Linear optimization

(P) $\max \{ c^T x \mid Ax = b,\ x \geq 0 \}$ $\quad A \in \mathbb{R}^{m \times n}$

(D) $\min \{ b^T y \mid A^T y \geq c \} = \min \{ b^T y \mid A^T y - s = c,\ s \geq 0 \}$

Theorem (Weak Duality)

If x is feasible for P, y feasible for D, then

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

Theorem (Strong Duality)

For each pair of linear programs, one of the following is true:

1. (P), (D) are both infeasible
2. (P) unbounded, (D) infeasible
3. (P) infeasible, (D) unbounded
4. (P) feasible, (D) feasible

$$\Rightarrow \quad b^T y^* = c^T x^* \quad \text{☺}$$

Thm (Complementary Slackness)

When x^*, y^* are optimal solutions of P, D, then

$$x_i^* s_i^* = 0 \quad , i = 1, \dots, n$$

Moral: Solving LP's results to solving the following system of equations and inequalities

$$Ax = b \quad , \quad A^T y - s = c \quad , \quad x^T s = 0$$

$$x \geq 0 \quad , \quad s \geq 0$$

Simplex Method (1947 George Dantzig)

Satisfy $Ax=b$, $A^Ty-S=c$, $x \geq 0$

Defn: X is a Basic Feasible solution if $\exists$ a submatrix A_B of A with rank m (invertible) such that

$$x_j = 0 \text{ if } j \notin B$$

(therefore $A_B x_B = b$, $x_B = A_B^{-1} b$)

Lemma: If there exists an optimal solution of (P), then there exists an optimal BFS of (P).
(Assume bounded solution)

proof: Need 2 things:

1. Every BFS = an extreme point of (P) and vice versa
2. Every point in P can be written as a convex combination of extreme points + direction of unboundedness.

hence $x^* = d + \sum \alpha_i v_i$

- note that $c^T d = 0$ since otherwise we could scale in that direction

then $c^T x^* = \sum \alpha_i c^T v_i \leq \max c^T v_i \Rightarrow c^T x^* = c^T v_i$

□

Representation

$$b = Ax = A_B x_B + A_N x_N \Rightarrow x_B = A_B^{-1} b - A_B^{-1} A_N x_N$$

$$z = c^T x = c_B^T x_B + c_N^T x_N$$
$$= c_B^T A_B^{-1} b + (c_N^T - \underbrace{c_B^T A_B^{-1}}_{y} A_N) x_N$$

$$c_j - y a_j = s_j$$

Pivot Rule: Select index from those non-basic variables that correspond to

$$s_j < 0$$

i.e. $c_j - y^T a_j = -s_j > 0$.

Many choices for pivot rules!!!

Entering variable: Call a_j the column vector of the entering variable $x_j = 0$.

increase x_j slowly by $t > 0$

$$x_B^{new} = x_B^{old} - t(A_B^{-1} a_j)$$

⇒ eventually one other variable will vanish
→ this variable leaves the basis
→ OR no variable vanishes ⇒ unbounded direction.

How to find first BFS? $Ax = b$, $x \geq 0$, $b \geq 0$ (otherwise multiply by -1)

Set $\bar{A} = (A, I_m)$, $\bar{x} = (x, x_{n+1}, \ldots, x_{n+m})$

Solve $\bar{A}\bar{x} = b$, $\bar{x} \geq 0$, using $\bar{x}_0 = (0, b)$

i.e.

$$\max \; -(x_{n+1} + \ldots + x_{n+m})$$
$$\text{s.t.} \quad \bar{A}\bar{x} = b$$

if $\max = 0$ ⇒ we found an initial BFS
if $\max < 0$ ⇒ original problem is infeasible.

Open Problem: Can you find a pivot rule s.t. the number of pivot rules necessary to get an optimal solution is polynomial in n & m?

Can you bound the diameter of a graph in n & m?

Interior Point Methods (1984 Karmarker)

Theorem: For all $\lambda > 0$, [redacted]* has a unique real solution $(x^*(\lambda), y^*(\lambda), s^*(\lambda))$ s.t.

1.) $x^*(\lambda)$ is an optimal solution to

$$\max\left(c^T x + \sum_{i=1}^{n} \lambda \log(x_i) \;\middle|\; Ax = b,\ x \geq 0\right)$$

2.) In the limit, the point $(x^*(0), y^*(0), s^*(0))$ is an optimal solution to (P).

Defn: The set of all real solutions of * as $\lambda \to 0$ is the Central Path

$$* \quad Ax = b,\ x_i s_i = \lambda,\ A^T y - s = c,\ x \geq 0,\ s \geq 0$$

Steps of IPM: 1. Set $\lambda = 1$, solve $*$, find initial point (x, y, s) (or close enough point)

Given $\epsilon > 0$,

while $\lambda > \epsilon$ REPEAT

$$\lambda \to \left(1 - \frac{1}{2\sqrt{n}}\right)\lambda$$

$$(x, y, s) \to (x + \Delta x, y + \Delta y, s + \Delta s)$$

$$A(x + \Delta x) = b \qquad A^T(y + \Delta y) - (s + \Delta s) = c$$

$$(x_i + \Delta x_i)(s_i + \Delta s_i) = \lambda$$

$$s_i + \Delta s_i \geq 0 \qquad x_i + \Delta x_i \geq 0$$

Cheat: $A(\Delta x) = 0$, $A^T(\Delta y) - \Delta s = 0$

$$s_i \Delta x_i + x_i \Delta s_i = \hat{\lambda}$$

LP Applications

- game theory
- combinatorial optimization
  - max flow / min cut
  - help Integer Programs
  - Approximation Algorithms

game theory:

