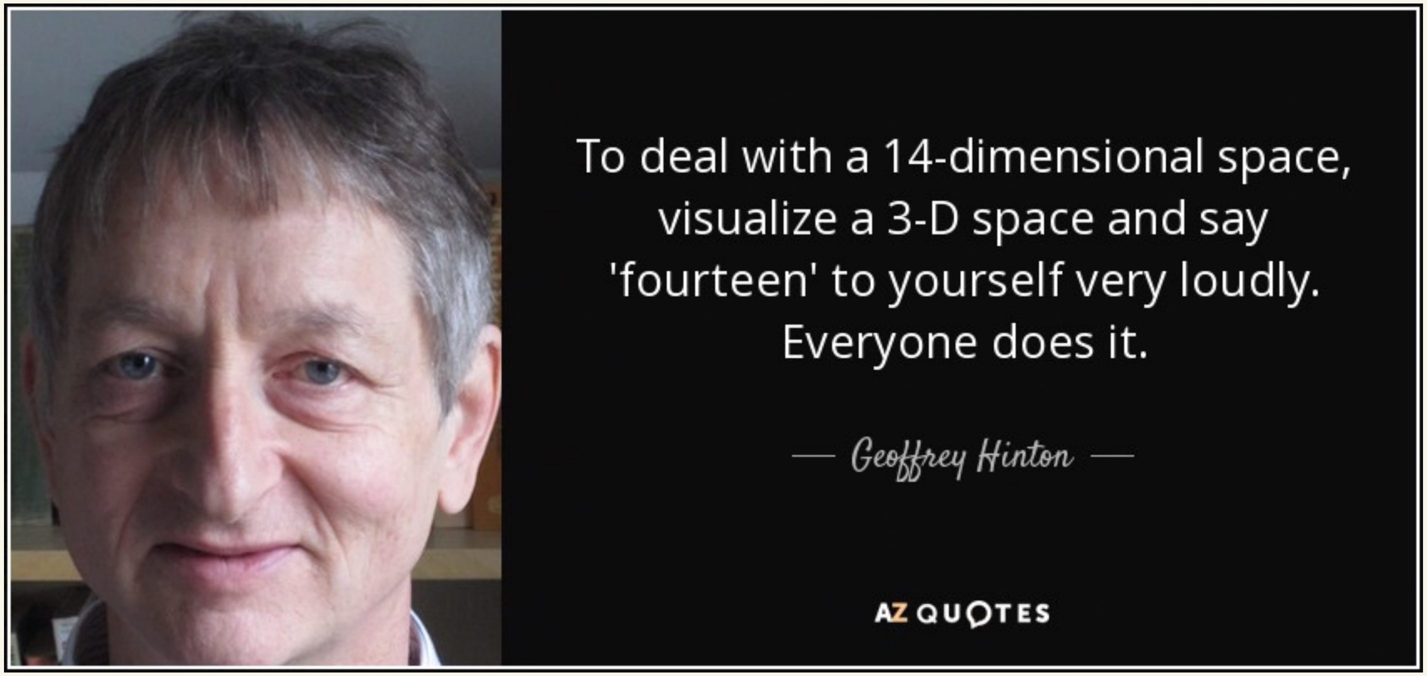
Algorithms - Spring '25

LPs - Duality

Recap:

- Make sure to sign up for HW 8 grading slot!
- Practice final is posted
- CIFs are live, & feedback is appreciated
- Block off time 2 days before final — keep an eye of webpage & slack

A moment of honesty to start lecture:



Linear optimization: Example

n foods, m nutrients

Let a_{ij} = amount of nutrient i in food j

r_i = requirement of nutrient i

x_j = amount of food j purchased

c_j = cost of food j

<u>Goal</u>: Buy food so you satisfy nutrients while minimizing cost

Can view as matrix $\longrightarrow$

$$A = \begin{array}{c|ccc} & 1 & j & n \\ \hline 1 & 0\ 0\ 0 & & \\ i & & a_{ij} & \\ m & & & \end{array}$$

$\vec{r} = (r_1, r_2, \ldots, r_m)$

$\vec{x} = (x_1, \ldots, x_n)$

$\vec{c} = (c_1, \ldots, c_n)$

So: minimize $\sum_{i=1}^{n} c_i x_i = \vec{c}\, \vec{x}^T$

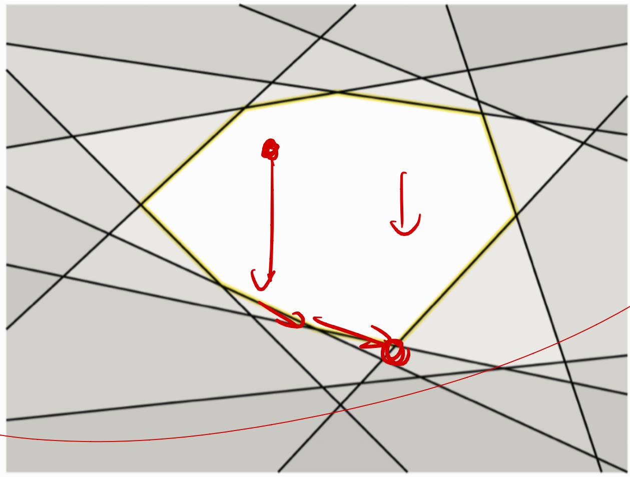
s.t.

