Optimization

based on the lectures of
Stephan THOMASSÉ

notes written by
Hugo SALOU

ENS Lyon – M1

Part 1. Linear optimization

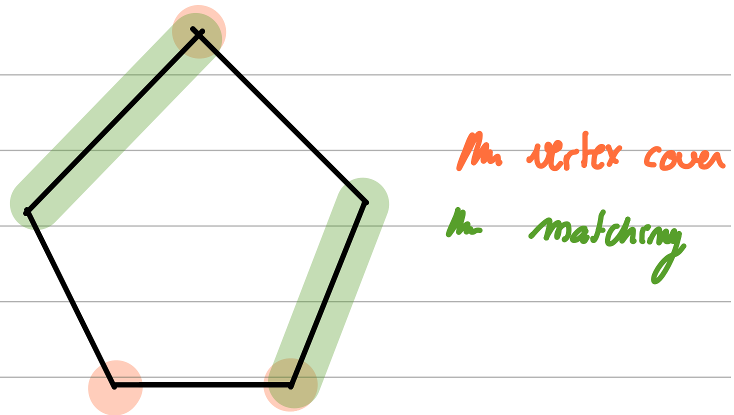
I Modelisation of problems

Example

A vertex cover in a graph $G=(V,E)$ is a set of vertices $X \subseteq V$ such that, for every edge $xy \in E$, $x \in X$ or $y \in X$. We will write $\tau(G)$ for the min size of a vertex cover.

For the pentagonal graph:

$$\tau(G) = 3$$
$$\nu(G) = 2$$



A matching is a set of disjoint edges. We will write $\nu(G)$ for the max size of a matching.

Computing a min size vertex cover is NP-hard; computing a max-size matching is very tricky but poly-time (Edmond's thm).

It is obvious that:

$$\nu \leq \tau$$

We will define fractional relaxations of these problems.

Let x_v be a variable for every vertex $v \in V$. We ask that

$$\forall uv \in E, \quad x_u + x_v \geq 1 \quad \text{and} \quad \forall u \in V, \; x_u \geq 0$$

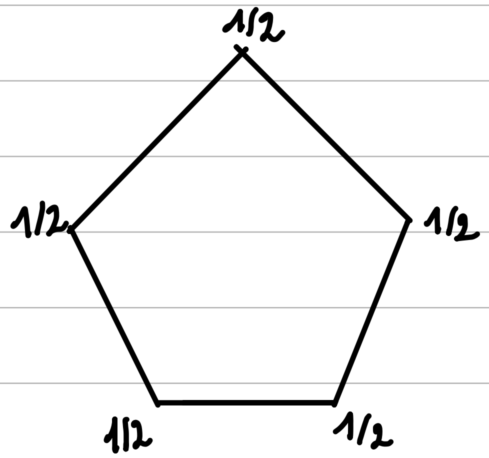
such that $\sum_{u \in V} x_u$ is minimal. We will write τ^* for the min.

For the max matching, we put a weight y_e for every edge e, such that

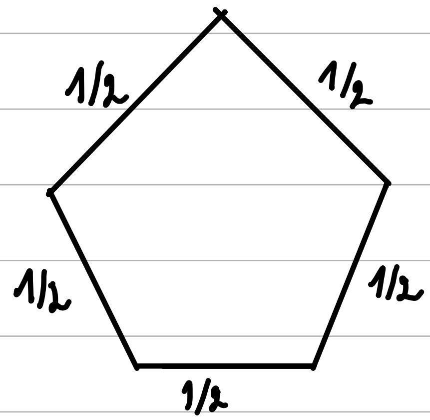
$$\forall e \in E, \quad y_e \geq 0 \qquad \text{and } \forall u \in V, \sum_{e \ni u} y_e \leq 1.$$

We will write ν^* for the max of $\sum_{e \in E} y_e$.

For the pentagon graph, we have :



$\tau^*(G) = \frac{5}{2}$



$\nu^*(G) = \frac{5}{2}$

In general, we have that

$$\nu \leq \underbrace{\nu^* = \tau^*}_{\text{"primal/dual" parameters}} \leq \tau.$$

Remark LP (linear programming) is in P. Linear solver programs can be done in poly-time, thus computing relaxed solutions is possible and useful.

Duality: LP come by pairs and parameters of primal & duals are equal.

Why is LP tractable?

1) The <u>simplex algorithm</u> is efficient but not in P.
2) There is a poly-time algo (using ellipsoids) but not useful in practice.
3) There is an algo that is both efficient and in P using interaction-point methods.

II The Simplex Algorithm.

Consider the following LP:

$$(P): \quad \max\ 5x_1 + 4x_2 + 3x_3$$
$$\text{s.t.} \left| \begin{array}{l} x_1, x_2, x_3 \geq 0 \\ 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + 3x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \end{array} \right.$$

We can try to increase $x_1, \dots, x_3$ but what's the next step?
Introduce <u>slack variables</u> $x_4, \dots, x_6$ one for each constraint.

We then transform (P) in

$$(D_0)\quad \begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - x_1 - 4x_2 - 2x_3 \\ z = 5x_1 + 4x_2 + 3x_3. \end{cases}$$

(P) is equivalent to max z st. (D_0) & $x_1, \dots, x_6 \geq 0$.

(initial) dictionnary

To a dictionnary we associate a solution by setting non-basic variables to 0 and getting solutions for the basic variables.

For (D_0), its solution is $\underbrace{(5, 11, 8)} = (x_4, x_5, x_6)$ for an objective of 0.

! if one of these would be negative, there would be a problem (eg. empty domain)

We can try by hand to increase z by increasing one and only variable: highest limitation is $x_1 \leq 5/2$ with constraint x_4.

Now... what's next? It's <u>PIVOT</u> time (Dantzig's idea). We call x_1 the leaving var and x_4 the entering var.
We can exchange the role of x_1 and x_4 and get

$$
(D_1) : \begin{cases} x_1 = 5/2 - x_4/2 - 3x_2/2 - x_3/2 \\ x_5 = 1 + 2x_4 + 5x_2 \\ x_6 = 1/2 + 3x_4/2 - x_2/2 \\ \eta = 25/4 - 5x_4/2 - 7x_2/2 + x_3/2 \end{cases}
$$

We have that (D_1) is equiv. to (D_0) and thus to (P).

We will now iterate the process, choosing a new entering var.

To increase η, we can only increase x_3, but we are constrained to $x_1 \leq 1$ by x_6's constraint. By pivot, we obtain

$$
(D_2) \begin{cases} x_1 = 2 - x_4 - 2x_2 + x_6 \\ x_2 = 1 + 2x_4 + 5x_2 \\ x_3 = 1 + 3x_4 + x_2 - 2x_6 \\ \eta = \underline{\underline{13}} - x_4 - 3x_2 - x_6 \end{cases}
$$

No entering variable! We are sitting on an optimal solution and the simplex algorithm.

This means the optimal for (P) is 13 as $x_1, \dots, x_6 \geq 0$.

! We also have a certificate of optimality.

(P) : $\max\ 5x_1 + 4x_2 + 3x_3$

s.t. $\left|\begin{array}{ll} x_1, x_2, x_3 \geq 0 & \\ 2x_1 + 3x_2 + x_3 \leq 5 & (1) \\ 4x_1 + 3x_2 + 2x_3 \leq 11 & (2) \\ 3x_1 + 4x_2 + 2x_3 \leq 8 & (3) \end{array}\right.$

In (D_2) we have

$$z = 13 - \underline{x_4} - 3x_2 - \underline{x_6}$$

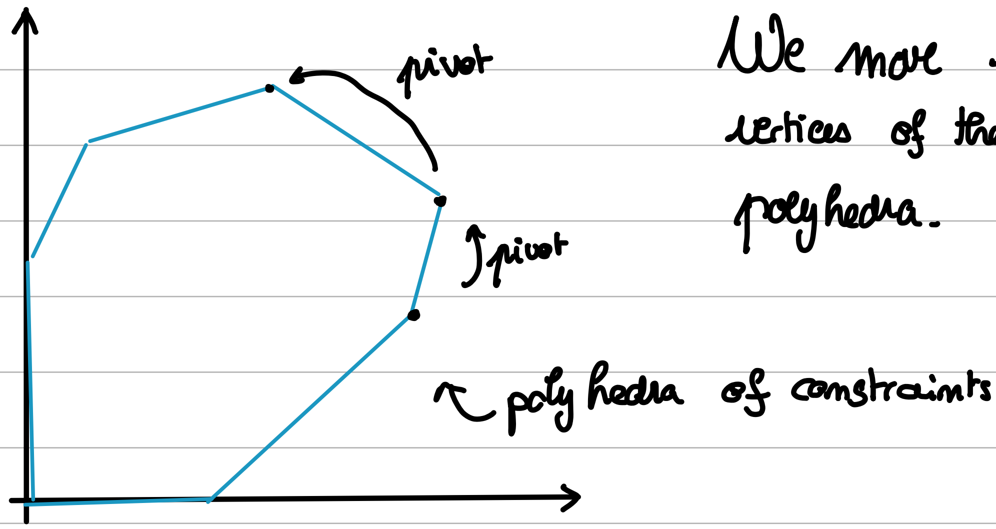
slack

and their coef is $-(1)$.

We sum $1 \times (1)$ and $1 \times (3)$ and get

$$\underbrace{5x_1 + 7x_2 + 3x_3}_{\text{Objective function}} \leq 13$$

Intuition: what are pivots?