2-person, 0-Sum game

win for one person = loss for other person

"robbers" Player 2 \ player 1 "cops"	1	2	3
1	4	-10	-10
2	-8	5	-8
3	-12	-12	9

pay off for player 1 = -(payoff for player 2)

$\Delta X = \{x \mid \sum x_i = 1\}$ $\Delta y = \{y \mid \sum y_i = 1\}$

Mixed strategy: Police patrol site i with probability x_i

robbers attack site i with probability y_i

$\sum x_i = 1$ $\sum y_i = 1$

Expected pay off: $\sum_{i,j} a_{ij} x_j y_i = y^T A x$ ← Police want to maximize this

Police try to find $\bar{x} \in \Delta x$ s.t.

$$\bar{x} = \arg\max_{x \in \Delta x} \left(\min_{y \in \Delta y} y^T A x \right)$$

$\Rightarrow$ Police can guarantee a payoff of at least

$$\text{Payoff} \geq \max_x \left(\min_y y^T A x \right)$$

Similarly, let $\bar{y} = \arg\min_{y \in \Delta y} \left(\max_{x \in \Delta x} (y^T A x) \right)$

Then $\max_x \min_y y^T A x \leq \bar{y}^T A \bar{x} \leq \min_y \max_x y^T A x$

Thm (Von-Neuman)

$$\max_x \min_y y^T A x = \min_y \max_x y^T A x$$

Corollary: $\bar{x}, \bar{y}$ is an equilibrium point

i.e. $\bar{y}^T A x \le \bar{y}^T A \bar{x}$

and $y^T A \bar{x} \ge \bar{y}^T A \bar{x}$

$\Rightarrow$ no one has incentive to move.

Pseudo-Proof of Theorem:

Consider $\max_{x \in \Delta x} \min_{y \in \Delta y} y^T A x$

If we fix a particular x^*, then $\min_y y^T A x^* = \min_i (Ax^*)_i$

("Put all chips in one bucket")

LP formulation:

(Primal)

$$v = \max z$$
$$\text{s.t.} \quad z \le q_i x \qquad q_i = i^{th} \text{ row of } A, \text{ it runs of } A.$$
$$\sum x_i = 1$$
$$x_i \ge 0$$

$\Rightarrow$ (Dual)

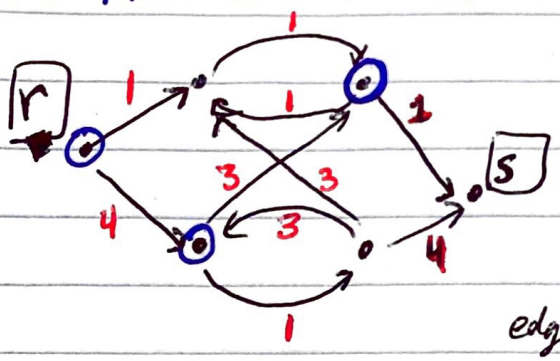
$$\min \gamma$$
$$\text{s.t.} \quad \gamma \ge y a^i$$
$$\sum y_i = 1$$
$$y \ge 0$$

Dual equivalent to

$$\min_y \max_x y^T A x.$$

Since $opt(Primal) = opt(dual)$ $\Rightarrow$ Result $\square$

Max-flow / min-cut Problem



Consider max flow from left to right

max flow ≥ min cut.

edges leaving ⊙ is 4

Theorem: Given a network/directed graph with vertices r, s, the maximum flow from r to s is exactly equal to the min cut between r & s.

Defn: A r-s cut is a partition of the vertices

$V = R \sqcup S$

s.t. $r \in R$, $s \in S$, $S \cap R = \emptyset$

Defn: For a vertex v, $\delta^+(v)$ = Set of all edges leaving v.

$\delta^-(v)$ = Set of all edges entering v.

LP formulation:

- let x_e for all $e \in E$ be the amount of flow along the edge.
- for a vertex v, $x(\delta^+(v)) = \sum_{e \in \delta^+(v)} x_e$

and $x(\delta^-(v)) = \sum_{e \in \delta^-(v)} x_e$

Max flow

$$\max \quad x(\delta^+(r)) - x(\delta^-(r))$$

s.t.

$$x(\delta^+(v)) = x(\delta^-(v)) \quad \forall\, v \neq r, s$$

$$x_e \geq 0 \quad , \quad x_e \leq c_e$$

Min Cut

$$\min \sum_{e \in E} c_e y_e$$

s.t. $z_r = 1, \; z_s = 0$

$$z_w - z_v + y_{vw} \geq 0 \quad \forall\, e \in E$$

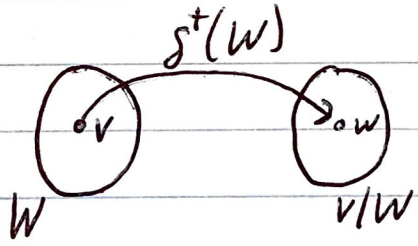
$$y_e \geq 0, \; z_v \in \mathbb{R}$$

Complementary Slackness

Let x^*, y^*, z^* be optimal solutions

then $y_e^*(c_e - x_e^*) = 0, \; (z_w - z_v + z_{vw}) \cdot x_{vw} = 0$

? $W = \{v \mid z_v^* \geq 1\}$, any edge in $\delta^+(W)$.