$$A\vec{x} \geq \vec{r}$$

In general, get systems like this:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{d} c_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{d} a_{ij} x_j \le b_i \quad \text{for each } i = 1 \,..\, p \\
& \sum_{j=1}^{d} a_{ij} x_j = b_i \quad \text{for each } i = p+1 \,..\, p+q \\
& \sum_{j=1}^{d} a_{ij} x_j \ge b_i \quad \text{for each } i = p+q+1 \,..\, n
\end{aligned}
$$

Geometric picture:



A two-dimensional polyhedron (white) defined by 10 linear inequalities.

History

Dates back to 1800's where studied by Fourier.

By 1940's: serious study, due to business/optimization demand

- Not known to be NP-Hard ← 60's
  (Karp actually listed it as key open question in original paper on NP-Hardness)

Canonical form:

Avoid having both $\leq$ and $\geq$.

Why?

So get something more like our first example:

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

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

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

Or, given a vector $\vec{c}$, matrix A + vector $\vec{b}$:

$$\max \ \vec{c}\,\vec{x}$$

$$\text{s.t.} \ A\vec{x} \leq \vec{b}$$

<u>Anything</u> can be put into canonical form:

The reduction:

① Avoid $=$:

$$\sum a_i x_i = b$$

2eqn

$$\sum a_i x_i \le b \text{ and } \sum a_i x_i \ge b$$

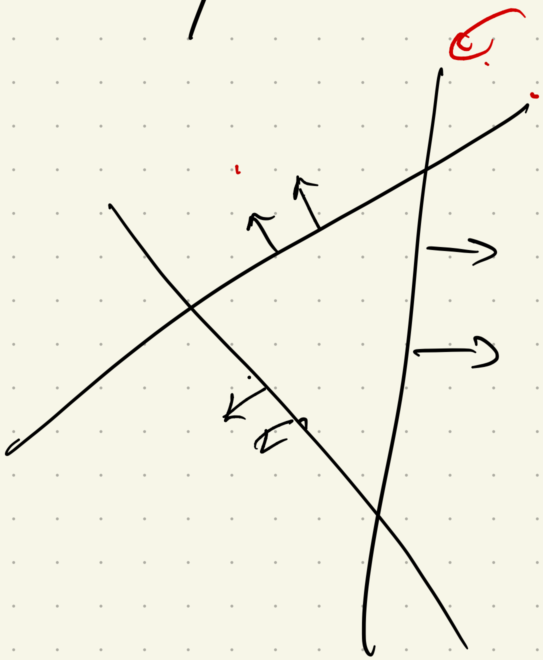
② Avoid $\overset{\checkmark}{\ge}$:

$$(-1) \cdot \left[\sum a_i x_i \ge b \right]$$

$$\sum -a_i x_i \le -b$$

How could these not have a solution?

