Overview of LP

Sunday, June 11, 2023 4:44 PM

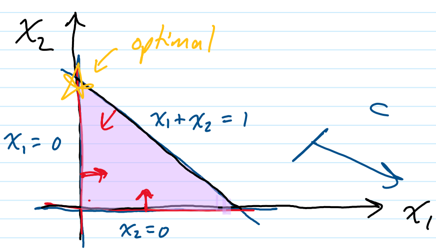
Linear Programming:

"Optimize a linear objective function over a $\underbrace{\text{set Linear constraints}}_{\text{a polyhedron}}$"

$$c^T x = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad c = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

ex)
$$\begin{pmatrix} \text{minimize} & 2x_1 - x_2 \\ \text{s.t.} & x_1 + x_2 \le 1 \\ & x_1 \ge 0 \\ & x_2 \ge 0 \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{x} \ge \underbrace{\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}}_{b}$$



In general,

$$\begin{pmatrix} \min & c^T x & & \\ \text{s.t.} & a_i^T x \ge b_i & i \in M_1 \\ & a_i^T x \le b_i & i \in M_2 \\ & a_i^T x = b_i & i \in M_3 \\ & x_j \ge 0 & j \in N_1 \\ & x_j \le 0 & j \in N_2 \end{pmatrix}$$

indexing sets

sign constraints

$A \in \mathbb{R}^{m \times n}$

$$A = \begin{pmatrix} - & a_1 & - \\ & \vdots & \\ - & a_m & - \end{pmatrix}$$

In "Standard Form",

$$\boxed{\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \ge 0 \end{pmatrix}}$$

$$Ax = \begin{pmatrix} a_1^T x \\ \vdots \\ a_n^T x \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

$x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

($x \ge 0$ means $x_i \ge 0 \ \forall i$)

all LPs can be reduced to standard form by

- eliminating free variables: $x \in \mathbb{R}^n \xrightarrow[\text{with}]{\text{replace}} (x^+ - x^-)$, $x^+, x^- \ge 0$
- eliminating inequalities:

$$a_i^T x \le b_i \xrightarrow[\text{"slack" variable}]{\text{add}} s_i + a_i^T x = b_i$$

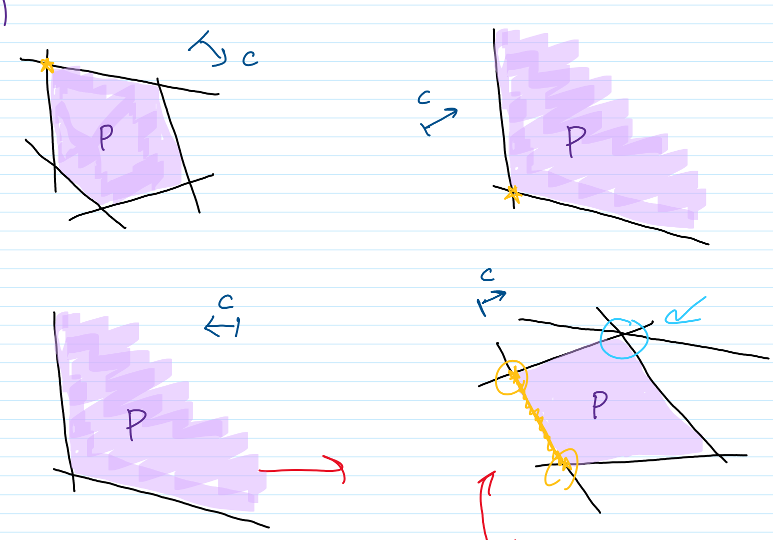
(these variables "absorbed" into new x vector)

Due to the geometry of the polyhedrons, we have:

Thm: If the polyhedral feasibility set has at least one vertex:

then there exists a vertex that is optimal for the LP,
or the optimal cost is $-\infty$

ex)



opt $= -\infty$

line segment of solns, w/ 2 vertices

$x \leq$

s

$\downarrow$

$$\bar{x} = \begin{pmatrix} x \\ s \end{pmatrix} \in \mathbb{R}^{n+1}$$

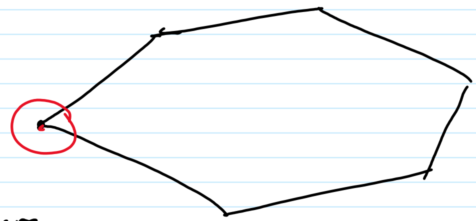
The Simplex Alg.