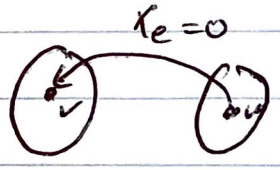


$z_v^* \geq 1, \; z_w^* < 1$

$(z_w^* - z_v^* + y_e^*) \geq 0 \rightarrow y_e^* > 0 \Rightarrow x_e^* = c_e$

"edge is at full capacity"

new case



for any edge in $e \in \delta^-(W)$, $e \in V/W$

$z_w^* - z_v^* + y_e^* \geq z_w^* - z_v^* > 0$

$z_v^* < 1, \; z_w^* \geq 1 \Rightarrow x_e = 0$

$\Rightarrow$ Capacity of the cut is equal to the flow value.

□

LP Solves IP

Theory of Totally Unimodular Matrices

Defn: A is called Totally Unimodular if every square submatrix has determinant $\pm 1, 0$.

Thm: $A = T.U.$ then

$$\min \{c^T x \mid Ax = b,\ x \geq 0\}$$

always has integral optimal solutions for every integral vector b.

Proof: Let x^* be the opt solution.

Then $\exists$ A_B $(m \times m)$ s.t. $x_B^* = A_B^{-1} b$, $x_N = 0$.

A_B^{-1} is integral.

$\Rightarrow$ by Cramers rule, x_B^* is integral.

Thm: Let A be a $\pm 1, 0$ matrix. If every column of A has at most one $+1$ and one -1, then A is T.U.

Proof: Let B be a submatrix of A.

if B has a 0 column (or row) then $\det(B) = 0$.

if B has a column w/ exactly 1, or -1, then expand about this column (induction)

if B has all 1 & -1 columns, then sum of rows $= 0$.

$\Rightarrow$ linearly dependent $\Rightarrow \det(B) = 0$.

D

Prop: A is T.U. $\iff \begin{bmatrix} A \\ I \end{bmatrix}$ is T.U. $\iff \begin{bmatrix} A \\ I \end{bmatrix}$ is unimodular.

Vertex Cover:

- Find a subset of vertices which touch all edges.

Approximate Solution:

given weights c_v, $x_v \in \{0,1\}$

$$\min \sum_{v \in V} c_v x_v$$

$$\text{s.t.} \quad x_v + x_w \geq 1 \quad \forall \ (v,w) \in E$$
$$0 \leq x_v \leq 1 \quad \forall \ v \in V.$$

Solve using LP to get x^*.

- If $x^* \in \mathbb{Z}^n$, done!
- If not, let $\boxed{S = \{v \mid x_v^* \geq 1/2\}}$

Claim 1: S is a vertex cover

Claim 2: $\boxed{\text{weight}(S) \leq 2\,\text{weight}(S^*)}$

$$= 2 \sum_{v \in V} c_v x_v^*$$

□

Ellipsoid Method (1979) Kachiyan

input: a convex set $S \subseteq \mathbb{R}^n$
given to you by a separation oracle
& $\in S$ s.t. $Vol(S) > \epsilon$.

Output: A point $x \in S$ or "$S = \emptyset$."

Idea: Assume $\exists$ an ellipsoid

$$E_{M,z_0} = \{x \in \mathbb{R}^n \mid (x-z_0)^T M (x-z_0) \leq 1\}$$

M = positive definite

$S \subseteq E_{M,z_0}$, z_0 = center of E

* Check if $z_0 \in S$.
→ if $z_0 \notin S$, find new ellipsoid that encompasses $H_k^+ \cap E_k$.

Construct $E_{M_{k+1}, z_{k+1}}$ & Repeat while

$$Vol(E_{M_k z_k}) > vol(S).$$

Lemma: one can explicitly construct from $E_{M_k z_k}$ & H_k,
the exact equations for $E_{M_{k+1} z_{k+1}}$

with minimum volume &

$$Vol(E_{M_{k+1} z_{k+1}}) < Vol(E_{M_k, z_k}) \cdot e^{\frac{-1}{2(n+1)}}$$

After k steps

$$Vol(E_{M_k z_k}) < Vol(E_{M_0, z_0})\, e^{\frac{-k}{2(n+1)}}$$

Note: $Vol(S) < Vol(E_{M_0 z_0})\, e^{-k/2(n+1)}$

$$\Rightarrow \quad k \leq 2(n+1) \ln\left(\frac{Vol(E_{M_0, z_0})}{Vol(S)}\right)$$

Also: If $P = \{y \mid A^T y \leq b\}$, $A = n \times n$ matrix,

if $P \neq \emptyset$, then it must contain a ball of radius $2^{-7\phi^3}$ where $\phi = O(m \cdot n \cdot \max(\log_2 |a_{ij}|, \log_2 |b_i|))$

History: 1989 (Renegar, Gonzaga, Roos\Vial)

find an optimal ~~so~~ Interior Point Method in $O(n^3 L)$

2000 (Deza, Terlaky) → best possible ever.

Simplex Counter example (exponential)

Klee-Minty Cube

$0 \leq x_1 \leq 1, \quad \epsilon x_{k-1} \leq x_k \leq 1 - \epsilon x_{k-1}, \quad k = 2, \ldots, n$

Other algorithms for LP:

1.) Criss-Cross Method
2.) Perceptron Algorithm
3.) Motzkin Relaxation Algorithm
4.) Ven pala Style Algorithms

Open: Prove any of these are polynomial

Curvature

Total curvature = length of the Gauss map

Curve $\gamma \rightarrow \frac{\dot{\gamma}}{\|\gamma\|}$ tangent vector

- Gauss map takes point on curve to tangent vector pointing on the sphere

Open: $2\pi n \geq$ total curvature

Strongly Polynomial = $O(n^k)$ k constant

* Holy grail for LP *

Polynomial Optimization

$$\begin{aligned} \max \quad & h(x) \\ \text{s.t.} \quad & f_1(x) = 0, \ \ldots, \ f_m(x) = 0, \\ & g_1(x) \geq 0, \ \ldots, \ g_k(x) \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

Almost every combinatorial optimization can be modeled like this

Vertex Cover

Find a minimum size vertex set that covers every edge.