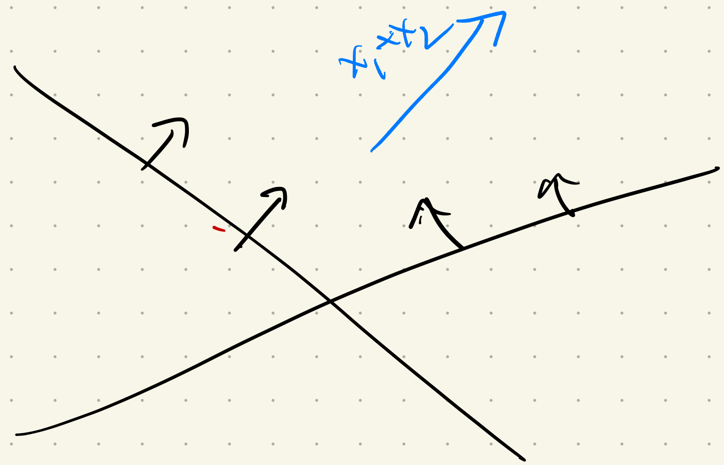
2 ways: 2d

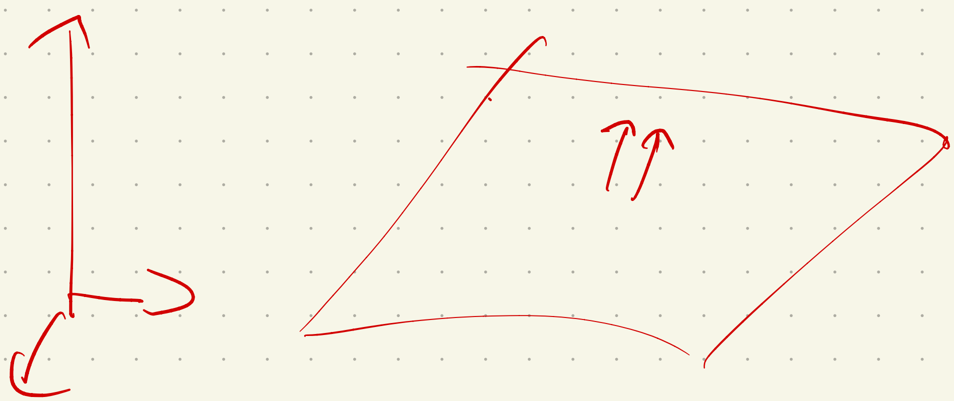


or

maximize x_1+x_2

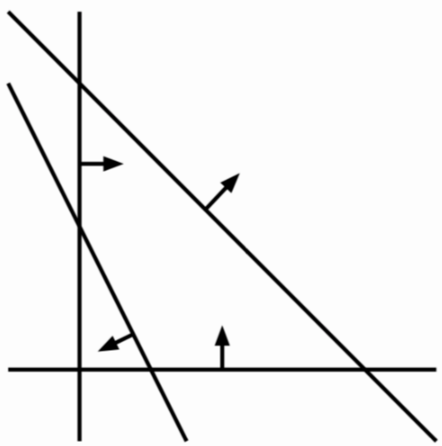
s.t.

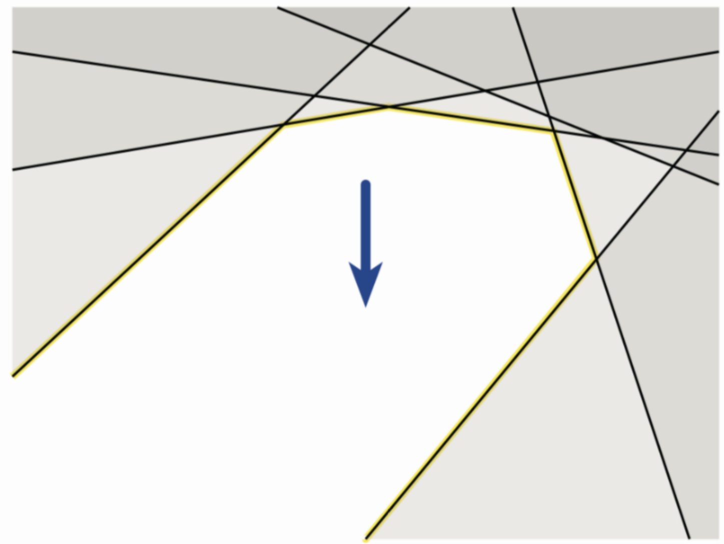




Better pictures (still 2-d):

$$\begin{aligned} &\text{maximize } x - y \\ &\text{subject to } 2x + y \leq 1 \\ &\qquad\qquad x + y \geq 2 \\ &\qquad\qquad x, y \geq 0 \end{aligned}$$

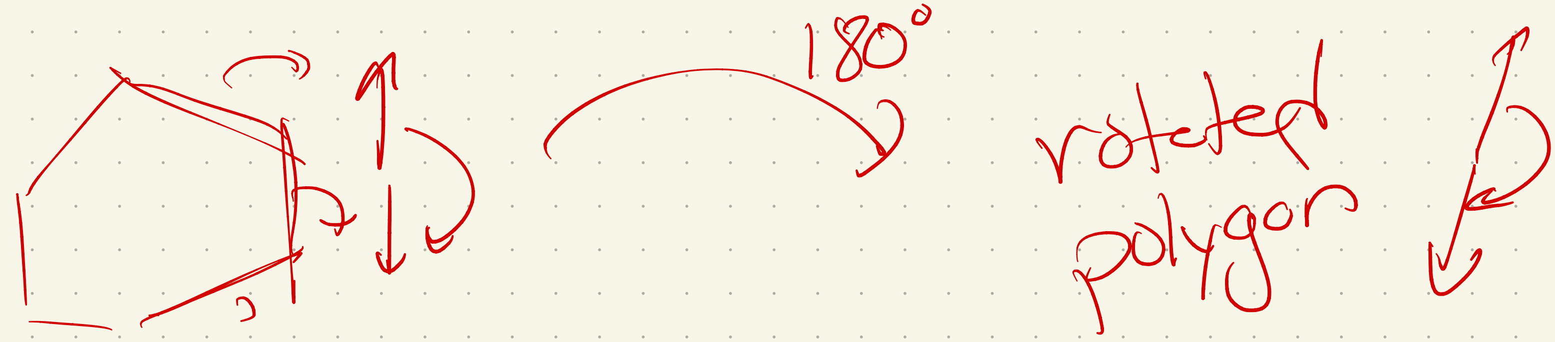




Note:

(1) Multiplying by -1 turns any maximization problem into a minimization one:

why?



(2) Can turn inequalities into equalities via slack variables:

$$\sum_{i=0}^{n} a_i x_i \le b \quad \Rightarrow \quad \sum_{i=0}^{n} a_i x_i + \underbrace{s}_{\text{adds one dim (new variable)}} = b$$

(3) Can change equalities into inequalities, also!

$$\sum_{i=1}^{n} a_i x_i = b$$

$$\Downarrow$$

Modeling Problems: Flows + Cuts

<u>Input</u>: directed G w/ edge capacities $c(e)$
+ $s, t \in V$

<u>Goal</u>: Compute flow $f: E \to \mathbb{R}$ s.t.