We move between adjacent vertices of the constraint polyhedra.

How many steps? Consider P with n vertices and m facets.
The <u>skeleton</u> of P is the graph obtained from vertices of P and facets of P.
An upper bound on the number of pivots is diam(skeleton).

For the cube, we have a $\dim^o$ of 3, with diameter of 3 and 6 facets.
For K_4 the complete 4-vertices-graph, we have a $\dim^o$ of 3 with 4 facets and a diameter of 1.

It is conjectured that

$$\text{diam} \leq \#\text{facets} - \dim^o \qquad \text{(Hirsch's conjecture)}$$

In general, this is false!

III Applications of linear programming

Consider the two-player game of Morra:

1) Alice hides one or two coins;
2) Bob hides one or two coins
3) Alice & Bob announce one or two coins.

Each player will have a pair $(i,j) \in [2] \times [2]$ where i is the hidden number of coins and j is the announced number.

This is a zero-sum game: the goal is for each player to guess the other's hidden num. of coins. If one of the player guesses correctly, but not the others, the winner wins all the hidden coins.

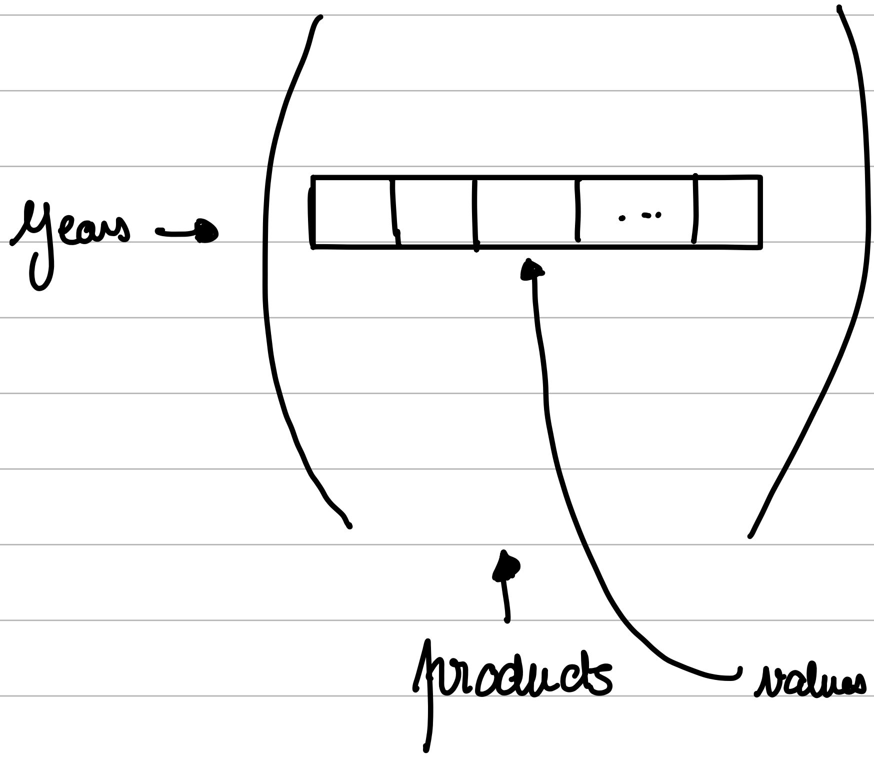
Alice has a pay off matrix:

$$
\begin{array}{cc}
 & \begin{array}{cccc} (11) & (12) & (21) & (22) \end{array} \leftarrow \text{Alice} \\
\text{Bob} \begin{array}{c} (11) \\ (12) \\ (21) \\ (22) \end{array} & \begin{pmatrix} 0 & -2 & 3 & 0 \\ 2 & 0 & 0 & -3 \\ -3 & 0 & 0 & 4 \\ 0 & 3 & -4 & 0 \end{pmatrix}
\end{array}
$$

In the general setting, $M = (m_{ij})_{i,j}$ with two players (the row and the column player). Each will choose one col ($\hookrightarrow i$) resp. row ($\hookrightarrow j$) and recieves m_{ij}.

<u>Example</u> Rock Paper Scisors

<u>Remark</u> We can represent how much you can bet on stock on year $n+1$.



Distribute money on products by assuming that year $n+1$ is a linear combination of the previous ones $\leadsto$ play safe.

Alice wants a probability vector which maximizes gain for every possible move of Bob. In Morra's game, it means maximizing

$$\min \begin{pmatrix} -2x_2 + 3x_3 \\ 2x_1 + 3x_4 \\ -3x_1 + 4x_4 \\ 3x_2 - 4x_3 \end{pmatrix}$$

s.t. $x_1 + x_2 + x_3 + x_4 = 1$

and $x_1, \dots, x_4 \geq 0$

This is not exactly a LP, but we can easily translate it into one:

$$\max y \text{ st } \left| \begin{array}{ll} -2x_2 + 3x_3 \geq y & (1) \\ 2x_1 + 3x_4 \geq y & (2) \\ -3x_1 + 4x_4 \geq y & (3) \\ 3x_2 - 4x_3 \geq y & (4) \\ x_1 + x_2 + x_3 + x_4 = 1 & (5) \\ x_1, \dots, x_4 \geq 0 & \end{array} \right.$$

We can find an unconventional solution (alternatively, we can use the Simplex algorithm to get a solution).
The game is symmetric thus $y = 0$. By $3 \times (2) + 2 \times (3)$, we get that $x_4 = 0$ and $x_1 = 0$. Finally, by $x_2 = 1 - x_3$, we can conclude that:

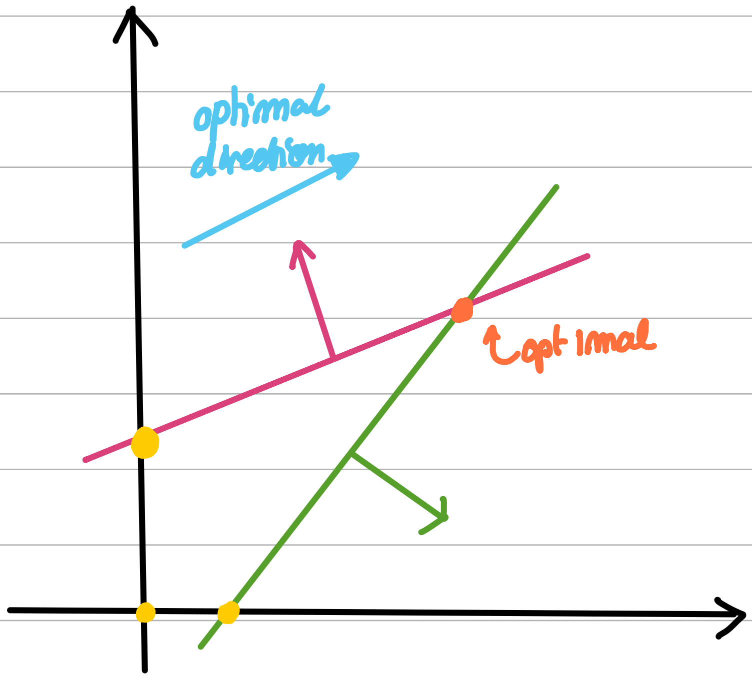
$$x_3 \geq \frac{2}{5} = 0{,}4 \quad \text{and} \quad x_2 \leq \frac{3}{7} = 0{,}\overline{428571}.$$

The optimal strategy for Alice is to pick $t \in [0.4, 0.\overline{42871}]$ and to play (2, 1) w/probability t and (1, 2) w/ probability $1-t$.

II Visualizing the pivots

Consider (P) : maximize $x_1 + x_2$

such that
$$x_1 - x_2 \leq 1 \qquad C_1$$
$$-x_1 + 2x_2 \leq 2 \qquad C_2$$
$$x_1, x_2 \geq 0$$



$$(D_0) \begin{cases} x_3 = 1 - x_1 + x_2 \\ x_4 = 2 + x_1 - x_2 \\ \hline z = x_1 + x_2 \end{cases}$$

Solution associated with (D_0) is obtained by $x_1 = 0$ and $x_2 = 0$

↳ the solution S_0 is in the intersection of n hyperplanes

x_1 enters

x_3 leaves

$$(D_1) \begin{cases} x_1 = 1 - x_3 + x_2 \\ x_4 = 3 + x_3 - x_2 \\ \hline z = 1 - x_3 - 2x_2 \end{cases}$$

S_1 is at the intersection of $x_2 = 0$ and $x_3 = 0$, that is, $x_1 + x_4 = 1$.

x_2 enters
x_4 leaves

$$(\mathcal{D}_2)\begin{cases} x_1 = 4 - 2x_3 - x_4 \\ x_2 = 3 - x_3 - x_4 \\ \hline z = 7 - 3x_3 - 2x_4 \end{cases}$$

S_2 is at the intersection of $x_1 - x_2 = 1$ and $-x_1 + 2x_2 = 2$.

Remark $3 \times C_1 + 2\, C_2$ gives us $x_1 + x_2 \leq 7$.

This is a certificate of optimality

Each pivot can be seen on the polyhedral domain as a move from one vertex v to another vertex v' such that:
- vv' is an edge of the domain

or
- $v = v'$ degenerate pivot which happens if v is represented by more than one facet of the domain.

