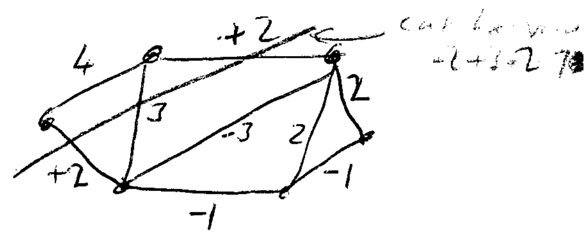
SEMI DEFINITE PROGRAMMING.

Eg: MAXCUT PROBLEM:

Given a graph $G=(V,E)$ and edge weight c_e, both positive and negative, divide G into two parts V_1 and V_2 so $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$

so as to maximize $\sum_{\substack{e \text{ with one end in} \\ V_1 \text{ and one end in } V_2}} c_e$.



Can model as an integer programming problem:

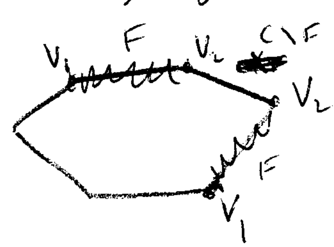
Let x be the incidence vector of a cut,

so $x_e = \begin{cases} 1 & \text{if } e \text{ is in the cut} \\ 0 & \text{otherwise} \end{cases}$

Solve: $\max \sum_{e \in E} c_e x_e$

s.t. x is the incidence vector of a cut

Cutting planes: Any cut and any cycle intersect in an even number of edges



Get constraints $x(F) - x(C \setminus F) \leq |F| - 1$

$\sum_{e \in F} x_e =$ ↗ for every cycle C and every subset $F \subseteq C$ of odd cardinality.

So solve: $\max \sum_{e\in E} c_e x_e$

s.t. $x(F) - x(C\setminus F) \le |F|-1$ $\forall$ cycles C, $\forall$ odd sets $F \subseteq C$.

~~$x(e)$~~ x_e binary.

~~This~~ Any feasible solution to this is the incidence vector of a cut.

LP relaxation: $\max \sum_{e\in E} c_e x_e$

s.t. $x(F) - x(C\setminus F) \le |F|-1$ $\forall F \subseteq C$, $|F|$ odd, $\forall$ cycles C

$0 \le x_e \le 1$.

Drawbacks: (1) Large number of constraints.
Sufficient to look at chordless cycles, but still a large number.

(2) Relaxation may not be very tight — need more constraints.

For K_n, n even & large: bound $\frac{z_{IP}}{z_{LP}} \approx \frac{3}{4}$.

$\left(\frac{n}{2}\right)\left(\frac{n}{2}\right)$ $\frac{n^2}{4}$

$\frac{2}{3}\binom{n}{2} = \frac{n(n-1)}{3}$

Alternative formulation:

Divide V into two sets V_1 and V_2.

Use a vector $x \in \mathbb{R}^{|V|}$ (not $x \in \mathbb{R}^{|E|}$)

Let $x_i = \begin{cases} 1 & \text{if } i \in V_1 \\ -1 & \text{if } i \in V_2. \end{cases}$

Then $\sum_{e \text{ in cut}} c_e = -\frac{1}{2}\sum_{\text{all edges}} c_{ij} x_i x_j + \frac{1}{2}\sum_{\text{all edges}} c_{ij} = \frac{1}{4}\sum_i \sum_j c_{ij}(1 - x_i x_j)$ since each pair i,j counted twice

since if edge is in cut: RHS is $\frac{1}{2}\left[-1\underbrace{(x_i)(x_j)}_{-1} + 1\right] = 1$

if edge is not in cut: RHS is $\frac{1}{2}\left[-1\underbrace{(x_i)(x_j)}_{+1} + 1\right] = 0$.

So, instead of maximizing $\sum_{e \in \text{cut}} c_e$,

we can instead

$$\min \sum c_{ij} x_i x_j$$

$$\text{s.t.} \quad x_i = \pm 1$$

This is a nonconvex quadratic ~~programming~~ integer programming problem.

Look at relaxations:

Let $X = xx^T$, so $X_{ij} = x_i x_j$

$C = \{c_{ij}\}$.