1. $0 \le f(e) \le c(e)$ ← edge constraints
2. $\forall v \ne s, t$,

$$\sum_u f(u \to v) = \sum_w f(v \to w)$$

vertex constraints

<u>Make an LP</u>: Maximize $\sum_{v \in V} f(s \to v)$

x_i's flow on edge i

s.t. 1. $x_i \ge 0$
$x_i \le c(e_i)$

2. flow in = flow out at each vertex

Create x_i for each edge $e_i = (u,v)$

(2) for each vertex v

$$\sum_{(u,v)=e_i} x_i - \sum_{(v,w)=e_j} x_j = 0$$

Conversion: V vertices, E edges

$\hookrightarrow$ E variables (or dimensions)

2 eqns per edge

1 eqn per vertex

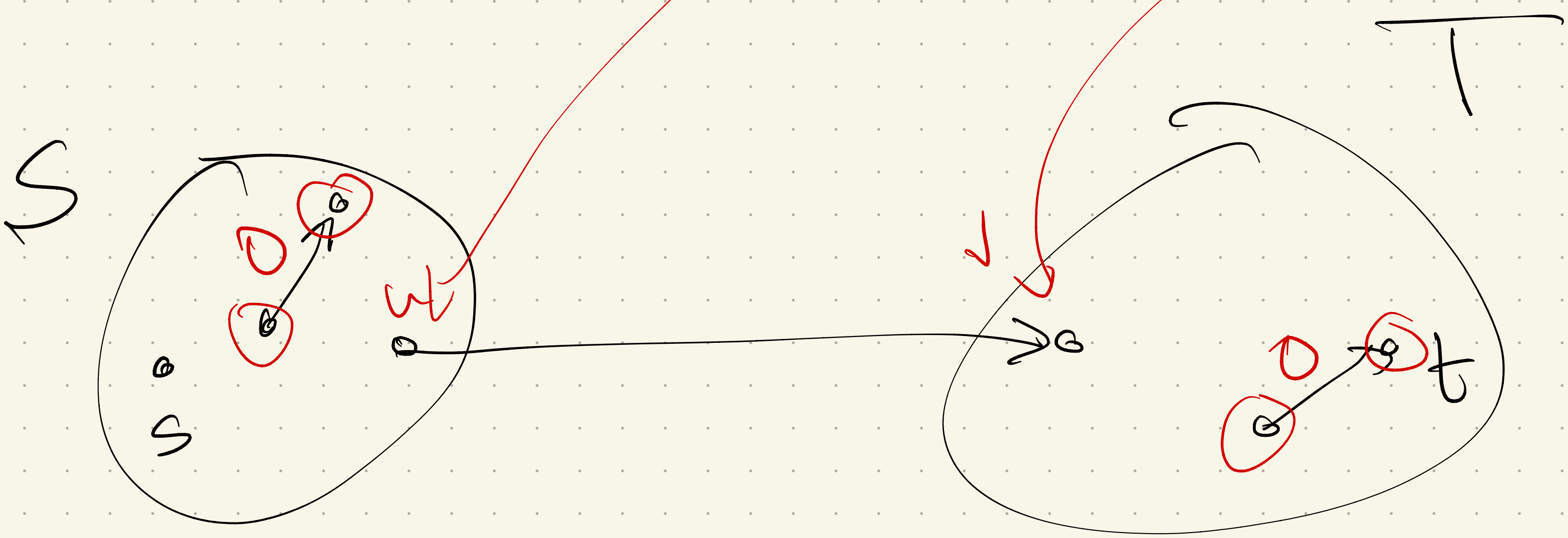
$\Big\} O(V+E)$

Related: Min cuts (S,T)

Use indicator variables: (ILP)

$S_v = 0$ or 1 ← on s side or t side

goal: $X_e = X_{(u \to v)} = 1$ if $u \in S$ and $v \in T$



The LP:

Minimize $\sum_{u\to v} c_{u\to v} \cdot x_{u\to v}$

1 if in cut

want few edges

s.t. $x_{u\to v} + s_v - s_u \geq 0 \quad \forall u \to v$

can be 0

≤ 0 if on same side

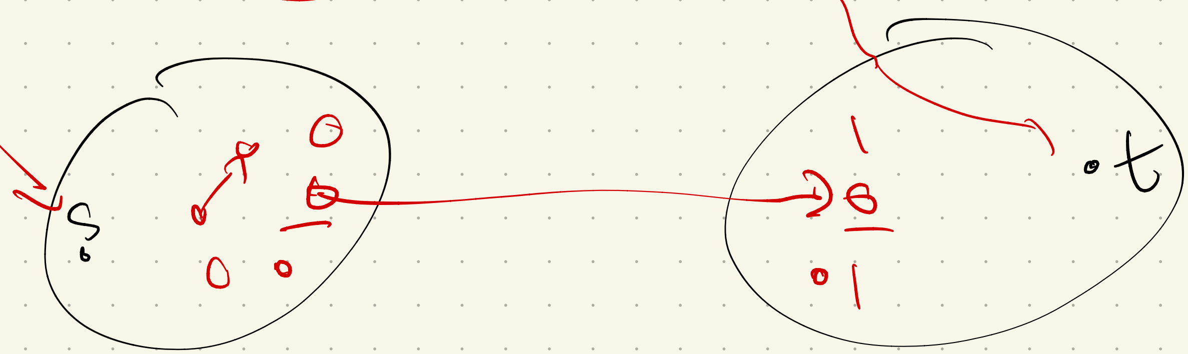
Which are forced?