Sunday, June 11, 2023 5:28 PM

Reformulate in Standard form: $\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{pmatrix}$ $x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

(Assume $m < n$, else overdetermined)

KEY INSIGHTS



- At any vertex, $\bar{x}$, of the polyhedron: $\exists$ m coordinates $\bar{x}_{B(1)}, \dots, \bar{x}_{B(m)}$ whose corresponding columns of A $(A_{B(1)}, \dots, A_{B(m)})$ are linearly independent

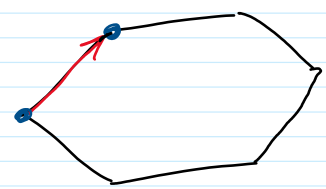
  (basic variables)

  indices

  $A = \begin{pmatrix} | & & | \\ A_1 & \cdots & A_n \\ | & & | \end{pmatrix}$

  The remaining $n-m$ coordinates of $\bar{x}$ are zero

  (non-basic variables)



- Moving from a vertex to an adjacent vertex swaps a basic variable with a nonbasic variable

- Can determine the change to the objective value per unit increase in one of the non basic variables while adjusting the basic variables to maintain feasibility, i.e.

$$\bar{c}_j = \underbrace{c_j}_{\text{cost of increasing variable } j} - \underbrace{c_B^T B^{-1} A_j}_{\text{cost of adjusting other basic variables to remain feas.}} \quad \text{(reduced cost of variable } x_j\text{)}$$

(Simplified)

Simplex Alg [Dantzig 1947]

1) Start @ a vertex
2) Compute $\bar{c}_j$ $\forall$ nonbasic x_j

   If $\bar{c}_j \geq 0$ $\forall j$: the current vertex is optimal

   Else choose some j' with $\bar{c}_{j'} < 0$
3) "Replace" the "appropriate" basic varible $x_{B(l)}$ by $x_{j'}$

Else choose some j with $c_j' < 0$

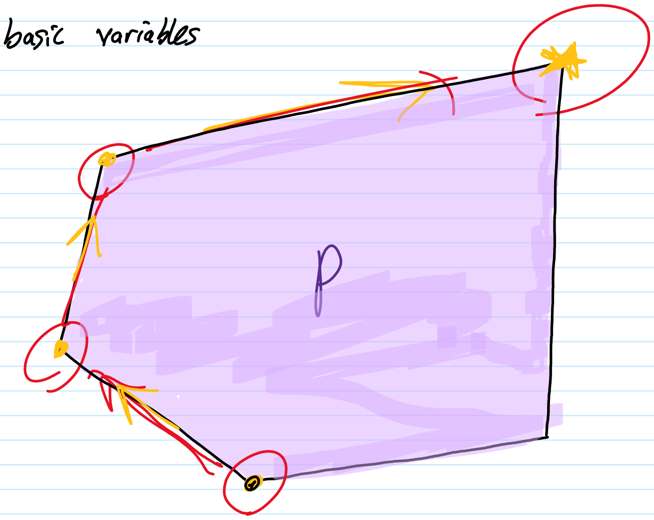
3) "Replace" the "appropriate" basic varible $x_{B(\ell)}$ by x_j

Set $x_{B(\ell)} = 0$ and update basic variables

go to (2)

(How far to go in chosen direction is also determined by alg., but we omit for brevity)

- see Intro to Linear programming by Bertsimas, Tsitsiklis for full details



Column Geometry of Simplex Method

$$\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & e^T x = 1 \\ & x \geq 0 \end{pmatrix}$$

convexity constraint $\to$ $x_1 + \cdots + x_n = 1$

(All LPs can be put in this form by scaling)

let $z = c^T x$ and $A = \begin{bmatrix} | & & | \\ A_1 & \cdots & A_n \\ | & & | \end{bmatrix}$