- $\min \ \sum x_i$
  $\text{s.t.} \quad x_i + x_j \geq 1 \quad (i,j) \in E$
  $x_i^2 = x_i \quad (x_i \in \{0,1\})$

  $x_i = \begin{cases} 1, \text{if } i \in V \\ 0 \text{ if } i \notin V \end{cases}$

OR.

- $\max \ \sum x_i$
  $\text{s.t.} \quad x_i \cdot x_j = 0$

  $x_i = \begin{cases} 0 \text{ if } i \in V \\ 1, \text{if } i \notin V. \end{cases}$

Binary Search

We instead solve the feasibility problem

$$\begin{aligned} & h(x) - \gamma \geq 0 \\ & f_i(x) = 0 \qquad i = 1, \ldots, m \\ & g_i(x) \geq 0 \qquad i = 1, \ldots, k \end{aligned}$$

When you know a range on values for objective function, we can do binary search with γ.

Goal: To understand the feasibility question with polynomial constraints.

Problem Setup: $\mathbb{K}$ = field, $\mathbb{K}[x_1, \dots, x_n] = \mathbb{K}[x] = R$

(ring of polynomials with coefficients in $\mathbb{K}$)

$x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ (monomial)

Solve $\begin{cases} f_i(x) = 0 & i = 1, \dots, m \\ g_i(x) \geq 0 & i = 1, \dots, k \end{cases}$

$p(x) = \sum_\alpha u_\alpha x^\alpha$, $\text{degree}(p(x)) = \max_{\alpha : u_\alpha \neq 0} \left(\sum_{i=1}^{n} \alpha_i \right)$

$F := \{ f_1, \dots, f_m \}$

$\langle F \rangle_R = \left\{ \sum_{i=1}^{m} \beta_i f_i \mid \beta_i \in R \right\}$ "Ideal generated by F"

Defn: A polynomial $s(x)$ is a Sum of Squares (SOS) if

$$s(x) = \sum_{i \in I} [g_i(x)]^2$$

Defn: Let $g_1, \dots, g_k \in R$

$$\text{Cone}(G) = \left\{ s_0 + \sum_i s_i g_i + \sum_{\{i,j\}} s_{ij} g_i g_j + \dots + s_{12\dots k} g_1 g_2 \dots g_k \right.$$

$$\left. s_\alpha \text{ is a SOS polynomial} \right\}$$

Lemma: Fredholm's Alternative

$\nexists\, x$ such that $Ax + b = 0 \iff \exists\, u$ such that $u^T A = 0$, $u^T b = 1$

Theorem Hilbert's (weak) Nullstellensatz

$\nexists\, x$ such that $f_i(x) = 0$, $i = 1, \dots, m$

$\iff \exists\, \beta_1, \dots, \beta_m \in R$ s.t. $\sum \beta_i f_i = 1$

$\iff 1 \in \langle F \rangle_R$

(Proof by Cox, Little, & O'Shea)

General Farkas' Lemma:

$\nexists\, x$ s.t. $Ax+b=0,\ Cx+d \geq 0$

$\Leftrightarrow \exists\, \mu, \lambda$ with $\lambda \geq 0$ $\nexists$

$$\mu^T A + \lambda^T C = 0$$
$$\mu^T b + \lambda^T d = -1$$

Proof uses: Hahn Banach Theorem = Separation Hyperplane Theorem

Theorem: Positivstellensatz (Stengle 1973)

$\nexists\, x$ such that $f_i(x)=0,\ i=1,\dots,m$

$\nexists\, g_i(x) \geq 0,\ i=1,\dots,k$

$\Leftrightarrow$

$\exists\ f \in \langle F \rangle_R,\ g \in \text{Cone}(G)$ such that $f+g=1$

Example: $f_1(x) = x_1^2 - 1$, $f_2(x) = 2x_1x_2 + x_3$, $f_3(x) = x_1 + x_2$, $f_4(x) = x_1 + x_3$

$\{x \mid f_i(x) = 0\}$??

Algorithm to find infeasibility certificate:

$$\mu_1(x^2-1) + \mu_2(2x_1x_2+x_3) + \mu_3(x_1+x_2) + \mu_4(x_1+x_3) = 1$$

- Assuming μ_i's are constant

$\Rightarrow$

$$\left.\begin{aligned} -\mu_1 &= 1 \\ \mu_2 + \mu_4 &= 0 \\ \mu_2 + \mu_4 &= 0 \\ \mu_3 &= 0 \\ \mu_1 &= 0 \end{aligned}\right\} \text{infeasible} \Rightarrow$$

try with μ_i's as linear functions, then quadratics, etc.

Theorem: $\exists$ an exponential bound on the degrees of β_i's in the Hilbert infeasibility certificate.

Algorithm: NuLA (uses linear algebra to solve system of polynomial equations)

for $d = 1, \ldots, D$ (exponentially bounded)

1. try to find β_i of degree d s.t. $\sum \beta_i f_i = 1$
2. If yes, Stop → report infeasible
3. If no, continue with $d = d+1$.

If END → report feasible.