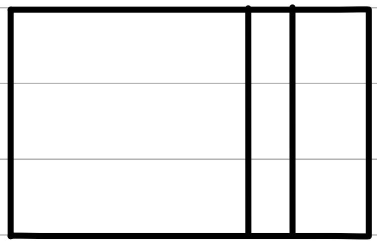
V Overview of the Simplex

- Start with an initial dictionnary $\mathcal{D}_0$
  Lo can be that some constants are < 0
  (c.f next part)

If not, then 0 is a solution and we can thus iterate the pivots.

When in dictionnary D_j there exists $c>0$ with x_j entering.

D_j



$z = \quad c x_j$

If all coeficients x_j in D_j are ≥ 0

then the LP is <u>unbounded</u>!

For example,

$$\left\{\begin{array}{l} x_3 = 4 + 2x_2 + x_4 \\ x_1 = 2 \qquad\quad - x_4 \\ \hline z = 6 + x_2 - x_4 \end{array}\right.$$

x_2 enters but there are no leaving variables

Putting $x_1 := t$, we get a half-line of solution given by

$$\vec{x} = \left(t, 0, 4+2t, 0\right)$$

with $z = 6+t$.

The solution is UNBOUNDED!

When there are no entering variables, all coefficients in z are ≤ 0. This means we have found the optimum!

The only question is TERMINATION.

<u>Remark:</u> A dictionnary is defined by the choice of the n possible non basic variables among $n+m$ variables.

(P) in dim° n with m variables has at most $\binom{n+m}{n}$ possible dictionnaries.

If the simplex does not terminate (and choices are deterministic) then it cycles into a sequence of dictionnaries

$$D_1 \to D_2 \to \cdots \to D_t \to D_2 \to \cdots$$

<u>Remark:</u> In such a case, the objective does not increase, thus it stays on the same vertex.

To see that pivot $D_1 \to D_2$ with z that doesn't increase.

$$\text{In } (D_1), \quad \begin{array}{l} \vdots \\ x_i = b_i - \dots a_{ij} x_j \dots \quad \text{with } a_{ij} > 0 \\ \vdots \\ z = \dots c_j x_j \dots \quad \text{with } c_j > 0 \end{array}$$

with x_j is entering and x_i is leaving.

We must have $b_i = 0$ otherwise z augments by

$$\frac{c_j - b_i}{a_{ij}}$$

Thus, in the solution associated to (D_1) we have $x_j = 0$.

solution

In S_2 associated to (D_2) we have $x_j = 0$ and also

$$(D_2)\begin{cases} \vdots \\ x_j = 0 + \cdots \\ \vdots \end{cases}$$

Moreover, (D_2) has all non-basic variables of (D_1) (save x_j) thus the solution is the same

$$S_1 = S_2.$$

Avoiding Cycling

We add a very small perturbation to all constraint. Every vertex of the domain is now derived by a unique set of n hyperplanes. To recover the original solution, you use rounding.

→ Do this formally with a sequence of infinitesimal

$$\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_n.$$

Bland's rule

- Choose every entering variable with lowest index (among the possible candidates).
- Same for leaving

Theorem Simplex does not cycle with Bland's rule.

How many steps? We do not even know is a poly(n+m) path exists from one vertex to another.

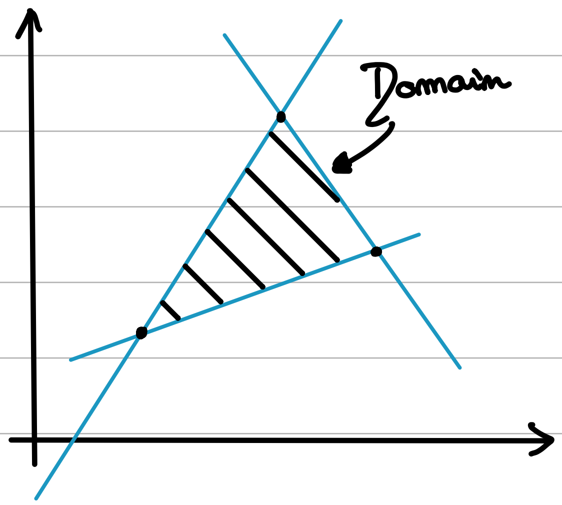
VI The first phase (Initialization)

How do we start when 0 is not a solution?

(P) max $2x_1 + x_2$

st
$$\begin{cases} -2x_1 + x_2 \leq -2 \\ x_1 - 2x_2 \leq -2 \\ x_1 + x_2 \leq 7 \\ x_1, x_2 \geq 0 \end{cases}$$

☹ 0 is not a solution.
((D_0) contains $x_3 = \boxed{-2} + \dots$)



We have to "jump" on one vertex of the domain.

To do that we just have to solve

$$\text{minimize } x_0 \quad (\text{i.e. maximize } -x_0)$$

such that
$$\begin{cases} -2x_1 + x_2 \leq -2 + x_0 \\ x_1 - 2x_2 \leq -2 + x_0 \\ x_1 + x_2 \leq 7 + x_0 \\ x_1, x_2 \geq 0 \end{cases}$$

The initial dictionnary of this other LP is

still < 0

$$(D_0') \begin{cases} x_3 = \boxed{-2} + 2x_1 - x_2 + x_0 \\ x_4 = \boxed{-2} - x_1 + 2x_2 + x_0 \\ x_5 = \boxed{7} - x_1 - x_2 + x_0 \\ \hline w = x_0 \end{cases}$$

Now we do an <u>illegal pivot!</u>

x_0 will enter and leaving is the one with min value, for instance x_3.

All values are ≥ 0 :)

$$(D_1') \begin{cases} x_0 = \boxed{2} - 2x_1 + x_2 + x_3 \\ x_4 = \boxed{0} - 3x_1 + 3x_2 + x_3 \\ x_5 = \boxed{9} - 3x_1 + x_3 \\ \hline w = -2 + 2x_1 - x_2 - x_3 \end{cases}$$

$\wr$

$\Big\downarrow$ Iterates pivots

$$(D_3')\quad \left\{\begin{array}{l} x_2 = 2 + 2x_4/3 - x_0 + x_3/3 \\ x_1 = 2 + x_4/3 - x_0 + 2x_3/3 \\ x_5 = 3 - x_4 + 3x_0 - x_3 \\ \hline w = -x_0 \end{array}\right.$$

$$\boxed{\text{If the optimum } w \text{ is } <0 \text{ then the domain is empty.}}$$

To solve (P) we use the initial dictionnary:

$$(D_0):\ \left|\begin{array}{l} x_2 = 2 + 2x_4/3 \cancel{-x_0} + x_3/3 \\ x_1 = 2 + x_4/3 \cancel{-x_0} + 2x_3/3 \\ x_5 = 3 - x_4 \cancel{+3x_0} - x_3 \\ \hline \eta = \underbrace{6 + 4x_4/3 + 5x_3/3} \end{array}\right. \quad \Bigg\}\ \text{from } (D_3') \text{ without the } x_0\text{'s}$$

$$\begin{aligned} \eta &= 2x_1 + x_2 \\ &= 4 + 2x_4/3 + 4x_3/3 \\ &\quad + 2 + 2x_4/3 + x_3/3 \\ &= 6 + 4x_4/3 + 5x_3/3 \end{aligned}$$

<u>Homework</u>: Code the Simplex algorithm

VII Duality

The goal is to certify the optimality of a solution.

Example: maximize $z := 4x_1 + x_2 + 5x_3 + 3x_4$
under the constraints

(P) primal

$$
\begin{aligned}
x_1 - x_2 - x_3 + 3x_4 &\le 1 && \Big) \times y_1 \geq 0 \\
5x_1 + x_2 + 3x_3 + 8x_4 &\le 55 && \Big) \times y_2 \geq 0 \\
-x_1 + 2x_2 + 3x_3 - 5x_4 &\le 3 && \Big) \times y_3 \geq 0 \\
x_1, \dots, x_4 &\geq 0
\end{aligned}
$$

OPT: $(0, 14, 0, 5)$

How to find an upper bound on (P)?

↳ Make linear combinations of the constraints (with non-negative coefficients, $y_1, \dots, y_n$) in such a way that the left hand terms "majorates" the objective function. Then,

$$y_1 + 55y_2 + 3y_3$$

will be an upper bound!

The best bound one can derive is a solution of the following DUAL linear problem:

(D) "dual" minimize $y_1 + 55y_2 + 3y_3$

with the constraints

$$\begin{cases} y_1 + 5y_2 - y_3 \geq 4 \\ -y_1 + y_2 + 2y_3 \geq 1 \\ -y_1 + 3y_2 + 3y_3 \geq 5 \\ 3y_1 + 8y_2 - 5y_3 \geq 3 \\ y_1, y_2, y_3 \geq 0 \end{cases}$$

OPT: $(11, 0, 6)$

Lemma (Weak Duality Theorem)

The value of a solution of (D) is always at least the value of any solution of (P).