negative if cut edge

$x_{u\to v} \geq 0 \quad \forall u,v$

$s_s = 1$

$s_t = 0$



Cuts: input = graph G

V + E variables

2E equations

+2

Note:

For flow/cuts, a solution would yield optimal LP solution.

The reverse is not obvious!

LP might have strange fractional answer which doesn't describe a cut.

It can be shown that this won't happen

↳ but not obvious...

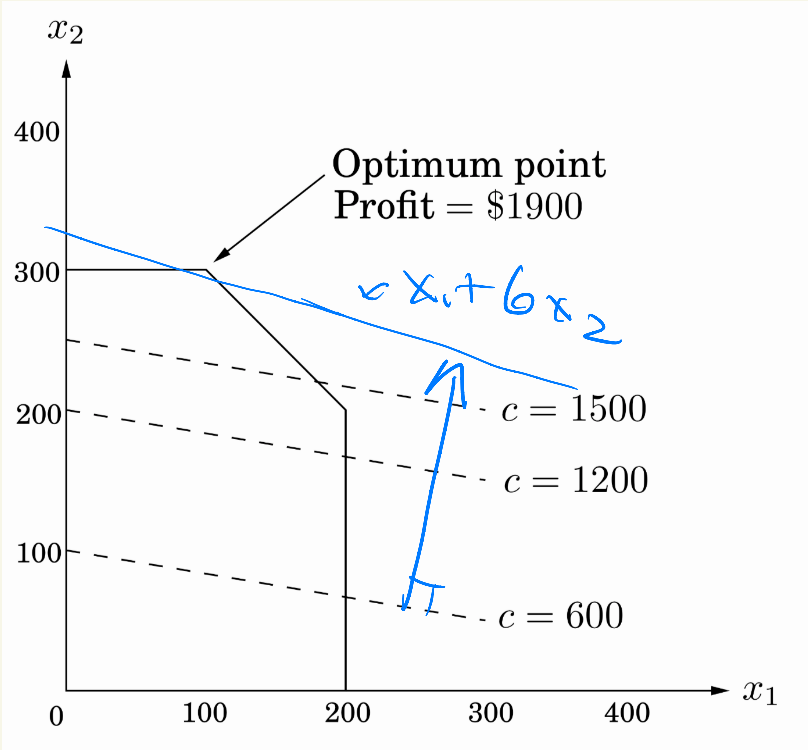
Duality:

Recall our chocolate:

LP: $\max\ x_1 + 6x_2$

s.t.

$$x_1 \le 200$$
$$x_2 \le 300$$
$$x_1 + x_2 \le 400$$
$$x_1, x_2 \ge 0$$



Can we check that this is best?

$$\max \quad x_1 + 6x_2$$
s.t.
$$x_1 \le 200 \quad ①$$
$$x_2 \le 300 \quad ②$$
$$x_1 + x_2 \le 400 \quad ③$$
$$x_1, x_2 \ge 0$$

$$\begin{array}{c} 1 \\ 6 \\ 0 \end{array}$$

Play w/ inequalities:

① + 6·② :

$$x_1 + 6x_2 \le \underbrace{200 + 6 \cdot 300}_{= 2000}$$

$$x_1 \le 200$$
$$6x_2 \le 1800$$

Interesting!

These 2 inequalities tell us that we couldn't ever beat \$2000.

But recall soln was \$1900

can we get a better combo?

$$\max \quad x_1 + 6x_2$$

s.t.

$$\begin{aligned} x_1 &\le 200 && ① \times 0 \\ x_2 &\le 300 && ② \times 5 \\ x_1 + x_2 &\le 400 && ③ \times 1 \\ x_1, x_2 &\ge 0 \end{aligned}$$

Play: $0 \cdot ① + 5 ② + 1 \cdot ③$

$$\left.\begin{aligned} 5x_2 &\le 1500 \\ x_1 + x_2 &\le 400 \end{aligned}\right| \text{add} \Rightarrow x_1 + 6x_2 \le 1900$$

These multipliers, $\boxed{(0,5,1)}$, are a <u>certificate</u> of optimality.

↳ No valid solution can ever beat $1900

<u>But</u> how do we find these magic values??

In this, we had three "≤" inequalities

↳ so goal is to find the right 3 multipliers: $\underline{y_1}$, $\underline{y_2}$, and $\underline{y_3}$

Let's try to rewrite...

