CS602 : Advanced ~~Applied~~ Algorithms

(Prof. Rohit Gurjar)

References : course webpage

1. Linear/Convex programming
(continuous optimization but useful in discrete)

2. Combinatorial Optimization

{LP duality}

Approximation algorithms, Online algorithms

NP-hard problems,
e.g max-sat (max # sat clauses)
(cut 3-sat is NP hard)
max-cut

partial input inc with time)
How good can we do
in an online setting vs
offline (approx algo)

Particular examples from
matching (online algorithms
matching keywords to advertisements)

Course structure

1. Introduction to LP/Convex programming
2. Application of LPs/Convex prog.
3. Algorithms for LPs/Convex prog.

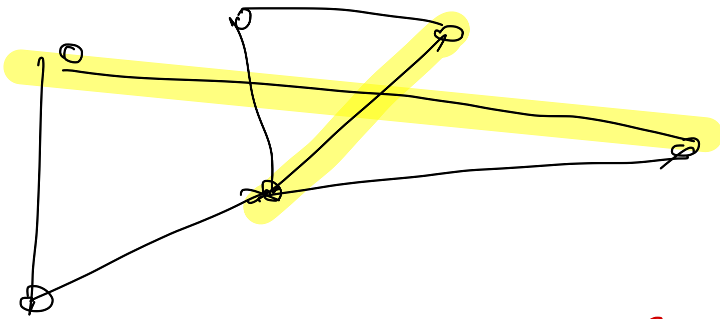
Prerequisites : CS218/CS601, Graph theory, Lin. Alg.

vectors, linear indep, rank, SOLE

Evaluation : 2 Assignments (5+5)
2 Exams (25+45)
Paper presentations (20) : Groups of 2 or 3

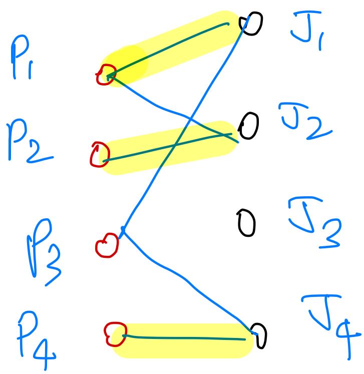
Some approximation algos are randomized algos

Matching



In a graph $G(V, E)$, $S \subseteq E$ is called a matching if they don't share endpoints

Bipartite matching

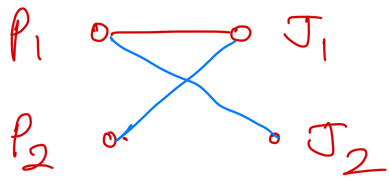


Basically,

1. Maintain a matching
2. Do swaps

(Polynomial time algorithm)

Augmenting path algorithm



P_2, J_2 not suitable, so $P_1 - J_2$, $P_2 - J_1$

This adjustment can be larger size.

A similar idea can work in a general matching, but the exact idea doesn't work (not as easy as it looks)

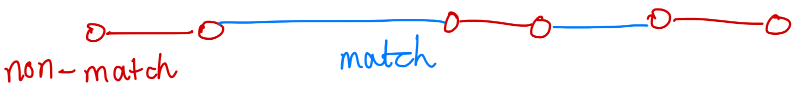
1965 : Edmonds' Algorithm (for matching) (G.O.A.T)

Running time is algebraic (polynomial) in number of vertices

Defs : Matched vertex : Vertex involved in one of the matching edges

Unmatched vertex
matching edge } self-explanatory

Augmenting path w.r.t a matching : A path which alternates between matching and non-matching edges and its endpoints are both unmatched



Let P be an augmenting path w.r.t a matching, then $M \Delta P$ (swap matching and non-matching edges to create a larger matching)

If max. matching then no augmenting path as size of matching cannot increase

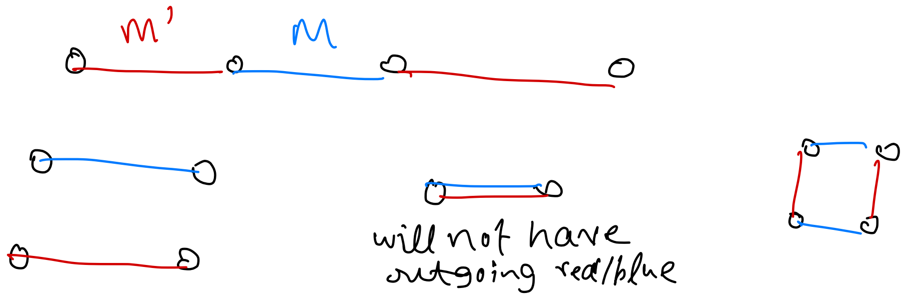
Claim: Let M be a matching which is not maximum, then there exists an augmenting path w.r.t M

Proof: Let M' be a maximum matching,

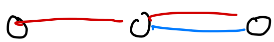
Consider $M' \Delta M$ (symmetric diff, throw out common edges)

$= M' \cup M - M' \cap M$

$M \cup M'$: every vertex has deg $0, 1, 2$.



Paths/cycles will always alternate between red & blue e.g.



is not possible

There must be a path where M' has more edges than M, hence, we get an augmenting path for M.

Algorithm 1. $M \leftarrow \emptyset$

while ( ) {

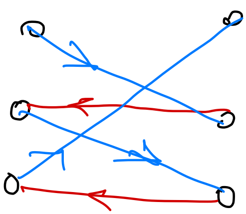
2. Find an augmenting path

- don't get → output M
- get → Swap matching edges on the path

}

Now, let's find a path (RIP)

Easy for bipartite case :

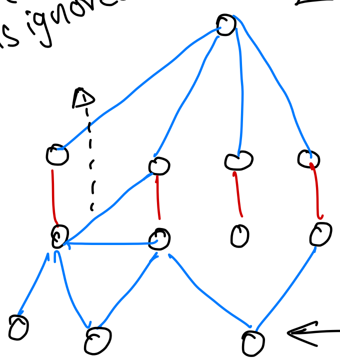


(assign directions)

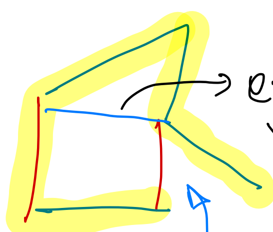
Now, this is a directed graph, find a path from unmatched to unmatched.

cross-edge is ignored

If path from s to u, u to v then path from u to v. But, in this situation,



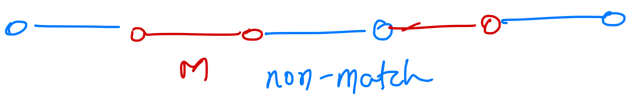
← free vertex, then we get augmenting path



even multiple visits won't help

issue!

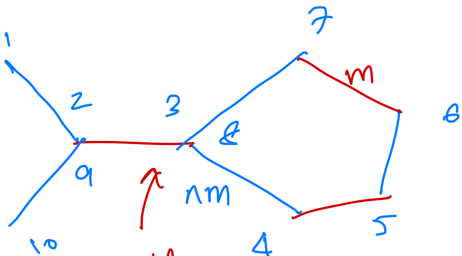
Fun fact : finding red-blue augmenting path in directed paths is NP-hard but we have a polynomial time algorithm undirected graphs



(augmenting path)

Finding an augmenting path

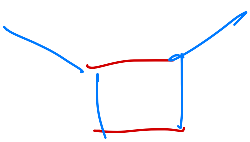
Note :



(not allowed, cuz walk not path, matching can't be extended

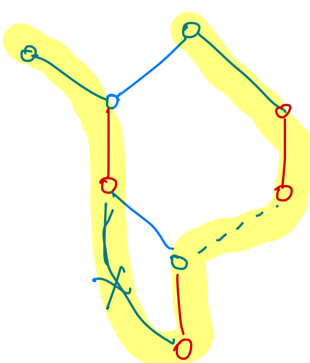
problem
if 2 visits are allowed

Larger matching cannot be found (cuz no augmenting path)



(even cycles are ok)

possible fix
ignore ancestor
(incorrect)



Keeping track of visited vertices

No such issues in SSSP cuz shortest path is shortest walk!

Unrelated problem : Directed graph where edges are colored blue/red. Source s, destination t.

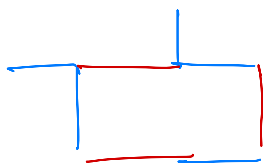
Q. find an alternating red-blue path from s to t

(red edges are arbitrary colored, not matching)

(harder problem) → NP-hard

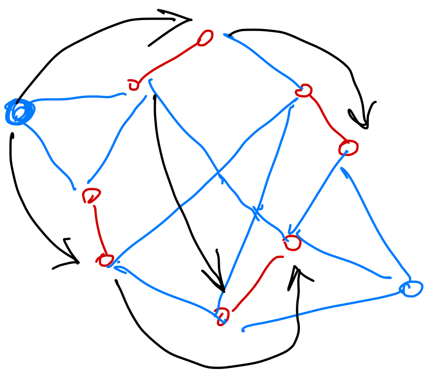
Edmond's Algorithm

1. Find shortest augmenting (shortest) walk



(we won't get things like this, as we'll get shorter augmenting walk)

• won't get even cycles



If red is adj to blue, we put a directed edge

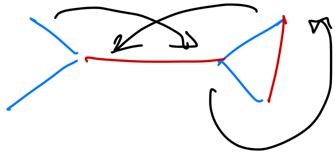
and so on,

In this graph, we want to reach neighborhood of another free vertex. (all pairs of free vertices)

Directed is imp cuz (BR ~~R~~B) is not allowed.

* what if other free vertex is same as the orignal one?

For the graph,

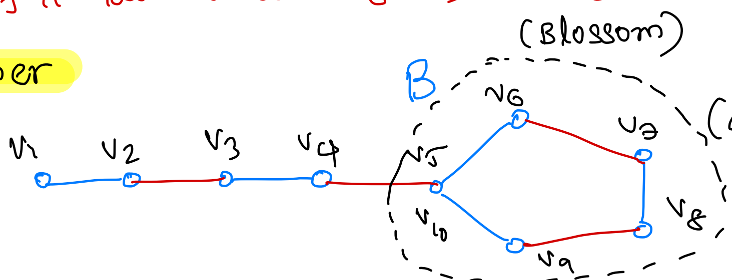


is a path for directed, but gives an augmenting walk (since DFS, we get shortest augmenting walk)

Now, what to do with an augmenting walk

1. If it's an augmenting path, we are done
2. If it has an odd cycle, we get a blossom.

Flower



(alternating path with alternating odd cycle)

$(v_1, v_2, \ldots, v_t)$ where $(v_1, v_2), (v_3, v_4), \ldots$ non-matching edges and $(v_2, v_3), (v_4, v_5), \ldots$ are matching edges and all vertices distinct except $v_t = v_i$, t is even, i is odd

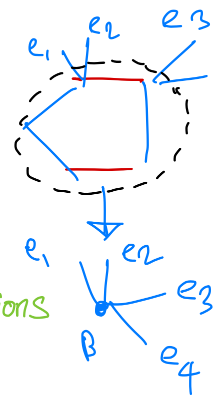
- first repeated vertex in an augmenting walk gives a blossom B

Now, contract B $\Rightarrow$ G/B (G contraction B)

Claim 1 : If G/B has an augmenting path $\Rightarrow$ augmenting path in G

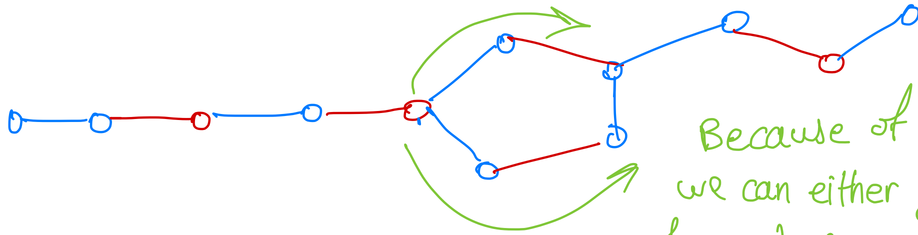
Claim 2 : If G has an augmenting path $\Rightarrow$ augmenting path in G

Imp : In reductions, it's always important to show claims in both directions



Proof of Claim 1

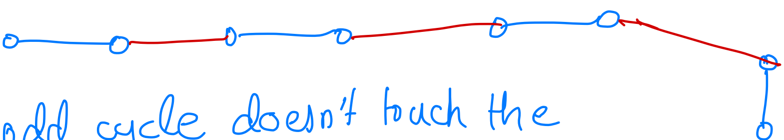
WLOG blossom vertex



Because of parity, we can either go from top or bottom (odd cycle)

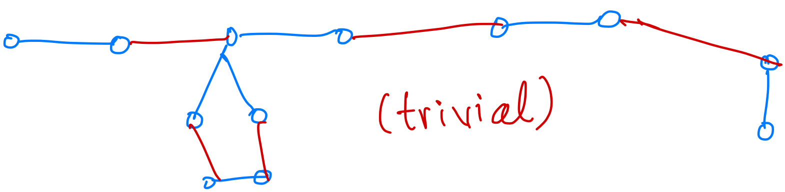
Proof of Claim 2 (not a part of algo, but a guarantee)

We have an augmenting path in G, to show G/B has an augmenting path.



1) if odd cycle doesn't touch the path, we're done.

2) if it touches,



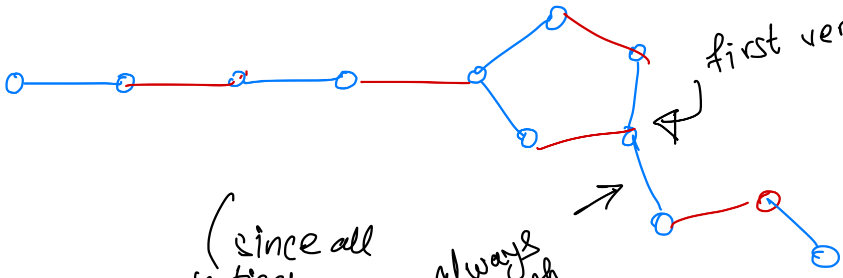
(trivial)

If it touches multiple edges, we need a general arguement (sections needn't be continuous)

First vertex where augmenting path touches the flower, (if it touches before the blossom, no problem)

combine with stem of flower

first vertex



(since all vertices are matched) in blossom

always enter with non-matching edge

Algorithm : Find a shortest augmenting walk

- Path → Done
- → Blossom B
  - Contract B → G / B
  - Find augmenting path and reconvert to G

Time :

Homework

1. Undirected graph edges red/blue source s destination t, finding a red-blue alternating path s-t (Reduce this to matching)

Up next : Linear Programming

Weighted Bipartite Matching : We want maximum weight matching

Opt quantity : unmatched − matched weights (minimize)

$15+5-10=10.$

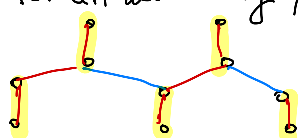
max weight matching of any size,

* Find max weight augmenting path (doesn't work for general graph, so LP makes things easier).

Ideas for Q1 : 1. For each vertex choose red edge & check if it results in augmenting i.e. blue edges between other endpoints (not efficient : n^n in worst case)

2. Find maximum matching in red subgraph and delete other red edges then, search (hope) for an alternating path

(Incorrect :



maximum matching fails to give s-t path)

3. Find alternating walk by directed graph construction combining red-blue edges. (Issue: No way to convert to augmenting path - Blossom worked because red edges were matching)

Idea : Say we want s–t path starting with red edge and ending with blue edge, delete other red edges on s, blue edges on t. (26/08/23)

Construction : For every vertex v different from s and t, create two vertices a red copy v_r, blue copy v_b, for s we only have s_r, for t, only v_b

If (u,v) in original graph, add edge (u_r,v_r) if edge was red, (u_b,v_b) otherwise.
Also add edge (u_r,u_b) for $\forall u \in V\setminus\{s,t\}$

Claim : Given graph has a desired s-t red-blue path if and only if the new graph has a perfect matching

Proof : $(\Leftarrow)$

Let the resultant graph have a perfect matching.
Delete all edges (u_r, v_b).

$s \mapsto u_r,\ u_b \mapsto v_b,\ v_r \mapsto \dots$ and so on till there is a free vertex.

If the final vertex $= t$, we're done.

If final vertex $= u_b \Rightarrow$ matching of $u_b = v_b$ was covered already

But v_b was covered $\Rightarrow u_b$ was covered or v_r was covered.

$v_r \mapsto v_b \mapsto u_b$. Hence, contradiction.

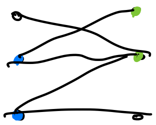
$(\Rightarrow)$

If we have an s-t red-blue path, let set of internal vertices in path $= V_1$. Then $\forall u \notin V_1 \cup \{s,t\}$, add (u_r,u_b) to matching edges.

$P=\{s, v_1, \dots, v_k, t\}$ then add $(s, v_{1_r}), (v_{1_b}, v_{2_r}), \dots, (v_{k_b}, t)$ to matching. Hence, we get a perfect matching ∎

Note : Bipartite rainbow s-t coloring reduces to s-t path as.

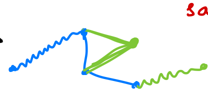
s



t

→ same color = adjacent to same vertex, same color.

$\xrightarrow{P}$



Linear Programming

(08/08/2023)

$x_1, x_2, \ldots, x_n \in \mathbb{R}$

e.g. $2x_1 + 3x_2 - 5x_3 \geq 10$
$x_3 + 2x_7 = 12$

} equations, inequalities are both fine

$\max \; 5x_2 - 9x_3 + 10x_4$

Mid-day meal:

	Calories	Protein	Carbohydrate
Rice	250		
Dal	150		
Eggs	80		

x_R, x_D, x_E (amount of rice, dal, eggs)

$250x_R + 150x_D + 80x_E$ has upper, lower limit.

$\|^{el}$ inequalities to other quantities.

Then, we want to minimize the cost.

$$\min \; c_R x_R + c_D x_D + c_E x_E$$

Roommate allocation

students, $\{1, 2, \ldots, n\}$. x_{ij} ← pair up i and j

$$x_{ij} = \begin{cases} 1 & (i,j) \in M \\ 0 & \text{otherwise} \end{cases}$$ → Integer linear Program (ILP)

i.e. $x_{ij} = \{0, 1\}$ (not a linear constraint)

$\sum_j x_{ij} \leq 1$ (1 student with atmost 1 other student) (for all i)

$x_{ij} = 0$ (if $(i,j) \notin E$)

$\max \sum x_{ij}$ (maximise pairings)

To do: Check if a LP exists for stable matching

<u>Integer Linear Program</u> : A LP where some subset of variables restricted to integers.

3-SAT reduces to ILP (but not LP), hence ILP is NP-hard.

Now, let's not restrict $x_{ij} \in \{0,1\}$, turns into a LP.

i.e. $0 \leq x_{ij} \leq 1$ (LP-relaxation)

$$\sum_{\substack{1 \leq j \leq n \\ (j \neq i)}} x_{ij} \leq 1 \quad \forall i \in \{1, 2, \dots, n\}$$

$$x_{ij} = 0 \text{ if } (i,j) \notin E$$

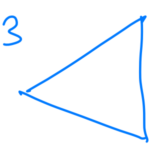
$$\max \sum_{(i,j)} x_{ij}$$

<u>LP relaxation</u> : Feasible solution set increases

Optimal value for the LP-relaxation $\geq$ OPT(ILP) = OPT(matching)

<u>Homework</u> : Matchings and feasible solutions of ILP have one-one correspondance.

e.g. K_3



OPT_{LP} : $x_{12} = x_{23} = x_{13} = 1/2$

$\sum x_{ij} = 3/2$, but $OPT_{ILP} = \underline{1}$

Maybe more linear constraints can make it work?
(depends, very problem specific)

Homework :

Claim : For bipartite graphs
LP OPT = optimal matching size
(some justification for why bipartite matching is easier)

Methods to deal with LP :

1. Add more constraints to get exact solution
(problem isn't NP-hard)

2. Approximate algorithms

Note : Exponentially many constraints in LP is req for matching in general graph

Problem : A set of numbers (+/-). Select an odd size subset s.t. sum is maximised

Ans X_S for odd length $S \subseteq [n]$

$\forall_S \quad X_S = \sum_{i \in S} a_i x_i$

$\forall_S \quad z_S = \sum_{i \in S} a_i$

$z \geq z_S \quad \forall S$

min z

(or)

$\sum X_S = 1$

max $\sum X_S z_S$

But, we have exponentially many vars (bad)

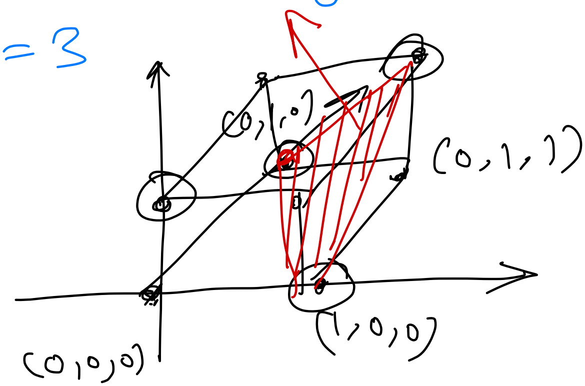
ILP $\quad x_1, x_2, \dots, x_n \in \{0,1\}$

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

Now, encoding "odd" is trouble some.