SOS $\quad p(x) = \sum_{i \in I} \{q_i(x)\}^2$

Theorem 1: $p(x) \in \mathbb{R}$ is SOS $\iff p(x) = z^T Q z$ where Q is PSD, z is vector of monomials

Example:

$$\begin{aligned} p(x_1, x_2) &= x_1^2 - x_1 x_2^2 + x_2^4 + 1 \\ &= \tfrac{3}{4}(x_1 - x_2)^2 + \tfrac{1}{4}(x_1 + x_2^2) + 1 \\ &= \tfrac{1}{6}\begin{bmatrix} 1 & x_2 & x_2^2 & x_1 \end{bmatrix} \begin{bmatrix} 6 & 0 & -2 & 0 \\ 0 & 4 & 0 & 0 \\ -2 & 0 & 6 & -3 \\ 0 & 0 & -3 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ x_2 \\ x_2^2 \\ x_1 \end{bmatrix} \end{aligned}$$

Defn: A symmetric matrix is PSD if $x^T A x \geq 0 \quad \forall x \in \mathbb{R}^n$

Theorem: Let A be a symmetric matrix.

A is PSD $\iff$ all eigenvalues of A are real & ≥ 0

$\iff A = C^T C$ where C is a $k \times n$ matrix.

Proof Theorem 1: ($\Leftarrow$) $p(x) = z^T Q z = z^T C^T C z = \sum_i (C_i z)^2 = \text{SOS}$ (rows of C)

($\Rightarrow$) $p(x) = \sum (q_i(x))^2 = x^T Q^T Q x$ where $Q = \begin{bmatrix} q_1 \\ \vdots \end{bmatrix}$ coefficients, $x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \end{bmatrix}$ all monomials

□

Semi-Definite Programming (SDP)

$$\max \sum_{ij} C_{ij} X_{ij} = \text{Trace}(C \cdot X)$$

$$\text{s.t.} \quad \text{Trace}(A_i \cdot X) = b_i \qquad i \in I$$

$$X \succeq 0 \qquad (X \text{ is a } n \times n \text{ matrix})$$

Thm: Grötschel-Lovász-Schriver (GLS)

Optimization / test feasibility over a convex set $S \subseteq \mathbb{R}^d$ is equivalent to separation over S.

(i.e. we need a separation oracle).

Let $S = \{\Lambda \in \mathbb{R}^{n \times n} \mid \Lambda \succeq 0\} \subseteq \mathbb{R}^{n \times n}$

Separation oracle for S

- given a matrix Q, decide if Q belongs to S

  OR Find a separating hyperplane.

→ find all eigenvalues of Q

→ if $\lambda_i \geq 0 \ \forall i$, then $Q \in S$

→ otherwise $\exists\, y \in \mathbb{R}^n$ s.t.

$$Qy = \lambda y, \quad \lambda < 0$$

$$\Rightarrow y^T Q y = y^T \lambda y = \lambda \|y\|_2^2 < 0$$

→ Separating hyperplane $y^T \Lambda y = 0$

i.e. $y^T \Lambda y \geq 0$ is valid for S.

Therefore, GLS $\Rightarrow$ SDP can be solved (By Ellipsoid Method)

(but we can also do this with interior point methods)

Application in Combinatorial Optimization

Stable Set Problem (NP-Complete)

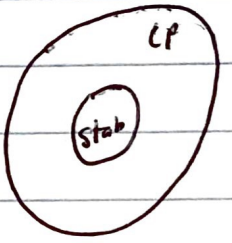
Find a maximum size stable set of vertices where no two vertices are connected by an edge

Prop: S is a stable set $\Leftrightarrow$ V/S is a vertex cover

$$\begin{aligned} \max \quad & \sum x_i \\ \text{s.t.} \quad & x_i + x_j \le 1 \qquad \forall (i,j) \in E \\ & x_i \in \{0,1\} \end{aligned}$$

$$LP(G) = \{x \mid x_i + x_j \le 1 \quad \forall (i,j) \in E,\ 0 \le x_i \le 1\}$$

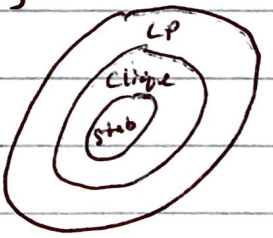
$$Stab(G) = Conv(\{x \mid x_i + x_j \le 1,\ x_i \in \{0,1\}\})$$



Defn: A clique is a set of vertices s.t. $i,j \in E$ forall $i,j \in C$ (the clique)

Observation: For every clique C, at most 1 vertex from C belongs to a stable set.

$$Clique(G) = \{x \mid \sum_{i \in C} x_i \le 1,\ 0 \le x_i \le 1\}$$



Recall that we can write binary variables using quadratic constraints

- one formulation is

$$x_i^2 - x_i = 0$$
$$x_i \cdot x_j = 0 \qquad \forall (i,j) \in E$$

Application in Combinatorial Optimization

Stable Set Problem (NP-Complete)

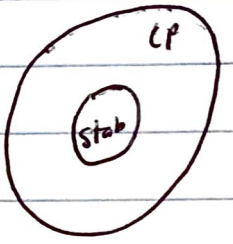
Find a maximum size stable set of vertices where no two vertices are connected by an edge

Prop: S is a stable set $\Leftrightarrow$ V/S is a vertex cover

$$\begin{aligned} \text{Max} \quad & \sum x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \qquad \forall (i,j) \in E \\ & x_i \in \{0,1\} \end{aligned}$$

$$LP(G) = \{x \mid x_i + x_j \leq 1 \quad \forall (i,j) \in E, \; 0 \leq x_i \leq 1\}$$