Proof A LP (P) reads

(P) maximize $c^T x$ such that $\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$

and the dual is

(D) minimize $b^T y$ such that $\begin{cases} A^T y \geq c \\ y \geq 0 \end{cases}$

Assuming that x is a solution of (P) and y

a solution of (D). we have:

$$\text{value of (P)} = c^T x \leq (y^T A)\, x = y^T (A x) \leq y^T b = \underset{(D)}{\text{val of}}$$

$\square$

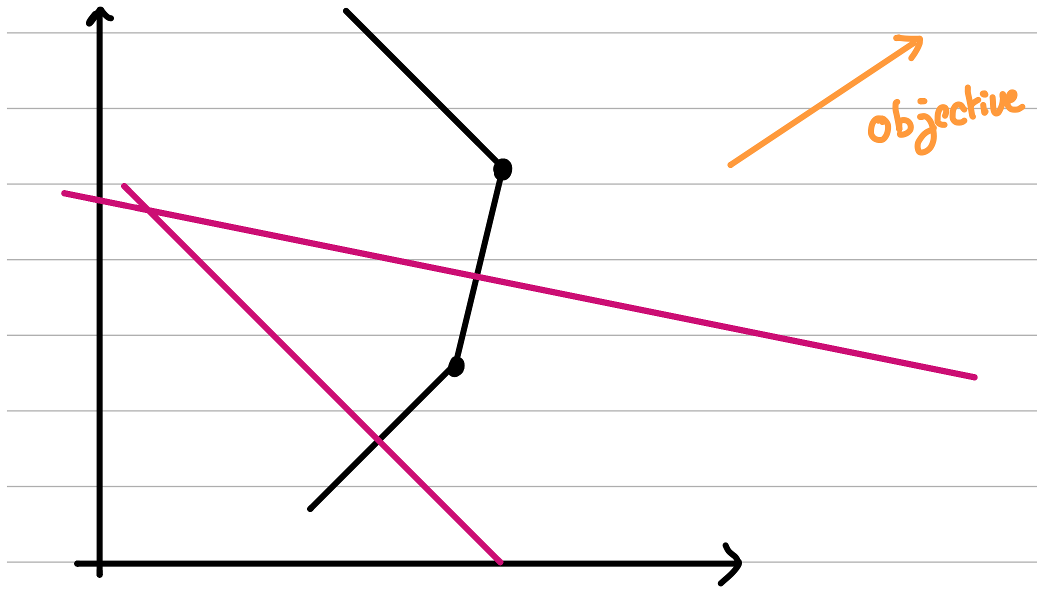
<u>Theorem</u> (Gale, Khun, Tucker in '51).

If (P) has an optimal solution then (D) has an optimal solution with the same value.

Many proofs; one can be made on the simplex and the fact that, if (P) has a solution, then the final dictionnary provides the linear combinations.

A vague idea is the following. When the simplex steps, the objective function in the positive cone of the (normal) vectors corresponding to the hyperplanes defined by the non-basic variables.

The coefs (i.e. the dual solution) can be read in the last dictionnary.



Consequences & Remarks

① Can always easily certify optimality

② "Decision is optimization"

$\boxed{\text{Decision LP}}$

Input : Given a set of inequalities $Ax \leq b$

Output : TRUE if there exists a solution x (return x)
FALSE otherwise

Remark that if DECISION LP is in $\mathbb{P}$ then so is Solving LP

We want to solve $\max\ c^T x$ st $Ax \leq b$ and $x \geq 0$

Form a decision version on variables (x,y) where y are the variables of the dual, and ask for a point in the polyhedron:

$$\text{Domain: } \left| \begin{array}{l} Ax \le b \\ A^T y \le c \\ c^T x = b^T y \end{array} \right.$$

Any solution (x,y) of this domain is an optimal solution of (P) and an optimal solution of (D).

All research to find an algorithm to solve LP points in the direction of solving Decision LP.

3 Certificate for Decision

A system of equalities $Ax = b$ has no solution IFF A linear combination of these equalities give $0=1$.

Gauss

A system of inequalities $Ax \leq b$ has no solution IFF A non-negative combination of these inequalities give $0 \leq -1$.

↑ Strong duality

A system of polynomial (multivariable)

$P_1(x_1 \ldots x_n)$

$\vdots$

$P_m(x_1 \ldots x_n)$

has a common zero IFF there exists multipliers $Q_1, \ldots, Q_m$ (polynomials) such that $\sum Q_i P_i = 1$

↑ Hilbert's Nullstellensatz

It is very to code 3SAT with polynomials but the size of the Q_i's are exponential.

Remark

- The dual of the dual is the primal.
- If a variable x_i in the primal is not constrained to be non-negative, it gives rise to an equality in the dual.

Conversely, if

$$(P) \quad \max \ldots \quad \text{st} \quad \ldots \quad a_i x = b \ldots$$

then the y_i's are not constrained to be ≥ 0.

VIII Two examples of duality

1) Matching is the dual of vertex cover

Given $G = (V, E)$ and I the incidence matrix of G

$I := (I_{v,e})_{v \in V, e \in E}$ with $I_{v,e} = 1$ iff $v \in e$.

The (Fractional) Minimum vertex cover is

$$\min \mathbb{1}^T x \quad \text{st. } Ix \geq 1 \quad x \geq 0$$

The dual is $\quad$ maximize $\mathbb{1}^T y$ st. $I^T y \leq 1$, $y \geq 0$.

Maximizing of weights on edges s.t. no vertex recieves total weight more than 1 on its incident edges

$\longrightarrow$ max fractionnal matching

2) Duality for max flow (bad case)

s is the source and t terminal
Flow is weight x_{uv} on each arc uv.

The relaxed max flow problem is

$$\text{maximize} \sum_{uv \text{ arc}} x_{uv}$$

$$\text{subject to } \forall v \notin \{s,t\}, \sum_{uv} x_{uv} - \sum_{vw} x_{vw} = 0 \quad (p_v)$$

$$\forall uv \text{ arc}, \quad 0 \leq x_{uv} \leq c_{uv} \quad (y_{uv})$$

Dualize two types of variables:

- p_v's : unconstrained "potential"
- $y_{uv} \geq 0$

The dual of flow is

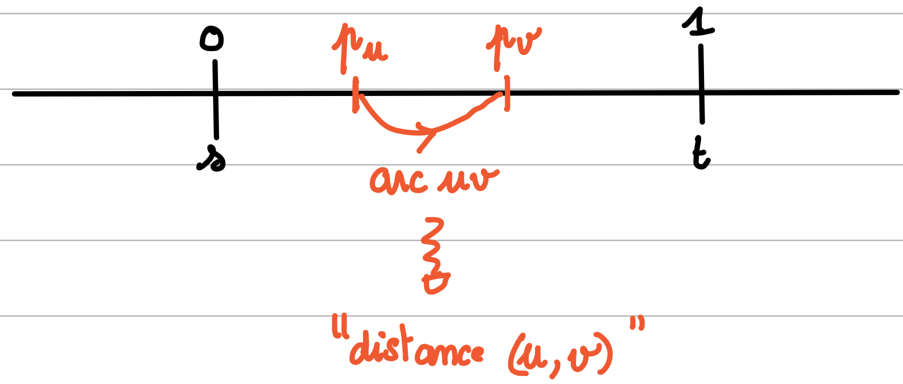
$$\text{minimize} \sum_{uv \text{ arc}} c_{uv} y_{uv} \quad \text{such that} \quad \begin{array}{|l} y_{uv} - p_v + p_u \geq 0 \ \forall uv \ {\scriptstyle u \neq s,\, v \neq t} \\ y_{ut} + p_u \geq 1 \quad \forall ut \\ y_{sv} - p_v \geq 0 \quad \forall sv \\ y_{uv} \geq 0 \quad \forall uv \\ p_v \text{ is not constrained} \end{array}$$

This corresponds to (by setting $p_t = 1$ and $p_s = 0$)

$$y_{uv} \geq p_v - p_u$$

The dual of flow involves finding a function p_v for every $v \neq s, t$ where $p_s = 0$ and $p_t = 1$

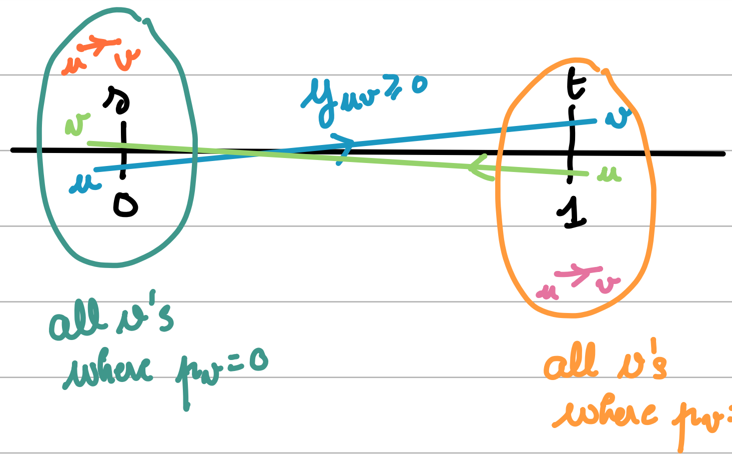
p is a potential



The cost of uv is $c_{uv}\, y_{uv}$ where $y_{uv} \geq \underbrace{p_u - p_v}_{\text{"distance"}}$

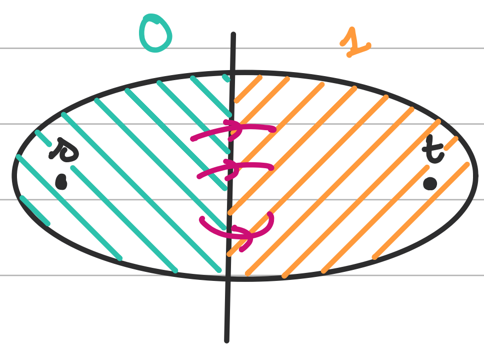
Unfortunately, backward arcs correspond to $y_{uv} = 0$.