(for 2 elements, is easy : $x_1 + x_2 = 1$)

for $n=3$



$x + y - z \leq 1$, similarly eliminate 4 other corners

$y + z - x \leq 1$

$x + z - y \leq 1$ $\qquad$ (2^{n-1} constraints)

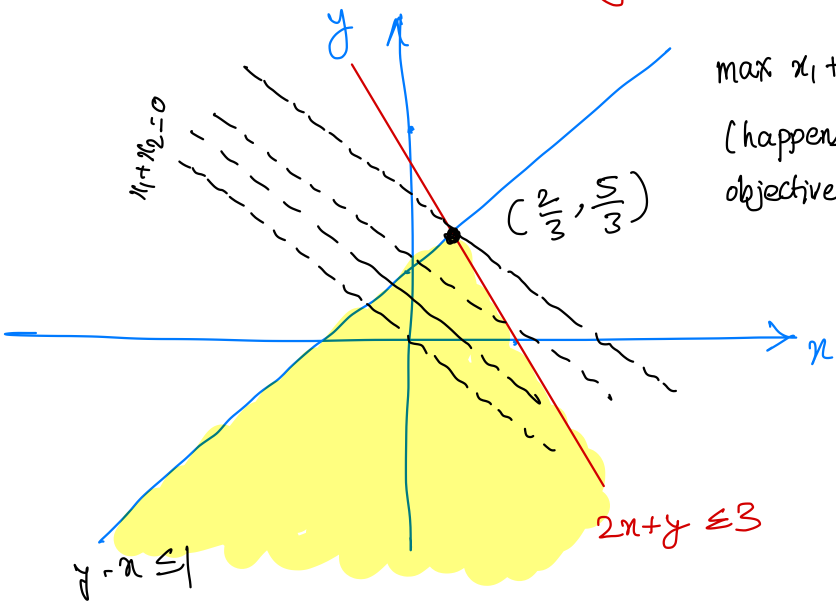
$x + y + z \geq 1$

exponentially many constraints, but description is simple

$$\sum_{i \in T} x_i - \sum_{i \notin T} x_i \leq |T| - 1 \quad \text{(all can't be 1)}$$

Number of constraints is always exp (easy to show, PT 2 pts can't be thrown out by same constraint)

Claim : The optimal solution for an LP are always at the boundary



max x_1+x_2

(happens because objective f^n is linear)

e.g. max $2x+y$ has multiple (∞ many) optimal sol^n.

A more formal proof of this fact follows later in course.

The feasible sol^n forms a polytope. (optimal sol^n as well, always a polytope)

Notation

$$x_1 + 3x_2 \le 4$$
$$2x_1 + x_2 \le 3$$
$$x_1 - x_2 \le 1$$
$$\max \; x_1 + x_2$$

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 1 & -1 \end{bmatrix}_{3\times 2}$$

row = constraint
column = variable

k constraints, n variables then
$A \in \mathbb{R}^{k\times n}$

$$b = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{n\times 1}$$

Shorthand notation : $Ax \leq b$.

for, $\sum_{j=1}^{n} A_{ij} x_j \leq b_i \;\; \forall i \in [k]$

Optimal condⁿ : $w \in \mathbb{R}^n$ (column vector)

$\max \; w^T x$

Standard forms of linear programs

1. $\max \; w^T x$

$Ax \leq b$

(can arbitrary LP can be converted into this form)

e.g. $2x_1 + x_2 \geq 3$

$-2x_1 - x_2 \leq -3$

$x_1 + x_2 + x_3 = 5$ (introduce $\leq$ & $\geq$)

$$\begin{cases} x_1 + x_2 + x_3 \leq 5 \\ -(x_1 + x_2 + x_3) \leq -5 \end{cases}$$

2. $\max \; w^T x$

$Ax = b$

$x \geq 0$

To do: Convert 1 to 2 and 2 to 1

Trick: for ILP if discrete domain is wanted, e.g.

$x \in \{n_1, n_2, \ldots, n_k\}$

then create vars $x_1, \ldots, x_k$

$$\left.\begin{aligned} & x_1 + x_2 + \cdots + x_k = 1 \\ & 0 \leq x_1, \ldots, x_k \leq 1 \\ & x = n_1 x_1 + n_2 x_2 + \cdots + n_k x_k \end{aligned}\right\}$$

an elaborate indicator function!

(11/08/2023)

Standard forms of LP

1. $\max\ w^T x$
   where $x \in \mathbb{R}^n$
   $Ax \leq b$

   $\left[\begin{array}{l} b \in \mathbb{R}^k, \\ A \in \mathbb{R}^{k \times n} \end{array} \right.$

2. $\max\ w^T x$
   where $x \in \mathbb{R}^n$
   $x \geq 0$
   $Ax = b$

   } feasible region (points which are allowed)

3. $\min\ w^T x$
   where $x \in \mathbb{R}^n$
   $Ax \geq b$

4. $\min\ w^T x$
   where $x \in \mathbb{R}^n$
   $Ax = b$
   $x \geq 0$

Claim : Any LP can be converted into any of these forms

e.g. $2x_1 + 3x_2 \leq 6 \longrightarrow 6 - (2x_1 + 3x_2) = y,\ y \geq 0$

(need to add positivity constraint for other vars)

$x_1 = x_{1,p} - x_{1,n}$

$x_{1,p} \geq 0$

$x_{1,n} \geq 0$

$6 - 2(x_{1,p} - x_{1,n}) - 3(\quad) = y$

$y \geq 0$

$x_{1,p} \geq 0$

$x_{1,n} \geq 0$

...

Trick : $x_1 \leq 0$. replace x_1 with $-y_1$, $y_1 \geq 0$

Geometric Picture

Polyhedron ← Feasible region

Polytope = bounded polyhedron

in 2D polygons
↓
nD polytopes $\xrightarrow{\text{unbounded}}$ polyhedron

Bounded : $\exists R \in \mathbb{R}$ s.t. $P \subseteq S(0, R)$

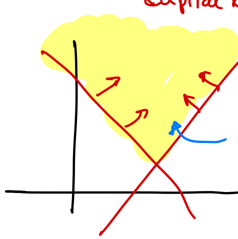
Halfspace $\{x \in \mathbb{R}^n : a^T x \leq b\}$, $b \in \mathbb{R}$

(one side of hyperplane)

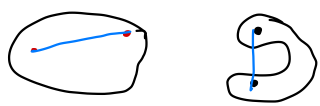
Notation : small letters = vector
capital letters = matrix

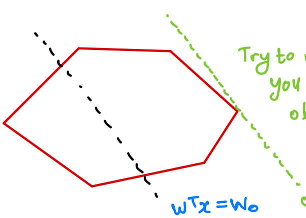
Polyhedron : Intersection of halfspaces
(and hyperplanes)

implied as hyperplane = intx of $a^T x \leq b$, $-a^T x \leq -b$



Note : A polyhedron is a convex set
(easy pf using lin. combinations)

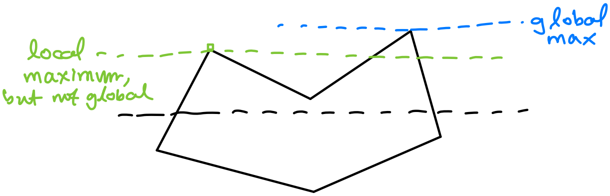




Try to move locally, if you cannot increase objective then it is the optimum

Note : In any convex set, any local optimal point is also an optimal point (for linear objective functions)

Proof : # To do (homework)

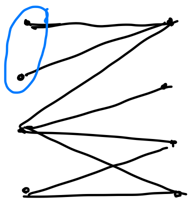


Local optimum (maximum)

x is a local optimum point if $\exists \varepsilon > 0$ s.t. for all y s.t. $\|y - x\| < \varepsilon$, y is feasible, then $w^T y \le w^T x$

Max flow / bipartite matching : (Hall's theorem)

If we can't get a perfect matching,



Easy proof for no perfect matching

Give an S s.t. $|N(S)| < |S|$ (easy certificate for optimality of solution)

e.g. max-flow

Proof of being optimal? Give a min-cut.

Now, consider a NP-hard problem, if someone gives a maximum independent set, checking its optimality is hard!

- easy proof of optimality correlates with an easy problem

Set of optimal points (a sub polytope at the boundary)

Face : For a polyhedron given by $Ax \le b$, a face is set of points in P (polyhedron) which satisfy some subset of constraints with equality (non-empty)

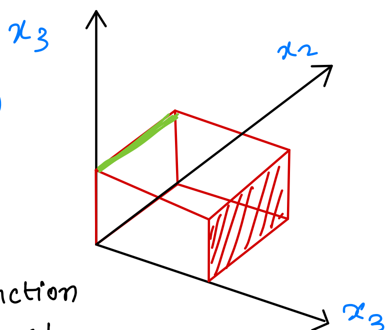
Face 1

$x_1 = 1$

(orignal inequalities remain)

Face 2

$x_1 = 0,\ x_2 = 1$



$0 \le x_1 \le 1$

$0 \le x_2 \le 1$

$0 \le x_3 \le 1$

Claim : for any linear function $w^T x$ and any polyhedron $Ax \le b$, set of points in P maximising $w^T x$ forms a face

Proof : Let S be the set of points maximizing $w^T x$. (clearly convex set, x_1, x_2 opt $\Rightarrow \lambda x_1 + (1-\lambda) x_2$ also opt.).

$a_1^T x \le b_1, \ldots, a_k^T x \le b_k$ (orignal polyhedron P)

Tight constraints (ineq. satisfied with equality)

Suppose all points in S satisfy (first l constraints) (without loss of generality)

$a_1^T x = b_1$

$\ldots$

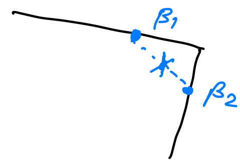
$a_l^T x = b_l$

l can be $0, 1, 2, \ldots, k$

for any $j = \ell+1, \dots, k$ $\quad \exists \beta_j \in S$ s.t. $a_j^T \beta_j < b_j$ $\quad$ (only the first ℓ constraints are tight)

$\beta \in S$ since convex S $\quad$ would be nice to get one point which gives strict ineq for all

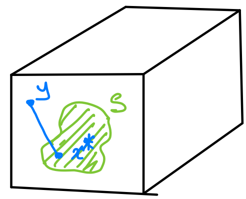
Define $\beta = \frac{1}{k-\ell}(\beta_{\ell+1} + \dots + \beta_k)$ $\quad \{\ell < k\}$



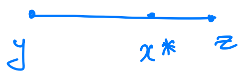
Claim : $\forall j = \ell+1, \dots, k;\ a_j^T \beta < b_j$

(Trivial, just plug this into given inequalities)

Claim : The face of P defined by $a_1^T x = b_1, \dots, a_\ell^T x = b_\ell$ is maximising $w^T x$



Proof : For contradiction assume $\exists y$ s.t. $a_1^T y = b_1, \dots, a_\ell^T y = b_\ell$ but $w^T y < w^T x^*$, where $x^* \in S$



$z = x^* + \epsilon(x^* - y)$ $\{$for $\epsilon > 0\}$

$w^T z = w^T x^* + \epsilon\,(\underbrace{w^T x^* - w^T y}_{>0})$ $\quad$ ($x^* = \beta$ everywhere)

$\Rightarrow w^T z > w^T x^*$

But, we need to show z remains inside polytope

final claim : $z \in P$

1) $j = 1, 2, \dots, \ell$ then, $\quad a_j^T z = a_j^T x^* + \epsilon(a_j^T x^* - a_j^T y)$

since $x^* \in S$, $y \in$ face both satisfy first ℓ with equalities.

Hence, $\quad a_j^T z = a_j^T x^* = b_j$

2) $j = \ell+1, \dots, k$

$a_j^T z = \underbrace{a_j^T x^*}_{< b_j} + \epsilon\,\underbrace{(a_j^T x^* - a_j^T y)}_{\text{we choose } \epsilon \text{ to be small enough s.t. this is still } < b_j}$

$< b_j$ (for appropriate ϵ).

Hence, $z \in P$

Claim : For every face F, there exists a $w^T x$ s.t. F is the face maximizing $w^T x$

[HW]

Linear Programming

(18/08/2023)

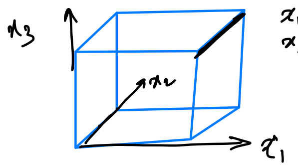
Claim 1: The set of points maximizing a linear function over a polyhedron forms a face

Claim 2: For a polyhedron $Ax \le b$, and any of its face F, there exists w s.t. $w^T x$ is maximised at F.

e.g. $0 \le x_1 \le 1$

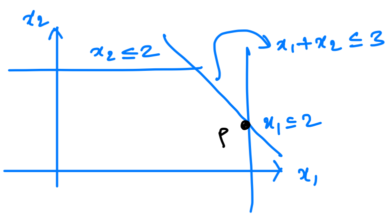
$0 \le x_2 \le 1$

$0 \le x_3 \le 1$



$x_1 = 1$, $x_3 = 1$

Here $f = x_1 + x_3$ works, as a maximising function.



At P, set of f^ns maximised are positive l.c. of x_1, $x_1 + x_2$. (i.e. x_1 dominates)

Claim 2 (contd): If F is defined by

$a_1^T x = b_1$

$\cdots$

$a_\ell^T x = b_\ell$

Then, $w = a_1 + a_2 + \cdots + a_\ell$

Proof: $f(x) = w^T x = (a_1^T + a_2^T + \cdots + a_\ell^T)x$

for any point $\alpha \in F$,

$f(\alpha) = b_1 + b_2 + \cdots + b_\ell$

for any point $\beta \notin F$, in the polyhedron,

(WLOG) $a_1^T \beta < b_1$, $a_i^T \beta \le b_i$ for $i = 2$ to ℓ.

Then $f(\beta) = (a_1^T + \cdots + a_\ell^T) < b_1 + \cdots + b_\ell = f(\alpha)$

Hence, $f(\beta) < f(\alpha)$

Fact: If there is an optimizing point, then there is a corner which is maximizing (if a corner exists)

Corner: 0-dimensional face

A face with n linearly independent tight constraints

equalities
||

e.g.

LD tight constraints

Claim : p is a corner of polyhedron iff only if, we cannot write $p = \mathrm{conv}(\alpha, \beta)$ for two distinct points α, β in the polyhedron

Proof: p is a corner, then there's a linear f^n $w^T x$ which is optimized precisely at p

For the sake of contradiction, let's take

$$p = \lambda \alpha + (1-\lambda)\beta$$

Then $f(\alpha), f(\beta) < f(p)$, since p is the unique maximising point

$$w^T \alpha < w^T p$$
$$w^T \beta < w^T p$$
$$\Rightarrow w^T(\lambda \alpha + (1-\lambda)\beta) < w^T p.$$ Hence, contradiction

p is not a corner

Consider all the tight constraints for p. Their rank $< n$

$$\text{rank} < n \quad \left\{ \begin{array}{l} a_1^T p = b_1 \\ \cdots \\ a_\ell^T p = b_\ell \end{array} \right.$$
$$a_{\ell+1}^T p < b_{\ell+1}$$
$$\vdots$$
$$a_k^T p < b_k$$

$\exists\, \alpha \neq \bar{0}$ s.t. $a_1^T \alpha = 0,\ a_2^T \alpha = 0, \ldots, a_\ell^T \alpha = 0$ (exists since rank $\begin{bmatrix} a_1^T \\ \vdots \\ a_\ell^T \end{bmatrix} < n$)

Now, let $q_1 = p + \varepsilon \alpha$, $q_2 = p - \varepsilon \alpha$ (where ε is small enough)

clearly $p = (q_1 + q_2)/2$.

Also, q_1, q_2 lie in the polyhedron as all constraints are satisfied.

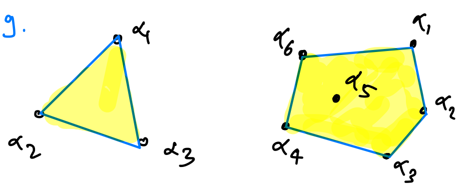
Observation : For any corner there is a linear function s.t. corner is unique maximising point.

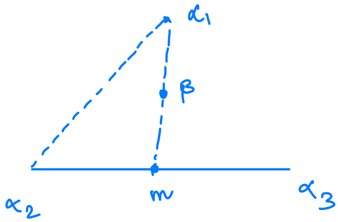
Convex hull (Definition)

For a set of points $\alpha_1, \alpha_2, \ldots, \alpha_r \in \mathbb{R}^n$,

$$\mathrm{Conv\ Hull}(\alpha_1, \alpha_2, \ldots, \alpha_r) = \left\{ \lambda_1 \alpha_1 + \cdots + \lambda_r \alpha_r \;\middle|\; \sum_{i=1}^{r} \lambda_i = 1,\ \lambda_i \geq 0\ \forall i \right\}$$

e.g.





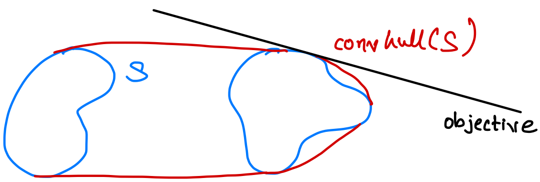
$m \in \text{conv}(\alpha_1, \alpha_2) \wedge \beta \in \text{conv}(\alpha_1, m) \Rightarrow \beta \in \text{conv}(\alpha_1, \alpha_2, \alpha_3)$ (trivial)

e.g. max lin f^n over spanning trees
then take LP over conv hull of spanning trees.

Note : while encoding max weight stuff, we encode a convex hull!

max $w^T x$ over set S = max $w^T x$ over conv hull(S)

Observation : $w^T a_1 = w^T a_2$
$= w^T(\lambda a_1 + (1-\lambda) a_2)$



Proof : Let $w^T(\Sigma \lambda_i \alpha_i)$ be maximum

But

$w^T(\Sigma \lambda_i \alpha_i) \leq \max(w^T \alpha_i)$ since $\Sigma \lambda_i = 1$, $\lambda_i \geq 0$ ■

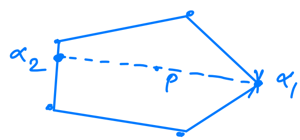
Observation : A polytope is the convex hull of its corners

Carothèodory's theorem : In an n-dimensional polytope, every point is a convex combination of n+1 corners.

Proof : (Induction)

$p \in \text{conv}(\alpha_1, \alpha_2)$, α_1 is a corner,

α_2 lies in a $(n-1)$ dimensional face



Hence, α_2 lies in $(n-1)$ dimensional polytope.

Using induction hypothesis, $\alpha_2 \in \text{conv}(n$ corners)

$\Rightarrow p \in \text{conv}(n+1$ corners) ■

Lecture (This friday, online class) 22/08/23

Discrete Optimization ⟶ Convex optimization

e.g. minimum spanning tree

for any $S \subseteq E$, $x_S \in \{0,1\}^E$. $(x_S)_e = \begin{cases} 0 & e \notin S \\ 1 & e \in S \end{cases}$.

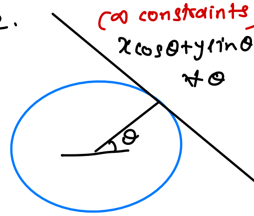
$(w_e)_{e \in E}$.

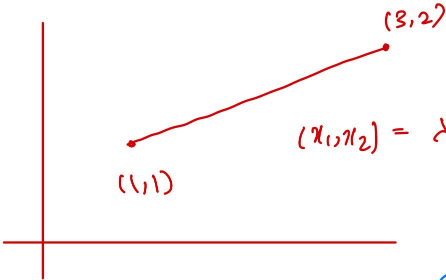
$\min\{\sum_{e \in S} w_e : S \text{ is a spanning tree}\} = \min\{\underbrace{w^T \cdot x_S}_{\sum w_e x_e} : S \text{ is a spanning tree}\}$

$= \min \{ w^T x : x \in \underbrace{\text{Conv hull } \{ \chi_S : S \text{ is spanning tree} \}}_{\hookrightarrow \text{description ?}} \}$

Claim: The convex hull of finite set of points is a polytope.
(cnt circle)

(∞ constraints)
$x\cos\theta + y\sin\theta \le r$
$\forall \theta$





$(x_1, x_2) = \lambda(1,1) + (1-\lambda)(3,2)$

$$\boxed{\begin{aligned} x_1 &= \lambda 1 + (1-\lambda)3 \qquad 0 \le \lambda \le 1 \\ x_2 &= \lambda(1) + (1-\lambda)(2) \end{aligned}}$$

Elimination of vars of equations using gaussian elim.

