Approximation Algorithms

Book - Shmoys & Williamson

NP - hard optimisation problems

An algorithm is polynomial time if its running time is bounded by a polynomial function of the input size.

Primality : Is n a prime?

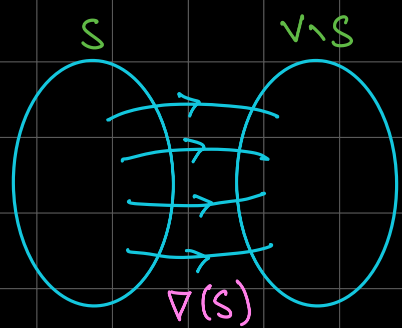
Polynomial time : $O(\log^k n)$

input size = $\log n$ ← bits

min-cut

$G = (V, E)$ $\quad c : E \rightarrow \mathbb{R}^+$

$c(S, \bar{S}) = \sum_{e \in \nabla(S)} c(e)$



[OPT] : What is the cost of the minimum cut?

[Decision] : Is there a cut of cost $\leq K$?

Vertex Cover : is a subset of vertices $S \subseteq V$ s.t. every edge in E is incident to a vertex in S

Min V.C. : Find the minimum vertex cover in G

≡ Art Gallery Problem

- min guards who can watch over all the galleries

Greedy :

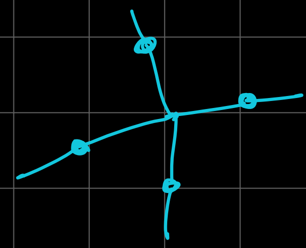
repeat :

Pick the vertex with max degree

Remove vertex & adjacent edges

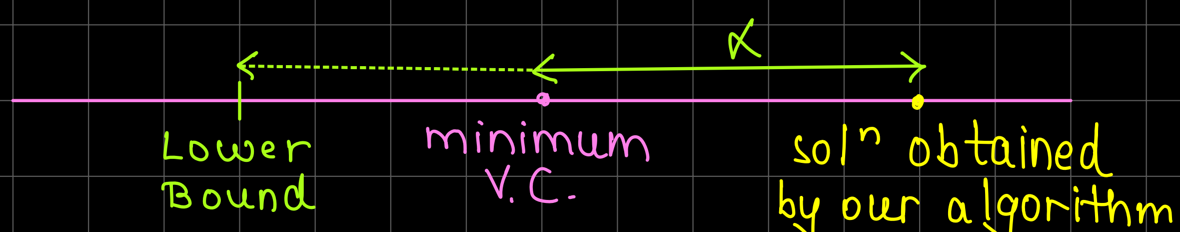
until :

no edges remain



Vertex cover returned by the algorithm is the set of vertices picked at each step

> H.W. : Find an example where the greedy algorithm for V.C. gives a solution which is much larger than the optimum solution.



Our algorithm A has an approximation factor $\alpha\,(>1)$ if $\forall$ instances, I, $A(I) \leq \alpha\, OPT(I)$

solution returned by algorithm A on instance I

optimum vertex cover for instance I

Lower Bound : a surrogate for the optimum value

Matching is a subset of edges $M \subseteq E$ s.t. M is indep

A set of edges is independent if no 2 edges in the set are incident at the same vertex