Then $\text{trace}(CX) = \sum_i (CX)_{ii} = \sum_i \left(\sum_j C_{ij} X_{ji}\right)$

$= \sum_i \sum_j C_{ij} X_{ij}$ since X symmetric

$= \sum_i \sum_j c_{ij} x_i x_j$

So: equivalently to max cut problem: $\min \text{trace}(CX)$
s.t. $X = xx^T$
$x_i = \pm 1$.

Note: If $x_i = \pm 1$ then $X_{ii} = 1$.

So: max cut equivalent to: $\min(\text{trace}(CX))$
s.t. $X = xx^T$,
$X_{ii} = 1 \; \forall i$.

Equivalent to: $\min(\text{trace}(CX))$

$\text{rank}(X) = 1$,

$X_{ii} = 1 \; \forall i$

X symmetric

Relax to: min trace(CX)

s.t. X is positive semi-definite

$X_{ii} = 1 \quad \forall i$.

This is a SEMI DEFINITE PROGRAMMING PROBLEM.

It can be solved by an interior point method.

Can show: bound given by this is within 0.87856.

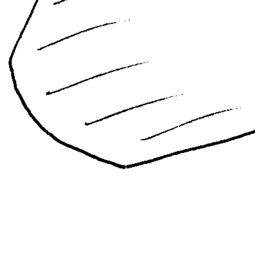
(ie, value of max cut $\geq .87856 \times$ bound given by this relaxation) if all $c_e \geq 0$.

By contrast, LP relaxation can only prove a bound of $.5$.
(Chvátal Polytope [illegible] for bad [illegible], [illegible])
— graphs where only half the edges are in a cut, but all cycles are long, (eg length at least k), so $x_e = \frac{k-1}{k}$ satisfies all the cycle ineqs.

Can get SDP relaxations for other problems.

Eg: There is an SDP relaxation of the node packing problem that is stronger than the LP relaxations we've looked at, even with all the families of facets. (Ie, Lovasz Theta).

~~Positive~~
Think of elements of X as one long vector — eg, write the columns one after another.
Instead of having a constraint that the elements should be nonnegative, require that $X \succeq 0$.
Looks like:

Eg: 

[illegible] { Piecewise continuous boundary, with quadratic and linear parts.
Feasible region is convex.

Goemans & Williamson Algorithm for ~~SDP~~ Max Cut.

1. Solve the SDP relaxation, obtaining X^*.

$$\min \ \text{trace}(CX)$$
$$\text{s.t. } X \text{ is positive semidefinite, symmetric.}$$
$$X_{ii} = 1 \quad \forall i$$

2. Form a ~~Cholesky~~ factorization ~~[illegible]~~. $X^* = V^T V$.

~~(e.g., if X^* is pos definite, can use Cholesky factorization.)~~
~~Let $v_1, \dots, v_n$ be rows of V.~~
~~Since $X_{ii} = 1$, we have $v_i^T v_i = 1 \ \forall i$.~~

V is a $d \times n$ matrix. (If soln to SDP is rank 1, then $d=1$

Let $v_1, \dots, v_n$ be the columns of V.
Note that $v_i^T v_j = X^*_{ij} = 1$ if $i=j$, so all the vectors are unit vectors

3. Generate a random vector ~~[illegible]~~ $r \in \mathbb{R}^d$, on the unit sphere.

4. If $v_i^T r \begin{cases} > 0 & \text{, place vertex } i \text{ on side 1 of the cut} \\ < 0 & \text{, place vertex } i \text{ on ~~other~~ side 2 of the cut.} \end{cases}$

Eg: If soln to SDP is $X^* = xx^T$ for some ± 1 vector x,

then we have n ~~vector~~ <u>scalars</u> v_i, each equal to ± 1.

~~If~~ The assignment in ~~B~~ Step 4 will ~~put all the~~ divide the vertices up as specified by x.

Thm. Assume $c_{ij} \geq 0 \ \forall i,j$. Let ZOPT be the optimal value for the maxcut problem, ~~and let RELAX be the value of the SDP relaxation, and let~~ let RELAX be the upper bound on the value of the ~~E~~ MAXCUT problem given by the optimal value of the SDP relaxation, and let ZSDP be the value of the solution returned by this algorithm. Then

$$\text{ZSDP} \geq 0.87856\,(\text{RELAX}) \quad \cancel{0.87856\,(\text{ZOPT})}$$
$$\geq 0.87856\,(\text{ZOPT}).$$

Proof uses properties of Arccos function.