Fourier Motzkin Elimination

$\lambda = \frac{3-x_1}{2}$
$\lambda = 2 - x_2$
$\lambda \ge 0$
$\lambda \le 1$

$\frac{3-x_1}{2} = 2 - x_2$
$2 - x_2 \ge 0$
$2 - x_2 \le 1$
$\cancel{\frac{3-x_1}{2} \ge 0}$ (redundant)

$$\boxed{\begin{aligned} 2x_2 - x_1 &= 1 \\ x_2 &\le 2 \\ x_2 &\ge 1 \end{aligned}}$$

$x_1, x_2, \ldots, x_n, \lambda$
$\lambda = E_1$ (linear expr.)
$\lambda = E_2$
$\lambda \ge E_3$
$\lambda \le E_4$
$\lambda \le E_5$
$n+1$ vars
System 1

$\xrightarrow[\text{(projection)}]{\text{eliminating } \lambda.}$

$E_1 = E_2$
$E_2 \ge E_3$
$E_2 \le E_4$
$E_2 \le E_5$
n vars
System 2

Claim:
$\exists \alpha$, $(p_1, p_2, \ldots, p_n, \alpha)$ satisfies system 1 iff the point $(p_1, p_2, \ldots, p_n)$ satisfies system 2.

Proof: trivial (believe)

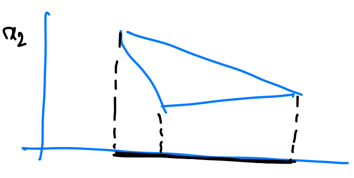
$\lambda \le E_1$
$\lambda \ge E_3$
$\ldots$

$\longrightarrow$ $E_3 \le E_1$ (if only inequalities)
$\ldots$

General algorithm: In each iteration, one variable is eliminated

Note: Eliminate r variables $\longrightarrow n^r$ (n = #constraints)

$\left\{ {}^nC_2 \text{ eq}^n \text{ new, so } n \text{ blowup each iteration} \right\}$



$\downarrow$ eliminate λ_2

For spanning trees, # vars $= O(2^n)$.
So, superexp. blowup.
(i.e. LP description as constraints can be huge)

[HW] Use FM elimination to solve linear programs. (Not efficient)

Algorithms for LP

1. Go over all corners (choose n inequalities & make them equalities, check for LI, sol^n lying inside polytope. $O({}^kC_n)$

2. FM elimination

Some facts so far:

① The convex hull of finite set of points is a polytope ($Ax \le b$)

② Any polytope is the convex hull of its corners (finite)

③ Set of optimizing points in a polytope form a face

④ There is always a corner optimizing a given linear function (polytope)

Linear Program Duality

① Feasibility : Given $Ax \le b$, is there a point satisfying the system

② Optimization : $\max\ w^T x$
s.t. $Ax \le b$

Claim : Optimization $\le$ feasibility
(reduces to)

Guess θ, ask whether $Ax \le b$, $w^T x \ge \theta$ is feasible.

yes $\rightarrow$ OPT $\ge \theta$

No $\rightarrow$ OPT $< \theta$

Binary Search : How large can θ be?

parameters : n = no. of variables
k = no. of constraints
ℓ = no. of bits in coefficients

Claim : $\theta^* \le \exp(n, k, \ell)$ [HW]

Pf : $\exists$ corner, coordinates of that corner $\le$?
bound determinants in inversion, etc.

(OPT $> \theta^*$ then, OPT $= \infty$).

Bit precision : ℓ bits of precision, then ℓ add^n rounds of binary search,
Running time $\propto$ desired bit precision.

Feasibility $\le$ Optimization

feasible $\to$ OPT

not $\to$ junk value

Puzzle : Suppose the optimization algorithm work like :

If $Ax \le b$ feasible $\to$ gives optimal value

If not feasible $\to$ junk value.

Solve feasability ?

idea :

$$a_1 x \le b_1$$
$$\dots$$
$$a_{\ell-1} x \le b_{\ell-1}$$
$$(a_\ell x \le b_\ell) \, ?$$

$\to$ min $a_\ell x$ if $\le b_\ell$ then done.

Doubt : but what if this gives junk value ?

FM elimination to solve LP.

$$a_1^T x \le b_1 \qquad x_n \le (\quad) \xrightarrow{FM} n-1 \text{ vars} \to \text{sol}^n$$
$$\dots \qquad \dots$$
$$a_\ell^T x \le b_\ell \qquad x_n \le (\quad)$$

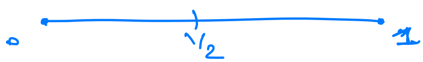
Repeat this process n times.

sol^n of LP $\begin{cases} a \le x_1 \le b \\ x_2 \le f(x_1) \\ \dots \end{cases}$

[HW] Puzzle : Suppose $\theta^* = \frac{p}{q}$ is an unknown rational number. $O(\log p + \log q)$

Queries : whether $\theta^* \ge \theta$. Goal is to find θ^*. find p and q.

Problem can be reduced to [0, 1] interval. WLOG $p_i < q_i$

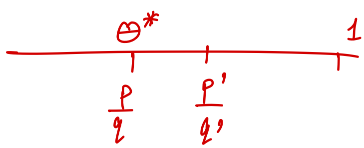


$$\frac{1}{2}\left(\frac{p_1}{2^i} + \frac{p_2}{2^j}\right) = \boxed{\frac{p_1 2^{j-i} + p_2}{2^{j+1}}}$$

say $j > i$ $\qquad$ p_1, p_2 are odd. Hence, gcd = 1.

Hence, $2^{j+1} = q > q_1, q_2$

$p_1 2^{j-i} + p_2 = p > p_1, p_2$.



q is valid if

$$\underbrace{\frac{1}{q} \le \theta^* \le 1 - \frac{1}{q}}$$

range increases with p, q

p is valid if

$$\underbrace{0 \le \theta^* \le} \frac{p}{p+1}$$

for each q, doing search for $p = q \log p$

for each p, doing search over $q = p \log q$

Hint : Continued fractions

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots \frac{1}{a_n}}} = [a_0, a_1, \ldots, a_n]$$

e.g. $5/3 = 1 + \cfrac{1}{1 + \frac{1}{2}} = [1, 1, 2]$

Thm : euclid's algo $O(\min(\log a, \log b))$

closely related to euclid's gcd algorithm, so same running time.

$0 < p/q < 1$

$(p, q) \rightarrow (p, q \% p) \rightarrow$

$p < q$

e.g. $(3, 5) \xrightarrow[\text{modulus}]{\text{quotient,}} (3, 2)$

$(2, 3) \rightarrow \cdots$

in $\underbrace{\text{modified}}$ setting,

finding this we need to find max a s.t. $1/a > 3/5$

$\underbrace{\log q}$ time

so, algo = $\boxed{\Theta(\log a + \log b)}$

$\downarrow$

$O(\log p \log q)$.

#ques : Ask how to improve to $\log p + \log q$.

claim : $\Theta^* \leq \exp(n, k, \ell)$ [HW]

n = no. of variables

k = no. of constraints

ℓ = no. of bits in coefficients

Let $A^T x \leq b$,

$w^T x = \theta^*$.

for a corner i.e. $\exists$ constraints

$a_1^T x = b_1$

$\cdots$

$a_n^T x = b_n$

$a_1, \ldots, a_n$ are L.I

$$\underbrace{\begin{bmatrix} a_1^T \\ \cdots \\ a_n^T \end{bmatrix}}_{A'} x = b \quad \Rightarrow x = (A')^{-1} b.$$

$$\Rightarrow \theta^* = \underbrace{w^T}_{1\times n} \underbrace{(A')^{-1}}_{n\times n} \underbrace{b}_{n\times 1}.$$

$$= w^T C b$$

$(A')^{-1} = C$

$$\Theta^* = \sum_{i,j} \underbrace{w_i c_{ij} b_j}_{n^2 \text{ terms}} = \frac{1}{\det A} \sum_{i,j} w_i \operatorname{cof}(a_{ij}) b_j \qquad c_{ij} = \frac{1}{\det} \operatorname{cof}(A_{ij})$$

cof = det of $(n-1)\times(n-1)$ matrix.

sum over all permutations = $(n-1)!$ terms,

each term = prod of $n-1$ terms

$\leq (2^\ell)^{n-1}$

$\operatorname{cof} \leq (n-1)!\,(2^\ell)^{n-1}$

Let $\overline{\det A} \leq 1$

$$\theta^* \leq \underbrace{\sum_{ij}}_{n^2 \text{ terms}} w_i (2^\ell)^{n-1} (n-1)!\, b_j \leq n^2 (2^\ell)^{n-1} (n-1)!\,(2^\ell)^2$$

$= \exp(n, \ell)$ if $k \geq n$. else $\exp(n, k, \ell)$.

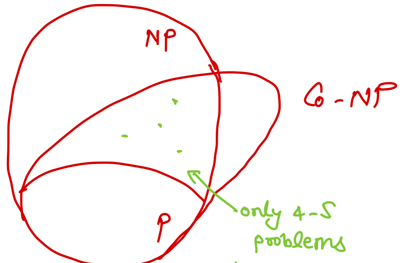
Lecture 25/08/2023

Feasibility $\equiv$ Optimization.

Complexity

Feasibility (yes/no question) is in NP because "yes" instance has polynomial-time certificate.

Proof = feasible point, verification = checking if all constraints are satisfied.



only 4-5 problems here.

(if any NP-complete problem here then NP = Co-NP which is not believed.)

Optimization: is there an x s.t. $Ax \le b$ such that $w^T x \ge w_0$

Also in NP, proof = point x

Co-NP: "no" instance has easily verifiable proof

Two boolean formulas $\phi = \psi$?

If this is false, an input which $\phi(x) \ne \psi(x)$ is an easily verifiable proof.

Feasibility $\in$ Co-NP? (if problem in NP, Co-NP $\Leftarrow$ problem in P) (necessary condn)

If $Ax \le b$ is not feasible then is there an easily verifiable proof for this.

<u>Note</u>: Problem $L \in NP \cap co\text{-}NP \overset{?}{\Rightarrow} L \in P$ (nothing known)

Given $Ax = b$ which is not feasible, is there an easy verifiable proof this? $\exists y \in \mathbb{R}^k$ $y^T A = 0$, $y^T b \ne 0$. Hence, this y is the certificate.

$k = \#$constraints

$A = k \times n$

Q. $\left.\begin{array}{l} Ax = b \\ x \ge 0 \end{array}\right\}$ one of standard forms of LP. Given this is not feasible, proof?

e.g. $x_1 + x_2 = 4 \quad \times(-1)$ $\rightarrow$ $-2x_2 = 4$

$x_1 - x_2 = 8 \quad \times(1)$ $\quad \le 0 \quad \ge 0$.

$x_1 \ge 0, x_2 \ge 0$.

<u>Claim</u>: If $\begin{array}{l} Ax = b \\ x \ge 0 \end{array}$ is not feasible, $\exists y \in \mathbb{R}^k$ s.t. $y^T A \ge 0$, $y^T b < 0$

(all elements +ve)

<u>Note</u>: This is an easily verifiable proof of non-feasability. since $y^T A x \ge 0$, $y^T b < 0$

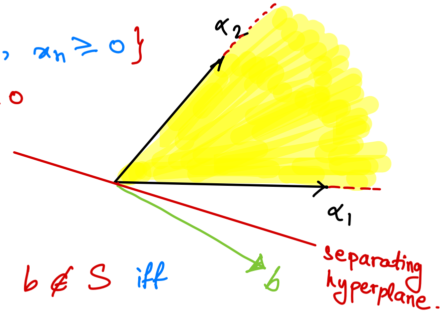
Farkas' lemma

Geometric interpretation : Let the columns of A are $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}^k$

if $\{ \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n = b, \; x_1 \geq 0, \dots, x_n \geq 0 \}$

not feasible then (iff) $\exists \, y \in \mathbb{R}^k$. $y^T \alpha_i \geq 0 \; \forall i, \; y^T b < 0$

separating hyperplane



Separating hyperplane theorem

Let S be a convex set in $\mathbb{R}^k$ and let $b \in \mathbb{R}^k$ s.t. $b \notin S$ iff

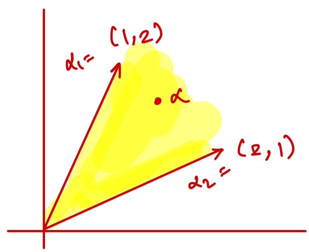
$\exists \, y \in \mathbb{R}^k$ s.t. $y^T b < 0$, $\forall x \in S, \; y^T x \geq 0$

In farkas lemma, the cone $\{ \sum \alpha_i x_i \mid x_i \geq 0 \}$ was the convex set since lin. comb. of any two points lies inside.

Proof of farkas lemma

$S = \{ \sum \alpha_i x_i \mid \forall_i \; x_i \geq 0, \; \alpha_i \in \mathbb{R}^k \}$ is a convex cone.

$\exists \, c_1, c_2, \dots, c_\ell \in \mathbb{R}^k$ s.t. $\alpha \in S$ iff $c_1^T \alpha \geq 0, \dots, c_\ell^T \alpha \geq 0$.



$\alpha_1 = x_1 + 2x_2$
$\alpha_2 = 2x_1 + x_2$
$x_1 \geq 0$
$x_2 \geq 0$

$\xrightarrow{\text{FM}}$ elimination. (elim x_1)

$x_1 = \alpha_1 - 2x_2$
$x_1 = \frac{1}{2}(\alpha_2 - x_2)$
$x_1 \geq 0$
$x_2 \geq 0$

$(\alpha_1 - 2x_2) = \frac{1}{2}(\alpha_2 - x_2)$
$\frac{1}{2}(\alpha_2 - x_2) \geq 0$
$x_2 \geq 0$
i.e.
$x_2 \leq \alpha_2$
$x_2 \geq 0$
$x_2 = \frac{2\alpha_1 - \alpha_2}{3}$

$\xrightarrow{\text{elim } x_2}$

$\frac{2\alpha_1 - \alpha_2}{3} \geq 0$
$\frac{2\alpha_1 - \alpha_2}{3} \leq \alpha_2$

$\boxed{\begin{aligned} \alpha_1 &\leq 2\alpha_2. \\ \alpha_2 &\leq 2\alpha_1 \end{aligned}}$

not same as α in lemma.

Hence, in general, FM elimination gives $c_1, \dots, c_\ell$ s.t. $c_1^T \alpha \geq 0, \dots$ $c_\ell^T \alpha \geq 0$.

if $b \notin S$ then for some i, $c_i^T b < 0$, but $c_i^T \alpha_j \geq 0 \; \forall \alpha_j$ gives a separating b. ∎

$Ax = b, \; x \geq 0$ System 1

$A^T y \geq 0, \; b^T y < 0$ (or) $\{ \underline{A^T y \geq 0, \; b^T y = -1} \}$ (you can scale y). System 2.

avoids strict inequalities

System 1 is feasible iff System 2 is infeasible. (farkas' lemma)

LP Duality

$x_1 \le 4$

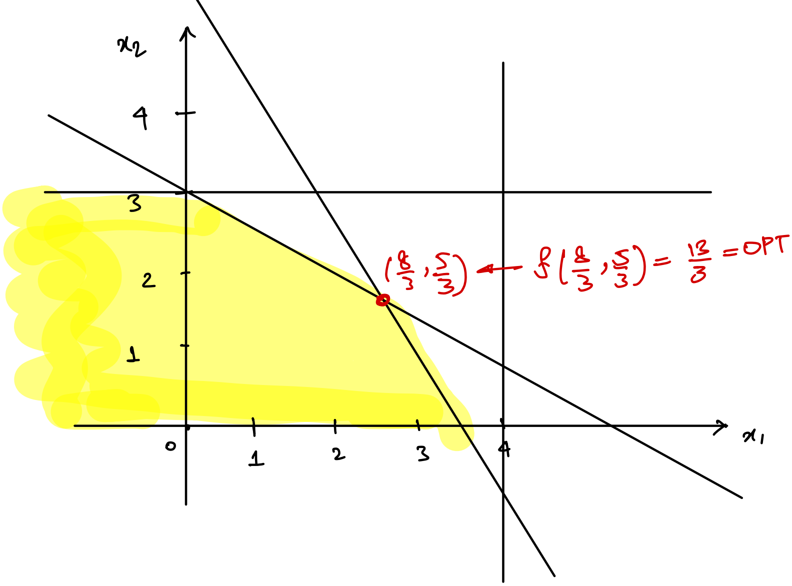
$x_2 \le 3$

$2x_2 + x_1 \le 6$

$2x_1 + x_2 \le 7$

$x_2 \ge 0$

Max $x_1 + x_2$



Lower bound.

(0,0) is feasible

$f(0,0) = 0$

(2,2) is feasible

OPT ≥ 4.

Adding $x_1 \le 4$, $x_2 \le 3$ → OPT ≤ 7 (upper bound)

Feasible point → lower bound, combination of constraints = upper bound

① + ③ : OPT ≤ 5.

③ + ④ : OPT $\le \frac{13}{3}$

Given n, j is the jth bit of smallest factor 0? ← NP ∩ co-NP but P?

If an NP-complete problem lies in co-NP ⇒ NP ⊆ co-NP. To show co-NP ⊆ NP,

$l \in$ co-NP ⇒ $l' \in$ NP ⇒ $l' \in$ co-NP ⇒ $l \in$ NP #

Also, if NP-complete ∈ co-NP, $l \xrightarrow[\text{poly}]{} s$, s is co-NP ⇒ $l \in$ co-NP

Lecture — Farkas lemma — 29/8/23

$Ax = b, x \geq 0$ | $A^T y \geq 0,\ b^T y = -1$

not feasible $\iff$ feasible

$Ax \leq b,\ x \geq 0$ | $\underbrace{y \geq 0}_{\text{multiplier}},\ A^T y \geq 0,\ b^T y = -1$

(subtracting ineq doesn't make sense)

$y_1, y_2, \ldots, y_k \geq 0.$

$\max\ w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$

$y_1 \{ a_{1,1} x_1 + a_{1,2} x_2 + \cdots + a_{1,n} x_n \leq b_1 \}$

$\vdots$

$y_k \{ a_{k,1} x_1 + \cdots + a_{k,n} x_n \leq b_k \}$

$$\underbrace{\sum_i a_{i,1} y_i}_{= w_1} x_1 + \underbrace{\sum_i a_{i,2} y_i}_{= w_2} x_2 + \cdots + \underbrace{\sum_i a_{i,n} y_i}_{= w_n} x_n \leq \sum y_i b_i$$

(then)

$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \leq \sum_i b_i y_i$

Dual LP

$y_1, y_2, \ldots, y_k \geq 0$

$\sum a_{i,1} y_i = w_1$

$\cdots$

$\sum_{i=1}^{k} a_{i,n} y_i = w_n$

$\min \sum y_i b_i$ ← best possible lower bound.

For any feasible x for primal LP, for any feasible y for Dual LP

$f(x) \leq g(y)$ {weak duality theorem}

Strong Duality Theorem

If the primal LP has an optimal solution, say x^*, $\exists$ dual feasible solution y^* (optimal)

such that $\sum_{j=1}^{n} w_j x_j^* = \sum_{i=1}^{k} b_i y_i^*$

↑ due to weak duality theorem.

Proof: (next pg)

e.g. $x_1 \le 4$

$x_2 \le 3$

$2x_2 + x_1 \le 6$

$2x_1 + x_2 \le 7$

$x_2 \ge 0$

i.e.

$x_1 \le 4$

$x_2 \le 3$

$x_1 + 2x_2 \le 6$

$2x_1 + x_2 \le 7$

$-x_2 \le 0$

max $x_1 + x_2$

Primal		Dual
Constraint	$\rightarrow$	variable
Variable	$\rightarrow$	constraints
objective	$\rightarrow$	Rightside number
Right side nos	$\rightarrow$	objective

Dual (Dual) = Primal

Dual LP

$y_1, y_2, \ldots, y_5 \ge 0$

$y_1 + y_3 + 2y_4 = 1$

$y_2 + 2y_3 + y_4 - y_5 = 1$

min $4y_1 + 3y_2 + 6y_3 + 7x_4$

suppose first l constraints are tight for x^*

$a_1^T x^* = b_1$

$a_2^T x^* = b_2$

$\ldots$

$a_l^T x^* = b_l$

$a_{l+1}^T x^* < b_{l+1}$

$\ldots$

$a_k^T x^* < b_k$

$\sum_{i=1}^{l} a_{i,1} y_i = w_1$, $y_1, \ldots, y_l \ge 0$

$\ldots$

$\sum_{i=1}^{l} a_{i,n} y_i = w_n$

Suppose this is not possible, from Farkas lemma (w and cone of vectors a_i seperabe)

$\exists\ \alpha_1, \alpha_2, \ldots, \alpha_n$ s.t.

$\alpha_1 w_1 + \cdots + \alpha_n w_n > 0$, and,

$\forall i \in [l]$, $\alpha_1 a_{i,1} + \alpha_2 a_{i,2} + \cdots + \alpha_n a_{i,n} \le 0$

Plan: $x^* \xrightarrow{\alpha} x'$, x' has larger value than x^* then done.

$x' = x^* + \varepsilon\alpha$, $\varepsilon > 0$ small (Imp: $a_i^T \alpha < 0$ for $i \le l$ only)

for $1 \le i \le l$

$a_i^T x' = a_i^T x^* + a_i^T \varepsilon \alpha$

$= b_i + (\le 0) \le b_i$

for $i > l$

$a_i^T x' = a_i^T x^* + a_i^T \varepsilon \alpha < b_i$ for small ε.

Hence x' is feasible.