A matching is maximal if it cannot be augmented

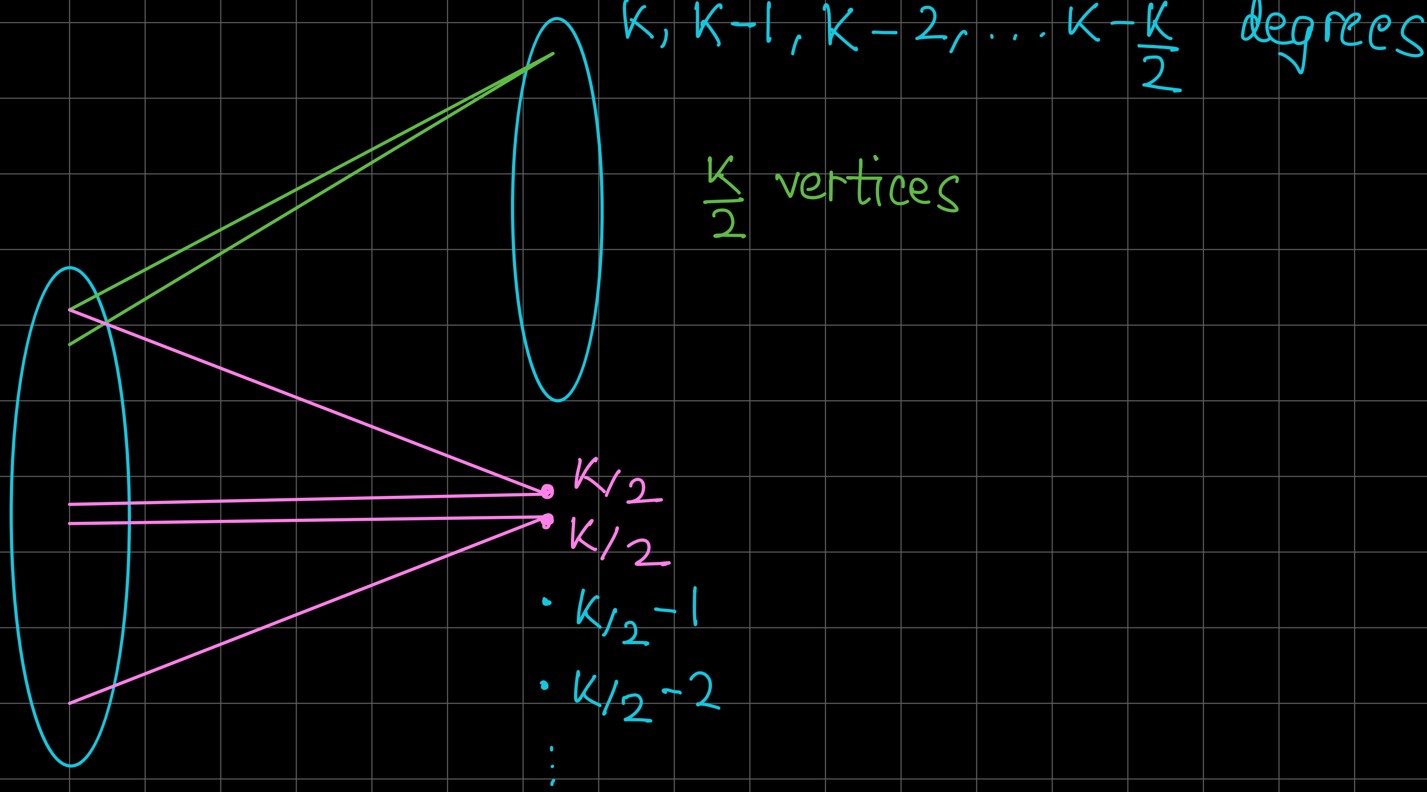
If G has a matching M & S is a vertex cover G then $|S| \geq |M|$

1. Find a maximal matching M in G
2. Let S be the vertices which are endpoints of edges in M
3. Output S

$$|S| = 2|M| \leq 2\,OPT$$

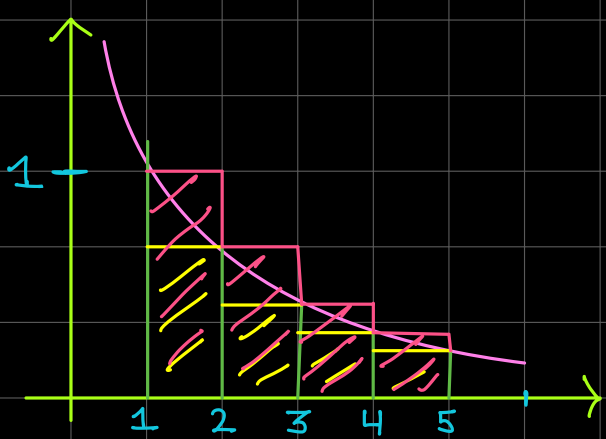
No better algorithm is known!