~~Note: The value of the SDP relaxation leads to an upper bound on ZOPT, [illegible]~~

In Step 3, the algorithm picks a random vector r.

With a poor choice of r, the algorithm may perform worse.

However: (1) The expected value is at least this good.

~~(2) The algorithm can be~~

(2) In practice, many vectors r can be chosen, and the best resulting cut selected.

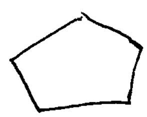
(3) The algorithm can be DERANDOMIZED, that is, there is a deterministic way to pick r that is guaranteed to be at least as good as the average.

~~How~~

Theorem (Håstad) It is NP-hard to approximate MAXCUT within $\frac{16}{17} + \varepsilon \approx 0.94117$ for any $\varepsilon > 0$.

Let RELAX be the upper bound

Conjecture The worst case ratio $\frac{Z_{SDP}}{RELAX}$ ~~performance~~ of Goemans-Williamson algorithm is 0.88446, given by the 5-cycle. (See, eg, Feige & Schechtman, 2002)



Value of MAXCUT = 2.

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

~~THM~~ Note: A similar approach can be used to give a 0.87856 algo for MAX2SAT.
Can be extended to a 0.931-approx algo for MAX2SAT.
Also to a 0.758-approx algo for MAXSAT.

Strengthening the relaxation.

If $X = xx^T$ and $x_i = \pm 1$, then $X_{ij} = x_i x_j$ and

$$X_{ij} + X_{jk} + X_{ik} = x_i x_j + x_i x_k + x_j x_k \geq -1$$

Since: $x_i = x_j = x_k = 1$ or $x_i = x_j = x_k = -1$ $\Rightarrow$ $x_i x_j + x_i x_k + x_j x_k = 3$

$\cancel{x_i = x_j = x}$

If exactly one of $x_i, x_j, x_k = -1$, others are $+1$, then

$$x_i x_j + x_i x_k + x_j x_k = -1$$

If exactly two of $x_i, x_j, x_k = -1$, others are -1, then

$$x_i x_j + x_i x_k + x_j x_k = -1.$$

Similarly: $X_{ij} - X_{jk} - X_{ik} \geq -1$

Adding these constraints as cutting planes helps computationally (although the theoretical bound is not strengthened.)

Branching

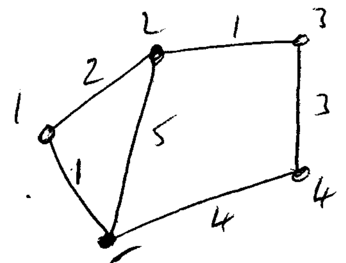
Branch using original variables:

~~Eg~~ Split into: $x_i = x_j$ vs $x_i \neq x_j$

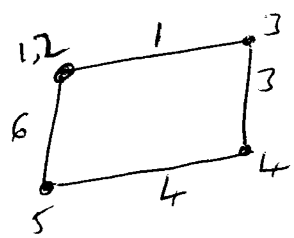
$x_i = x_j$ → i and j on same side of cut → ~~replace ea~~ remove vertex j, replace edges from i to other vertices with edges with cost $c_{ik} + c_{jk}$

$x_i \neq x_j$ → i and j on opposite sides of the cut. → remove vertex j, replace edges from i to other vertices with edges with cost $c_{ik} - c_{jk}$, add $\sum_k c_{ij}$ to value of cut.

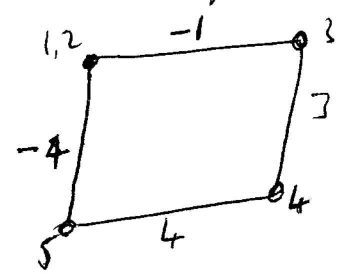
Eg:



Branch so 1 & 2 on same side of cut.
Get new graph:



Branch so 1 & 2 on opposite sides of cut.
Get new graph:



Add 1+2+5=8 to size of cut.