$w^T x' = w^T x^* + \varepsilon \underbrace{w^T \alpha}_{>0} > w^T x^*$, hence contradiction. , i.e. $\exists\ y$ which is feasible for dual LP.

Now, we need to show dual objective has equal optimal to primal.

$y^* = (y_1^*, y_2^*, \ldots, y_\ell^*, \underbrace{0, \ldots}_{k-\ell \text{ zeros}})$, where y^* is OPT of dual LP

$$\sum_{i=1}^{k} b_i y_i^* = \sum_{i=1}^{\ell} b_i y_i^* = \sum_{i=1}^{\ell} \left(\sum_{j=1}^{n} a_{i,j} x_j^*\right) y_i^*$$

$$= \sum_{j=1}^{n} x_j^* \left(\sum_{i=1}^{\ell} a_{i,j} y_i^*\right)$$

$$= \sum_{j=1}^{n} x_j^* w_j \quad , \text{ since } f(x) \le g(y) \Rightarrow f(x) = g(y) \qquad \blacksquare$$

Note: $x \to$ primal feasible, $y \to$ dual feasible, $f(x) = g(y) \Rightarrow$ x is optimal and y is optimal.

Primal LP		Dual LP
$Ax \le b$, max $w^T x$		$A^T y = w$, $y \ge 0$, min $b^T y$
unbounded	$\Rightarrow$	Infeasible (since dual sol^n gives upper bound of primal)
Bounded	$\Leftarrow$	Feasible
Infeasible	$\Leftarrow$	Unbounded
Feasible	$\Rightarrow$	Bounded

Note: It is possible that both are infeasible.

(good to remember)

Primal LP	Dual LP
$Ax \le b$ max $w^T x$	$A^T y = w$ $y \ge 0$ min $b^T y$
$Ax \le b$ $x \ge 0$ max $w^T x$	$A^T y \ge w$ $y \ge 0$ min $b^T y$
$Ax = b$ $x \ge 0$ max $w^T x$	$A^T y \ge w$ $y \ge 0$ min $b^T y$

Convention: max programs, constraints are $=$ or $\le$.

max $\longleftrightarrow$ min

constraint $\longleftrightarrow$ variable

$=$ $\longleftrightarrow$ $+/-$

$\le$ $\longleftrightarrow$ $y \ge 0$ (non-neg comb.)

variable free $\longleftrightarrow$ equation

$x \ge 0$ $\longleftrightarrow$ Ineq. constraint

for min, ineq ≥ 0.

Lecture 01/09/23

Economic Interpretation

wheat and chickpeas. Quantities $x_w, x_c \geq 0$. x_w = no. of quintals of wheat, x_c = no. of quintals of chickpeas

(acre/quintal)

$0.05x_w + 0.1x_c \leq 5$ (Land)

$0.5x_w + 2x_c \leq 70$ (Fertilizer)

$x_w + 11x_c \leq 80$ (Electricity)

Max $2500x_w + 6000x_c$
(price/unit)

Dual LP {mechanical process, not always interpretable}

y_L = revenue/unit land

min $5y_L + 70y_F + 80y_E$

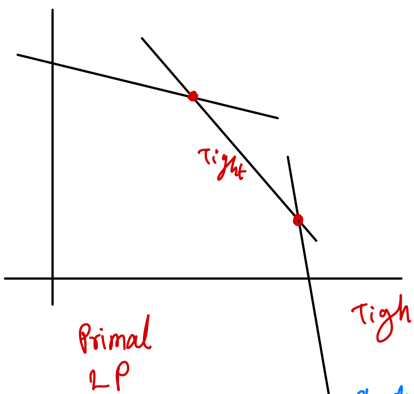
(for every primal variable, dual constraint)

(shadow price/unit wheat)

(wheat) $\boxed{0.05y_L + 0.5y_F + y_E} \geq 2500$

(chickpeas) $0.1y_L + 2y_F + 11y_E \geq 6000$

$y_L, y_F, y_E \geq 0$



Some land unused, fertilizer, electricity used.

Primal optimal

$x_w = 57.24$

$x_c = 20.68$

Dual optimal

Tight

$y_F = 2241$

$y_E = 1379$

$y_L = 0$ (not tight, so zero) ← To be proved later

shadow pricing

↓

if you increase fertilizer by δ, increase in revenue $y_F\delta = 2241\delta$

min $5y_L + \underline{(70+\delta)}y_F + 80y_E$

since tight constraints don't change, dual solution doesn't change by much.

Shortest Path

Directed graph with edge weights with source s, destination t

(no incoming edges) (no outgoing edges)

$x_e \in \{0,1\}$ for $e \in E$

$$\min \sum_{e \in E} w_e x_e$$

• Since weights are positive, LP will remove cycles by itself

$$\sum_{e \in out(s)} x_e = 1 \quad , \quad \sum_{e \in in(t)} x_e = 1$$

For $v \in V$, $\neq s,t$: $\sum_{out(v)} x_e = \sum_{in(v)} x_e$ {shortest walk $\Rightarrow$ shortest path}

(we shouldn't expect to come up with LP for path exactly cuz we could encode longest path otherwise).

For negative weights, shortest $s-t$ path is NP-hard, so negative weights not handled.

to convert to LP, $0 \le x_e \underbrace{\le 1}_{\text{not needed}}$ $(x_e \geq 0)$ (≤ 1 taken care of by LP)

• The optimal value still gives the shortest path! (proof using duality)

Claim: LP optimal = weight of the shortest path

Primal

$$\min \sum_{e \in E} w_e x_e$$

for $v \in V \setminus \{s,t\}$

$$\sum_{e \in in(v)} x_e - \sum_{e \in out(v)} x_e = 0$$

$$-\sum_{e \in out(s)} x_e = -1$$

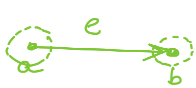
$$\sum_{e \in In(t)} x_e = 1$$

Observation: $\{y_v\}_{v \in V}$

$$\max \; y_t - y_s$$

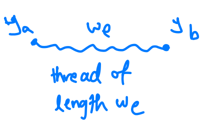
for $e=(a,b) \in E$, (look at x_e, in $= -1$, out $= +1$)

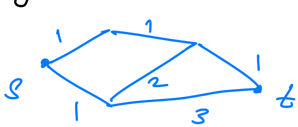
$$-y_a + y_b \le w_e$$

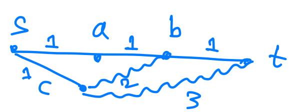


(Updates used for number at each vertex in Dijkstra's algorithm).

Note: Dijkstra's algorithm basically solves the dual LP.







Proof : $x \leftarrow$ primal feasible $\quad \sum_e w_e x_e = y_t - y_s \Rightarrow$ indeed optimal. (both x and y)

$y \leftarrow$ dual feasible

Let $\underbrace{x = \text{shortest path}}_{\text{still holds for neg. weights}}$, for y, $y_a = \underbrace{\text{distance from } s}_{\text{breaks for neg. weights}}$, now proof follows easily.

[HW] Write an LP for max-flow, show that dual LP min-cut, and using this show max-flow min-cut

* Dilworth, hall, mirsky, König, etc. all come from LP duality.

Complementary Slackness

$$\max\ w^T x \qquad\qquad \min\ b^T y$$
$$Ax \leq 0 \qquad\qquad A^T y \geq w$$
$$x \geq 0 \qquad\qquad y \geq 0$$

Claim :

x^* is an optimal solution, y^* is a dual optimal solution iff

(1) $\sum_i a_{ij} y_i^* > w_j \Rightarrow x_j = 0$ (j^{th} primal var $\longleftrightarrow$ j^{th} dual constraint)

$x_j > 0 \Rightarrow \sum_i a_{ij} y_i^* = w_j$

(2) $y_i^* > 0 \Rightarrow \sum_j a_{ij} x_j^* = b_i$

$\sum_j a_{ij} x_j^* < b_i \Rightarrow y_i^* = 0$

Proof : $\sum_{j=1}^n w_j x_j \leq \sum_{j=1}^n \left(\sum_{i=1}^k a_{ij} y_i\right) x_j$ (from dual constraints)

$= \sum_{i=1}^k \left(\sum_{j=1}^n a_{ij} x_j\right) y_i$

$\leq \sum_{i=1}^k y_i b_i$

At optimal solution, we have equalities & if equalities, optimal.

$\sum_j \underbrace{x_j}_{\geq 0} \big(\underbrace{\sum_i a_{ij} y_i - w_j}_{\geq 0} \big) = 0$. Hence, each term is zero, i.e.

(iff x, y is optimal)

$x_j = 0$ or $\sum_i a_{ij} y_i = w_j$

Lecture

$Ax \le b$, $x \ge 0$, $\max w^T x$ $\qquad$ $A^T y \ge w$, $y \ge 0$, $\min b^T y$ $\qquad$ 05/09/23

Feasible solutions x, y are said to satisfy CS conditions if

CS = complementary slackness

1) $\forall_i$ $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j = b_i$

2) $\forall_j$ $x_j = 0$ or $\sum_{i=1}^{k} a_{ij} y_i = b_j$

Theorem : x, y satisfy CS iff x, y are optimal solutions

If LP changes,

$Ax = b$ $\qquad$ $A^T y \ge w$

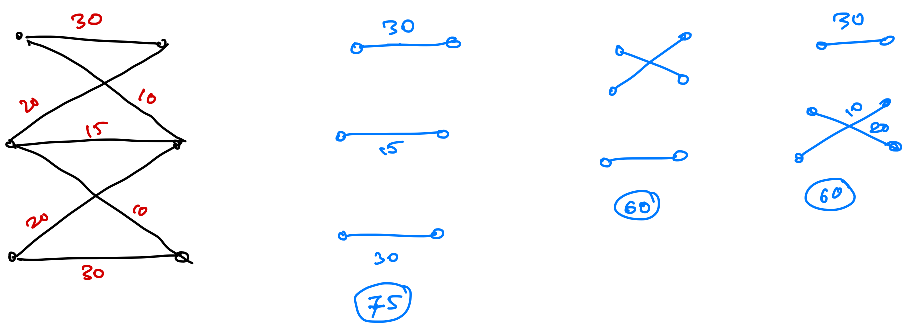
$x \ge 0$ $\qquad$ $\min b^T y$

$\max w^T x$

Condition 1 : unchanged

Ask : Cond^n 2 ?

Minimum weight Perfect matching in Bipartite Graphs

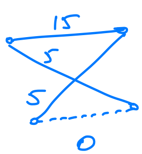


TA matching : weight = preference of course for TAs.

If TA can have multiple courses, just add copies of TA in left side.

Maximum weight matching bipartite $\le$ min weight perfect matching bipartite.

e.g.



max weight match = 15

if we want min-weight p.m,

adding 0-edges = max-weight perfect matching

now make edge weights negative $\Rightarrow$ min weight perfect matching

ILP for min-weight perfect matching

for $e \in E$, $x_e \in \{0,1\}$

$\forall v \in V$, $\sum\limits_{e \text{ incident on } v} x_e = 1$

$\min \sum\limits_{e \in E} x_e w_e$

LP

for $e \in E$, $0 \leq x_e \boxed{\leq 1}$ not required as non-neg. x_e sum to 1.

$\forall v \in V$, $\sum\limits_{e \text{ incident on } v} x_e = 1$

$\min \sum\limits_{e \in E} x_e w_e$

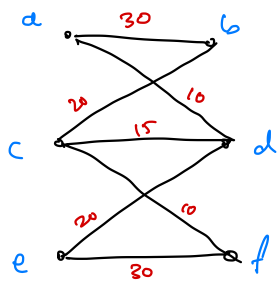
- This LP exactly gives minimum weight perfect matching!

Dual LP

y_v for $v \in V$

$\max \sum\limits_{v \in V} y_v$

for $e = (a,b) \in E$, $y_a + y_b \underset{\text{max prog.}}{\leq} w_e$



$y_a + y_b \leq 30$

$y_a + y_d \leq 10$

$y_c + y_b \leq 20$

$y_c + y_d \leq 15$

$y_c + \cancel{y_f} \leq 10$

$y_e + y_d \leq 20$

$y_e + y_f \leq 30$

CS conditions

$x_e = 0$ or $y_a + y_b = w_e$

Equivalently,

$y_a + y_b < w_e \Rightarrow x_e = 0$

(non-tight)

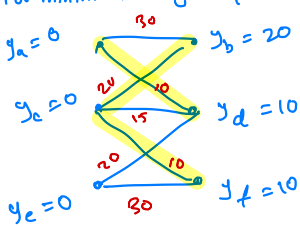
Primal - Dual Algorithm

Dual feasible solution y

→ Try to construct a primal solution x s.t. (x, y) satisfy CS.

- succeed → OPTIMAL
- fail → we get a way to improve dual solution (better objective)

For minimum weight perfect matching,



Tight edges (constraints) are highlighted.

dual vars $\xrightarrow{\text{Primal.}}$ Try to construct perfect matching among tight edges.

no p.m. exists

⇓

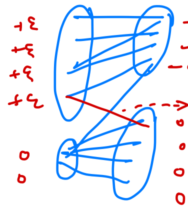
Better dual solution?

In bipartite graph H if there is no perfect matching then $\exists\, S \subseteq L$ s.t. $|N(S)| < |S|$

choose $S = \{e\}$, $N(S) = \phi$

For $v \in S$, $y_v \leftarrow y_v + \varepsilon$

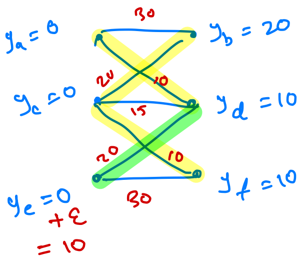
$v \in N(S)$ (tight), $y_v \leftarrow y_v - \varepsilon$ } to maintain feasibility.



not tight (may violate constraint)

so choose ε small enough

This yields a feasible solution with better optimal value



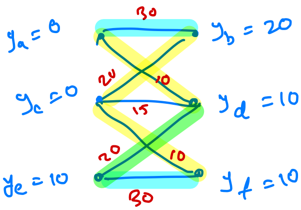
→ no p-m. in tight edges

$S = \{a, e\} \rightarrow \{d\}$

Note: earlier tight edges can become loose!

$(\varepsilon = \min\{10, 10\} = 10)$

$10 = \varepsilon + y_a = 0$



$20 = \varepsilon + y_e = 10$

Perfect matching ←

Yes! (take any = OPT)

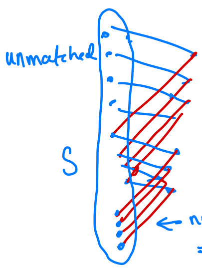
Questions:

1. Termination → Dual objective strictly increases (bounded by min weight p-m)

Change in dual objective $\sum_{u \in V} y_u = \varepsilon|S| - \varepsilon|N(S)| > 0$.

2. Running Time

How to find S



← no augmenting path
⇒ ends in left side at matched.

change $\geq +1$. (integral change).

max dual objective = min weight perfect matching

$O(\sum_e w_e)$ ← pseudo polynomial. (poor analysis)

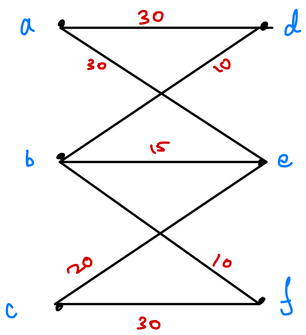
Better analysis (assignment)

3. Optimality → Complementary slackness

Lecture

08/09/23

Taxis Passenger



Edge weights = cost of taxi reaching passenger

dual. $y_a + y_d \leq 30$

$y_a + y_e \leq 20$

....

y_i = cost paid by person/driver

constraints ↦ matching with min y_i chosen given other y_j s are fixed.

Taxi-company : get maximum money out of everyone

max $\sum y_i$

LP exact formulations

- Bipartite Matching
- Shortest walk
- Max flow
- Interval scheduling
- Interval coloring (e.g. assign lecture halls to courses with slots, overlap ⇒ different colors)

LP exact formulation (non-trivial)

- Spanning trees

$1 \geq x_e \geq 0$ for $e \in E$

min $\sum w_e x_e$

for all $S \subset V$, $\sum_{e \in (S, \bar{S})} x_e \geq 1$
$S \neq \phi$

$\sum_{e \in E} x_e = n - 1$

$x_e \in \{0, 1\}$

Then, it is correct ILP,

but LP can have fractional solutions

(not exact)

#ques : Counter-example?

Exact LP formulation

$1 \geq x_e \geq 0$ for $e \in E$

min $\sum w_e x_e$

for all $S \subset V$, $\sum_{e \in (S, \bar{S})} x_e \leq |S| - 1$
$S \neq \phi$

$\sum_{e \in E} x_e = n - 1$

LP-OPT = MWST

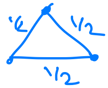
Non-trivial proof

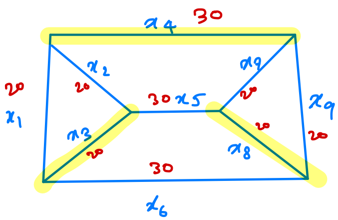
{possible presentation topic}

- Perfect Matching in non-bipartite Graphs (MWPM)

min weight perfect matching

Usual LP won't work, cuz triangle.





MWPM = 70

Non-integral sol^n : $x_1 = x_2 = x_3 = 1/2$

$x_4 = x_5 = x_6 = 0$

$x_7 = x_8 = x_9 = 1/2$

$\sum w_e x_e = \frac{1}{2} \times 6 \times 20 = 60 < \text{MWPM}.$

$x_1 + x_2 + x_4 = 1$ & so on.

So, we need more constraints,

Edmonds [Exact LP formulation]

$\forall\ S \subseteq V,\ |S| = \text{odd}$

$$\sum_{e \in (S, \bar{S})} x_e \geq 1$$

LP-OPT = MWPM

Edmonds primal-dual algorithm $\longrightarrow$ only known algorithm for MWPM

Maximum matching

$\max \sum x_e$

$\forall S \subseteq V,\ |S| = \text{odd} \geq 3$

$$\sum_{e \in E[S]} x_e \leq \frac{|S|-1}{2}$$

$$\sum_{e \in in(v)} x_e \leq 1$$

Note : This is an exact LP formulation and primal-dual algorithm is similar to edmonds - blossom algorithm

Approximation Algorithms using Linear programs

Ideas :

1. Primal-Dual Algorithm
   - Relax complementary slackness
2. Rounding
   - solve LP
   - Round the fractional solution (problem-specific) $\longrightarrow$ Integral solution (may not be optimal)
   - Prove that rounding doesn't lose much $\longrightarrow$ approximation factor.

Minimum weight Vertex cover (NP-hard)

So, we look for an approximation algorithm

$ALG \leq OPT + \beta$ additive approximation

$ALG \leq \underbrace{\alpha}_{\text{constant}} \cdot OPT$ multiplicative approximation

None of these is guaranteed & one doesn't guarantee others

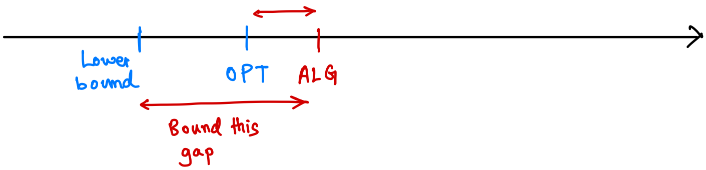
(or) $\left(1+\frac{1}{\sqrt{n}}\right)$ ← better than constant

(or) $(1+\varepsilon)$ ← ε is any real > 0

2-approximation algorithm

Nothing better is known (e.g. 1.99 approx is open)

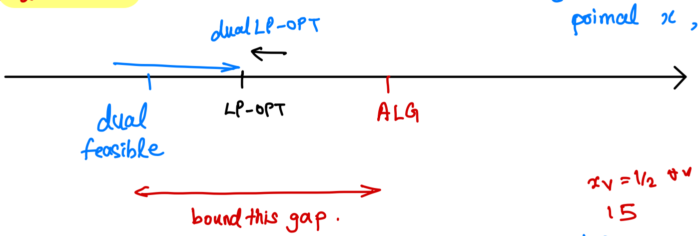
For comparing ALG with OPT, we need some lower bounds on OPT, and compare ALG with lower bound.



1. Any matching, for every matching edge 1 vertex must be picked.. (sum of lower vertex weights)

2. dual LP

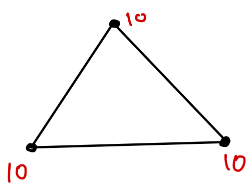
Any feasible dual y, feasible primal x, $g(y) \leq f(x)$



$x_v = 1/2 \;\forall v$ choose 2 vertices

15 20

LP-OPT $\neq$ MWVC



Primal LP (relaxed from ILP)

$0 \leq x_v \quad \forall v \in V$

for $(u,v) \in E$, $x_u + x_v \geq 1$

$\min \sum_u x_u w_u$

Can we write an exact LP formulation for vertex cover if we allow exp. many constraints?] unlikely (NP-hard prob. not believed to be in co-NP)

each constraint is easy to check.