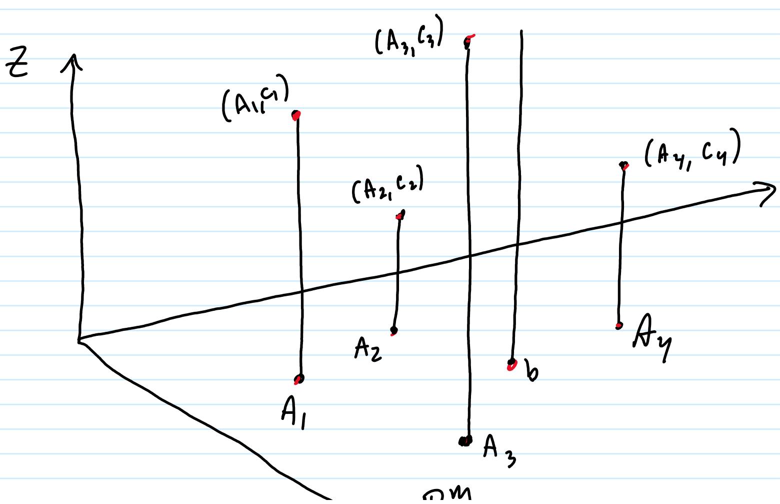
$\Updownarrow$

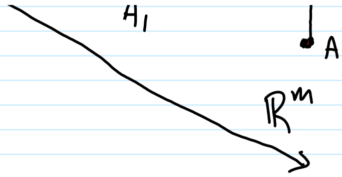
minimize z s.t. $x \geq 0$, $e^T x = 1$, and

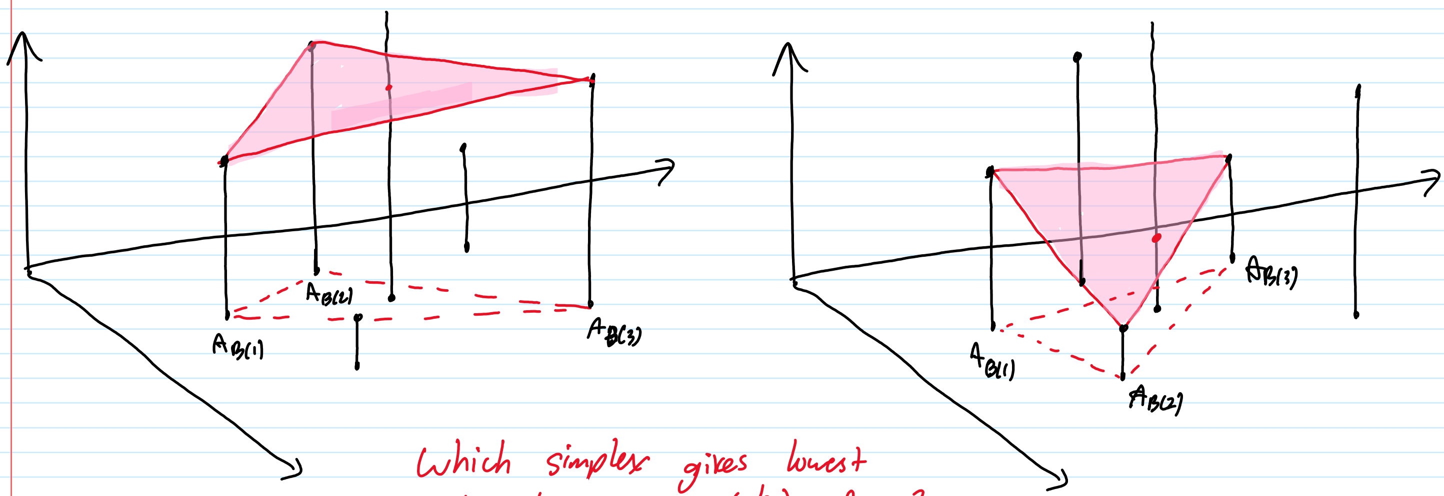
$$x_1 \begin{bmatrix} A_1 \\ c_1 \end{bmatrix} + \cdots + x_n \begin{bmatrix} A_n \\ c_n \end{bmatrix} = \begin{bmatrix} b \\ z \end{bmatrix}$$

$x_1 A_1 + \cdots + x_n A_n = b$

$x_1 c_1 + \cdots + x_n c_n = z$







Which simplex gives lowest intersection with $\binom{b}{z}$ line?