Works since: $\bar{i}$ = supervertex

$$\sum_k \hat{c}_{\bar{i}k} x_{\bar{i}k} = \sum_k (c_{ik} - c_{jk})(x_{ik})$$ since i is on same side as $\bar{i}$

modified costs

$$= \sum_{k \neq i,j} (c_{ik} x_{ik}) - \sum_{k \neq i,j} c_{jk}(1 - x_{jk})$$ since $x_{jk} = 1 - x_{ik}$

$$= \sum_{k \neq i} c_{ik} x_{ik} + \sum_{k \neq i} c_{jk} x_{jk} - \sum_{k \neq i} c_{jk}.$$

Handling other problems.

Eg: Equipartition problem:

Need equal numbers of vertices on each side of the cut.

So $\sum_{i=1}^{n} x_i = 0$, since half of x_i's are $+1$ and half of x_i's are -1.

Write this as $e^T x = 0$

Then $e^T x x^T e = 0$

ie, $e^T X e = 0$, a linear constraint on the elements of X,

namely, $\sum_{i,j} X_{ij} = 0$.

Any linear constraint:

$a^T x = b$ (b is a scalar, a is a vector)

$\Rightarrow a^T x x^T a = b^2 \rightarrow a^T X a = b^2$. Again, a linear constraint on the elements of X.

Lovász θ - Number.

Semidefinite programming relaxation for node packing: (unweighted)

$G=(V,E)$, $|V|=n$. IP formulation: $\max \sum x_i$ s.t. $x_i + x_j \leq 1 \ \forall (i,j) \in E$, x_i binary

X is $n \times n$.

Assume we have a packing of size p. Define $\bar{x}_i = \begin{cases} 1/\sqrt{p} & \text{if } i \text{ in packing} \\ 0 & \text{otherwise.} \end{cases}$

Let $\bar{X} = \bar{x}\bar{x}^T$

So $\bar{X}_{ij} = \begin{cases} 1/p & \text{if } i,j \text{ both in packing} \\ 0 & \text{otherwise} \end{cases}$ $\quad \bar{X} = \left[\begin{array}{c|c} 1/p & 0 \\ \hline 0 & 0 \end{array}\right] \begin{array}{l} \} \ p \text{ nodes in packing} \\ \} \text{ remaining nodes} \end{array}$

block of $1/p$'s

$\#$ nonzeroes in $\bar{X} = p^2$ (reorder rows/columns if nec.)

Sum of entries in $\bar{X}$: $p^2\left(\frac{1}{p}\right) = p$.

$$\operatorname{trace}(\bar{X}) = \sum_{i=1}^{n} \bar{X}_{ii} = p\left(\tfrac{1}{p}\right) + (n-p)(0) = 1.$$

Suggests:

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij}$$

$$\text{s.t.} \quad X_{ij} = 0 \quad \text{if } (i,j) \in E \qquad (P\theta)$$

$$\sum_{i=1}^{n} X_{ii} = 1$$

$$X \succeq 0.$$

$\bar{X}$ is feasible in this problem, with value p.

Why does this give a good bound?

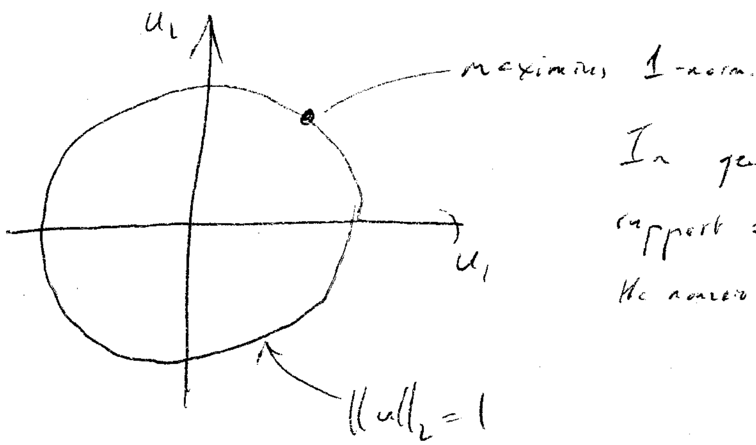
If X is rank one, so $X = uu^T$:

Now, $\underbrace{X_{ij}}_{u_iu_j} = 0$ if $(i,j) \in E$, so the support of u defines a node packing

Also, $\text{trace}(X) = \sum_{i=1}^{n} X_{ii} = \sum u_i^2$, so $\|u\|_2 \le 1$.