<u>Integer solutions</u> involve setting p_v to 0 or 1



arcs in the same "group" have $y_{uv} = 0$

This corresponds exactly to the minimum st-cut problem



The value of the function is $\sum_{\substack{p_u=0 \\ p_v=1}} c_{uv}$

IX Concrete interpretation of dual variables

Given raw materials A, B, C needed to produce products x_1, x_2, x_3. The respective compositions are

In our stock we have the following amount:

A is 5

B is 4

C is 6

	A	B	C
x_1	1	2	3
x_2	2	3	1
x_3	3	2	1

We can sell x_1 for 1

x_2 for 2

x_3 for 2

(and we admit rational solutions)

(P) maximize $x_1 + 2x_2 + 2x_3$

such that

$$\begin{cases} x_1 + 2x_2 + 3x_3 \leq 5 \\ 2x_1 + 3x_2 + 2x_3 \leq 4 \\ 3x_1 + x_2 + x_3 \leq 6 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

OPT: $\left(0, \frac{5}{2}, \frac{7}{2}\right)$ for a value of $\frac{18}{5}$

(D) minimize $5y_1 + 4y_2 + 6y_3$

such that

$$\begin{cases} y_1 + 2y_2 + 3y_3 \geq 1 \\ 2y_1 + 3y_2 + y_3 \geq 2 \\ 3y_1 + 2y_2 + y_3 \geq 2 \\ y_1, y_2, y_3 \geq 0 \end{cases}$$

OPT: $\left(\frac{2}{5}, \frac{2}{5}, 0\right)$ for a value of $\frac{18}{5}$

<u>Interpretation</u> The value of the dual correspond to the cost of raw materials from your point of view (at optimum of (P)).

Suppose the value of A in the market is 0.5. Then, maybe it is better to sell $\varepsilon > 0$ amount of A. $\rightsquigarrow$ (P_ε).

In (P_ε), the first constraint is now

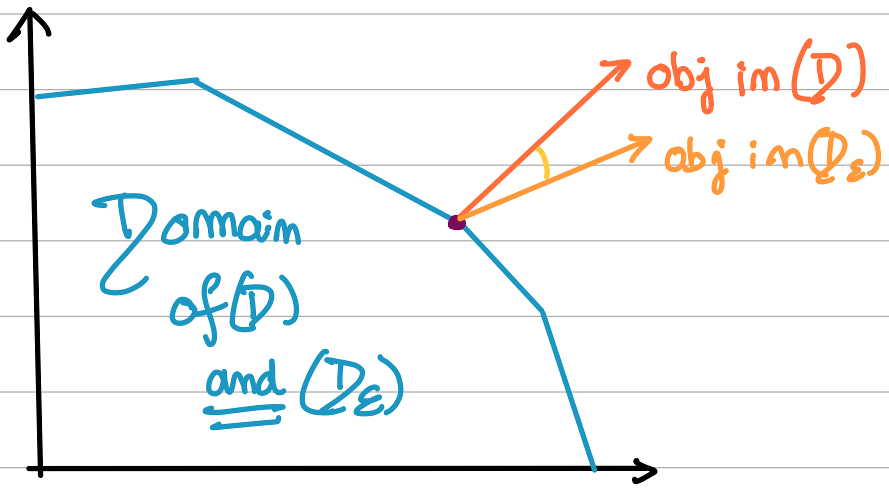
$$x_1 + 2x_2 + 3x_3 \leq 5 - \varepsilon$$

☹ It is very hard to see how (P) evolves since the domain is changing.

<u>From the point of view of (D):</u>

(D_ε) minimize $(5-\varepsilon)y_1 + 4y_2 + 3y_3$
such that | EXACTLY THE SAME CONSTRAINTS AS (D)

The domain is fixed but only the objective function is tilted by ε.



Under the (natural) hypothesis that the optimal solution of (D_ε) does not change, the value of (D) is

$$0.4 \times (5-\varepsilon) + 0.4 \times 4 +$$

Thus $obj(D_\varepsilon) - obj(D)$ is -0.4ε

Since the ε part is sold for 0.5ε we gain 0.1ε.

How large do we choose ε without the optimal solution of (D) changing? We will use Complementary Slackness.

X Complementary Slackness.

<u>Goal:</u> given a program (P) and a potential optimal solution x, decide if x is actually optimal.

! EASY : Just amount to solve a system.

Consider (P): maximize $x_1 + 2x_2 + 2x_3$

such that
$$\left|\begin{array}{l} x_1 + 2x_2 + 3x_3 \leq 5 \\ 2x_1 + 3x_2 + 2x_3 \leq 4 \\ 3x_1 + 2x_2 + x_3 \leq 6 \\ x_1, x_2, x_3 \geq 0 \end{array}\right.$$

How do we check if $\left(0, \frac{2}{5}, \frac{3}{5}\right)$ is optimal? Check constraints

C_1 is equality

C_2 is equality

C_3 is strict $\frac{11}{5} < 6$

From the dual's point of view: y_1, y_2, y_3 are multipliers of constraints which certify equality of OPT in (P) and (D).

This means you cannot use a strict inequality, thus $y_3 = 0$.

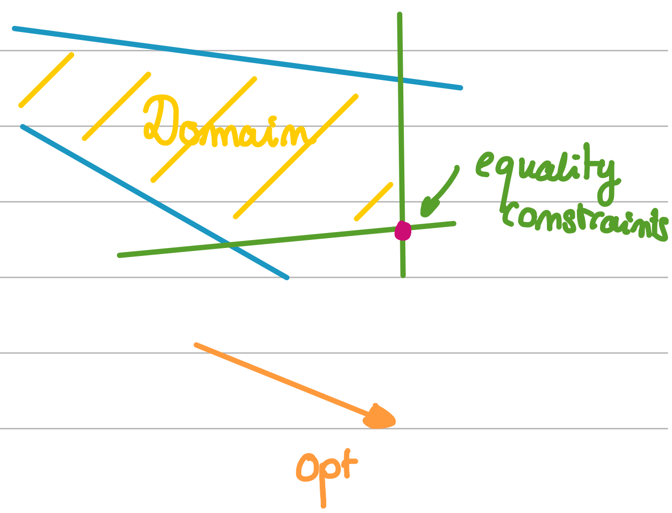
Now, let us calculate the (potential) optimal solution of (D).

Since $x_2 \neq 0$, the second constraint of (D) is an equality
Since $x_3 \neq 0$, the third constraint of (D) is an equality

We are left with the system

$$\begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Thus $$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 2/5 \\ 2/5 \end{pmatrix}$$



Complementary Slackness theorem

(i) Either $x_i = 0$ or the i^{th} constraint of (D) is equality
(ii) Either $y_i = 0$ or the i^{th} constraint of (P) is equality

Let us go back to ε. How do we compute the largest possible value of ε?

We wanted to check that $(2/5, 2/5, 0)$ stays on the optimal solution of (D_ε).

We apply Complementary Slackness.

In (D_ε), 1st constraint must be strict,
2nd & 3rd constraints must be equality.

Thus x_1 must be 0.

Because of $\left(\overset{y_1}{\underset{\neq 0}{\frac{2}{5}}}, \overset{y_2}{\underset{\neq 0}{\frac{2}{5}}}, \overset{y_3}{\underset{= 0}{0}}\right)$, the two first constraints must be equality.

We only have to solve:

$$\begin{cases} 2x_2 + 3x_3 = 5 - \varepsilon \\ 3x_2 + 2x_3 = 4 \end{cases}$$

This gives $x = \left(0, \frac{2}{5} + \frac{2}{5}\varepsilon, \frac{7}{5} - \frac{3\varepsilon}{5}\right)$. This is a valid until x_3 becomes negative (strictly): $\frac{7}{5} - \frac{3}{5}\varepsilon \geq 0$, ie. $\boxed{\varepsilon \leq \frac{7}{3}}$

Complementary Slackness is first used in "Sensitivity Analysis" and secondly in the primal dual algorithm.

Part # 2

Integer Polytopes & Total unimodularity

I Polytopes & Polyhedra

A polytope is a convex hull of a finite number of points in $\mathbb{R}^n$.
A polyhedron is a finite intersection of half spaces.

The dimension of a polyhedron P is
- the largest dimension of a hull included in P;
- the smallest dimension of an affine space containing P.

A face of P is a subset $H \cap P \subseteq P$ where
- H is a hyperplane
- P is contained in one of the half spaces defined by H: H^+ or H^-

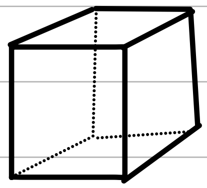
$$H = \{ x \mid b^T x = c \} \qquad H^+ = \{ x \mid b^T x \geq c \}, \quad H^- = \{ x \mid b^T x \leq c \}.$$

Faces of P are polyhedra:

faces of dimension 0 are vertices of P
faces of dimension 1 are edges of P

If P has dimension p, facets of P are faces with dimension $d-1$.