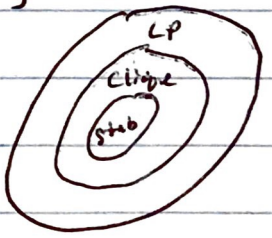
$$Stab(G) = Conv(\{x \mid x_i + x_j \leq 1, \; x_i \in \{0,1\}\})$$



Defn: A clique is a set of vertices s.t. $i,j \in E$ forall $i,j \in C$ (the clique)

Observation: For every clique C, at most 1 vertex from C belongs to a stable set.

$$Clique(G) = \{x \mid \sum_{i \in C} x_i \leq 1, \; 0 \leq x_i \leq 1\}$$



Recall that we can write binary variables using quadratic constraints

- one formulation is

$$x_i^2 - x_i = 0$$
$$x_i \cdot x_j = 0 \qquad \forall (i,j) \in E$$

★ Introduce "linearizing variables" y_{ij}

$y_{ij} = x_i x_j$, $y_{jj} = x_j^2 = x_j$, $y_{0j} = y_{j0} = x_j$, $y_{00} = 1$

$\Rightarrow$ $Y = \begin{pmatrix} 1 \\ x \end{pmatrix} (1 \; x^T)$

★ Y is a positive semidefinite matrix

$\Rightarrow$ Relaxation: $Y \succeq 0$

i.e. need not be that $Y = (\;)(\;)$ for rank 1 matrices.

$\Rightarrow$ cannot recover x from the constraint.

$$TH(G) = \{x \mid Y \succeq 0,\ y_{ij} = 0 \ \forall (i,j) \in E,\ y_{0j} = y_{j0} = y_{jj} = x_j,\ y_{00} = 1,\ 0 \le x_j \le 1\}$$

"Theta body" from Lovász

Theorem

$$Stab(G) \subseteq TH(G) \subseteq Clique(G)$$

proof: Clearly $Stab \subseteq TH(G)$ by construction.

WTS. $TH(G) \subseteq Clique(G)$.

Assume $x \in TH(G)$ $\Rightarrow$ $\exists\ Y \succeq 0$ in $TH(G)$ definition.

$\Rightarrow$ $v^T Y v \ge 0$ $\forall$ $v \in \mathbb{R}^{n+1}$

Take any clique C

set $v_i = \begin{cases} 1 & \text{if } i = 0 \\ -1 & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases}$

Then $0 \le v^T Y v = \sum_{i,j} y_{ij} v_i v_j$

$= \sum_{i,j \in C} y_{ij} \underbrace{v_i v_j}_{(-1)(-1)} + \sum_{i=1}^{n} y_{i0} \underbrace{v_i v_0}_{1} + \sum_{j=1}^{n} y_{0j} \underbrace{v_0 v_j}_{1} + y_{00} \underbrace{v_0^2}_{1}$

$= 0 + \sum_{j=1}^{n} y_{jj} - \sum_{i \in C} x_i - \sum_{i \in C} x_i + 1$

(★ $y_{ij} = 0$ if $(i,j) \in E$)

$\Rightarrow$ $\sum_{i \in C} x_i \le 1$ ☺ i.e. $x \in Clique(G)$ □

★ Introduce "linearizing variables" y_{ij}

$y_{ij} = x_i x_j$, $y_{jj} = x_j^2 = x_j$, $y_{0j} = y_{j0} = x_j$, $y_{00} = 1$

$\Rightarrow$ $Y = \begin{pmatrix} 1 \\ x \end{pmatrix} (1 \; x^T)$

★ Y is a positive semidefinite matrix

$\Rightarrow$ **Relaxation:** $Y \succeq 0$

i.e. need not be that $Y = (\;)(\;)$ for rank 1 matrices.

$\Rightarrow$ cannot recover x from the constraint.

$$TH(G) = \{x \mid Y \succeq 0,\ y_{ij} = 0 \ \forall (i,j) \in E,\ y_{0j} = y_{j0} = y_{jj} = x_j,\ y_{00} = 1,\ 0 \le x_j \le 1\}$$

"Theta body" from Lovász

Theorem

$$Stab(G) \subseteq TH(G) \subseteq Clique(G)$$

proof: Clearly $Stab \subseteq TH(G)$ by construction.

WTS. $TH(G) \subseteq Clique(G)$.

Assume $x \in TH(G)$ $\Rightarrow$ $\exists\ Y \succeq 0$ in $TH(G)$ definition.

$\Rightarrow$ $v^T Y v \ge 0$ $\forall$ $v \in \mathbb{R}^{n+1}$

Take any clique C

set $v_i = \begin{cases} 1 & \text{if } i = 0 \\ -1 & \text{if } i \in C \\ 0 & \text{otherwise} \end{cases}$

Then $0 \le v^T Y v = \sum_{i,j} y_{ij} v_i v_j$

$= \sum_{i,j \in C} y_{ij} \underbrace{v_i v_j}_{(-1)(-1)} + \sum_{i=1}^{n} y_{i0} \underbrace{v_i v_0}_{1} + \sum_{j=1}^{n} y_{0j} \underbrace{v_0 v_j}_{1} + y_{00} \underbrace{v_0^2}_{1}$

$= 0 + \sum_{j=1}^{n} y_{jj} - \sum_{i \in C} x_i - \sum_{i \in C} x_i + 1$

(★ $y_{ij} = 0$ if $(i,j) \in E$)

$\Rightarrow$ $\sum_{i \in C} x_i \le 1$ ☺ i.e. $x \in Clique(G)$ □

Max-Cut

Given a graph $G=(V,E)$ and weights on the edges, find a partition of $V = S \sqcup V/S$ such that the total weight of the edges crossing the partition is maximized.