If you have a "nice" LP description even with exponentially many constraints
$\Rightarrow$ problem is in co-NP

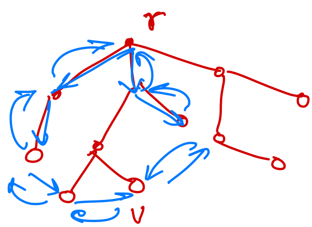
Is there a v.c. of size k, answer is no,
then we have a system of inequalities with $\sum_u w_u x_u \le k-1$
which is infeasible, i.e. $\exists y$ s.t. $A^T y = -1, b^T y > 0$. (farkas lemma)

Lecture 12/09/25

Spanning Tree Polytope

$$x_e \geqslant 0 \quad e \in E$$

$$\sum_{e \in E} x_e = n-1$$



paths for directed = easy. incoming = outgoing

$$\sum_{e \in out(r)} x_e = 1$$

$\forall u \in V \setminus \{r, v\}$,

$$\sum_{e \in in(u)} x_e = \sum_{e \in out(u)} x_e$$

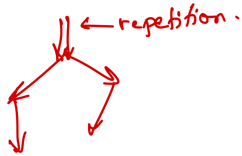
$$\sum_{e \in in(v)} x_v = 1$$

encodes path (walks) from r to v

For every vertex, make copies of x_e

$$y_e^1, \ldots, y_e^n. \qquad e \in \vec{E} = 2|E|$$

write this for all v.



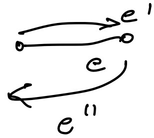
$$\sum_{e \in out(r)} y_e^i = 1$$

$$\sum_{e \in in(v)} y_e^i = \sum_{e \in out(v)} y_e^i \quad v \notin \{r, i\}$$

$$\sum_{e \in in(i)} y_e^i = 1$$

$z_{e'} \geq y_{e'}^{i}$ $\quad 1 \leq i \leq n$

$z_{e''} \geq y_{e''}^{i}$ $\quad 1 \leq i \leq n$



$x_e = z_{e'} + z_{e''}$

$\min \sum w_e x_e$

#constraints = $O(n^2)$, #variables = poly(n)

Last class, we saw

For every S,

$$\sum_{e \in E[S]} x_e \leq |S| - 1$$

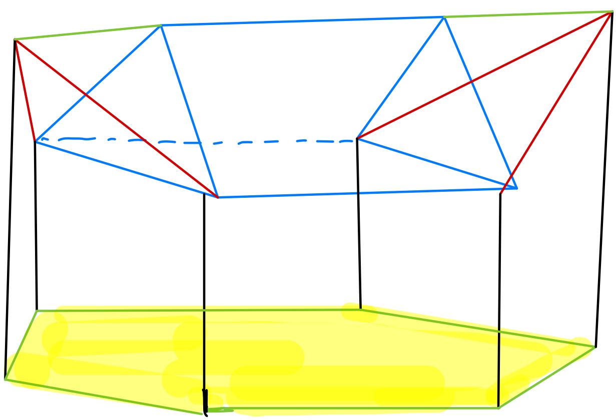
Has #constraints = $O(2^n)$.

Hence, path description is better! (and it is an exact LP formulation)

Interior point / simplex $\leftarrow$ no. of constraints

Ellipsoid $\leftarrow$ doesn't depend on no. of constraints (separation oracle)

these are extended LP formulations



3 vars

triangular prism : 5 faces, 5 constraints

Projection = hexagon

6 constraints

2 vars

Example:

$x_1 \le y_1$
$x_1 \le y_2$
$x_2 \le y_1$
$x_2 \le y_2$
$x_3 \le y_1$
$x_3 \le y_2$

$\rightarrow$

$x_1 \le z$
$x_2 \le z$
$x_3 \le z$
$z \le y_1$
$z \le y_2$

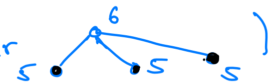
Note: # vars > # constraints then usually redundancy in variables

Same feasible region since $2 \rightarrow 1$ by F.M. elimination

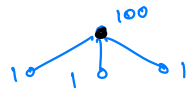
Primal-Dual Approx. (Min weight vertex cover)

2-approximation

Greedy approaches:

1. min weight vertex (countereg: star 

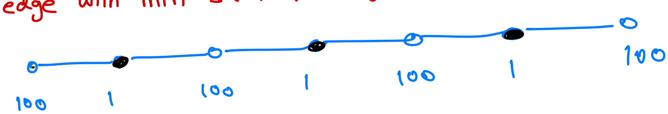
)

2. Largest degree (

)

3. min $\frac{w_v}{d_v}$ $O(\log n)$ approximation (tight)

4. Pick edge with min sum of weights on end pts, choose both end pts



(covers 100-1 edges instead)

LP

$\min \sum w_u x_u$

$x_u \ge 0$

for $e=(a,b)$, $x_a + x_b \ge 1$

Dual LP

$y_e \ge 0$

for $u \in V$,

$\sum_{e \in in(u)} y_e \le w_u$

$\max \sum y_e$

x_u with coeff 1 for all $e \in incident(u)$

Complementary slackness

① $x_u = 0 \Leftarrow \sum_{e \in in(u)} y_e < w_u$ (Equivalent to $x_u > 0 \Rightarrow \sum_{e \in in(u)} y_e = w_u$)

② $e=(a,b)$ $y_e > 0 \Rightarrow x_a + x_b = 1$. (approximately true, trivially)

Note: Every v.c. satisfies $2 \ge x_a + x_b \ge 1$

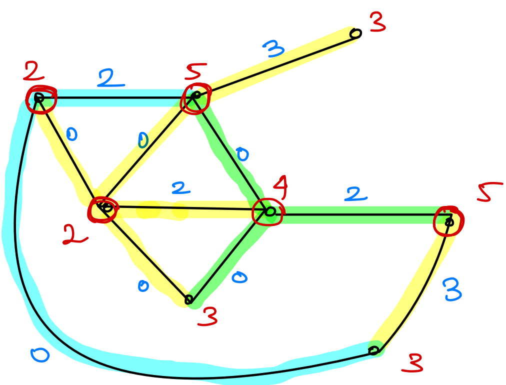
Primal - Dual

① Feasible dual

② If any dual constraint is tight, pick that vertex in vertex cover

③ Try to increase any y_e for any edge which is not yet covered

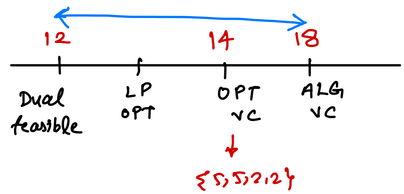
Example



Can't increase highlighted edges as their vertices are tight

ALG : weight = 18

Dual objective = 12



$$ALG = \sum_{u \in S} w_u$$ (for selected vertices $w_u = \sum_{e \in in(u)} y_e$)

$$= \sum_{u \in S} \sum_{e \in in(u)} y_u$$ appears at most twice

$$= \sum_{u \in V} x_v \sum_{e \in in(u)} y_u = \sum_{e=(a,b) \in E} y_e (x_a + x_b) \leq 2 \sum_{e \in E} y_e$$

$$\leq 2 \cdot (OPT - VC) \blacksquare$$

Hence, we get a 2-approximation,

Analysis works if both vertices are chosen when both are tight.

Lecture (15/09/23)

Game Theory (Application of LP duality)

- Multiple players
- Each player chooses a strategy
- Payoff of each player depends on all players' strategies

Nash Equilibrium (1951)

A Tuple of strategies is a Nash equilibrium if no player can gain by unilaterally changing strategy.

Example Prisoner's Dilemma

Two prisoners
- Silent
- Approver

	S	A
S	-1, -1	-4, 0
A	0, -4	-2, -2

Only nash eqbr^m (but not global optimum)

Example : Rock, Paper, Scissor

	R	P	S
R	0, 0	-1, 1	1, -1
P	1, -1	0, 0	-1, 1
S	-1, 1	1, -1	0, 0

No PSNE, but $\exists$ a MSNE.

(Nash) $\exists$ Mixed Strategies which is a nash eqbr^m.

In this example, $(1/3, 1/3, 1/3)$, $(1/3, 1/3, 1/3)$ is a MSNE.

If R, $\mathbb{E}[\text{payoff}] = 0$ (11^y for P, S). Hence, any strat. leads to $\mathbb{E}[\text{payoff}] = 0$.

Two-player zero sum Game

We show NE exists and, this is equivalent to LP duality.

NE exists $\longleftrightarrow$ LP duality

Red payoff matrix $= R$

Blue payoff matrix $= B$

$$B, R \in \mathbb{R}^{m \times n}, \quad B + R = 0$$

Blue player $\rightarrow$ n pure strategies $\{x \in \mathbb{R}^n \mid \sum x_i = 1, x_i \geq 0\}$

Red player $\rightarrow$ m pure strategies $\{y \in \mathbb{R}^m \mid \sum y_i = 1, y_i \geq 0\}$

R	B	a	b	c
y_1	0	20, −20	10, −10	30, −30
y_2	1	−10, 10	20, −20	10, −10
		x_1	x_2	x_3

If Blue player decides first $x \in \mathbb{R}^3$ $(y \in \mathbb{R}^2)$

$$\max_{y \in \mathbb{R}^2} f(x,y) = \max_{y \in \mathbb{R}^2} y^T R x$$

Blue would like to minimize red payoff

$$= \min_{x \in \mathbb{R}^3} \max_{y \in \mathbb{R}^2} f(x,y) \quad \{\text{Payoff for the red player}\}$$

If red player decides first, payoff for red player $= \max_{y \in \mathbb{R}^2} \min_{x \in \mathbb{R}^3} f(x,y)$

If Blue decides first, better for red.

$$\min_x \max_y f(x,y) \geq \max_y \min_x f(x,y)$$

Note: For pure strategies, min max $\neq$ max min, but for mixed strategies min max = max min

For any x, define $g(x) = \max_y f(x,y)$

for any y, define $h(y) = \min_x f(x,y)$

Claim: $\forall x, y$, $g(x) \geq h(y)$ (and hence, $\min_x g(x) \geq \max_y h(y)$)

Proof: $g(x) =$ max val in x^{th} row

$h(y) =$ min val in yth column

$g(x) \geq R(x,y) \geq h(y)$ for pure strat.

$\Rightarrow$ holds for lin combinations

Claim: Suppose $\exists\, x^*, y^*$ s.t. $g(x^*) = h(y^*)$ then, (x^*, y^*) is a nash equilibrium.

Proof:

$$\left.\begin{aligned} f(x^*, y^*) &\geq f(x^*, y') \\ f(x^*, y^*) &\leq f(x', y^*) \end{aligned}\right\} \text{subclaims} \leftarrow \text{easy to show.}$$

Showing Duality $\rightarrow$ NE.

Blue plays first: $\min\ z$

$z \geq 20x_1 + 10x_2 + 30x_3$

$z \geq -10x_1 + 20x_2 + 10x_3$

$\left(\min_x \max(\text{Red } 0, \text{Red } 1)\right)$

$x_1 + x_2 + x_3 = 1$

$x_1, x_2, x_3 \geq 0$

Red first (y_1, y_2): $\min\,(20y_1 - 10y_2,\ 10y_1 + 20y_2,\ 30y_1 + 10y_2)$

$\max\ w$

$w \leq 20y_1 - 10y_2$

$w \leq 10y_1 + 20y_2$

$w \leq 30y_1 + 10y_2$

$y_1, y_2 \geq 0$

$y_1 + y_2 = 1$

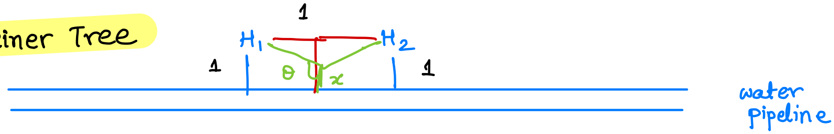
Since, these LPs are dual to each other.
Hence, by strong duality, these values are equal $\blacksquare$

26/09/23

Lecture

- Presentation topics (shared on moodle) ~20%

Steiner Tree



x is minimised at $\theta = 120°$, length $\approx 1.86 < 2$ (red, blue solutions)

Euclidean Steiner Tree Problem (NP-hard)

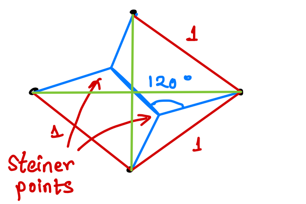
Given a set of points in $\mathbb{R}^2$ we want to connect them with minimum total length of line segments

Approximate Algorithm

Compute $\binom{n}{2}$ distances

Compute minimum spanning tree

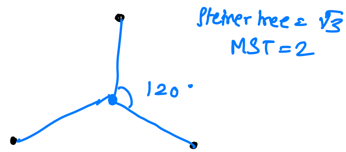
Conjecture: $2/\sqrt{3}$ (open problem)



Sol^n: $2\sqrt{2}$, $1+\sqrt{3}$, 3

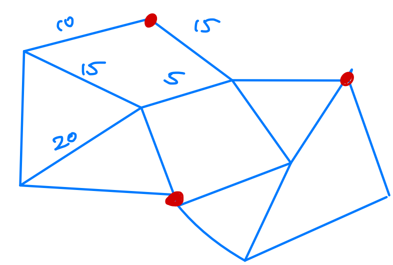
MST = 3

Steiner tree = $1+\sqrt{3}$



120° for balancing equal forces

Graph Steiner Tree



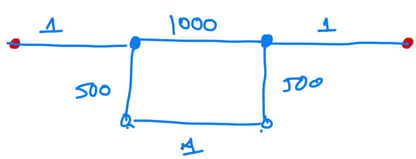
Input: $G(V, E)$

Terminal set $T \subseteq V$

$\{c_e\}_{e \in E}$

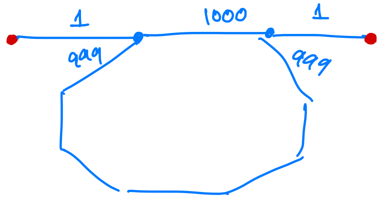
Output: Minimum cost sub-set of edges s.t. all terminals are connected.

Proposal 1: Compute MST for G, Throw out any edge which does not lie in a pair of terminals, i.e. any of the $\binom{|T|}{2}$ paths



Steiner = 1002 (1000+1+1)

MST pruning = 1+500+1+500+1 = 1003



any attached even cycle skips edge 1000

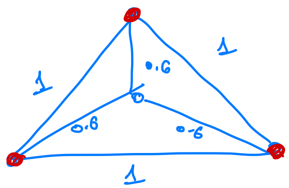
So, difference $= 999n - 1000 =$ unbounded.

Proposal 2: $T = \{t_1, t_2, \ldots, t_K\}$. Shortest path $t_1 \sim t_2$

shortest path $t_3 \sim$ any node on $t_1 \sim t_2$

...

$t_i \sim$ run dijkstra's algorithm, connect first point you hit in current tree



Example for non-optimal

HW: Show this is 2-approximation or show an example which is worse than a 2-approx.

Proposal 2': Allow t_i to hit other terminals

Proposal 3: Compute all pair shortest paths $\binom{|T|}{2}$ distances.

Compute complete graph H using edge weights as above distances

Compute MST on H

Take the union of corresponding shortest paths in G to edges in MST

This gives a 2-approximation. [HW]

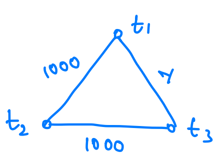
Graph Steiner Forest

Input: $G(V, E)$

set of terminal pairs e.g. (t_1, t_5), (t_3, t_4), (t_2, t_5)

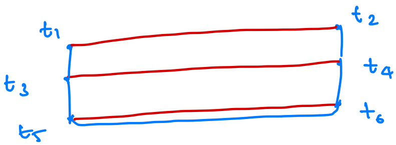
Output: Minimum cost subset of edges s.t. for every given pair (t_i, t_j), t_i is connected to t_j

Proposal 3 can be v. bad here



(t_1, t_2), (t_2, t_3)

Proposal : Input pairs $\longrightarrow$ disjoint subsets of terminals
For each subset run proposal 3
Take union



(t_1, t_2)
(t_3, t_4)
(t_5, t_6)

3x Blue ≈ Red
Worst case : $O(n)$ worse

ques : Why not prune proposal 3 ?

Steiner Forest : Primal-Dual (2-approx)

Hint : For writing LP express steiner forest as a covering problem
{1 edge from each of buckets ⇒ steiner forest }
Exponentially many buckets are fine.

03/10/23

(missed Rec. 29/09)

Lecture

Minimum Steiner Forest

Input : $G(V, E)$ $\{w_e\}_e$
$(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$

Def^n: $U \subseteq V$ is a separating cut if for some i, $s_i \in U, t_i \notin U$

LP relaxation

$\min \sum_e w_e x_e$
$x_e \geq 0$

$\sum_{e \in \delta(U)} x_e \geq 1$, U is sep. cut

$\delta(U)$: set of cut-edges

(similar to spanning tree limiting U s)

Dual LP

$\max \sum_U y_U$
$y_U \geq 0$

$\sum_{U: e \in \delta(U) \text{ and } U \text{ is sep. cut}} y_U \leq w_e$

Primal dual Algo.

① $F \leftarrow \emptyset$
$y \leftarrow 0$

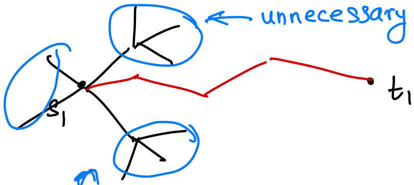
do {

- Let $C_1, C_2, \ldots, C_n$ be the connected components in the tight edge subgraph, which separate some terminal pair, then increase $y_{C_1}, y_{C_2}, \ldots, y_{C_n}$ simultaneously
- Include tight edges one-by-one not creating cycles in F

}
till every (s_i, t_i) pair is connected

Pruning : Remove any edge from F that can be removed

Eg.



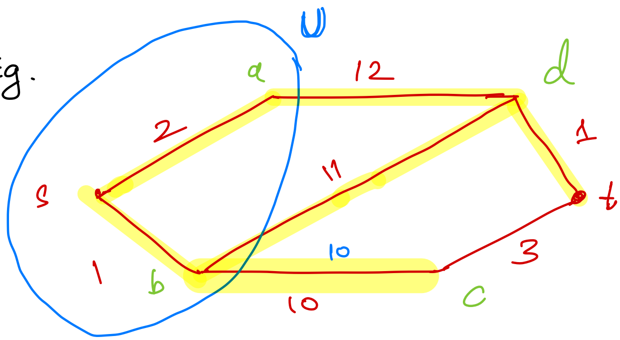
← unnecessary

{some kind of dijkstra}

Hence, pruning is required

edges keep getting included until t_1 is reached.

Eg.



$U_1 \leftarrow \{s, a, b\}$

y_u in 3 constraints

$y_{u_1} \longleftarrow 10$

Next choice can't have this as a cut-edge

$U_2 \leftarrow \{s, a\}$

$y_{u_2} \leftarrow 1$

$U_3 \leftarrow \{s, b, c\}$

$y_{U_3} \leftarrow 1$

$e = (b, d)$ has contri from U_1 already, so inc by atmost 1

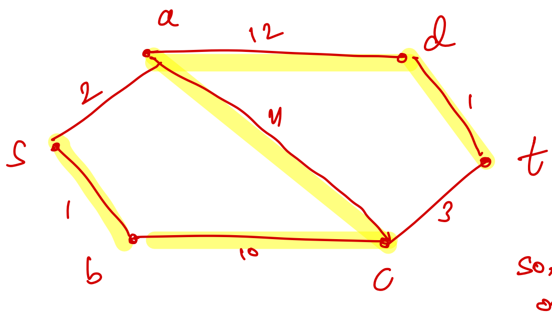
$U_4 \leftarrow \{s, b, c, d\}$

$e = (s, a)$ has contri 1

$y_{U_4} \leftarrow 1$

$e = (a, d)$ had contri 11

Eg.



$U_1 = \{s, a, b\}$ $y_{U_1} = 10$

$U_2 = \{s, a\}$ $y_{U_2} = 1$

$U_3 = \{s, a, b, c\}$ $y_{U_3} = 1$

$U_4 = \{s, a, b, c, d\}$ $y_{U_4} = 1$

So, choosing dual variables in arbitrary order can have v. bad approximation

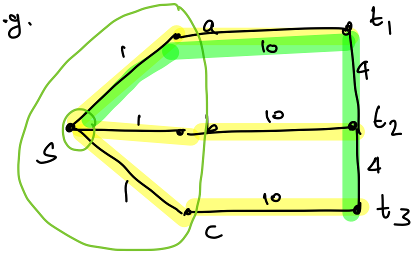
Hence, ordering of dual variables matter.

minimal U : $\{s\} \longrightarrow \underbrace{\{s, b\}}_{}$ $\longrightarrow \{s, a, b\} \rightarrow \ldots$ gives legit shortest path b/w $\{s_1, t_1\}$

(inclusion-wise)

both must be inc, else tight.

E.g.



done.

But, this is a bad sol^n

Green sol^n is better.

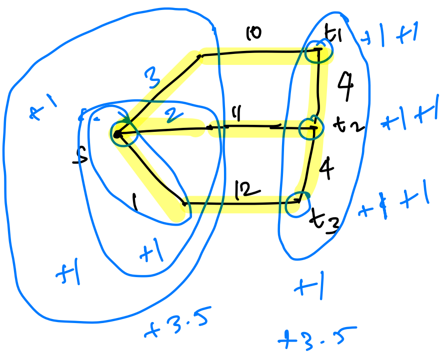
Yellow sol^n can be arbitrarily bad.