This gives a vertex cover. Suppose it doesn't, then $\exists$ an edge which does not share any vertex with the edges in the matching but then this edge should have been in the matching. A contradiction



$$\text{Total vertices} = \frac{K}{2} + \left(\frac{K}{2}-\frac{K}{3}\right)2 + \left(\frac{K}{3}-\frac{K}{4}\right)3 + \ldots$$

$$= \frac{K}{2} + \frac{K}{3} + \frac{K}{4} + \ldots \approx K\log K$$



yellow

$$\frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{K} < \log K$$

red

$$1 + \frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{K} > \log K$$

$$\log K - 1 < \frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{K} < \log K$$

$$\boxed{\log K < H_K < \log K + 1}$$

SETCOVER problem

$U = \{e_1, e_2, \dots e_m\}$

$S_i \subseteq U, \quad 1 \leq i \leq n$

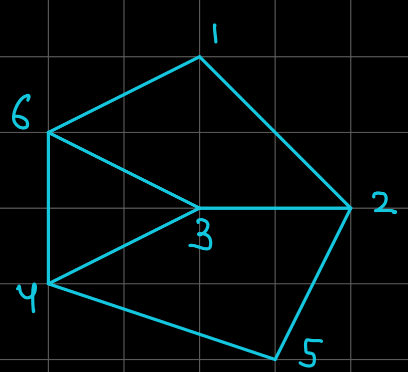
Find the smallest number of sets
$\mathcal{S} = \{S_i, i \in [n]\}$ whose union is U

Greedy Algorithm for Set Cover:

$U' = U$
repeat
  pick set $S \in \mathcal{S}$ which covers most no. of elements in U'
  set of all uncovered elements
  $U' = U \setminus S$
until $U' = \phi$

instance of VC $\longrightarrow$ instance of SC

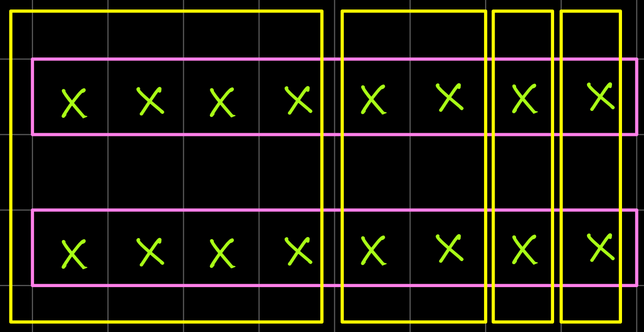
elements are edges.
sets are edges incident on each vertex



$U = \{(1,2), (2,3), (2,5), (5,4), (3,4), (4,6), (3,6), (1,6)\}$

$S_1 = \{(1,2), (1,6)\} \dots$ for all vertices

$S_i = \{e \in E \mid i \text{ is an endpoint of } E\}$



Greedy chooses yellow
optimal is pink

extend to 16, 32 . . .

optimal is 2
greedy chooses $\log_4 n$

v_i is the volume (non-negative) of element i such that

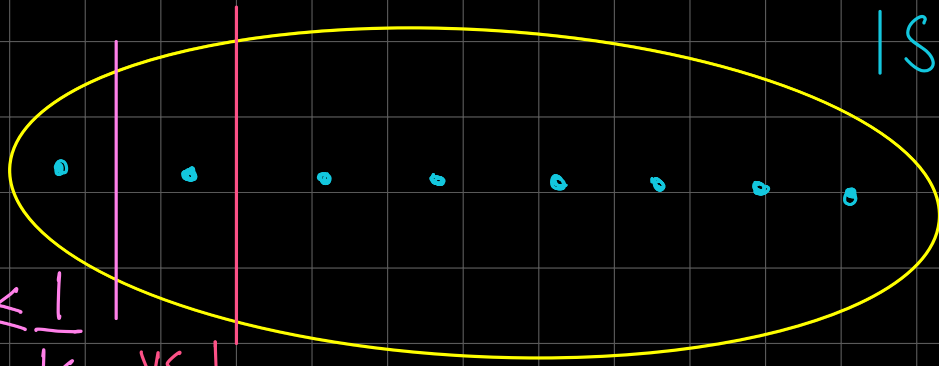
for any $S_j \in \mathcal{S}$, $\sum_{e_i \in S_j} v_i \leq 1$