Multiplier		Inequality
y_1	$\times$	$(x_1 \le 200)$
y_2	$\times$	$(x_2 \le 300)$
y_3	$\times$	$(x_1 + x_2 \le 400)$

Result $y_1 x_1 + y_2 x_2 + y_3(x_1 + x_2)$

$\le 200y_1 + 300y_2 + 400y_3$

$(y_1 + y_3)x_1 + (y_2 + y_3)x_2 \le 200y_1 + 300y_2 + 400y_3$

Rewrite: Make left side look like the original max/min goal, so right will be an upper bound

So here:

$$(y_1+y_3)x_1 + (y_2+y_3)x_2 \leq 200y_1 + 300y_2 + 400y_3$$

Means:

$$\boxed{1x_1 + 6x_2} \leq \boxed{200y_1 + 300y_2 + 400y_3}$$

if:
$$\begin{cases} y_1, y_2, y_3 \geq 0 \\ y_1+y_3 \geq 1 \\ y_2+y_3 \geq 6 \end{cases}$$

] b/c if negative, inequalities flip!

} b/c original eqn.

Any y_i's would give an upper bound!

We want the best one

↳ ie minimize another LP!

Duality:

$$\begin{array}{lll} & \max & x_1 + 6x_2 \\ & \text{s.t.} & \\ y_1 & & x_1 \le 200 \\ y_2 & & x_2 \le 300 \\ y_3 & & x_1 + x_2 \le 400 \\ & & x_1, x_2 \ge 0 \end{array} \quad \overset{\text{dual}}{\Longleftrightarrow} \quad \left\{ \begin{array}{ll} \min & 200y_1 + 300y_2 + 400y_3 \\ \text{s.t.} & \\ & y_1 + y_3 \ge 1 \\ & y_2 + y_3 \ge 6 \\ & y_1, y_2, y_3 \ge 0 \end{array} \right.$$

Any solution to bottom is upper bound to top LP.

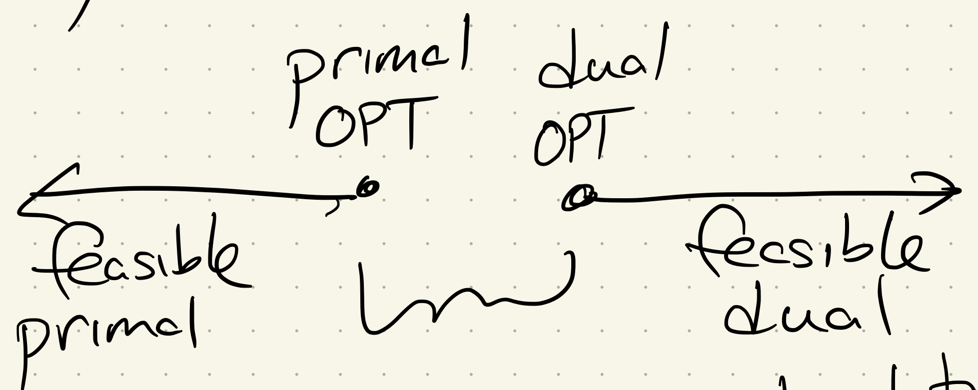
$\Longrightarrow$ If we can find primal/duals that are equal, both are OPT

Here, 1900: primal $(x_1, x_2) = (100, 300)$

Dual: $(y_1, y_2, y_3) = (0, 5, 1)$

This is just like max flow/min cut duality, in a way.

Works for any LP:



this gap – the duality gap – is $= 0$.

For ILP $\rightarrow$ can be > 0.

In general:

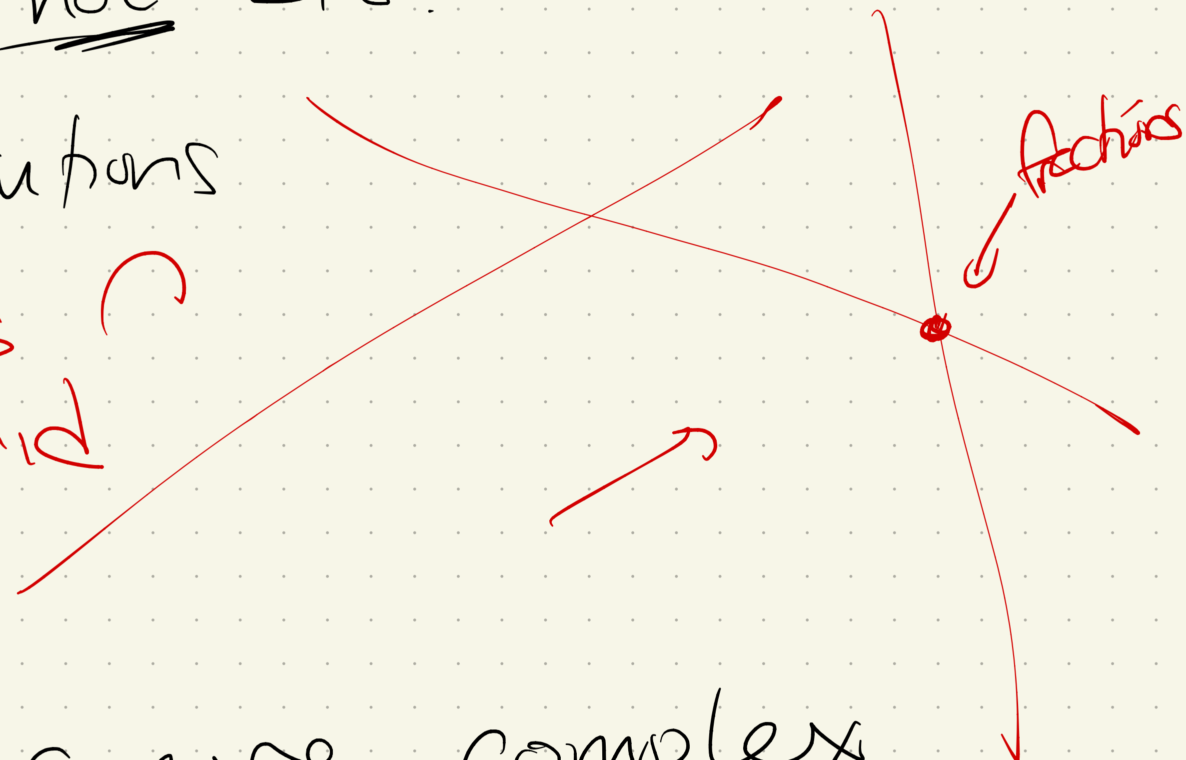
Primal LP	Dual LP
$\max \; c^T x$	$\min \; y^T b$
s.t.	s.t.
$A\vec{x} \leq \vec{b}$	$y^T A \geq c^T$
$\vec{x} \geq 0$	$y \geq 0$
$\vec{c}, \vec{x}, \vec{b}$ vectors	$\vec{c}, \vec{b}, \vec{y}$, vectors