$10 \rightarrow$ any n.

also, m arms instead of 3.

So m-approx.

Idea : Start exploration from all minimal Us and increase all simultaneously



$(\varepsilon = 1)$

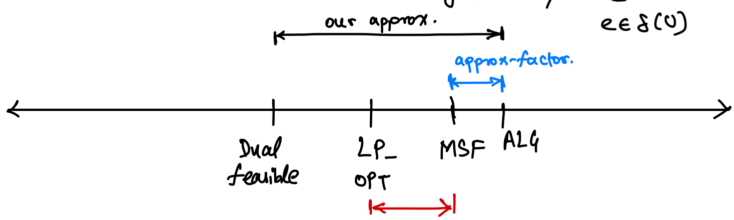
Issue : Many edges becoming tight simultaneously

Sol^n : Include new tight edges one-by-one.

Approximate C.S.

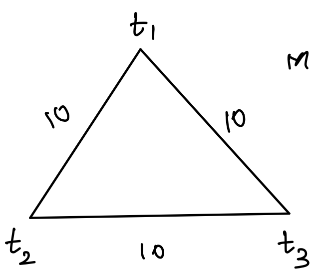
$x_e > 0 \Rightarrow \sum_{\substack{U \\ e \in \delta(U)}} y_u = w_e$ holds

but $y_u > 0 \not\Rightarrow \sum_{e \in \delta(U)} x_e = 1$ } can be very large



Integrality gap ; Gap b/w LP-OPT and integral optimal.

Note : Approximation factor can't be better than integrality gap due to proof strategy. Actually no rel^n b/w approx factor, integrality gap.



t_1, t_2, t_3 should be connected.

MSF = 20

$x_1 = x_2 = x_3 = 1/2$ (feasible primal)

LP-OPT = 15 (no better sol^n)

Integrality gap $= \frac{4}{3}$.

e.g.

K_n, $x_e = 1/_{n-1}$ $\forall e$

$\sum x_e = \frac{n(\cancel{n-1})}{2} \cdot \frac{1}{\cancel{n-1}}$

$IG = 2\left(\frac{n-1}{n}\right) \rightarrow 2$ as $n \rightarrow \infty$

[HW] Construct an example where integrality gap $\to 2$

Hence, by our pf-strategy we cannot hope for better than 2-approximation. We have a graph G with int gap $\to 2$ $(2-\varepsilon,\ \varepsilon = O(\frac{1}{n}))$

<u>Fact</u>: Integrality gap < 2 for minimum steiner tree problem.

Now, let's show 2-approximation.

<u>Proof of 2-approx</u>

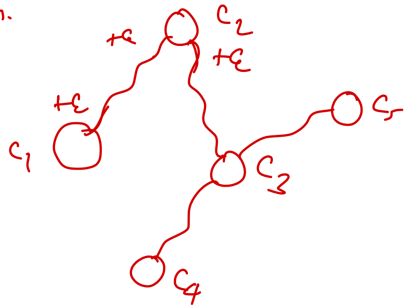
<u>Claim</u>: $w(F) \leq 2 \sum_v y_v$

<u>Idea</u>: use induction.

ΔD = increment in dual objective $= q \cdot \varepsilon$

$$w(F) = \sum_{e \in F} \underbrace{w_e}^{\text{tight}} = \sum_{e \in F} \sum_{\substack{v: \\ e \in \delta(v)}} y_v$$

Now, for the final forest, these components each have exactly 1 path b/w any pair.



Hence, $\Delta w(F) = \overbrace{(q-1)}^{\#\text{pairs.conn.}} \times 2\varepsilon$

Thus, $\dfrac{\Delta w(F)}{\Delta D} \leq 2$

Lecture — LP rounding

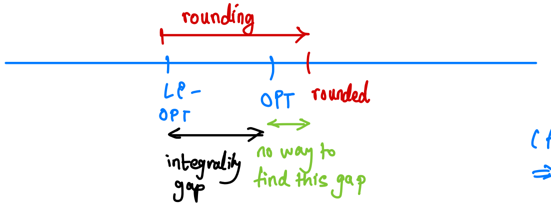
{06/10/23}

1. Write problem as an ILP
2. Relax integer constraint
3. Solve it using LP solvers
4. fractional $\xrightarrow{\text{rounding}}$ integral → need to argue feasibility

LP-OPT

Need to prove :

1) feasibility



So, rounding bounds the gap

(feasible, integral ⇒ sol^n ≥ OPT)

(not dependent on rounding) ← Best hope for approximation factor.

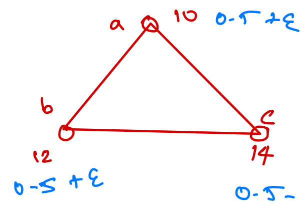
Min vertex cover

$$\min \sum w_u x_u$$

$$x_u \geq 0$$

$$\forall e=(u,v) \quad x_u + x_v \geq 1$$

x^* is an LP-OPT solution.



$x_a = 1$

$x_b = 1$

$x_c = 0$

OPT = 22

$x_a^* = 1/2, \; x_b^* = 1/2, \; x_c^* = 1/2$

OPT = 18

LP-OPT → +8ε.

we can show this is LP-optimal. (show a dual solution)

Rounding

Take u if $x_u^* \geq 1/2$

Don't take u if $x_u^* < \frac{1}{2}$ } → S

since $x_u + x_v \geq 1 \Rightarrow x_u$ or $x_v \geq 1/2 \Rightarrow$ every edge is covered.

$$w(S) = \sum_{u \in S} w_u$$

We want $w(S) \leq 2 \times$ LP-OPT

$$\text{LP-OPT} = \sum_u w_u x_u^*$$

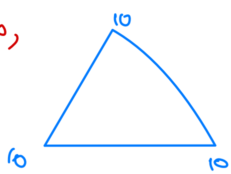
$(x_u^* \geq 1/2)$

$$w(S) = \sum_{u \in S} w_u \cdot 1 \leq \sum_{u \in S} 2 w_u x_u^*$$

$$\leq 2 \sum_u w_u x_u^*$$

Approx ratio is tight for triangle

Now,



LP-OPT = 15

OPT = 20

so, approx ratio = $\frac{4}{3}$

This example shows integrality gap $\geq \frac{4}{3}$ ⇒ we can't get better than $\frac{4}{3}$-approx.

Construct an example with integrality gap more than $4/3$

K_n with $n \to \infty$: cost $\to 2$, so, LP+rounding cannot give better than 2-approximation.

No. approx algorithm is known with better than 2-approx.

Max-SAT (NP-hard).

$(x_1 \vee \overline{x_2})$, $(x_3 \vee \overline{x_1})$, $(\overline{x_2} \vee \overline{x_3})$, x_4, $(\overline{x_4} \vee x_2)$ $\quad$ x_i are boolean variables

Max number of clauses that can be satisfied.

$x_1 = T$, $x_2 = T$, $x_3 = F$, $x_4 = T$

Weighted version : clauses have weights.

Greedy approaches

Approach 1 :
- Pick clause with maximum weight
- Set a literal in this clause to make it satisfied.
- Remove all satisfied clauses

Gives a bad-approx : Not constant

$ALG \geq \alpha \cdot OPT$

Approach 2 :
- Pick a variable, pick T or F s.t. total weight of satisfying clauses is maximised.

$(x_1 \vee \overline{x_2})$, $(x_3 \vee \overline{x_1})$, $(\overline{x_2} \vee x_3)$, x_4, $(\overline{x_4} \vee x_2)$; $x_3 = T$ or $x_2 = F$

- Remove the satisfied clauses and repeat.

Claim : This is a $\frac{1}{2}$-approximation. $\quad ALG \geq \frac{1}{2} \cdot OPT$. $\quad$ [HW]

Convention : for max-problems, $\alpha \leq 1$, for min, $\alpha \geq 1$

Randomized algorithm

set $x_i = \binom{T}{F}$ w/p $\frac{1}{2}$ independently for every variable.

$W \leftarrow$ total weight of satisfied clauses

$$\mathbb{E}[W] = \sum_{\sigma \in \{T,F\}^n} \frac{1}{2^n} \left(\text{weight of satisfied clauses on } \sigma\right)$$ $\leftarrow$ analysis of this is unclear.

Linearity of Expectation

$Z = Z_1 + Z_2$

$\mathbb{E}[Z] = \mathbb{E}[Z_1] + \mathbb{E}[Z_2]$

Idea : Write W as sum of other random variables.

$$W = \sum_{c \in \text{clauses}} w_c Y_c \quad , \quad Y_c = \begin{cases} 1, & \text{if clause } c \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbb{E}[W] = \sum_c w_c \, \mathbb{E}[Y_c]$$

$$\mathbb{E}[Y_c] = p(c \text{ is satisfied}) = 1 - p(c \text{ not satisfied})$$
$$= 1 - 2^{-\ell(c)} \qquad \ell(c) = \#\text{variables in the clause}$$
$$\geq 1/2$$

$$\Rightarrow \mathbb{E}[W] \geq \frac{1}{2} \sum_c w_c \geq \frac{1}{2} \text{OPT}$$

Hence, this algorithm gives a $\frac{1}{2}$-approximation

Note : Ratio is tight for just all clauses single variable.

Note : This can be converted to a deterministic algorithm.

Method of Conditional Expectations

$$\mathbb{E}[W \mid x_1 \leftarrow \text{True}] = \sum_{\sigma \in \{T, F\}^{n-1}} \frac{1}{2^{n-1}} \, (\text{weight of satisfied clause with } \sigma, x_1 \leftarrow \text{True})$$

Again, using linearity of $\mathbb{E}$, we get a simple.

Set $x_1 \leftarrow \text{True}$ if $\mathbb{E}[W \mid x_1 \leftarrow \text{True}] \geq \mathbb{E}[W \mid x_1 \leftarrow \text{False}]$

Remove satisfied clauses and continue with next variable.

This is not same as greedy ALG involving x_1.

observe :

$$\mathbb{E}[W] = \frac{1}{2} \underbrace{\mathbb{E}[W \mid x_1 = \text{True}]}_{A} + \frac{1}{2} \mathbb{E}[W \mid x_1 = \text{False}] \quad \text{, where } \mathbb{E}[W] \geq \frac{1}{2} \text{OPT}$$

$$\leq \mathbb{E}[A] = \frac{1}{2} \mathbb{E}[A \mid x_2 = T] + \frac{1}{2} \mathbb{E}[A \mid x_2 = F] \leq \ldots$$

Inductively show for all x_i assignments, we get $\mathbb{E}[W] \leq \text{ALG}$.

so, $\text{ALG} \geq \mathbb{E}[W] \geq \frac{1}{2}$

Remark : If all clauses have ≥ 2 literals $\Rightarrow \frac{3}{4}$ approx

≥ 3 literals $\Rightarrow \frac{7}{8}$ approx.

[HW] Work out an LP for max-SAT.

Lecture 10/10/23

Presentations

- Find a slot (1 day / week for 3 weeks)

Maximum weight satisfiability

Greedy $\frac{1}{2}$-approx

Randomized $\frac{1}{2}$-approx ← better for larger clauses

$\left(1-\frac{1}{2^l}\right)$-approx if each clause has atleast l-literals.

Converted into deterministic using conditional expectations.

(LP-rounding)

$C_1, \ldots, C_m$

$w_1, \ldots, w_m$

$C_i = x_2 \vee \overline{x_3} \vee x_5$

$1 \le i \le m, \quad y_i \in \{0,1\} \rightarrow 0 \le y_i \le 1$

$\text{Max } \sum w_i y_i$

$1 \le j \le n \quad t_j \in \{0,1\} \quad 0 \le t_j \le 1$

$y_1 \le t_3 + t_5$

$y_1 \ge t_3, \; y_1 \ge t_5$ } not necessary

$C_1 = x_3 \vee x_5$

0	0	0
1	1	0
1	0	1
1	1	1

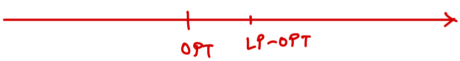
$C_2 = x_4 \vee \overline{x_5} \vee x_1$

$y_2 \le t_4 + 1 - t_5 + t_1$

Use LP solver

$\text{LP-OPT} = (y_i^*, t_j^*)$

If it is integral $\rightarrow$ OPT = LP-OPT



If not integral $\rightarrow$ Rounding

Option 1 $\quad t_j^* \ge 0.5$ set $x_j = T$ else $x_j = F$

option 2 $\quad t_j^* \ge \theta \Leftrightarrow x_j = T$

$C_1 = \overline{x_1} \vee \overline{x_2} \quad 100$

$C_2 = x_1 \vee x_2 \quad 1$

$\max 100 y_1 + y_2$

$t_1 = 0.4, t_2 = 0.6$

choose $\theta = 0.4$ to get both variables true. (bad sol^n)

$(\varepsilon, 1-\varepsilon$ for $\theta = \varepsilon)$

(or)

$C_1 = x_1 \vee x_2 \vee \cdots \vee x_n$

$x_n = 1/n$

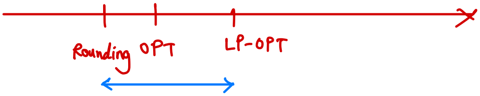
n acc. to θ.

gives 0, but should give 1

Option 3 : Set $x_j = T$ with prob t_j^*

$x_1 = T \quad p = 0.5$
$x_2 = T \quad p = 0.5$

$\mathbb{E}[W] = \frac{3}{4} \times 100 + \frac{3}{4} \times 1 = \frac{3}{4} \times 101.$



P_j = set of +ve literals in C_j
N_j = " — " -ve " — "

$\mathbb{E}[W] = \sum_i w_i \Pr[C_i \text{ satisfied}] = \sum_j w_j \left(1 - \prod_{j \in P_j}(1-t_j^*) \prod_{j \in N_j} t_j^*\right)$

$\left\{ \Pr[C_j \text{ satisfied}] = 1 - \Pr[C_j \text{ not sat}] = 1 - (1-t_1^*)(1-t_2^*) \ldots \right\}$

Compare y_i^* with $\Pr[C_j \text{ satisfied}]$

For clause i, $y_i \le \sum_{j \in P_i} t_j + \sum_{j \in N_i} 1 - t_j$

Would like to show

$$1 - \prod_{j \in P_i}(1-t_j^*) \prod_{j \in N_i} t_j^* \ge \alpha\, y_i^*$$

GM $\le$ AM

$\theta_1 \cdots \theta_l \le \left(\frac{\theta_1 + \cdots + \theta_l}{l}\right)^l$

$$\ge 1 - \left(\frac{\sum_{j \in P_i}(1-t_j^*) + \sum_{j \in N_i} t_j^*}{l_i}\right)^{l_i}$$

$l_i = |P_i| + |N_i|$

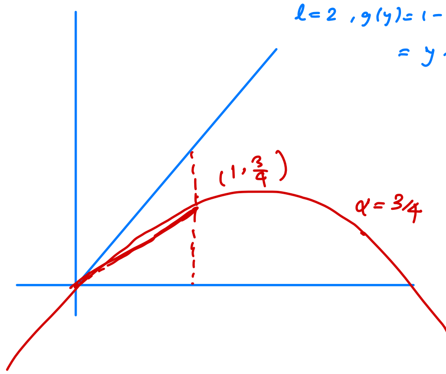
$$= 1 - \left(\frac{l_i - \left(\sum_{j \in P_i} t_j^* + \sum_{j \in N_i}(1-t_j^*)\right)}{l_i}\right)^{l_i}$$

$$\ge 1 - \left(1 - \frac{y_i^*}{l_i}\right)^{l_i}$$

$g(y) = 1 - \left(1 - \frac{y}{l}\right)^l$

we want $g(y) \ge \alpha y$ for some α.

$l = 1, \; g(y) = y$
$l = 2, \; g(y) = 1 - \left(1 - \frac{y}{2}\right)^2$
$= y - \frac{y^2}{4}$



In general

$\frac{g(y)}{y} \ge \alpha = 1 - \left(1 - \frac{1}{l}\right)^l$. $\{g(y) \text{ concave } f^n\}$

Claim : $g(y) \ge g(1)\, y$

Note : This ratio can be pushed to 0.91

$\alpha = 1 - \left(1 - \frac{1}{l}\right)^l \to 1 - \frac{1}{e}$ as $l \to \infty$
≈ 0.63

Homework

1. De-randomise using conditional expectations with same (or better) α
2. LP $\leftarrow$ large clauses give worse approx factor

   Randomized $\leftarrow$ large clauses better α

   Give a hybrid algorithm $\rightarrow \frac{3}{4}$ approx. (for $l=2$ both give $\frac{3}{4}$)

   (Note: conv. to 3-SAT doesn't work since clauses change)
3. Integrality gap for MAX-SAT (construct example with gap $= \frac{3}{4}$)

   $OPT = \frac{3}{4}$ LP-OPT

Lecture — Semi-Definite Programming — 13/10/23

Convex programming

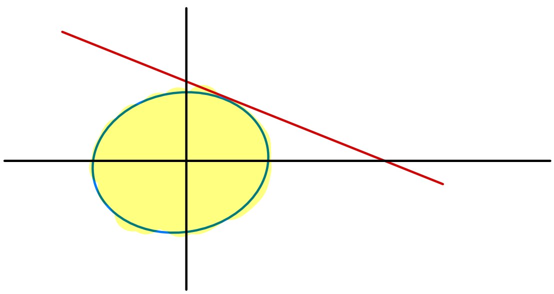
$\min f(x)$ — f is convex

$x \in \mathbb{R}^n$ — C is convex

$x \in C$

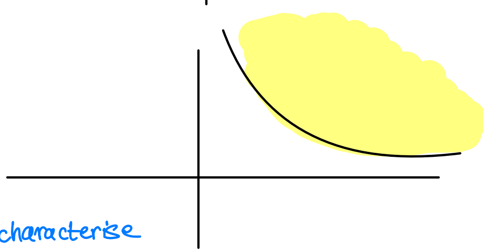
Ex: $2x+3y$

$x^2+y^2 \leq 1$



Ex: $xy \geq 1$, $x \geq 0$, $y \geq 0$

Sets described by polynomial constraints (semi-algebraic sets)

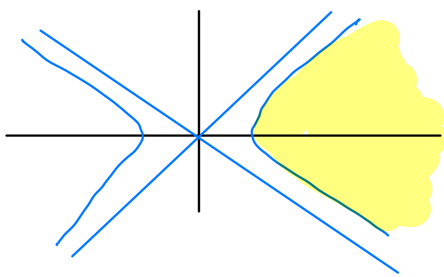


Degree 2 constraints: we know to characterise convex sets

Ex: $xy \leq 1, x \geq 0, y \geq 0$ (non-convex region)

Ex: $x^2-y^2 \geq 1$, $x \geq 0$ is convex,

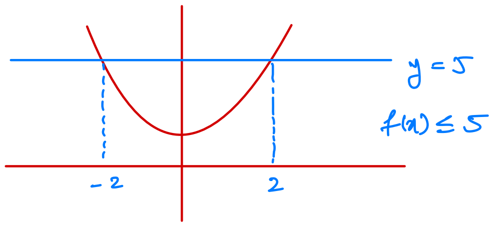
but $x^2-y^2 \leq 1, x \geq 0$ is not!



Claim: If $f(x)$ is a convex function then $K = \{x \in \mathbb{R}^n : f(x) \le \alpha\}$ is convex

Pf: Follows from def^n of a convex function.

Eg: $f(x) = x^2 + 1$ is convex.



Convex f^n

$$f(tx_1 + (1-t)x_2) \le t f(x_1) + (1-t) f(x_2)$$

$\forall x_1, x_2 \in X$

Examples of convex functions (Deg 2)

- $x^2 + 1$
- $1 - xy$ is convex in $x \ge 0$, $y \ge 0$
- $x^2 + y^2$
- $(x+y)^2 + (2x-y)^2$

Thm: Deg 2 function $f(x)$ is convex in $\mathbb{R}^n$ iff $f(x) = \sum_i (a_{i1}x_1 + \cdots + a_{in}x_n + b_i)^2 + c_1x_1 + \cdots + c_nx_n + b$

Proof: $(\Leftarrow)$

$f(x) = x^2$ then $\left(\frac{\alpha+\beta}{2}\right)^2 \le \frac{\alpha^2+\beta^2}{2}$ (AM $\le$ QM) and any other

Linear f^n, run same proof.

Claim: f_1 is convex, f_2 is convex then $f_1 + f_2$ is convex

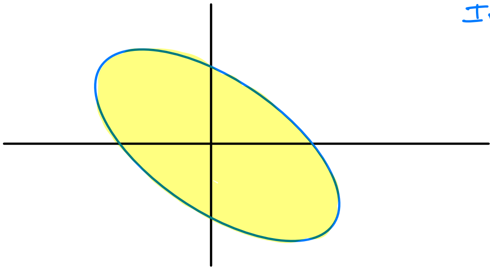
Pf: Follows from def^n.

Hence, any sum of squares of linear functions is convex

$(\Rightarrow)$

Let $f(x)$ be convex in $\mathbb{R}^2$.