$\sum_i v_i$ is a lower bound on optimum

> give every element of U' covered by S a volume of $\frac{1}{|S \cap U'|}$

$\sum v_i$ = # sets picked by greedy

$|S_i| = k$



$v \le \frac{1}{k}$ $\quad$ $v \le \frac{1}{k-1}$

The greedy algo chose some other set $\Rightarrow$ volume assigned must be $\le \frac{1}{k}$ (the other set has size atleast k to be picked)

$\forall S_j \in \mathcal{S}, \quad \sum_{e_i \in S_j} v_i \le$ $\qquad$ $\ln|S_j| \le \ln(m)$

> give every element of U' covered by S a volume of $\frac{1}{|S \cap U'|} \times \frac{1}{\ln(m)}$

$$\sum_i v_i = \frac{\#\text{ of sets picked by greedy}}{\ln(m)} \le OPT$$

Since $\sum v_i$ is a lower bound

Scheduling: Minimizing makespan on parallel (identical) machines

m machines, n jobs : p_j, $1 \le j \le n$

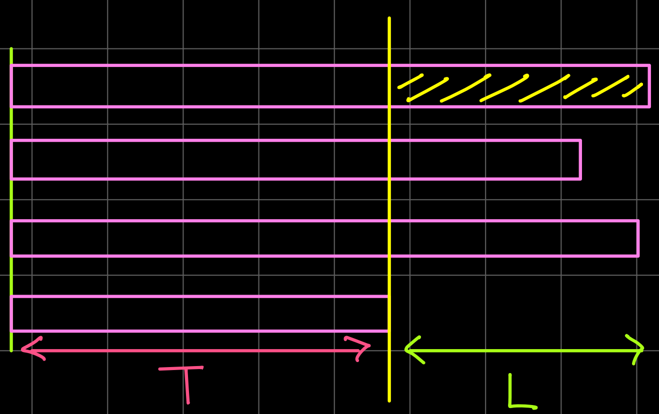
makespan : time at which all jobs complete

Graham : list scheduling

Consider jobs in arbitrary order
Assign job to least loaded machine
} Greedy Algorithm

4, 5, 2, 3, 6, 7, 8, 2, 9, 4, 6, 1, 3

m4	3 + 7 + 6 = 16
m3	2 + 6 + 4 + 3 = 15
m2	5 + 2 + 9 = 16
m1	4 + 8 + 1 = 13



$L \le OPT$ — ① ($L \le \max(p_i) \le OPT$ otherwise we would have scheduled it on the empty machine)

$\sum p_i \le mT$ - from the greedy algorithm

$\frac{1}{m} \sum p_i \le OPT$ — ② (obvious from construction of T and L)

$M = T + L \le OPT + OPT \le 2 \cdot OPT$

Network Design Problem

Steiner Tree Problem

Given n-points in the 2-D plane, find the cheapest way of connecting these points

$G=(V,E)$, $C: E \to \mathbb{R}^+$, $T \subseteq V$ is the set of terminals

Find a minimum length subgraph which connects all the terminal vertices.

Non-terminal vertices are also called Steiner Vertices

NOTE:

- We can assume C satisfies the Δ inequality.
- This is not required in the problem but we can always assume that those not satisfying will never be a part of the solution.

- The solution is always a tree (if some solution has a cycle then we can remove any edge and have a subgraph of lesser cost which still spans T)

Trivial Cases:

- If $T=V$, the solution is a MST
- If T is 2 nodes, the solution is the shortest path

Metric Steiner Tree:
Restriction of Steiner Tree where the graph G is complete and the cost function is metric i.e. $u, v, w \in V$, $c(u,w) \leq c(u,v) + c(v,w)$

Algorithm: Compute a Minimum Spanning Tree in the subgraph $G[T]$ induced by the terminal set T.