Limits of LP:

Many things are not LPs!

Ex: Integer solutions

If eqns not on grid points!

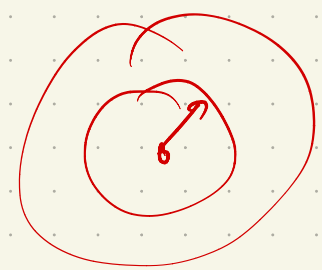


Ex: Quadratic or more complex constraints

→ distance fcns

x_1, y_1 ∘ ∘ x_2, y_2

$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ not linear



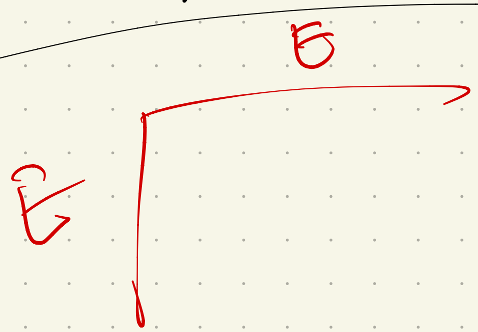
Often other approaches could give a better runtime!

Ex: Flows & cuts!

Orlin: $O(VE)$ via a combinatorial approach

$\boxed{\begin{array}{l}\underline{\text{LP}}: \\ E \text{ variables,} \\ V+E \text{ equations}\end{array}}$ $\Rightarrow$

- LP w/simplex alg. is exponential
- with "better" algorithm matrix multiply time (so ~ E^3)



The algorithm: Simplex (Dantzig 1947)

Assumes canonical form:

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

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

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

So:

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

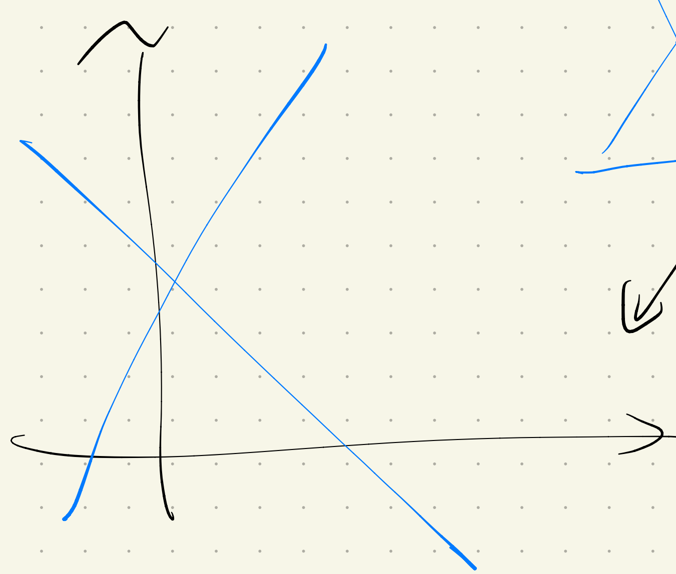
Klee-Minty, 1973: some feasible polytopes have $O(n^{\lfloor d/2 \rfloor})$ vertices