Also,
$$\sum_{i=1}^{n}\sum_{j=1}^{n} X_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j$$
$$= \left(\sum_{i=1}^{n} u_i\right)^2$$

So, we want to maximize the 1-norm of u,
subject to the 2-norm is ≤ 1
and the support of u gives a node packing.



In general, if the cardinality of the support of u is p, then we want to have the nonzero elements of u all equal to $1/p$.

Dual problem:

A semidefinite program has a dual program that is also an SDP.

Eg: $\min \sum_i \sum_j c_{ij} X_{ij}$

s.t. $\sum_i \sum_j a_{ij}^k X_{ij} = b_k \quad k=1,\dots,m \qquad (P)$

$X \succeq 0$

has dual: $\max \sum_k b_k y_k$

s.t. $\sum_k y_k a_{ij}^k + S_{ij} = c_{ij} \quad \forall i,j \qquad (D)$

$S \succeq 0$

Can write these in matrix terms:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} X_{ij} = \sum_{i=1}^n \sum_{j=1}^n c_{ij} X_{ji} \quad \text{since } X \text{ symmetric}$$

$$= \sum_{i=1}^n (CX)_{ii}$$

$$= \operatorname{trace}(CX)$$

$$=: C \bullet X$$

Similarly, $\sum_i \sum_j a_{ij}^k X_{ij} = \operatorname{trace}(A_k X) = A_k \bullet X$

So (P) & (D) are:

$\min \; C \bullet X$ s.t. $A_k \bullet X = b_k \quad k=1,\dots,m$, $X \succeq 0 \qquad (P)$

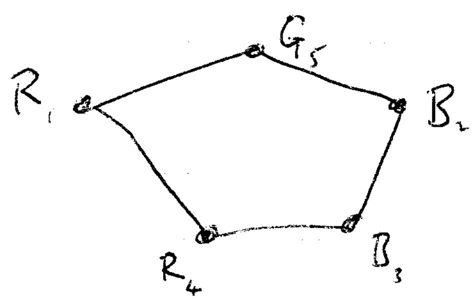
$\max \; b^T y$ s.t. $\sum_k y_k A_k + S = C$, $S \succeq 0 \qquad (D)$

Look at dual of $(P\theta)$

$$
\begin{array}{llll}
\min & \tau & & \\
\text{s.t.} & \tau - s_{ii} = 1 & & \\
y_{ij} & -s_{ij} = 1 & \text{if } (i,j) \in E & (D\theta) \\
 & -s_{ij} = 1 & \text{if } (i,j) \notin E & \\
 & S \succeq 0 & &
\end{array}
$$

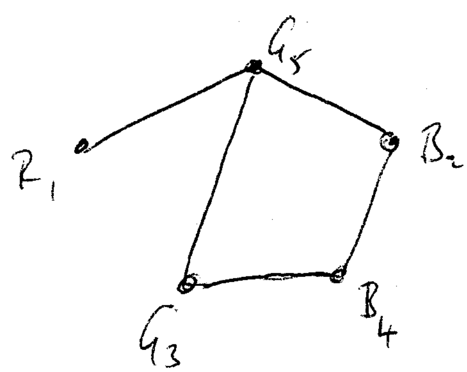
Look at vertex colourings (really, in the complement):

Given a node packing, assign each node a different colour.
For remaining nodes: pick a colour so that if $(i,j) \notin E$ then i and j have different colours.
May need to introduce more colours.
So $(i,j) \notin E \Rightarrow i,j$ have different colours.



Need three colours here.
Best node packing has size 2.

(odd hole)



Need three colours.
Best node packing also has three colours.

$\equiv$ $\boxed{\text{minimum number of cliques required to cover all the nodes of } G.}$

So min number of colours $\geq$ max node packing.

OK since i,j have ~~different~~ same colors $\Rightarrow$ $(i,j) \notin$ complement of E $\Rightarrow (i,j) \in E$.

Given a colouring:

Let τ = number of colours

Let $S_{ii} = \tau - 1$ for $i = 1, \ldots, n$

Let $S_{ij} = -1$ if ~~$(i,j) \notin E$~~ i,j have different colors

Let $S_{ij} = \tau - 1$ if ~~$(i,j) \in E$~~, i,j have same colors

and $y_{ij} = \tau$ if ~~$(i,j) \in E$~~ i,j have same colors

and $y_{ij} = 0$ if i,j have different colors.