Theorem: This is a 2-approximation for Metric Steiner Tree

Proof:
in the original graph G
- Consider an optimal Steiner Tree B^* i.e. $C(B^*) = OPT$
- Duplicate all edges of B^* $\Rightarrow$ Eulerian (multi-) graph B' with cost $c(B') = 2 \cdot OPT$
- Find an Eulerian Tour T' in $B' \Rightarrow c(T') = c(B') = 2 \cdot OPT$

- Find a Hamitonian Path H in $G[T]$ by "short-cutting" Steiner vertices and already visited terminals (in the Eulerian Tour)

$\Rightarrow c(H) \leq c(T') = 2 \cdot OPT$ since G is metric

MST B of $G[T]$ has cost $c(B) \leq c(H) \leq 2 \cdot OPT$
since H is a spanning tree of $G[T]$

Euler Walk : a walk which traverses every edge exactly once

↳ possible iff: 1) every vertex has an even degree
2) exactly 2 vertices with odd degree (start at one, end at the other)

Theorem: There is an approximation preserving reduction from SteinerTree to Metric SteinerTree

Instance I_1 of SteinerTree, $G_1 = (V, E_1)$, weights c_1, $V = T \cup S$

Metric Instance $I_2 = f(I_1)$
Complete graph $G_2 = (V, E_2)$, same partition of V

cost $c_2(u,v)$ = Length of shortest u-v path in G_1

↳ G_2 is the "metric closure" of G_1

1)
Let B^* be an optimal Steiner Tree for I_1, B^* is also a feasible solution for I_2 ($E_1 \subseteq E_2$, same V)

$$OPT(I_2) \leq c_2(B^*) \leq c_1(B^*) = OPT(I_1)$$

2)

Let B_2 be a Steiner Tree of G_2

Construct $G_1' \subseteq G_1$ from B_2 by replacing each edge (u,v) of B_2 by a shortest u-v path in G_1.

Keep ≤ 1 copy per edge

$c_1(G_1') \leq c_2(B_2)$ – G_1' connects all terminals, might not be a tree

Consider spanning tree B_1 of G_1' $\rightsquigarrow$ steiner tree B_1 of G_1

Thus $c_1(B_1) \leq c_1(G_1') \leq c_2(B_2)$

$\therefore$ By constructing the metric closure G' of G and finding its MST we get a Steiner Tree on G' which we can convert into a Steiner Tree B on G

$$c(B) \leq c'(B') \leq 2 \cdot OPT(G') \leq 2 \cdot OPT(G)$$

$\Rightarrow$ $c(B) \leq 2 \cdot OPT(G)$, hence we have a 2-approximation

Facility Location

Given n points (clients) on a 2D-plane, find k locations to open facilities such that the max distance of a client to nearest open facility is minimised.

"k-center problem"

Let r^* be the minimum radius

We can view the k-center problem as a covering problem by k disks with some radius r. The optimal radius r^* must be an interpoint distance (otherwise we could have made it smaller)

Suppose we know r^*
repeat:
    Take an arbitrary point P which is not covered.
    Place a disk of radius $2r^*$ at P and remove all points covered by this disk
until:
    All points are covered

$< k$ disks required $\Rightarrow$ 2-OPT Approximation

Claim: If the disks have a radius $2r^*$ then the above algorithm needs $\leq k$ disks

Suppose we need more than k disks of $2r^*$, then these k points cannot be covered by $\leq k$ disks of radius r^*.