Goemans-Williamson Approximate Algorithm (SDP)

There is an efficient (polytime) algorithm which gets within 82% of the optimal max cut.

- ∄ any algorithm better than this unless $P=NP$

Formulation: $X_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \notin S \end{cases}$, $i \in V$.

$X_i \in \{\pm 1\} \iff X_i^2 = 1$

$X_i \neq X_j \iff \frac{1 - X_i X_j}{2} = 1$

$$\Rightarrow \max \sum w_{ij} \left(\frac{1 - X_i X_j}{2}\right)$$
$$\text{s.t. } X_i^2 = 1$$

Max-Cut

Given a graph $G = (V, E)$ and weights on the edges, find a partition of $V = S \sqcup V/S$ such that the total weight of the edges crossing the partition is maximized.

Goemans-Williamson Approximate Algorithm (SDP)

There is an efficient (polytime) algorithm which gets within 82% of the optimal max cut.

- ∄ any algorithm better than this unless $P = NP$

Formulation: $x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \notin S \end{cases}$, $i \in V$.

$x_i \in \{\pm 1\} \iff x_i^2 = 1$

$x_i \neq x_j \iff \frac{1 - x_i x_j}{2} = 1$

$$\Rightarrow \max \sum w_{ij} \left(\frac{1 - x_i x_j}{2}\right)$$
$$\text{s.t. } x_i^2 = 1$$

SOS Theory

Defn: f is convex if $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$

$\forall\ x, y \in \mathbb{R}^n,\ \lambda \in [0,1]$

Prop: if f is twice-differentiable, then

$$f \text{ is convex} \iff H_f = \left[\left(\frac{\partial f}{\partial x_i \partial x_j}\right)\right] \succeq 0 \qquad \forall\ x \in \mathbb{R}^n$$

"the Hessian is PSD for all $x \in \mathbb{R}^n$"

Polynomial Matrices $P(x) = [p_{ij}]$ $(m \times m)$

p_{ij}: polynomial in x

Defn: $P(x)$ is a PSD matrix if $P(x)$ is PSD $\forall\ x \in \mathbb{R}^n$

Fact: $P(x)$ is PSD $\iff$ $y^T P(x) y \geq 0$ as an element of $\mathbb{R}[x, y]$

$y \in \mathbb{R}^m$

Defn: $P(x)$ is an SOS matrix if $\exists$ a polynomial matrix $M(x)$

such that $P(x) = M^T(x) M(x)$

Fact: $P(x)$ is SOS $\Rightarrow$ $P(x)$ is PSD

Fact: $P(x)$ is an SOS matrix $\iff$ $y^T P(x) y$ is a sum of squares polynomial.

Example PSD-Matrix not SOS-matrix

$$A(x) = \begin{bmatrix} x_1^2 + 2x_2^2 & -x_1x_2 & -x_1x_3 \\ -x_1x_2 & x_2^2 + 2x_3^2 & -x_2x_3 \\ -x_1x_3 & -x_2x_3 & x_3^2 + 2x_1^2 \end{bmatrix}$$

$A(x)$ is PSD, but $A(x)$ is not SOS

★ $A(x)$ is not a Hessian matrix

since $\frac{\partial}{\partial x_3}\left(\frac{\partial^2 p}{\partial x_1^2}\right) \neq \frac{\partial}{\partial x_1}\left(\frac{\partial p^2}{\partial x_1 \partial x_3}\right)$

Note: if $n=1$, then PSD = SOS

Deciding complexity of a polynomial function:

Defn: $P(x)$ is SOS-convex if $H(x)$ is an SOS-matrix

Clearly SOS-convex $\Rightarrow$ convex

Theorem (Amir Ali & Pablo Parrilo)

$\exists$ a polynomial that is convex, but not SOS-convex.

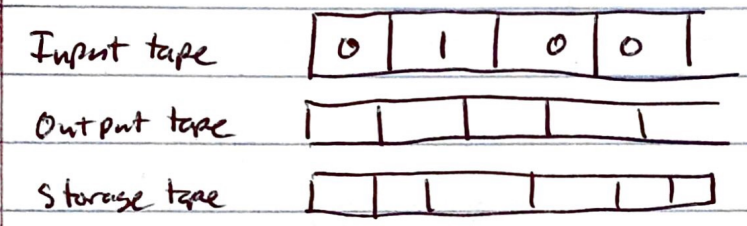
Note: Deciding $H(x) = \text{SOS}$ is easy since we just show that $y^T H(x) y$ is a SOS polynomial

(use SDP to solve)

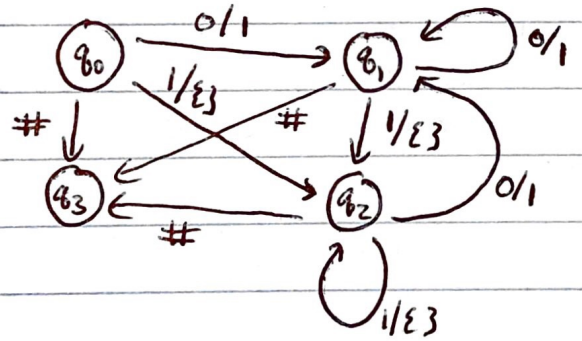
Complexity Theory

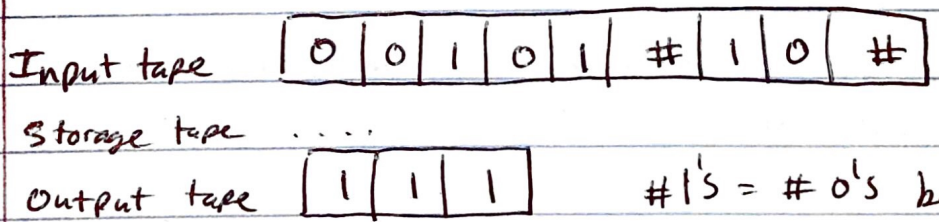
Defn: A Turing Machine includes the following