For the Hypercube polyhedron, we have



vertices are $\{0,1\}^n$

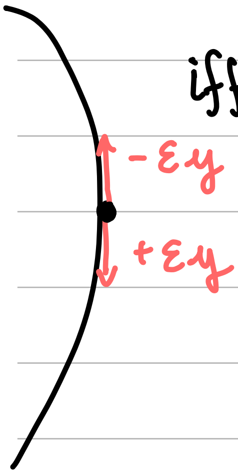
facets are defined by $\left| \begin{array}{l} x_i \leq 1 \\ \text{OR} \\ x_i \geq 0 \end{array} \right.$ $\quad i = 1, \dots, n$

<u>Remark</u> (i) A point x of (P) is a vertex of P

iff $\forall$ interval $[u,v] \subseteq P$ such that $x \in [u,v]$

then, $x = u$ or $x = v$.

iff $\forall y \neq 0$, if $[x - \varepsilon y, x + \varepsilon y] \subseteq P$ then $\varepsilon = 0$



<u>Theorem</u> Bounded polyhedra are exactly polytopes

(ii) When trying to model a discrete problem with LP, one first consider the solutions of it as integer points (usually, 0,1 coordinates) and consider convex hull.

<u>Example</u> Consider F a FNC on variables $(x_1, \dots, x_n)$,

S is the set of all solutions to F

seen as $\{0,1\}$ vectors of $\mathbb{R}^n$.

P is the convex hull of S.

The central question is:

Can we describe P as a polyhedron?

That is, find the facets.

(1) If possible with a polynomial number of facets

↳ then it is solvable in poly time, just use LP.

(2) If P has an exponential number of facets, but there is a Separation Oracle then it is poly-time solvable.

A Separation Oracle is a blackbox which takes as input $x \in \mathbb{R}^n$, and answers

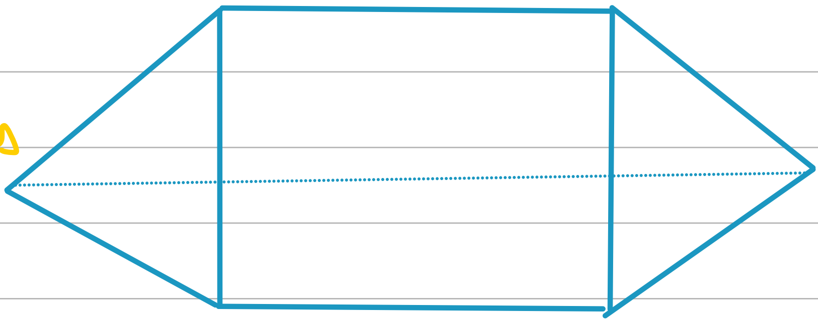
- TRUE if $x \in P$
- If not, output valid constraints $a^T x \leq b$ and $a^T x > b$.

⇝ Ellipsoid Algorithm

(3) It is sometimes the case that P has an exponential number of facets but P is the projection of a polyhedron Q (in higher dimension) such that Q has a polynomial number of constraints.

Solve LP in Q
project the solution } Poly-time

Extended Formulations



projection of  $\subseteq \mathbb{R}^3$ in $\mathbb{R}^2$.

Tent in $\mathbb{R}^3$ has 5 facets but the projection has 6.

__Example__ The spanning tree polytope can have exponential number of facets but it has $O(n^3)$ facets in higher dim.

II The bipartite matching polytope

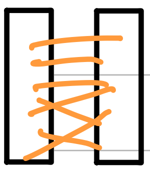
Every weight e has a weight w_e. Find a matching with maximum weight. Consider the polytope M given as the convex hull of points p of $\mathbb{R}^{|E|}$ such that

$$p = (\underbrace{x_1, \dots, x_m}_{\substack{e_1 \; \dots \; e_m \\ \text{edges}}}) \quad \text{and } \{e_i \mid x_i = 1\}$$

is a matching.

M is a $0,1$ polytope : all coordinate of vertices are 0 and 1.

! The number of vertices of M is potentially exponential.



$K_{n,n}$ complete

n n

has $n!$ perfect matchings

Let us guess the facets :

- All $x_i \geq 0$ $\quad i = 1, \dots, m$

(M'): • For every vertex v:
$$\sum_{v \text{ in edge } e_i} x_i \leq 1$$

These are valid constraints and,

Theorem These are exactly the facets of M

<u>Proof</u> We want to show $M = M'$.

1) We have $M \subseteq M'$ since those are valid constraints

2) To show that $M' \subseteq M$, it suffices to show that every vertex of M' is $0,1$-valued. Indeed, it gives that it is a matching, hence in M.

Assume, for contradiction, x is a vertex of M' and not $0,1$-valued.

Then consider the set S of all coordinates of x which are not $0,1$-valued. These coordinates are edges of G.

We have that S is a subgraph.

III Totally Unimodular Matrices

A $(0, 1, -1)$ - matrix is **totally Unimodular** (TU) if the determinant of all its submatrices is 0, 1, or -1.

Example

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

yes!   yes!   no!

det = 2

for the full matrix

Theorem (Seymour)

Checking TU is in P.

↳ there is a poly-time algorithm

He characterized all TU matrices by

- basic ones
- operations to build new

Theorem Given a polyhedron

$$P = \{ x \in \mathbb{R}^n \mid Ax \leq b \}$$

if b is integer and A is TU then all vertices of P are integer-valued.

Proof

$$\left.\begin{array}{l} Ax \leq b \\ x \text{ is a vertex} \end{array}\right] \Rightarrow \begin{array}{l} x \text{ is a solution of } A'x = b \\ \text{for a non singular matrix of } A \end{array}$$

$\boxed{\text{CRAMER:}}$ The ith coordinate of x is

$$x_i = \frac{\det(A_i \mid b)}{\det(A)}$$

← integer

← full rank & TU so 1 or -1

where "$A_i \mid b$" is A's ith column was replaced by b.

<u>Apply to bipartite matching</u>

Consider

Examples of TU matrices

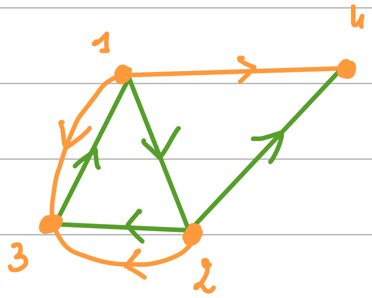
1) incidence matrices of bipartite graphs (0, 1 - valued)
2) incidence matrices of oriented graphs (-1, 0, 1 - valued)
3) 0, 1 - matrices where 1's are consecutives in columns

This explains why
  1) matchings
  2) flows
  3) orderings of tasks on a single machine

are easy to find.

Tutte proposes a general construction: network matrices.

- We are given an oriented tree $T = (V, A)$ on the set of nodes V $\rightsquigarrow$ network.
- We are given a set of queries seen as an oriented graph $G = (V, E)$ where $xy \in E$ means "x sends one unit of flow to y".



We define a matrix M of size $|A| \times |E|$ where

$$M_{ae} = \begin{cases} 1 & \text{if edge } a \text{ is traversed in the positive sense in the unique } xy\text{-path in } T \\ -1 & \text{if edge } a \text{ is traversed in the negative sense in the unique } xy\text{-path in } T \\ 0 & \text{otherwise} \end{cases}$$

where $e = xy$

$$M = \begin{array}{c} \\ 14 \\ 13 \\ 23 \end{array} \begin{array}{c} \begin{array}{cccc} 12 & 31 & 23 & 24 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & +1 \\ +1 & -1 & 0 & 0 \\ -1 & 0 & +1 & -1 \end{pmatrix} \end{array}$$

Proposition: Network matrices are TU.

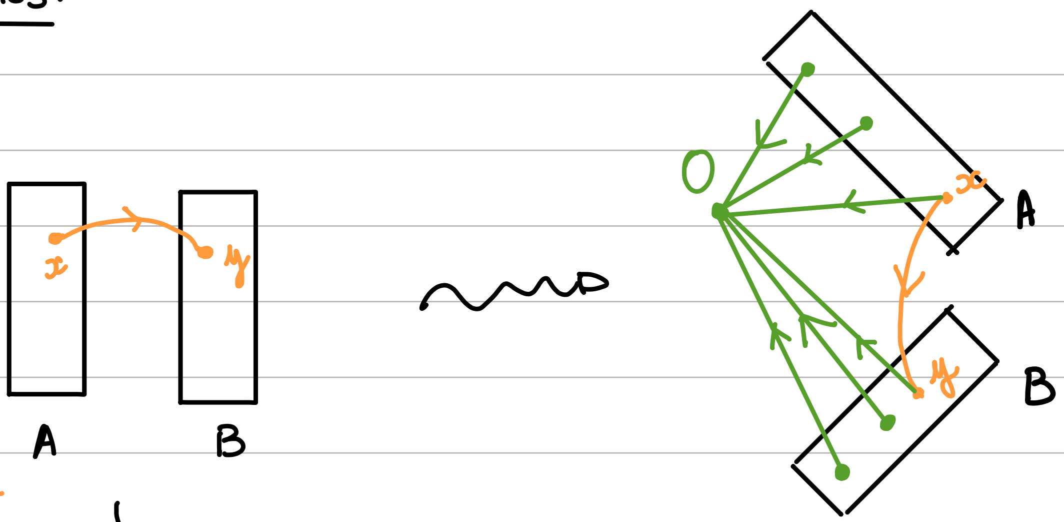
proof: Easy. Note that a submatrix is also a network matrix.

- → removing a column means removing a request
- → removing a line means contracting an edge of the network.

"Contraction / Deletion" are at the heart of Tutte's work in algebraic methods in combinatorics.

Examples.