$(x+y)^2 + (2x-y)^2 \leq 4$ must be convex by above claims.



In higher dimensions, this generalises to ellipsoids.

Using SDP, we can minimize convex f^n over SoS of linear functions

But, SDPs capture much more than this.

Linear Algebra Basics

$A \in \mathbb{R}^{n \times n}$ is symmetric iff $A = A^T$

eigenvalues. eigenvectors = columns of U

__Fact__: For any symmetric matrix A, $A = UDU^T$ where U is an orthonormal matrix over $\mathbb{R}$

any two cols are orthogonal, unit len. $\leftarrow$ $UU^T = I$ (or) $U^T = U^{-1}$

__Claim__: Eigenvalues of a symmetric matrix are real

$$Av = \lambda v, \quad \lambda \in \mathbb{C}, v \in \mathbb{C}^n$$

$$v^\dagger A v = \lambda v^\dagger v = \lambda \|v\|_2^2$$

$$(v^\dagger A v)^\dagger = \lambda^* \|v\|_2^2 = v^\dagger A^T v = v^\dagger A v = \lambda \|v\|_2^2$$

$$\Rightarrow \boxed{\lambda = \lambda^*} \quad \text{hence, } \lambda \in \mathbb{R}$$

__Example__: $\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = M$. if M has a real eigenvector, direction is preserved, but it rotates everything by $45°$.

So, no real eigenvector.

__Example__: $\begin{bmatrix} -1 & 0 \\ 0 & +1 \end{bmatrix} \rightarrow$ gives mirror image along y-axis.

__Positive Semidefinite Matrices__ : Symmetric matrix with non-negative eigenvalues.

__Ex__: $\begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$ is psd. ($\det A \geq 0$, $tr(A) \geq 0$). $\begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix}$ is not psd. $tr(A) = 0$ $\det(A) \neq 0$.

__Characterisations__ :

- $x^T A x \geq 0$, $x \in \mathbb{R}^n$
- All principal minors are non-negative (principal minor = det of same subset of rows, columns)
- $A = B^T B$, $B \in \mathbb{R}^{n' \times n}$ $\{n' \leq n\}$

that is, $A_{ij} = b_i^T b_j = \langle b_i, b_j \rangle$ ← useful in algorithms.

↑ inner product matrix

Semi-Definite Programs

$$x_{11}, x_{12}, x_{13}, \ldots x_{1n}, x_{22}, \ldots, x_{nn}$$

$\binom{n+1}{2}$ variables $X_{ij}, \; 1 \le i \le j \le n$

$$\max \; w^T X \quad , w \in \mathbb{R}^{\binom{n+1}{2}}$$

$$Cx = d \quad , \quad C \in \mathbb{R}^{k \times \binom{n+1}{2}} \quad , d \in \mathbb{R}^k$$

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \\ x_{12} & x_{22} & x_{23} & \cdots \\ x_{13} & x_{23} & \ddots & \\ & & & x_{nn} \end{bmatrix} \succeq 0 \quad \text{(p.s.d.)}$$

<u>Claim</u>: Set of $n \times n$ PSD matrices forms a convex cone

<u>Proof</u>: $A_1, A_2 \in PSD. \Rightarrow \lambda A_1 + (1-\lambda) A_2 \in PSD$ (add eigenvalues)

$\alpha \ge 0 \Rightarrow \alpha A \in PSD$ (mult. eigenvalues).

<u>Ex</u>: $\boxed{\begin{bmatrix} \boxed{x} & z \\ z & \boxed{y} \end{bmatrix}} \succeq 0 \Rightarrow \left.\begin{matrix} x \ge 0 \\ y \ge 0 \\ xy - z^2 \ge 0 \end{matrix}\right\}$ 2x2 p.s.d matrices cone. (convex region)

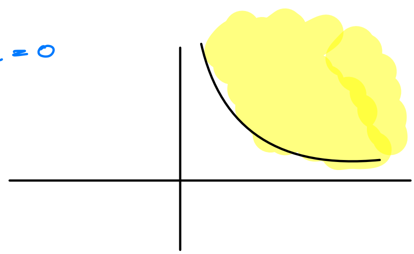
SDP is cutting this cone with a hyperplane.

<u>Ex</u>: Intersect $X \succeq 0$ with $z = 0$

$x \ge 0$

$y \ge 0$

$xy - 1 \ge 0$



<u>Ex</u>: $x^2 + y^2 \le 1$

MI:

$$\det \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix} = abc + 2hgf - g^2 b - f^2 a - h^2 c \ge 0$$

$$g^2 b + f^2 a + h^2 c \le abc + 2hgf$$

$$h = 0, a = 1, b = 1, c = 1$$

$a \ge 0, b \ge 0, c \ge 0$ ✓

$ab - h^2 \ge 0, \; bc - f^2 \ge 0, \; ac - g^2 \ge 0$ ✓

M2: $1-x^2-y^2 \geq 0$

$(1-x)(1+x) - y^2 \geq 0$

$\det \begin{bmatrix} 1-x & y \\ y & 1+x \end{bmatrix}$ $\qquad$ $\begin{bmatrix} 1-x & y \\ y & 1+x \end{bmatrix} \succeq 0$

Express this as proper form.

[HW] Express sum of squares as SDP, ellipsoid as SDP.

Sol^n: Use M1 above with different coefficients. for ellipsoid as SDP.

17/10/23

Lecture

Semi-definite program

$X_{11}, X_{12}, \ldots, X_{nn}$

$(X_{ij})_{i \leq j}$ $\qquad$ $\frac{n(n+1)}{2}$ variables

$$\min \sum_{i \leq j} w_{ij} X_{ij}$$

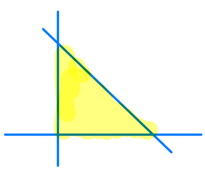
$$\left.\begin{aligned} &X \succeq 0 \quad \text{(psd cone)} \\ &\sum_{i \leq j} a_{h,i,j}\, x_{i,j} = b_h \quad \text{for } 1 \leq h \leq k \end{aligned}\right\} \text{convex set.}$$

Linear Programs

$x \geq 0$

$y \geq 0$

$x + y \leq 1$



$$\begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & 1-x-y \end{bmatrix} \succeq 0$$

$\downarrow$ std form

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{12} & X_{22} & X_{23} \\ \cdots & & X_{33} \end{bmatrix} \succeq 0$$

eliminate x, y $\left\{ \begin{aligned} &X_{11} = x, \; X_{33} = 1 - X_{11} - X_{22} \\ &X_{22} = y \\ &X_{33} = 1 - x - y \end{aligned} \right.$

$X_{12} = X_{13} = X_{23} = 0$

Another standard form

$x_1, x_2, \ldots, x_n \in \mathbb{R}$

$Ax = b$

$$\begin{bmatrix} x_1 - x_2 & 2x_3 + x_5 & \\ 2x_3 + x_5 & x_4 - x_5 & \\ & & \ddots \end{bmatrix} \succeq 0$$

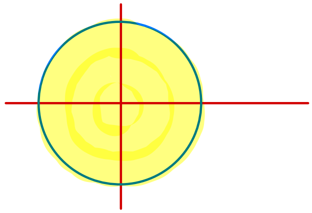
$x^2 + y^2 \leq 1$

$1 - x^2 - y^2 \geq 0$

$\begin{bmatrix} 1-x & y \\ y & 1+x \end{bmatrix} \succeq 0$

Gives $1 - x^2 - y^2 \geq 0$

$\left.\begin{matrix} 1 - x \geq 0 \\ 1 + x \geq 0 \end{matrix}\right\}$ redundant



Another method,

$\begin{bmatrix} 1 & x & y \\ x & 1 & 0 \\ y & 0 & 1 \end{bmatrix} \succeq 0$

$1 - x^2 \geq 0$

$1 - y^2 \geq 0$

$1 - x^2 - y^2 \geq 0$

Quadratic convex programs with quadratic constraints

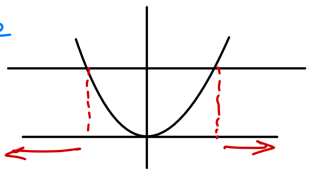
$\hookrightarrow$ Can be encoded as $\sum L_i^2 + L$

e.g. $(2x - 3y + 5)^2 + (3x - 2y - 1)^2 + 2x - y + 5 \leq 0$

which is convex, and can be encoded using SDP. [HW]

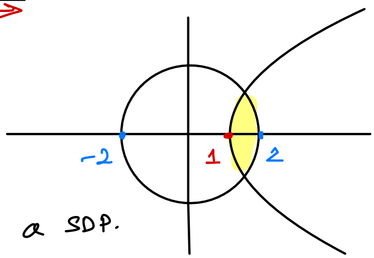
Note: for a convex function f, $f(x) \leq C$ is convex but $f(x) \geq C$ need not be convex.

$f(x) = x^2$



$f(x) \geq 4$

not convex! (not an interval)



$x^2 + y^2 \leq 4$

$y^2 \leq x^2 - 1$

$x \geq 0$

This region can be encoded as a SDP.

Trick: Put one set of constraints in each block

$\begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_n \end{bmatrix}$ is PSD $\Leftrightarrow \forall i \; A_i$ is PSD

$\begin{bmatrix} 2-x & y & & \\ y & 2+x & & \\ & & x-1 & y \\ & & y & x+1 \end{bmatrix} \succeq 0$

$x - 1 \geq 0 \Rightarrow x \geq 0$. So x is not required on diag.

Note: For left half just replace x with $-x$.

Recall : Matrix M is said to be PSD ($M \succeq 0$)

1. • if $v^T M v \geq 0$ for all $v \in \mathbb{R}^n$
2. • if $\text{Det}(M_{S,S}) \geq 0 \ \forall S \subseteq \{1,2,\ldots,n\}$
3. • all eigenvalues are non-negative
4. • $\exists u_1, u_2, \ldots, u_n \in \mathbb{R}^n$ s.t. $M_{ij} = u_i^T u_j$ ($M = U^T U$ iff M is psd)

$(4 \Rightarrow 1)$ $v^T M v = v^T U^T U v = (Uv)^T (Uv) \geq 0$.

$(1 \Rightarrow 3)$ $v^T M v \geq 0$. Take v as eigenvector. Then $v^T M v = \lambda \geq 0$.

$(3 \Rightarrow 4)$ $M = UDU^T = (U\sqrt{D})(\sqrt{D}U^T) = (U\sqrt{D})(U\sqrt{D})^T$

Alternate form of SDP

$x_{11}, x_{12}, \ldots, x_{nn} \in \mathbb{R}$

$$\sum_{i \leq j} w_{ij} x_{ij}$$

$$\sum a_{ij}^k x_{ij} = b_k$$

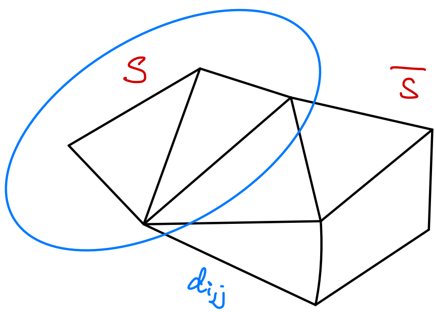
$\exists u_1 \ldots u_n$

$$x_{ij} = u_i^T u_j$$

MAX-CUT (NP-hard)

Graph with edge weights. Find a subset $S \subseteq V$ which maximizes

$$\sum_{e \in E(S, \bar{S})} w(e)$$



Randomized $\frac{1}{2}$-approx algorithm

Pick $v \in S$ with prob. $1/2$.
By linearity of expectation,

$$\mathbb{E}[S] = \frac{1}{2} \sum_e w_e \geq \frac{1}{2} \text{OPT}.$$

NP-hardness

Max-cut can solve maximum independent set problem

Linear Program

$$\max \sum w_{ij} x_{ij}$$

$x_{ij} \leq z_i + z_j$ (if both zero, $x_{ij} = 0$)

$x_{ij} \leq 2 - z_i - z_j$ (if both one, $x_{ij} = 0$)

$0 \leq z_i \leq 1$

$0 \leq x_{ij} \leq 1$

Rounding scheme

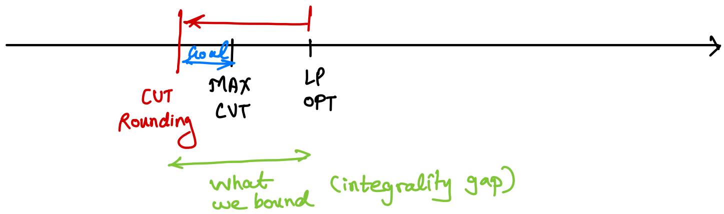
z^*, x^*

$z_i^* \geq 0.5$ then $i \in S$

$z_i < 0.5$ then $i \notin S$

Issue : Integrality Gap = 1/2 for this LP.

Example where MAX-CUT = $\frac{1}{2}$ LP-OPT [HW]



Using this LP, best hope is $\frac{1}{2}$-approximation

1995 : Goemans Williamson SDP based algorithm : 0.878 approx.

$$\max \sum_{i,j} w_{i,j} x_{i,j}$$

$$z_i \in \{-1, 1\}$$

$$x_{ij} = \begin{cases} 1 & \text{if } z_i \neq z_j \\ 0 & \text{if } z_i = z_j \end{cases} \rightarrow x_{ij} = \frac{1 - z_i z_j}{2}$$

i.e. $\max \sum_{i,j} w_{ij} \left(\frac{1 - z_i z_j}{2} \right)$

$$z_i \in \{-1, 1\}$$

n = no. of vertices, since $X = Z^T Z$, so, $Z \in \mathbb{R}^n$.

We now relax this,

Vector program

$$\boxed{\begin{array}{l} z_i \in \mathbb{R}^n \quad \langle z_i, z_i \rangle = 1 \\ \max \sum_{i,j} w_{ij} \left(\frac{1 - \langle z_i, z_j \rangle}{2} \right) \end{array}}$$

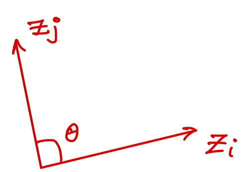
if opp, $\langle z_i, z_j \rangle = -1$
if same direction, $\langle z_i, z_j \rangle = +1$

Note : This is a relaxation as we can put $z_i' = (\pm 1, 0, \ldots, 0)^T$
Hence OPT $\geq$ Max-cut.

Up-next : Why this is an SDP and how we can get a good approximate cut.

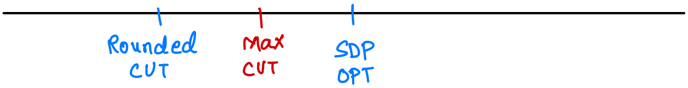
Note : Quadratic programs can't exactly be written as SDP. cuz
$u_1 u_2 = (u_1 \cdots 0)(u_2\, 0 \cdots 0)^T = u_1'^T u_2'$ but u_1' can admit several other solutions

$$x_{ij} = \frac{1 - \langle z_i, z_j \rangle}{2} = \frac{1 - \cos\theta}{2}$$



Lecture 20/10/23

Claim : Any feasible solution of the LP gives a feasible solution of SDP.



(fractional sol^n. $\langle z_i, z_j \rangle$ can be fractional)

1. Use some SDP solver to find (x^*, z^*)
2. Rounding scheme $(x^*, z^*) \rightsquigarrow$ cut.

M is PSD iff $\exists\, u_1, u_2, \dots, u_n \in \mathbb{R}^n$ s.t. $u_i^T u_j = M_{ij}$

$$\max \sum_{i \le j} w_{ij}\, x_{ij}$$

$$z_i^T z_j = 1 - 2x_{ij}$$

$$z_i^T z_i = 1$$

$$i \rightarrow \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & 1-2x_{ij} & & \ddots & \\ & \uparrow j & & & 1 \end{bmatrix} \succeq 0$$

Note : Why not go for more than n dimensions ? $\rightarrow$ no new information

Note : If $z_i \in \mathbb{R}^2$, $Z^T Z$ is PSD, but this does not cover the space of $n \times n$ PSD matrices $X \succeq 0$

E.g. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \succeq 0$ but $u_1, u_2, u_3 \in \mathbb{R}^2$ s.t. $\langle u_i, u_j \rangle = \mathbb{1}_{i=j}$ are not possible because $\mathbb{R}^2$ can't have 3 orthonormal vectors.

E.g. $u_1, u_2, u_3 \in \mathbb{R}^2$

$$M_{ij} = u_i^T u_j$$

$$\left.\begin{matrix} M \succeq 0 \\ \text{Rank} \le 2 \end{matrix}\right\}$$

$$\begin{bmatrix} u_1^T \\ u_2^T \\ u_3^T \end{bmatrix}_{3\times 2} \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}_{2\times 3} \rightarrow \mathbb{R}^{3\times 3}$$

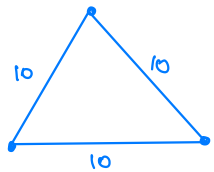
But $a_1 A_1 + a_2 A_2 = A_3$ $\quad A_i \in \mathbb{R}^{3\times 3}$

rank $A_i \le 2$ can have A_3, rank $= 3$. Hence, this is not a convex set!

Claim : Feasible solutions for VP $\Leftrightarrow$ feasible solutions for SDP

Proof: HW (easy enough)

Now, how to do this rounding? $(x^*, z^*) \rightsquigarrow$ CUT.

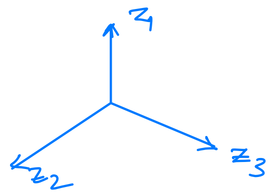


$\max \ 10(x_{12} + x_{23} + x_{13})$

$z_1, z_2, z_3 \in \mathbb{R}^3$

$x_{12} = \frac{1 - \underbrace{z_1^T z_2}_{\text{minimize}}}{2}$

$x_{12} = x_{13} = x_{23} = 3/4$



$10(x_{12} + x_{23} + x_{13}) = 90/4 = 22.5$, OPT $= 20$

Ideas : $x_{ij} \geq 0.5$ then round to 1 (but 1,1,1 is not a valid CUT)

Sol^n: Round using z_i

Randomized rounding procedure

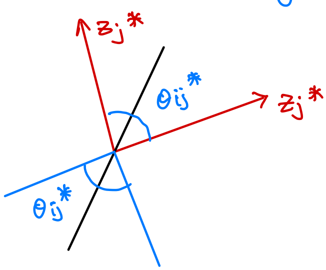
choose random hyperplane $h \in \mathbb{R}^n$

$h^T z_i \geq 0$ vertex i on left side

$h^T z_i < 0$ vertex i on right side

say $\sum w_{ij} \left(\frac{1 - \langle z_i, z_j \rangle}{2}\right) = \sum w_{ij} \left(\frac{1 - \cos\theta_{ij}^*}{2}\right)$

After rounding, use linearity of expectation to find $p(h_{ij}$ is a cut edge$)$



$P(z_i^*, z_j^*$ are on different sides of hyperplane$) = \frac{\theta_{ij}^*}{\pi}$

$\mathbb{E}[\text{cut}] = \sum w_{ij} \, P(\text{edge } (i,j) \text{ is in cut})$

$= \sum w_{ij} \frac{\theta_{ij}^*}{\pi}$

We want $\frac{\theta_{ij}^*}{\pi} \geq \alpha \left(\frac{1 - \cos\theta_{ij}^*}{2}\right)$ for a good approximation ratio

So, $\min_\theta \frac{2\theta}{(1-\cos\theta)\pi} = \alpha$ gives $\alpha = 0.878$

Hence, we get a 0.878 - approximation algorithm ∎

Note : 0.94 - approximation is NP-hard.

< Simplex lecture missed > : read reference.

Lecture

31/10/23

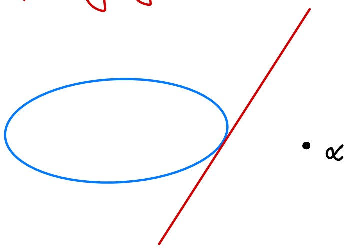
Ellipsoid Algorithm (1979)

Also works for more general convex programs?

- Optimization : $K \leftarrow$ convex set , $f(x)$ linear function
  min $f(x)$ over K
- Feasibility : Is K non-empty
- Membership : Given $x \in \mathbb{R}^n$ and convex set K does $x \in K$?
- Separation : If $x \notin K$ then give a separating hyperplane

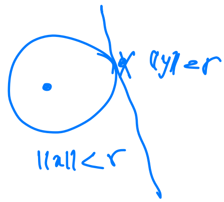
Separating Hyperplane

a, b s.t. $a^T x \le b$ for all $x \in K$



Separating Hyperplane Thm (guarantees such a plane exists) for closed convex sets

$$\text{Optimization} \underset{\text{(binary search)}}{\le} \text{Feasibility} \underbrace{\le \text{Separation}}_{\text{Ellipsoid algorithm}}$$



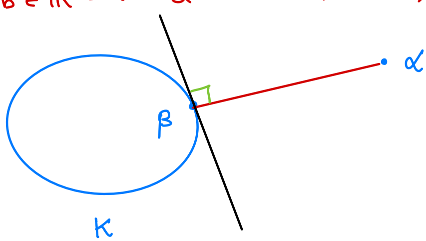
Que : Algorithm for optimization $\Rightarrow$ Algorithm for separation ?