Let Γ denote the matrix of all ones. (Size may vary)

Write S as:

$$S = \begin{array}{cl} \begin{bmatrix} (\tau-1)\Gamma & -\Gamma & \cdots & -\Gamma \\ -\Gamma & (\tau-1)\Gamma & \cdots & -\Gamma \\ \vdots & & \ddots & \vdots \\ -\Gamma & -\Gamma & \cdots & (\tau-1)\Gamma \end{bmatrix} & \begin{matrix} \text{colour } 1 \\ \text{colour } 2 \\ \vdots \\ \text{colour } \tau \end{matrix} \end{array}$$

$= \text{sum of } \binom{\tau}{2}$ matrices of the form

$$\begin{bmatrix} 0 & & 0 & & \\ & \Gamma & & -\Gamma & \\ & & 0 & & \\ & -\Gamma & & \Gamma & \\ 0 & & 0 & & 0 \end{bmatrix}$$

(Each off-diagonal block shows up once in the sum. Each diagonal block shows up $(\tau - 1)$ times, in combination with each other block on the row)

$= \text{sum of } \binom{\tau}{2}$ psd matrices

$\succeq 0.$

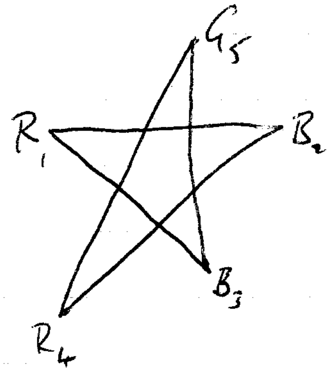
So τ gives an upper bound on the max size of a node packing.

Complementary graphs

$G=(V,E)$ has a complement $\bar{G}=(V,\bar{E})$,
where $(i,j)\in E \Leftrightarrow (i,j)\notin \bar{E}$.

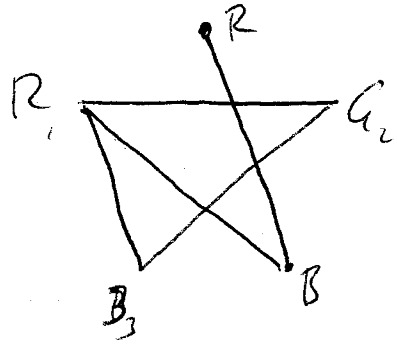
A node packing in G corresponds to a clique in $\bar{G}$.

The colouring we defined in the discussion of $(D\theta)$ corresponds to the standard node colouring in $\bar{G}$.



No clique of size > 2
Min colouring has size 3

(odd antihole



Clique of size 3
Colouring with three colours.

Defn A graph $G=(V,E)$ is PERFECT if the cardinality of the maximum clique in G is equal to the minimum number of colours in a colouring of the graph where no two adjacent nodes have the same colour.

Thm A graph is perfect if and only if its complement is perfect (Reg 59, page 578)

STRONG PERFECT GRAPH CONJECTURE.

Thm A graph is perfect iff it contains no odd holes and no odd antiholes (CHUDNOVSKY, SEYMOUR, ROBERTSON, THOMAS, 2002)

Another SDP relaxation of node packing:

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

Multiply each constraint by each x_k, $1-x_k$:

$$\begin{aligned}
\max \quad & \sum x_i^2 \\
\text{s.t.} \quad & x_i(x_i + x_j) \le x_i && \forall (i,j) \in E \\
& (1-x_i)(x_i + x_j) \le 1 - x_i && \forall (i,j) \in E \\
& x_k(x_i + x_j) \le x_k && \forall (i,j) \in E, \ \forall k \in V \setminus \{i,j\} \\
& (1-x_k)(x_i + x_j) \le 1 - x_k && \forall (i,j) \in E, \ \forall k \in V \setminus \{i,j\} \\
& x_i \text{ binary} && \forall i \in V.
\end{aligned}$$

Let $\cancel{Z_{ij} = x_i x_j}$ $Z = xx^T$.