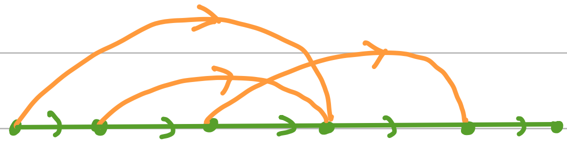
1)



$$\begin{array}{c|c} & xy \\ & 0 \\ & \vdots \\ & 0 \\ O_x & 1 \\ & 0 \\ & \vdots \\ & 0 \\ O_y & 1 \\ & 0 \\ & \vdots \\ & 0 \end{array}$$

and the network matrix is exactly the incidence matrix of the bipartite graph

2)

3) $\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ $\rightsquigarrow$ 

<u>Theorem</u> Deciding if a matrix is TU is in P.

Seymour: every TU matrix is made from network matrices and a 5×5 sporadic matrix, using elementary operations.

IV Rounding

A. Hitting Set

We define the problem:

$\boxed{\text{Minimum Cost Hitting Set}}$

<u>Input</u>: $S_1, \dots, S_n$ subsets of $[n]$

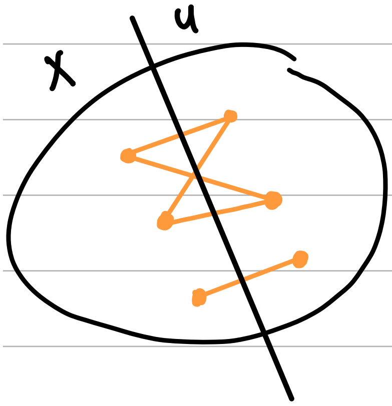
cost function $c : [n] \ni i \mapsto c_i \in \mathbb{N}$

<u>Output</u>: $I \subseteq [n]$ such that
- for all j, $S_j \cap I \neq \emptyset$
- $\sum_{i \in I} c_i$ is the smallest

This problem covers half of combinatory optimization (the other half is the dual problem).

Example: Minimum Weight Spanning Tree

Take $[n]$ = set of edges of the graph. The S_j's are the cuts.



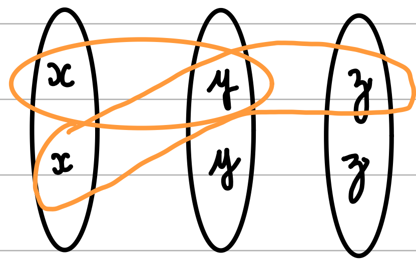
S_j is the set of edges from X to Y.

Example: SAT

$F = (x \vee y) \wedge (\bar{x} \vee y \vee z) \wedge \dots$

Take $[n]$ = all litterals.

There exists a hitting set of size |variables|
iff F is sat



S_x S_y S_z

Example: Set Cover

B. Deterministic rounding

We fix a threshold s and we round x_i's to 1 over the threshold (we choose, in the solution, the i's such that $x_i \geq s$).

Proposition If all the S_i's have a size $\leq k$ then this rounding gives a k-approximation of Min-Hitting Set when using a threshold $1/k$.

proof Let $x^* := (x_i^*)$ be the optimal solution with a value OPT^*. In particular, $OPT^* \leq OPT$.

The rounded solution A has that

$a_i = 1$ if $x^*_i \geqslant 1/k$
$a_i = 0$ otherwise

So the value ARR of the rounded solution has that ARR $\leq k \cdot OPT^*$.
And all the S_i's have that $\sum_{j \in S_i} x_j \geqslant 1$ so there exists $j \in S_i$ such that $x_i \geqslant \frac{1}{k}$ and thus $j \in A$.

We thus have ARR $\leq k \cdot OPT$.

In the $k = 2$ case, the problem is exactly vertex cover which doesn't admit $(2-\varepsilon)$ approximations (unless UGC).

C. Randomized rounding.

As $x_i \in [0,1]$, we can naturally consider it as the probability of choosing i in the solution.

If (x_i^*) is the solution of optimal value OPT^*, the expected value of the cost of a randomly chosen solution is exactly

$$\mathbb{E}\left[\sum_{i=1}^{n} c_i x_i^*\right] = OPT^*.$$

<u>Problem:</u> The solution is not necessarily a hitting set.
↳ The algorithm consists of repeating $\lceil 2 + \log n \rceil$ times this randomized algorithm by taking the union of all the solutions found.

We thus have

1) $\rightarrow$ with probability $\geq \frac{3}{4}$, a hittin set

2) $\rightarrow$ with probability $\geq \frac{3}{4}$, the total cost is $\leq 4(\log n)$ OPT.

$\Rightarrow$ with probability $\geq 1/2$, on a un hitting set qui est une $(4 \log n)$ - approximation.

Let us show 1).

For 2), we use the Markov bound: as $\mathbb{E}[\text{cost for } 1 \text{ iter}^\circ] = OPT^*$ then $\mathbb{E}[\text{cost for } \log n \text{ iter}^\circ] = \log n$, thus

$$\mathbb{P}(R \geq 4 \log n \; OPT^*) \leq \frac{1}{4}.$$

There is no $o(\log n)$ - approximation for MAX SAT.

D) Randomized Rounding for MAX SAT.

Input A CNF formula.

$$F = \underbrace{(x_1 \vee \overline{x_2})}_{c_1} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee x_3)}_{c_2} \wedge \underbrace{(x_4)}_{c_3} \wedge \cdots$$

and a function

The rest will be filled later...

Primal Dual Algorithm

The goal is to solve <u>Min Cost Hitting Set</u>. We have

Inputs
- E a set
- $T_1, \dots, T_n$ a family of subsets of E
- $c : E \ni e \mapsto c_e \in \mathbb{N}$ a cost function

Output $X \subseteq E$ with min cost and $|X \cap T_i| \geq 1$.

As a LP problem, we want

Hitting Set minimize $\sum_{e \in E} c_e x_e$ subject to $\left| \begin{array}{l} x_e \geq 0 \quad \forall e \\ \sum_{e \in F_i} x_e \geq 1 \quad \forall i \end{array} \right.$

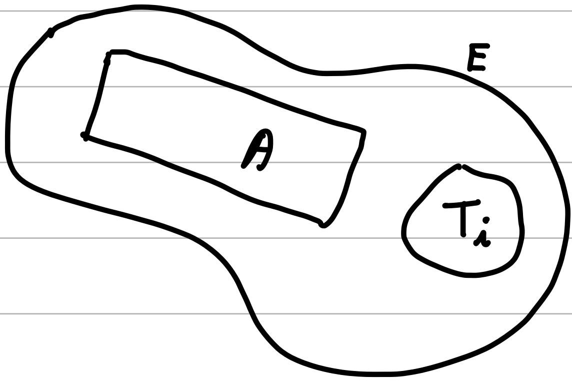
whose dual is

Packing maximize $\sum_{i=1}^{n} y_i$ subject to $\left| \begin{array}{l} y_i \geq 0 \quad \forall i \\ \sum_{F_i \ni e} y_i \leq c_e \quad \forall e \end{array} \right.$

The Primal Dual algorithm is to keep a pair (y, A)
- y is dual feasible
- $A \subseteq \{ e \in E \mid \sum_{T_i \ni e} y_i = c_e \}$. "Sharp equalities in (D)".

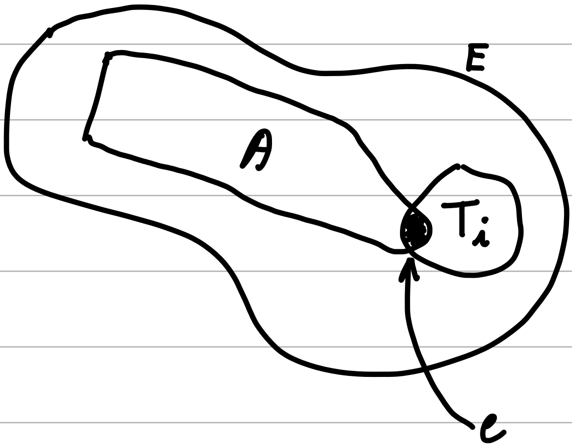
We maintain such pairs (y, A), starting from $y = 0$ and $A = \emptyset$.

$\boxed{\text{Key observation}}$ We want to stop the process when A is solution of the original Hitting Set problem. To define a step (if A is not a solution), observe that there exists a T_i such that $A \cap T_i = \emptyset$.



Here, in the dual, T_i corresponds to y_i. Thus increase y_i until some $e \in T_i$ reaches

$$\sum_{T_i \ni e} y_i = c_e$$



Then, put e in A.

$y \leftarrow 0$
$A \leftarrow \emptyset$
while $\exists k,\ A \cap T_k = \emptyset$
  increase y_k until $\exists e \in T_k,\ \sum_{T_i \ni e} y_i \leq c_e$
  $A \leftarrow A \cup \{e\}$
Return (y, A)

__Correctness__

At each step, a NEW element $e \in E$ is added to A. The set A which is returned is a solution of HS.

How good is A? What is the approximation ratio?

The cost of A:

$$c(A) = \sum_{e \in A} c_e \overset{\text{def of } A}{=} \sum_{e \in A} \sum_{T_i \ni e} y_i \overset{\text{collecting the } y_i\text{'s}}{=} \sum_{i=1}^{n} |T_i \cap A|\, y_i$$

Demote OPT the best integer solution of HS, and OPT* its relaxation.