$\therefore$ there are atleast $k+1$ points s.t. distance b/w each is $> 2r^*$ $\Rightarrow$ would require atleast $k+1$ disks of $2r^*$ (a single disk cannot cover 2 points $> 2r$ distance apart)

This is a contradiction to the fact that r^* is OPT

$\Rightarrow$ our r^* is wrong $\rightarrow$ Do a Binary Search

NOTE: $\varepsilon > 0$

A $(2-\varepsilon)$ approximation for k-center is NP-Hard.
If I can find a $2-\varepsilon$ approximation for k-center then I can solve the vertex-cover problem.

k-center:

Given a set of n points V and $d: V \times V \to \mathbb{R}^+$ that satisfies the Δ inequality, find a $S \subseteq V$, $|S| = k$ which minimizes

set of centres

$$\max_{v \in V \setminus S} d(v, S), \quad d(v, S) = \min_{u \in S} d(v, u)$$

minimize the maximum distance of a point from its closest center

GAP Instance

Dominating Set - A set of vertices $S \subseteq V$ is a dominating set if all vertices in $V \setminus S$ are adjacent to some vertex in S.

Dominating Set Problem:
Given G, k, does there exist a dominating set of size $\leq k$

Scheduler: Multiprocess Scheduling, Makespan Minimization

Given n jobs with processing times $p_1, p_2, \ldots p_n$ and m m/cs, assign jobs to m/cs s.t. $\underbrace{\text{maxload on any m/c is minimized}}_{\text{makespan}}$.

BIN-PACKING:
n objects of sizes $s_1, s_2, \ldots s_n$ $\quad 0 < s_i \leq 1$
bins of size 1
Find min. no. of bins required to pack all objects

Suppose we have n_i objects of size s_i and k distinct sizes. We can find minimum no. of bins in time $n^{O(k)}$. n^{2k} (polynomial for a fixed k)

$(o_1, o_2, \dots o_k)$ is a valid configuration if

$$\left.\begin{array}{l} o_1 \text{ objects of size } s_1 \\ o_2 \text{ objects of size } s_2 \\ \vdots \\ o_k \text{ objects of size } s_k \end{array}\right\} \begin{array}{l} \text{fit into a bin} \\ 0 \le o_i \le n_i \end{array} \qquad \sum_{i=1}^{k} o_i s_i \le 1$$

$$\# \text{ of valid configs} \le \prod_{i=1}^{k} (1+n_i) \le n^k$$

choose how many of each type to put in bin

Θ is the set of all valid configurations

$b(t_1, t_2, \dots t_k) =$ min no. of bins needed to pack t_i objects of size s_i, $1 \le s_i \le k$

$0 \le t_i \le n_i$, $\prod_{i=1}^{k}(1+n_i) \approx n^k$

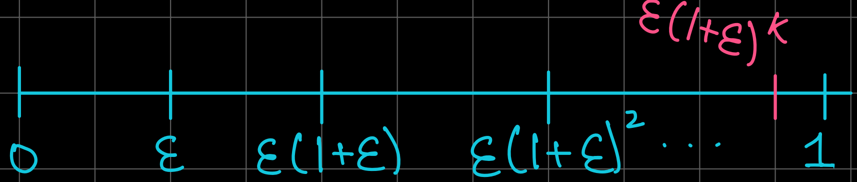
$$= 1 + \min_{o_1, o_2, \dots o_k \in \Theta} (t_1 - o_1, t_2 - o_2, \dots t_k - o_k)$$

1. Guess the makespan T
2. $\bar{p}_i = p_i / T \Rightarrow 0 \le \bar{p}_i \le 1$ as $0 \le p_i \le T$
3. Solve bin packing with object sizes $\bar{p}_i$

T is the smallest value s.t. $B(\bar{p}) = m$ (Binary search)

Solve bin packing (n^k); i.e. reasonable solving only if distinct packing sizes is small. These correspond to distinct processing times.