LPs $\leftarrow$ separation oracle $\equiv$ constraint violated

Separating Hyperplane Thm

$K \subseteq \mathbb{R}^n$ is a closed , convex set , bounded (compact) and $\alpha \in \mathbb{R}^n$. If $\alpha \notin K$ then $\exists\, a \in \mathbb{R}^n, b \in \mathbb{R}$ s.t. $a^T x \le b \ \forall x \in K$, $a^T \alpha > b$.

Proof :



Note : More generally, two disjoint compact sets can be separated by a plane.

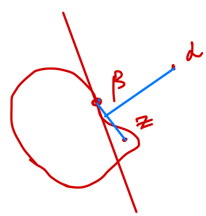
$$H : (\alpha - \beta)^T x - (\alpha - \beta)^T \beta = 0$$

Putting $x = \alpha$, $H(\alpha) = (\alpha - \beta)^T (\alpha - \beta) > 0$ ($\alpha \neq \beta$ since $\alpha \notin K, \beta \in K$)

want $\forall x \in K$, $(\alpha - \beta)^T x - (\alpha - \beta)^T \beta \le 0$

For contradiction, $\exists\, z \in K$

$$(\alpha - \beta)^T z - (\alpha - \beta)^T \beta > 0$$



Define $y = \beta + \varepsilon(z-\beta)$

$$(\alpha-y)^T(\alpha-y) = ((\alpha-\beta) - \varepsilon(z-\beta))^T((\alpha-\beta) - \varepsilon(z-\beta))$$

$$= (\alpha-\beta)^T(\alpha-\beta) - 2\varepsilon\underbrace{(z-\beta)^T(\alpha-\beta)}_{>0} + \varepsilon^2\underbrace{(z-\beta)^T(z-\beta)}_{>0}$$

Hence take ε small enough such that $2(z-\beta)^T(\alpha-\beta) > (z-\beta)^T(z-\beta)\varepsilon$

$\Rightarrow \|\alpha - y\| < \|\alpha - \beta\|$

Hence, we get a contradiction, since $y \in K$ but $d(y,\alpha) < d(\beta,\alpha)$ ∎

Note: Separating Hyperplane $\Rightarrow$ Farkas Lemma

$Ax = b$, $x \geq 0$ not feasible $\Rightarrow \exists y$ $A^Ty \geq 0$, $b^Ty < 0$

$S = \{z : z = Ax, x \geq 0\}$ and in-feasible $b \notin S$ then use separating hyperplane theorem.

convex, closed

Definition : Ellipsoid

Start with a Ball

- Stretch along axes
- Rotate
- Translate

$B(0,1)$ = ball with center $\bar{0}$, radius 1.

$E = \{Lx : x \in B(0,1)\}$ for some L $n\times n$ matrix

centred at zero

$E = \{Lx + c : x \in B(0,1)\}$ is centred at c.

$$E = \{Lx : x^Tx \leq 1\}$$

$$= \{y = Lx : x^Tx \leq 1\}$$

$$= \{y : y^T\underbrace{(L^{-1})^TL^{-1}}_{Q}y \leq 1\}$$

$$= \{y : y^TQy \leq 1\} \text{ where } Q \text{ is P.S.D}$$

$$E(c,Q) = \{y : (y-c)^TQ(y-c) \leq 1\}$$

$$\mathrm{Vol}(E) = |\det(L)|\,\mathrm{vol}(B_n(0,1))$$

-ve det corresponds to a "flip".

Note: Even in n-dim only one flip (distinguishes)

$\mathrm{Vol}(B_n(0,1))$ = constant

$$V_{2k}(R) = \frac{\pi^k}{k!}R^{2k}$$

$$V_{2k+1}(R) = \frac{2k!\,(4\pi)^k}{(2k+1)!}R^{2k+1}$$

Ellipsoid Algorithm

Input: a separation oracle for $K \subseteq \mathbb{R}^n$

Output: a point $x \in K$

Requirements :

- $\exists R > 0$ s.t. $K \subseteq B(0, R)$ (not too far)
- $\exists r > 0$ s.t. $B(c, r) \subseteq K$ (not too small) and full dimensional

(full dimensional)

Equivalently,

$\exists r, R$ s.t. $B(c,r) \subseteq K \cap B(0,R)$

Running time : $O(n^2 \log \frac{R}{r})$

For LP, $Ax \leq b$

Discussed before $\exists x^*$ such that bit size is not too large.
(corner)

$x^* = (A')^{-1} b'$, determinants can be bounded,

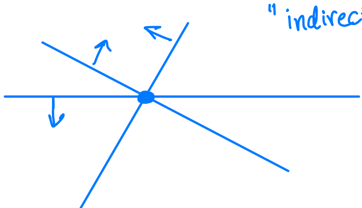
So $\exists$ a solution not too far

$R \leftarrow \exp(n,$ bit-size of $A, b)$

Now, feasible set can be in a subsp or a single point $\Rightarrow$ $B_n(c, r) \subseteq K$ does not exist

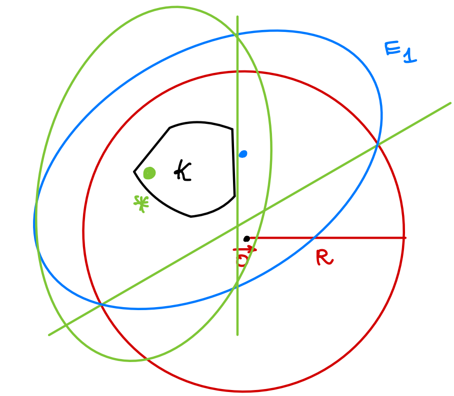
Solution : 1. Eliminate all equations (gaussian elimination)

Can have "indirect" smaller dimensions.



2. Random Perturbation $(x_1 + x_2 \geq 5 \rightarrow x_1 + x_2 \geq 5 - \varepsilon)$

Lecture



Separation oracle

Given a point P,

either a. $P \notin K$, give a, b such that $a^T P > b$, $a^T x \le b \ \forall x \in K$

or b. $P \in K$

Ellipsoid Algorithm

Input: a separation oracle for $K \subseteq \mathbb{R}^n$

Output: a point $\alpha \in K$

$E_0 = B(0, R)$

while ($vol(E_t) \ge vol(B(c, r))$)

t^{th} iteration: Let c_t be center of E_t

check whether $c_t \in K$ (sep-oracle)

Yes → output c_t

No → oracle gives a, b s.t. $a^T x \le b \ \forall x \in K$

Compute smallest ellipsoid E_{t+1} s.t.

$E_{t+1} \supseteq \underbrace{E_t \cap \{x : a^T x \le b\}}_{\text{Half-ellipsoid}}$

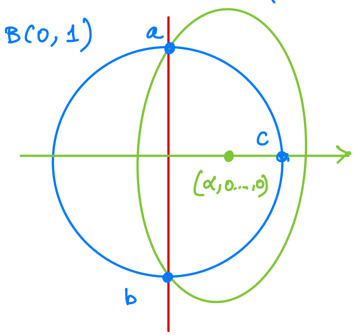
<u>Requirements</u>:

- $\exists R > 0$ s.t. $K \subseteq B(0, R)$
- $\exists r > 0$ s.t. $B(c, r) \subseteq K$ for some c (full dimensional)

Running time: $O(n^2 \log \frac{R}{r})$

<u>Invariant</u>: $K \subseteq E_t$

Construct new ellipsoid containing intx of ellipsoid, half-plane.



If sep. hyperplane doesn't pass through center, one half has smaller volume, so w.l.o.g hyperplane passes through center.

$x_1^2 + \cdots + x_n^2 \le 1$

$x_1 \ge 0$

New center $(\alpha, 0, \ldots, 0)$

$$\frac{(x_1 - \alpha)^2}{\gamma^2} + \frac{x_2^2}{\beta^2} + \cdots + \frac{x_n^2}{\beta^2} \le 1$$

(Due to symmetry $\beta_2 = \cdots = \beta_n = \beta$, $\beta_1 = \gamma$)

Ellipsoid should pass through $(0, 1, 0, \ldots 0)$ .... $(0, \ldots, 1)$
$(0, -1, 0, \ldots 0)$ $(0, \ldots, -1)$
and $(1, 0, \ldots, 0)$

$\Rightarrow \dfrac{(1-\alpha)^2}{r^2} = 1 \Rightarrow r^2 = (1-\alpha)^2$ Putting $(1, 0, \ldots 0)$

(Let $\alpha < 1$)

$\boxed{r = 1-\alpha}$

Put $0, 1, 0, \ldots, 0$

$\Rightarrow \dfrac{\alpha^2}{r^2} + \dfrac{1}{\beta^2} = 1 \Rightarrow \boxed{\beta^2 = \dfrac{1}{1 - \frac{\alpha^2}{(1-\alpha)^2}} = \dfrac{(1-\alpha)^2}{1-2\alpha}}$

For any $0 \le \alpha < 1/2$

$E_\alpha : \dfrac{(x_1-\alpha)^2}{(1-\alpha)^2} + (x_2^2 + \cdots + x_n^2)\left(\dfrac{1-2\alpha}{(1-\alpha)^2}\right) \le 1$

Note: $\alpha = 0$ gives orignal ball back.

Claim: E contains $B(0,1) \cap x_1 \ge 0$ [HW]

We want to minimize vol^m of E_α

$$\mathrm{Vol}(E_\alpha) = \mathrm{Vol}(B(0,1))\,(1-\alpha)\left\{\frac{(1-\alpha)}{\sqrt{1-2\alpha}}\right\}^{n-1}$$

$$\frac{(1-2\alpha)\,n\cancel{(1-\alpha)}^{n-1}(-1) + (1-\alpha)^{n}\left(\frac{n-1}{\cancel{2}}\right)\cancel{(1-2\alpha)}^{\frac{n-3}{2}}(\cancel{-2})}{(1-2\alpha)^{n-1}} = 0$$

$(1-\alpha)(n-1) = n(1-2\alpha)$

$n-1-\alpha(n-1) = n - 2n\alpha$

$\alpha(n+1) = 1$

$\boxed{\alpha = 1/n+1}$

minimize $\dfrac{(1-\alpha)^n}{(1-2\alpha)^{n-1/2}}$, we get $\alpha = \dfrac{1}{n+1}$

That is, $\mathrm{Vol}(B(0,1))\ \dfrac{\left(\frac{n}{n+1}\right)^n}{\left(\frac{n-1}{n+1}\right)^{\frac{n-1}{2}}} = \dfrac{n^n}{(n+1)^{\frac{n+1}{2}}(n-1)^{\frac{n-1}{2}}} < 1$ (by AM-GM)

Note:

If required:

- Shift the center
- Rotate (hyperplane inclined)
- Scaling of axis

to convert E_t to $B(0,1)$ find E_{t+1}, convert back to orignal system.

Ratio works for any ellipsoid.

Make T iterations

$$\left[\frac{n^n}{(n^2-1)^{\frac{n-1}{2}}(n+1)}\right]^T = \left(\frac{n}{n+1}\left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}}\right)^T = \frac{n}{n+1}\left(1+\frac{1}{n^2-1}\right)^{\frac{n-1}{2}}$$

$$\le \frac{n}{n+1}\, e^{\frac{1}{n^2-1}\cdot\frac{n-1}{2}} \qquad \{1+y \le e^y\}$$

$$= \frac{n}{n+1}\, e^{\frac{1}{2(n+1)}} \qquad \{1-y \le e^{-y}\}$$

$$\le e^{-\frac{1}{n+1}}\, e^{\frac{1}{2(n+1)}} = e^{-\frac{1}{2(n+1)}}$$

Hence, after $2(n+1)$ iterations, volume decreases by factor of e^{-1} (atleast)

Initial vol^m $\sim R^n$, final $\sim r^n$

Hence, $\#$iterations $\leq \log\left(\frac{R}{r}\right)^n \cdot 2(n+1)$

$$\leq O\left(n^2 \log \frac{R}{r}\right)$$

Linear Program

$Ax \leq b$ $\quad R = ?$

Corner: $A'x = b'$. Estimate how large entries in A'^{-1} can be, gives estimate of x.

Entries of A^{-1} can be much larger than those of A. $(c \rightarrow c^n)$, that gives R.

Idea: Take 2 different corners, and show their distance is large enough for r.

Homework:

Steiner Tree

$$\min \sum w_e x_e$$
$$x_c \geq 0$$
$$\sum_{e \in \delta(S)} x_e \geq 1$$

$S \leftarrow$ terminal separating cut.

Q. Solving this LP using ellipsoid algorithm, design a separation oracle for this LP.

Hint: Use min s-t cut algorithm, to somehow check all constraints.

Up next: Interior point methods.

Lecture — Algorithms for LPs / SDPs — 7/11/23

- Ellipsoid algorithm

  Separation oracle for SDPs

  $$X \succeq 0$$

  if X is not PSD, then we need a separating hyperplane

  find a vector $v \in \mathbb{R}^n$ s.t. $v^T X v < 0$

Test for PSD :

- Do gaussian elimination (two-sided). Same operation on rows, columns to reach a diagonal matrix.

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 2 \\ 0 & -3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & -3 \end{pmatrix}$$

$R_2 \leftarrow R_2 - 2R_1$

$C_2 \leftarrow C_2 - 2C_1$

Claim: M is p.s.d iff diagonal entries are non-negative. [HW]

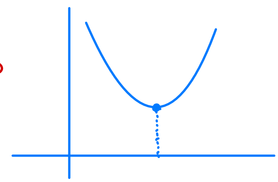
$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} & \\ & \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}^T \quad \text{So, } D = L M L^T$$

row vecs of L give $v^T M v = D_{vv} < 0$

Interior Point Methods (1984) {Karmarkar}

Unconstrained Convex Minimization

$\min_{x \in \mathbb{R}^n} f(x)$ minimized at a point x^*, where $\nabla f(x^*) = 0$



$$\nabla f = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \cdots \\ \partial f / \partial x_n \end{pmatrix}$$

Naive method : Solve $\nabla f = 0$ as a polynomial system

Might be complicated.

Iterative method : Move closer to the optimum

e.g. gradient descent $x_{t+1} \leftarrow x_t - \gamma \nabla f(x_t)$

γ ← step size

$f(x) \approx f(x_0) + \nabla f(x)^T (x - x_0)$ [if x is close to x_0]

we want to choose direction which min. $f(x) - f(x_0)$, so we choose $x - x_0$ in direction $-\nabla f(x)$.

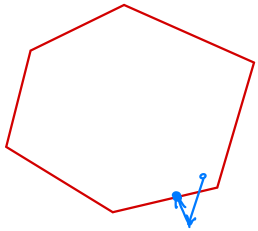
No. of iterations $= O(1/\varepsilon)$ where $|f(x) - f(x^*)| < \varepsilon$ (x gets ε-close to x^*)

ℓ-bits of precision, then $\varepsilon = \frac{1}{2^\ell}$, i.e. #iterations $= O(2^\ell)$. So, it's pseudo-polynomial

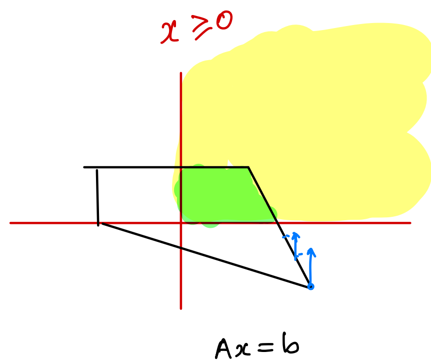
Constrained Minimization

$$\min w^T x$$
$$x \geq 0$$
$$Ax = b$$

Projected Gradient descent : Project the current point to the feasible region



Feasibility $\left.\begin{matrix} Ax = b \\ x \geq 0 \end{matrix}\right\}$ feasible or not



$$Q = \{x : x \geq 0\}$$
$$d_Q^2(x) = \text{distance}^2(x, Q) = \sum_{i : x_i < 0} x_i^2$$

$$\min \underbrace{d_Q^2}_{\text{convex } f^n.}(x) \text{ over } Ax = b$$

Claim : $\min d_Q^2(x) = 0$ iff $\begin{matrix} Ax = b \\ x \geq 0 \end{matrix}$ is feasible.

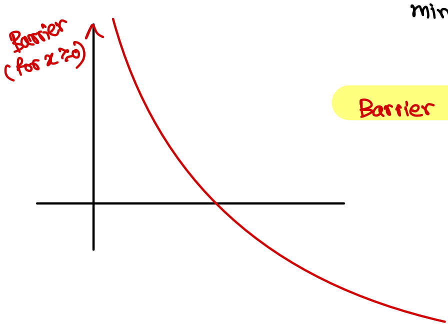
$$\nabla d_Q(x)_i = \begin{cases} 2x_i & \text{if } x_i < 0 \\ 0 & \text{otherwise}. \end{cases}$$

This is pseudo-polynomial time $O(2^l)$.

Constrained $\longrightarrow$ Unconstrained.

$$\min 2x + 3$$
$$x \geq 0$$

Barrier function.



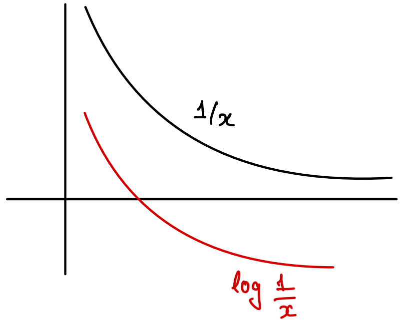
<u>Idea</u> : $\min 2x + 3 + B(x)$

$B(x) \to \infty$ as $x \to 0^+$, hence, we never go to $x < 0$.

So, we add a barrier for every constraint and then solve an unconstrained optimization problem.

Barrier function

- Go to ∞ as we move closer to the boundary
- Convex function inside the feasible region
- Decrease as we move away from the boundary



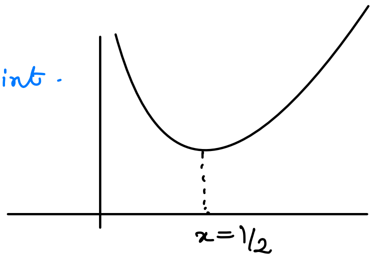
As it turns out, $\log \frac{1}{x}$ is the right choice.

Slow growing functions are good, since they allow us to move closer to the boundary.

Now, min $2x + 3 - \log x$

$$\frac{df}{dx} = 2 - \frac{1}{x} = 0$$

$x = 1/2$ is the minimization point.



Clearly $x = 1/2$ is not "correct", but it is close to 0.

In general, for every constraint

$$a_i^T x \le b_i \quad , \quad b_i - a_i^T x \ge 0 \quad \text{for } 1 \le i \le k$$

So, min $\eta\, w^T x - \sum_{i=1}^{k} \log (b_i - a_i^T x)$

where η decides weightage for the orignal objective function.

min $w^T x$

$a_i^T x \le b_i$ for $1 \le i \le k$

x^*

min $\eta w^T x - \sum_{i=1}^{k} \log (b_i - a_i^T x)$

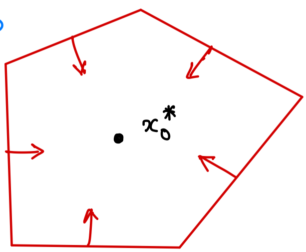
x_η^*

If $\eta \to$ large, $x_\eta^* \to x^*$.

So, if we want x ε-close to x^*, we choose appropriate η

If $\eta = 0$

x_0^* is some center of polytope



Claim : $\eta \to \infty$, then $w^T x_\eta^* \longrightarrow w^T x^*$

Proof: Given $\varepsilon > 0$, s.t. $|w^T x_\eta^* - w^T x^*| < \varepsilon$

$$\phi_\eta(x) = \eta w^T x - \sum_{i=1}^{k} \log(b_i - a_i^T x)$$

$$\nabla \phi_\eta(x) = \eta w - \sum_{i=1}^{k} \frac{1}{b_i - a_i^T x}(-a_i) = 0$$

$$\Rightarrow \quad \sum_{i=1}^{k} \frac{a_i}{b_i - a_i^T x_\eta^*} = -\eta w$$

Then, $$w^T x^* = \frac{1}{\eta} \sum_{i=1}^{k} \frac{-a_i^T x^*}{b_i - a_i^T x_\eta^*} \geq \frac{1}{\eta} \sum_{i=1}^{k} \frac{-b_i}{b_i - a_i^T x_\eta^*}$$

$$w^T x_\eta^* = \frac{1}{\eta} \sum_i \frac{-a_i^T x_\eta^*}{b_i - a_i^T x_\eta^*}$$

Hence, $$w^T x_\eta^* - w^T x^* \leq \frac{1}{\eta} \sum_{i=1}^{k} \frac{b_i - a_i^T x_\eta^*}{b_i - a_i^T x_\eta^*}$$

$$\leq \frac{k}{\eta}$$

Choose $\eta = k/\varepsilon$, to get $w^T x_\eta^* - w^T x^* < \varepsilon$

Hence,
$\min \frac{k}{\varepsilon} w^T x - \sum_{i=1}^{k} \log(b_i - a_i^T x)$ is ε-close to $\min w^T x$ s.t. $a_i^T x \leq b_i$