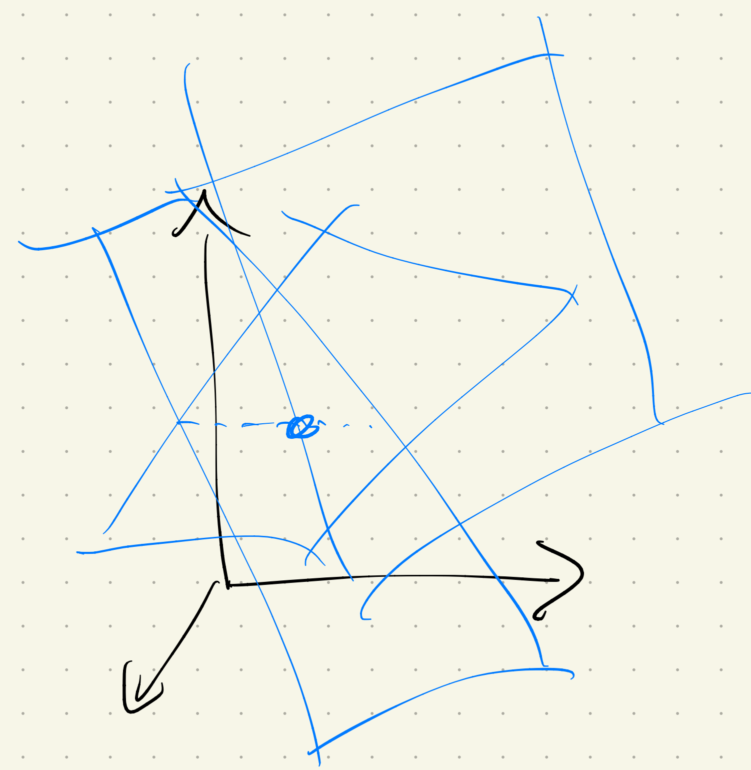
Algorithm (simplex):

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

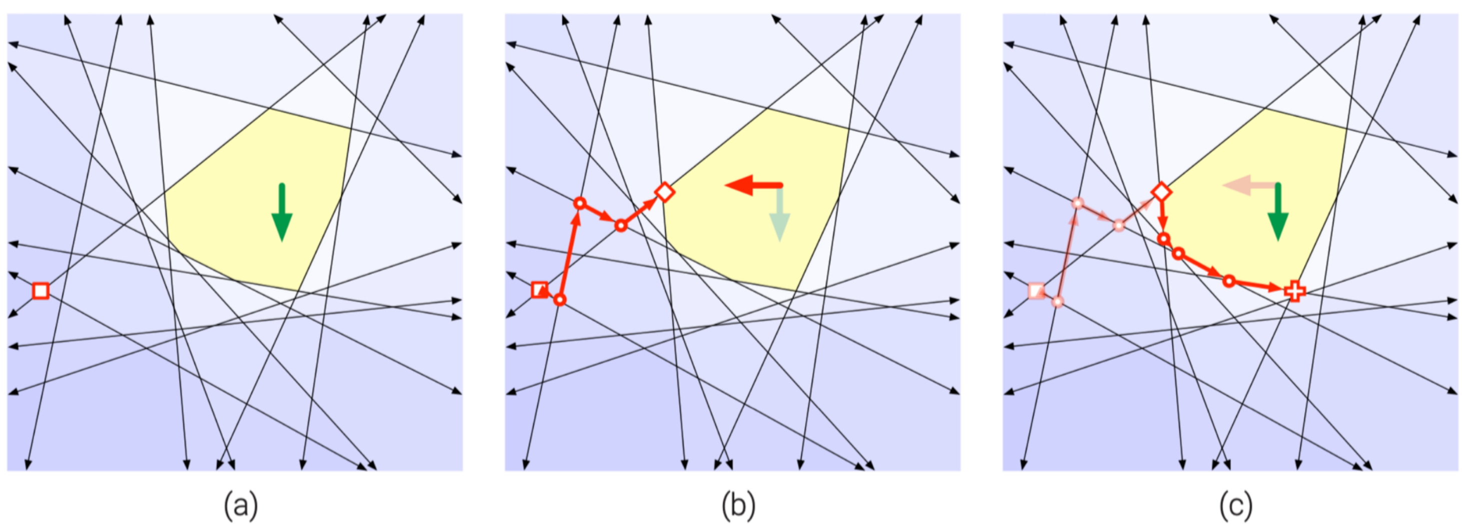


Figure I.2. Primal simplex with dual initialization: (a) Choose any basis. (b) Rotate the objective to make the basis locally optimal, and pivot "up" to a feasible basis. (c) Pivot down to the optimum basis for the original objective.

How to pick where to move?

No ideal way!

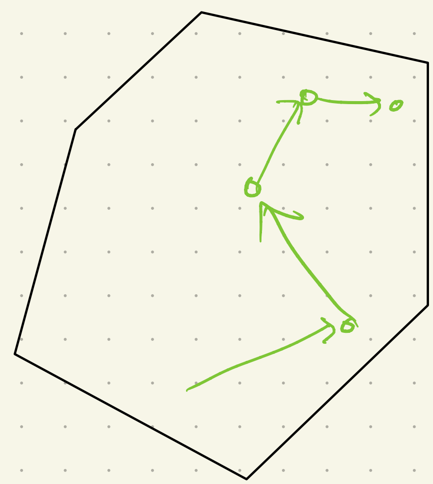
Many proposed, but for almost every one, there is <u>some</u> input polyhedron that needs an <u>exponential</u> number of pivots.

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