Note that $x_i^2 = x_i$ and $Z_{ii} = x_i^2$, $Z_{ij} = x_i x_j$.

Write problem in terms of Z:

$$\begin{aligned}
\max \quad & \sum z_{ii} \\
\text{s.t.} \quad & z_{ij} \le 0 && \forall (i,j) \in E \\
& z_{ii} + z_{jj} \le 1 + z_{ij} && \forall (i,j) \in E \\
& z_{ik} + z_{jk} \le z_{kk} && \forall (i,j) \in E, \ \forall k \in V \setminus \{i,j\} \\
& z_{ii} + z_{jj} + z_{kk} \le 1 + z_{ik} + z_{jk} && \forall (i,j) \in E, \ \forall k \in V \setminus \{i,j\} \\
& Z \succeq 0.
\end{aligned}$$

Could also add: $z_{ij} \ge 0 \quad \forall i \in V, \ \forall j \in V$ | $z_{ii} + z_{jj} \le 1 + z_{ij} \quad \forall i \in V, \ \forall j \in V$ (from $(1-x_i)(1-x_j) \ge 0$.)

$z_{ij} = 0 \quad \forall (i,j) \in V$

Solving SDPs

Use barrier function:

$$\begin{aligned} \min \quad & C \bullet X - \mu \ln \det X \qquad \mu > 0. \\ \text{s.t.} \quad & A_k \bullet X = b_k \quad k = 1, \dots, m \\ & X \succ 0 \end{aligned}$$

Let $\mu \to 0_+$.

Gradient of objective is $C - \mu X^{-1}$.

The barrier function is self-concordant, so Newton's method can lead to soln in $O(\sqrt{n})$ iterations.

Duality

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i, \quad i = 1, \ldots, m \\ & X \succeq 0 \end{aligned}$$

Lagrangian relaxation:

$$\theta(y) := \min_{X \succeq 0} \; C \bullet X + \sum_i y_i (b_i - A_i \bullet X)$$

or: $$\min_{X \succeq 0} \; b^T y + \left(C - \sum_i y_i A_i\right) \bullet X$$

Gives a lower bound for any y.
Try to get best lower bound.

$$\max \theta(y) \equiv \max \; b^T y + \min_{X \succeq 0} \left(C - \sum_i y_i A_i\right) \bullet X$$

If $C - \sum_i y_i A_i$ is psd then $\left(C - \sum_i y_i A_i\right) \bullet X \geq 0 \quad \forall X \succeq 0$,
so min value is zero.

If $C - \sum_i y_i A_i$ is not psd then $\exists X \succeq 0$ so that $\left(C - \sum_i y_i A_i\right) \bullet X < 0$.
So can drive min to $-\infty$.

Thus, Lagrangian dual is:
$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & \sum_i y_i A_i + S = C \\ & S \succeq 0 \end{aligned}$$

Quadratically Constrained Quadratic Programs

$$\begin{aligned} \min\ & \tfrac{1}{2}x^T Q_0 x + c^T x \\ \text{s.t. } & \tfrac{1}{2}x^T Q_i x + a_i^T x \le b_i \quad i=1,\dots,m \end{aligned} \Bigg\} \; Q_i \text{ can be nonconvex.}$$

Eg: Max Cut:

$x_i = \pm 1$ can be represented as the constraint $x_i^2 = 1$.

Eg: Binary variables x_i can be represented by $x_i = x_i^2$.

Get SDP relaxation:

$$\min \ \tfrac{1}{2}\begin{bmatrix} 0 & c^T \\ c & Q_0 \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix}$$

$$\text{s.t.} \quad \tfrac{1}{2}\begin{bmatrix} -2b_i & a_i^T \\ a_i & Q_i \end{bmatrix} \bullet \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \le 0 \qquad i=1,\dots,m$$

$$X = xx^T.$$

Relax "$X = xx^T$" to $X \succeq xx^T$, which is

equivalent to $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$ (Schur complement)

(Proof: Assume $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \Rightarrow \begin{bmatrix} v_0 \\ v \end{bmatrix}^T \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \begin{bmatrix} v_0 \\ v \end{bmatrix} \ge 0 \Rightarrow v_0^2 + 2v_0 v^T x + v^T X v \ge 0$.
For given v, LHS is minimized by $v_0 = -v^T x$. So $v^T X v \ge (v^T x)^2$.
Other direction is similar.)