Computational Efficiency

☆ Exponential Runtime in Worst Case ☆

ex) Klee-Minty Cube

$D = 2$ case :

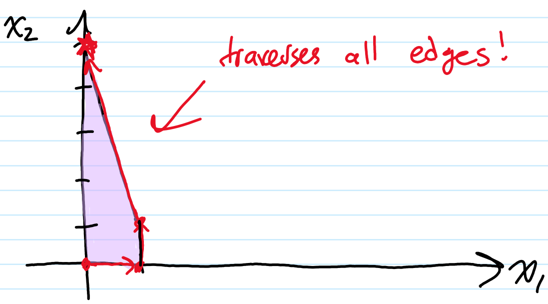
$$\begin{pmatrix} x_1 \le 5 \\ 4x_1 + x_2 \le 25 \\ x_1, x_2 \ge 0 \end{pmatrix} =: P$$

Consider :

running simplex on

$$\begin{pmatrix} \max & 2x_1 + x_2 \\ \text{s.t.} & x \in P \end{pmatrix}$$

starting @ $x = (0,0)$



HOWEVER:

Simplex perform well in practice

¿Why?

Duality

Sunday, June 11, 2023 5:52 PM

$$\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{pmatrix} \quad (LP)$$

Let x^* be the optimal

$\downarrow$ penalty relaxation

$$\min_{x \geq 0} \; c^T x + y^T (b - Ax) \quad =: \quad g(y)$$

claim: $g(y)$ lower bounds LP $\forall y$

pf: $$g(y) = \min_{x \geq 0} c^T x + y^T (b - Ax) \leq c^T x^* + \cancel{y^T (\underbrace{b - Ax^*}_{=0})}$$
$$= c^T x^*$$

Let's find the maximum lower bound.

Rewriting, $$g(y) = y^T b + \underbrace{\min_{x \geq 0} (c^T - y^T A) x}$$

$= 0$ if $c^T - y^T A \leq 0$

$= -\infty$ otherwise

Thus, $\max_y g(y)$

$\Updownarrow$

$$\begin{pmatrix} \max & b^T y \\ & A^T y \leq c \end{pmatrix} \quad (DP) \quad \text{dual problem}$$

Thm (Weak Duality): $(DP) \leq (LP)$

ie. $\forall$ feasible x for LP and feasible y for DP:

$b^T y \leq c^T x$

ie, $\forall$ feasible x for LP and feasible y for DP:

$$b^T y \le c^T x$$

pf: Notice

- $y^T\overbrace{(Ax - b)}^{=0} = 0$
- $\underbrace{(c^T - y^T A)}_{\ge 0}\underbrace{x}_{\ge 0} \ge 0$

Thus,

$$0 \le (y^T A x - y^T b) + (c^T x - y^T A x)$$
$$= -y^T b + c^T x$$

Thm (Strong Duality):