1. Q: a finite set of states $\quad Q = \{q_0, \ldots, q_m\}$ (q_0: initial, q_m: final)
2. Σ "alphabet" - a set of symbols
3. f "transition function" $\quad f: Q \times \Sigma \to Q \times \Sigma \cup \{e\}$ (e: empty symbol)
4. Tape : Input, Storage, & Output



Example: Suppose $\Sigma = \{0, 1, \#\}$





#1's = # 0's before first #

"Any reasonable model of computation is equivalent to a turing machine"

Time Complexity n = size of input

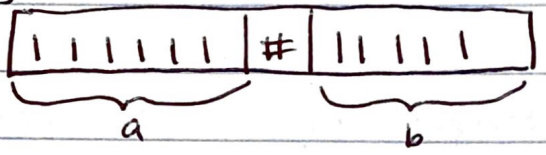
$T(n)$ = # of steps/transitions the turing machine makes

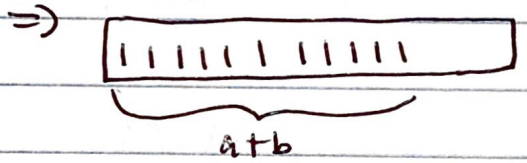
Space complexity

$S(n)$ = amount of storage required

Example: important to consider your input!!!

Adding: $a+b$





"unary encoding"

$T_1(n) = c_1 n$

length $a+b$.

versus:

a base 2 b base 2

a+b base 2

"binary encoding"

$T_2(n) = c_2 n$

only $\log(a) + \log(b)$ length

★ both linear time algorithms, but T_2 is exponentially better than T_1.

Time Complexity n = size of input

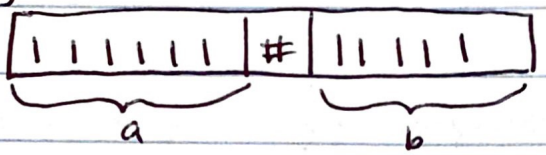
$T(n)$ = # of steps/transitions the turing machine makes

Space complexity

$S(n)$ = amount of storage required

Example: important to consider your input!!!

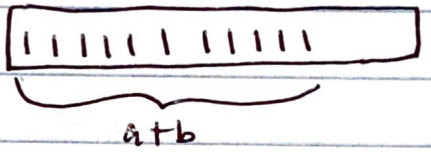
Adding: $a+b$



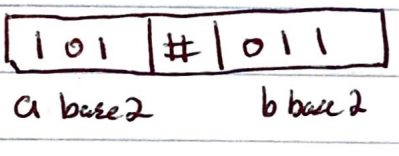
"unary encoding"

$T_1(n) = c_1 n$

length $a+b$.



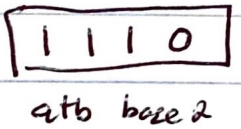
versus:



"binary encoding"

$T_2(n) = c_2 n$

only $\log(a) + \log(b)$ length



★ both linear time algorithms, but T_2 is exponentially better than T_1.

Defn: A problem is a set of strings on Σ

An instance is a particular string from the problem

Defn: The class of polynomial time problems P is the set of problems for which there exists a turing machine with polynomial time complexity.

Reduction: Problem B reduces to problem A if for all instances of B there exists a map

$R: B \to A$, $R(b) \in A$, s.t. R is polytime.

Defn: A and B are polynomially equivalent if A reduces to B and B reduces to A.

Example: Vertex Cover

- Optimization asks for the minimum vertex cover
- Feasibility asks for a vertex cover of size $\leq k$

Stable Set

- optimization asks for the maximum stable set
- feasibility asks for stable set of size $\geq n-k$

All of these are polynomially equivalent

Defn: A non-deterministic polynomial time (NP) turing machine has a map

$$f: Q \times \Sigma \to 2^Q \times \Sigma \cup \{e\}$$

it can carry out parallel computations.

Alternative Defn: A problem is NP if there exists a polynomial time algorithm to check a given certificate.

Clearly $P \subseteq NP$. Probably $P \neq NP$.

Defn: A problem is a set of strings on Σ

An instance is a particular string from the problem

Defn: The class of polynomial time problems P is the set of problems for which there exists a turing machine with polynomial time complexity.

Reduction: Problem B reduces to problem A if for all instances of B there exists a map

$R: B \to A$, $R(b) \in A$, s.t. R is polytime.

Defn: A and B are polynomially equivalent if A reduces to B and B reduces to A.

Example: Vertex Cover

- Optimization asks for the minimum vertex cover
- Feasibility asks for a vertex cover of size $\leq k$

Stable Set

- optimization asks for the maximum stable set
- feasibility asks for stable set of size $\geq n-k$

All of these are polynomially equivalent

Defn: A non-deterministic polynomial time (NP) turing machine has a map

$$f: Q \times \Sigma \to 2^Q \times \Sigma \cup \{e\}$$

it can carry out parallel computations.

Alternative Defn: A problem is NP if there exists a polynomial time algorithm to check a given certificate.

Clearly $P \subseteq NP$. Probably $P \neq NP$.