We have $OPT \geq OPT^* = \sum_{i=1}^{n} y_i^*$ for the optimal sol° y^*

y is dual feasible thus $\sum_{i=1}^{n} y^*_i \geq \sum_{i=1}^{n} y_i$, and so

$OPT \geq \sum_{i=1}^{n} y_i$.

If one can get any upperbound c on $|T_i \cap A|$ we have

$$c(A) \leq \sum_{i=1}^{n} c\, y_i \leq c \sum_{i=1}^{n} y_i \leq c \cdot OPT$$

Thus we have a c-approximation of Hitting-Set.

__A particular case:__ When all T_i's have size at most k then the primal dual algorithm gives a k-approximation for Hitting Set.

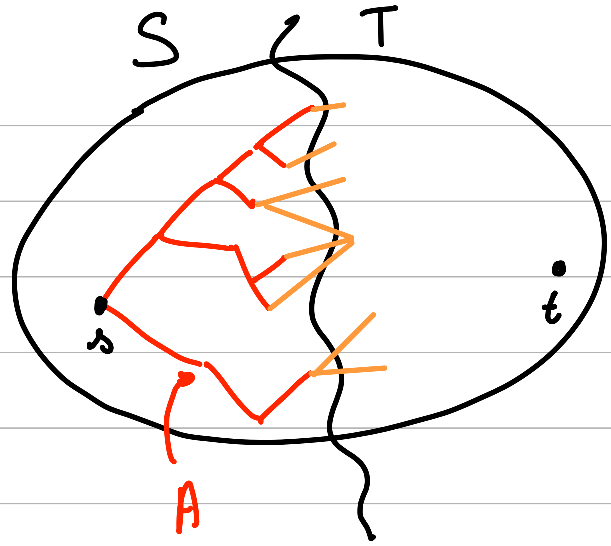
- → we know it is the best possible
- → same as deterministic rounding (but no LP is involved)

__Remark:__ • We do not need an explicit list of T_i's. We just need a separation Oracle:

__Input:__ A

__Output:__ | True if A is a hitting set
| Otherwise give a T_i such that $T_i \cap A = \emptyset$.

• A particular case is Dijkstra's algorithm: we want a min length st-path



T_i is the set of edges of st-cuts

min ℓ^0 hitting set of T_i's
$\shortparallel$
shortest path

There is an exponential number of cuts. If the separation oracle give the cut $T_i = (S, T)$ where S is exactly the set of vertices attained by s using edges in A.

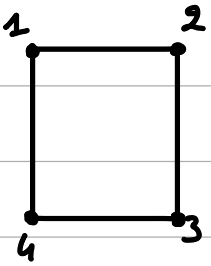
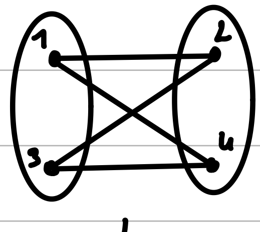
⇝ This is precisely Dijksta's algorithm.

Little catch it returns a tree containing a shortest st-path.
It needs a post process which deletes useless $e \in A$.
(Usually do this starting by the last element added to A)

Primal dual methods are very efficient in approximation or even for exact algorithms.
It requires a good separation oracle & good post treatment.

The Goemans-Williamson MAXCUT algorithm.

MAX CUT Input A graph $G = (V, E)$
Output A bipartition (X, Y) of V such that the number of edges $e(X, Y)$ from X to Y is max.

4

It is one of the 21 NP-hard problems of Garey-Johnson.

How to approximate? There is an easy 2-approx:

- RANDOMIZED Pick a random partition (X,Y)

  $\Pr[e \text{ is in } (X,Y)\text{-cut}] = 1/2$

  $$\mathbb{E}[\text{size of cut}] = \underbrace{\overset{\#\text{edges}}{m}/2}_{\text{Solution}} \geq OPT/2.$$

- DETERMINISTIC

  Start with any (X,Y)

  While there exists v which has more neighbors in its part than in the other

      Move v to the other part

  Return (X,Y)

  \# Steps $\leq m$

The returned (X,Y) has at least $m/2$ edges.

For 20 years, no better approximation was known, until a $0.87865\ldots$-approximation came...

The GW algorithm

- Not l.p. relaxation: usually, we would introduce (real) variables v_x for every $x \in V$. Here, we associate to every $x \in V$ a unit vector v_x in $\mathbb{R}^d$ (where d is "unspecified").

The objective function is to maximize the sum of the angles $\underbrace{\widehat{V_x O V_y}}_{= \theta_{xy}}$ for all $xy \in E$. But this is NP-hard.

Instead, we maximize $\displaystyle\sum_{xy \in E} \frac{1 - \langle v_x | v_y \rangle}{2}$

(with the unit vector representation of V)

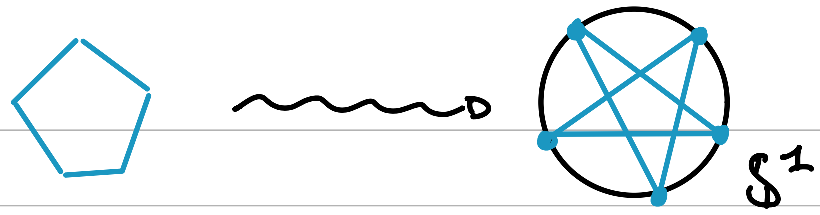
Denote OPT^* for the optimum of this max problem.
Note that $OPT^* \geq OPT$ since if OPT is attained by (X, Y) give $V_x = -1 \ \forall x \in X$
$V_y = 1 \ \forall y \in Y$

$\rightarrow \dfrac{1 - V_x V_{x'}}{2} = 0$ if $x, x' \in X$ and $xx' \in E$

$\rightarrow \dfrac{1 - V_x V_y}{2} = 1$ if $x \in X, y \in Y$ and $xy \in E$

This gives a particular representation with value OPT.

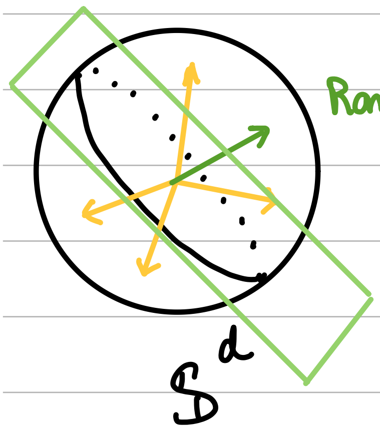
Example



Remark OPT^* can be approximated in poly time to every precision $\leadsto$ Semi Definite Programming.

This idea come from Lovász work on the Shannon Capacity (read the litterature)

How to go back to a real-world solution?



Random vector c

$\downarrow$ normal vector of

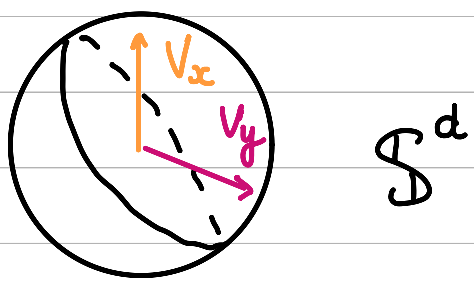
Random hyperplane that gives the separation

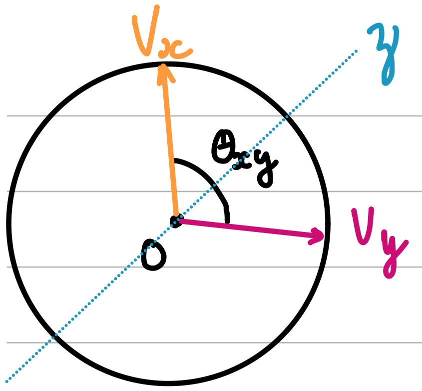
$$\hookrightarrow \left.\begin{array}{l} X = \{ V_x \mid \langle V_x \mid c \rangle \geq 0 \} \\ Y = \{ V_y \mid \langle V_y \mid c \rangle < 0 \} \end{array}\right\} S^0$$

What is the approximation ratio ?

What is the expected number of edges in this random cut?

With an edge $xy \in E$, we just have to consider the projection on the plane V_x, V_y





The probability that a random line (via θ) separates V_x and V_y?

$\longrightarrow \theta_{xy}/\pi$

Expected size of cut is $\sum_{xy \in E} \theta_{xy}/\pi = GW$

We know $\sum_{xy} \frac{1 - \overbrace{\langle V_x | V_y \rangle}^{\cos \theta_{xy}}}{2} \geq OPT$

and we are given $\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \theta / 1 - \cos\theta$

$(\alpha = 0.87865\ldots)$,

Thus $GW \geq \alpha\, OPT^* \geq \alpha\, OPT$.

Remark. Assuming the Unique Game Conjecture (which also implies that no $(2-\varepsilon)$-approx for Vertex Cover exists unless P = NP) there is no $(\alpha+\varepsilon)$-approx for MAX Cut for $\varepsilon > 0$ unless P = NP.

A semi-definite program is an optimization problem of the form

$$\max \sum_{1 \leq i,j \leq n} c_{ij}\, y_{ij} \quad \text{such that } (y_{ij}) \text{ is a positive semi-definite matrix. (PSD)}$$

Definition: A matrix $M \in \mathbb{R}^{n \times n}$ is PSD if $\exists\, v_1 \ldots v_n \in \mathbb{R}^d$ such that $m_{ij} = \langle v_i | v_j \rangle$.

Can