If LP has an optimal soln, so does DP, and the optimal objective costs are equal

pf: See Bertsimas '97

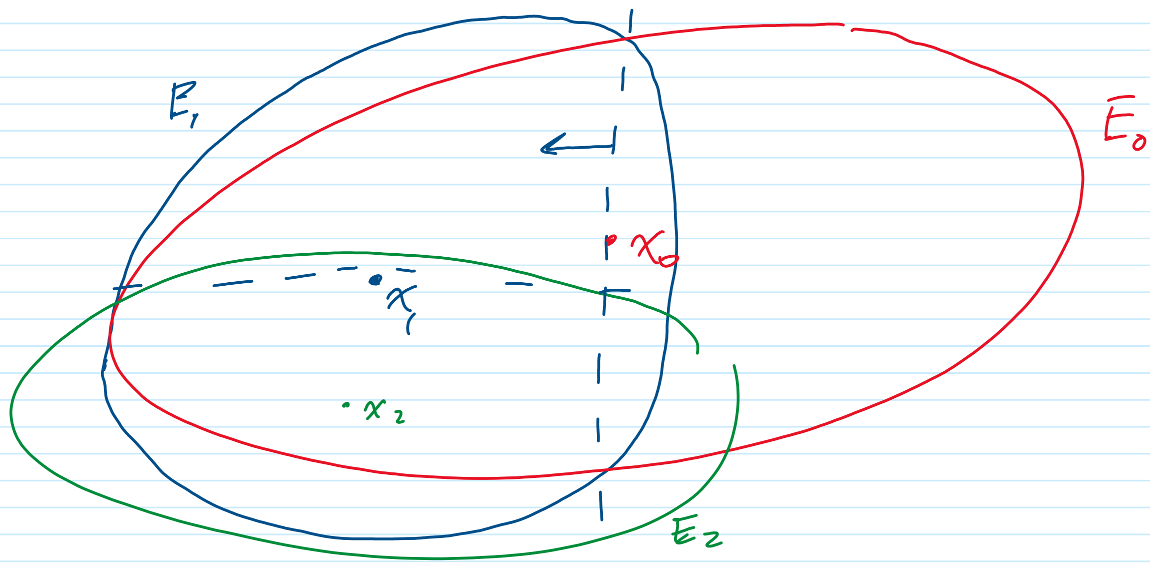
Other LP alg.s

Sunday, June 11, 2023 5:45 PM

Ellipsoid Algorithm

- Determines if polyhedron P is nonempty and if so returns an $x \in P$

ie solves $\begin{pmatrix} \text{find } x \\ \text{s.t. } x \in P \end{pmatrix}$



How this solve LP's?

$$\underbrace{\begin{pmatrix} \min & c^T x \\ \text{s.t.} & Ax \geq b \end{pmatrix}}_{LP}, \quad \underbrace{\begin{pmatrix} \max & b^T y \\ \text{s.t.} & A^T y = c \\ & y \geq 0 \end{pmatrix}}_{DP}$$

By duality, LP and DP have optimal solns iff the following system is feasible:

$$P := \begin{cases} b^T y = c^T x & \text{strong duality} \\ Ax \geq b & \text{LP feas.} \\ A^T y = c & \\ y \geq 0 & \text{DP feas.} \end{cases}$$

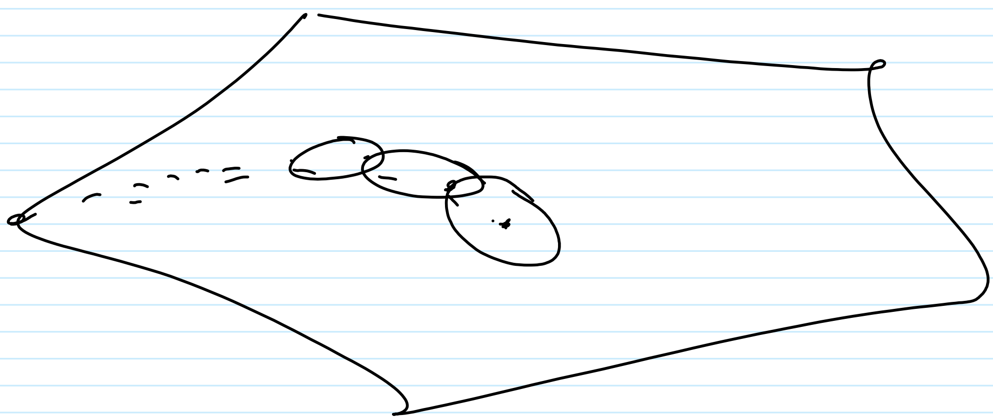
$$P := \left\{ \boxed{\begin{array}{l} A^T y = c \\ y \geq 0 \end{array}} \right.$$ DP feas.

Complexity:

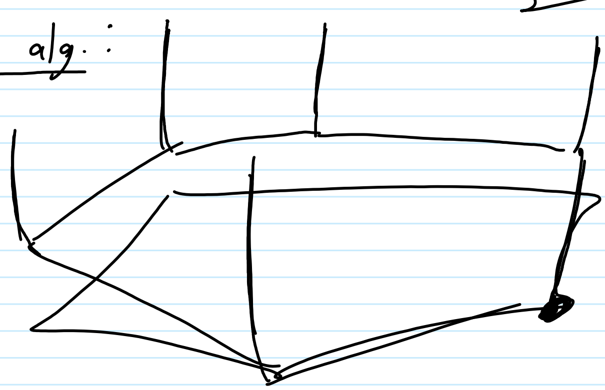
- polynomial runtime
- HOWEVER: Inefficient in practice :(

Interior Point Methods

- **Affine Scaling alg:**



- **Path Following alg.:**



Complexity:

- poly. runtime!
- Efficient in practice :)
- Spearheaded by [Karmarkar 1984]

"Karmarkar's algorithm"

Smoothed Analysis

Sunday, June 11, 2023 5:52 PM

- Why does simplex work so well?
- Exponential runtime

~~Often~~ outperforms poly-time competitors

KEY OBSERVATIONS [Spielman and Teng 2001]

- The worst cases (exp. runtime) are VERY rare

↓

- Worst cases are VERY fragile
  - run in poly time given random perturbation (perturbations to data: A, b)

↓

- Every input has a neighborhood containing inputs on which the Alg will perform well