Completely Positive Programs.

(Burer, MP 120, pp 479-495, 2009,

DEFN Closed, full-dimensional convex cone of $n \times n$ COPOSITIVE MATRICES is
$$C_n := \{X \in \mathbb{R}^{n\times n} : X = X^T, v^T X v \geq 0 \quad \forall v \in \mathbb{R}^n_+\}$$

DEFN The closed full-dimensional convex cone of $n \times n$ COMPLETELY POSITIVE MATRICES is
$$C_n^* := \left\{X \in \mathbb{R}^{n\times n} : X = \sum_{k\in K} z^k (z^k)^T \text{ for some finite } \{z^k\}_{k\in K} \subseteq \mathbb{R}^n_+\right\}$$

LEMMA C_n and C_n^* are dual cones, under the Frobenius inner product.

(So, $C_n^* = \{X \in \mathbb{R}^{n\times n} : X \bullet Y \geq 0 \quad \forall Y \in C_n\}$,

and vice versa.) (No proof)

$C_n^* \subseteq S_n \subseteq C_n$, S_n = cone of psd $n \times n$ matrices.

A <u>completely positive program</u> is of the form

$$\begin{aligned} \min \quad & C \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i, \quad i = 1, \dots, m \\ & X \in C_n. \end{aligned}$$

Assume we have a problem of the form:

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} x^T Q x \\ \text{s.t.} \quad & a_i^T x = b_i, \quad i = 1, \ldots, m \\ & x \geq 0 \\ & x_j \in \{0, 1\}, \quad \forall j \in B. \end{aligned} \qquad (P)$$

Any nonconvex quadratic program with a mixture of binary and continuous variables can be put in this form.

Delay this to the theorem

(Assume the constraints $a_i^T x = b_i$ include the constraints $x_j + s_j = 1 \quad \forall j \in B$, with s_j restricted to be nonnegative.)

We can relax (P) by introducing a $n \times n$ matrix X. Initially set $X = xx^T$, and then relax the assumption that X is rank one

(P) is equivalent to:

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} Q \bullet X \\ \text{s.t.} \quad & a_i^T x = b_i \quad i = 1, \ldots, m \\ & a_i^T X a_i = b_i^2 \quad i = 1, \ldots, m \\ & X = xx^T \\ & x \geq 0 \\ & x_j \in \{0, 1\} \quad \forall j \in B \end{aligned}$$

With the relaxation, get:

$$\begin{aligned}
\min \quad & c^T x + \tfrac{1}{2} Q \bullet X \\
\text{s.t.} \quad & a_i^T x = b_i \quad \forall i \\
& a_i^T X a_i = b_i^2 \quad \forall i \\
& X_{jj} = x_j \quad \forall j \in B \\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in C^*_{n+1}
\end{aligned} \qquad (C)$$

Thm Under the assumptions, (C) is equivalent to (P).

In particular, (i) opt(C) = opt(P)

(ii) if (x^*, X^*) is optimal for (C) then x^* is in the convex hull of optimal solutions for (P).

(Proof uses convex analysis, looking at recession cones for (P) and (C).)

Corollary Problem (C) is NP-hard.

Proof Nonconvex quadratic programming is NP-Complete. ∎

Example

① Standard quadratic programming, i.e.,

$$\min \ \tfrac{1}{2} x^T Q x \quad \text{s.t.} \quad e^T x = 1, \ x \geq 0$$

Equivalent to
$$\min \ \tfrac{1}{2} x^T Q x \quad \text{s.t.} \quad e^T x x^T e = 1, \ x \geq 0$$

Get equivalent CPP:
$$\min \ Q \bullet X \quad \text{s.t.} \quad e^T X e = 1, \ X \in C_n^*.$$

② Node packing:

Can show maximum stable set number is
$$\max \{ x^T e e^T x : x \geq 0, \|x\|^2 = 1, x_i x_j = 0 \ \forall (i,j) \in E \}.$$

Get CPP:
$$\begin{aligned} \max \quad & ee^T \bullet X \\ \text{s.t.} \quad & I \bullet X = 1 \\ & X_{ij} = 0 \quad \forall (i,j) \in E \\ & X \in C_n^*. \end{aligned}$$