Idea: Reduce distinct no. of processing times (doing approximation)



$\bar{p}_i \leftarrow$ different each iteration

Elimination of small objects

increase by a factor of $(1+\varepsilon)$

$$\varepsilon(1+\varepsilon)^{K-1} < 1$$

$$K \leq 1 + \log_{1+\varepsilon} 1/\varepsilon$$

$$K \leq 1 + \frac{\ln 1/\varepsilon}{\ln(1+\varepsilon)}$$

$O(n^{2/\varepsilon^2}) \leftarrow$ small ε

$$K \leq 1 + \frac{1}{\varepsilon} \ln \frac{1}{\varepsilon} \leq \frac{1}{\varepsilon^2}$$

PTAS - Polynomial Time Approximation Scheme

$T \leq (1+\varepsilon) T_{OPT}$ $\quad$ $O(n^{2/\varepsilon^2} \log)$

$\hookrightarrow$ binary search

Now the remaining small ε jobs: schedule them greedily (least loaded). Suppose if adding these shift T, then we know all machines are almost full $\Rightarrow$ at the end, machines are at most ε empty from our schedules' makespan

makespan of our schedule $\leq OPT + \varepsilon$

$$\leq OPT + \varepsilon \cdot OPT$$

$$= OPT(1+\varepsilon)$$

$$\Rightarrow OPT(1+\varepsilon)^2 = OPT(1+2\varepsilon)$$

Fully Polynomial Time Approximation scheme

best one can do for NP-Hard problems with arbitrary closeness

Knapsack Problem

- Knapsack which can hold objects of total weight W
  (w_i, v_i) is (weight, value) of the i^{th} object. $1 \le i \le n$

Find a subset $S \subset [n]$ s.t. $\sum_{i \in S} w_i \le W$ &
$\sum_{i \in S} v_i$ is maximum

$v_i \in \mathbb{Z}^+$, $w_i \in \mathbb{R}^+$, $V = \sum_{i=1}^{n} v_i$

	0		v_1		j	V
1	0	∞	w_1	∞	∞	∞ ...
i						
n					$<W$	
				X		

$X(i,j)$: minimum weight of a subset of $[1 \dots i]$ w/ value j

$$X(i,j) = \min\left(X(i-1, j),\ X(i-1, j-v_i) + w_i\right)$$

size $= nV$
in time $O(nV)$ can fill table

> find the rightmost entry in the last row which is less than W.

$\underbrace{v_1}\ \underbrace{v_2}\ \dots\ \underbrace{v_n}$ $\quad |I| \le n \log v_{max}$

size of the input

Running time is atleast v_{max} ← exponential in $|I|$

$$I' = \left(w_i, \left\lceil \frac{v_i}{K} \right\rceil\right) = (w_i, v_i') \ \forall\ i$$

Run the DP on instance I' to get the best subset S'

$$\sum_{i \in S} w_i \le W$$

$$\sum_{i \in S} \left\lceil \frac{v_i}{K} \right\rceil = \sum_{i \in S} v_i' = OPT(I')$$

$$\sum_{i \in S} \frac{v_i}{K} - 1 \le \sum_{i \in S} \left\lceil \frac{v_i}{K} \right\rceil \le \sum_{i \in S} \frac{v_i}{K}$$

Let S be the best solution for instance I.

$$OPT(I) = \sum_{i \in S} v_i \ge \sum_{i \in S'} v_i \ge \sum_{i \in S} v_i - K|S|$$

$$OPT(I') = \sum_{i \in S'} v_i$$

$$\ge K \sum_{i \in S'} v_i'$$

$$\ge K \sum_{i \in S} v_i' \ge \sum_{i \in S} v_i - K|S|$$

$$\sum_{i \in S'} v_i \geq OPT(I) - K|S| \quad , \quad K|S| \approx \varepsilon \cdot OPT(I)$$

$$= OPT(I) - \varepsilon \cdot \frac{V_{max}|S|}{n}$$

$$V_{max}\frac{|S|}{n} < OPT$$ (Assume every object has $w < W$)

$$\geq (1-\varepsilon)\, OPT$$

$$O\left(n\frac{V}{K}\right) , \quad \frac{nV}{K} = n\frac{V \cdot n}{\varepsilon \cdot V_{max}} \leq \frac{n^3}{\varepsilon}$$

Knapsack: we obtained a $(1-\varepsilon)$ approximation running in time $O(n^3/\varepsilon)$

$\log(1/\varepsilon)$ bits $\therefore$ Not poly time algo? X

Represent $1/\varepsilon$ in unary

$$w_i, v_i, 1/\varepsilon \longrightarrow \boxed{}$$

$$|I| = n + \frac{1}{\varepsilon}$$

FPTAS: has a running time which is polynomial in input size & $1/\varepsilon$ & gives a $(1-\varepsilon)$ $(1+\varepsilon)$ approximation

n objects sizes $0 \le S_i \le 1$, $1 \le i \le n$
Bins are of size 1. Find no. of bins required to pack all objects.

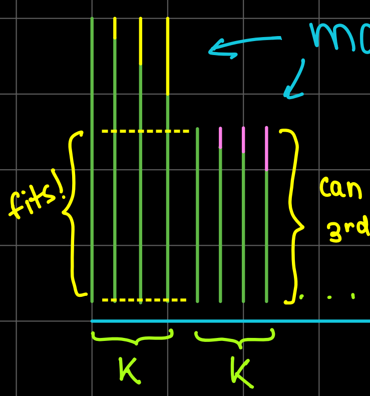
If there are k-distinct sizes then $\exists$ a DP which computes optimum in time $O(n^{2K})$

Eliminating small objects: remove all objects of size $< \varepsilon$

Suppose m objects remain

Linear Grouping

make size of each element the maximum in the group



can pack these in 1st group
3rd in 2nd ... so on

$\#$ distinct sizes $= \left\lceil \frac{m}{K} \right\rceil$

I' is the instance w/ sizes increased

$\#$ of bins I need $= OPT(I') \le OPT(I) + K$

can't pack the 1st group with our method. ∴ K more bins for them

$K = \varepsilon^2 m$

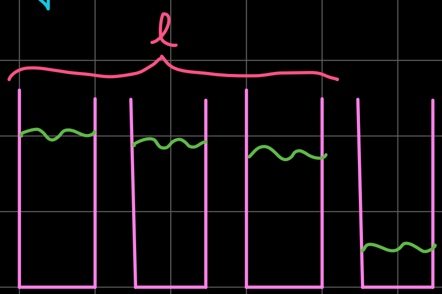
$OPT(I) \ge \varepsilon m$

$\#$ of distinct sizes $= \left\lceil \frac{m}{K} \right\rceil = \left\lceil \frac{1}{\varepsilon^2} \right\rceil$

$$OPT(I') \le OPT(I) + K$$
$$= OPT(I) + \varepsilon \cdot \varepsilon m \le (1+\varepsilon)\, OPT(I)$$

For the smaller objects : if they can fit in an existing bin, we just put it there else open a new bin.

If we open a new bin while packing small objects then



each bin (except 1) is full to an extent of $(1-\varepsilon)$

total size of all objects $\ge (1-\varepsilon)\,\ell$

$\Rightarrow$ $OPT \ge (1-\varepsilon)\,\ell$

$\#$ of bins in my solution $= \ell + 1$

$$\le \frac{OPT}{1-\varepsilon} + 1$$
$$\simeq (1+\varepsilon)\, OPT + 1$$

APTAS : Asymptotic Polynomial Time Approx. Scheme

Bin packing

$0 < s_i < 1$

Given n-objects of size $s_1, s_2, \dots s_n$ can all these objects be packed in 2 bins of size 1?

2-BIN-PACKING is NP-Hard

$\Rightarrow$ There can be no algorithm which for all instances I, packs object into at most $\left(\frac{3}{2} - \varepsilon\right) OPT(I)$ bins

I is an instance of 2-Bin Packing $\rightarrow$ $\boxed{\text{A is a 1.49 approx. for Bin-Pack}}$

> I is a YES instance

$\Rightarrow OPT(I) = 2 \Rightarrow$ A packs objects in < 3 bins $= 2$ bins!

if such an algorithm exists then we have solved an NP-Hard problem in polynomial time

> I is a NO instance

$\Rightarrow OPT(I) \geq 3 \Rightarrow$ A packs objects in ≥ 3

Most problems have a scaling property, hence the additive term does not play a role.

1.5 OPT + 3

10   A gives 18

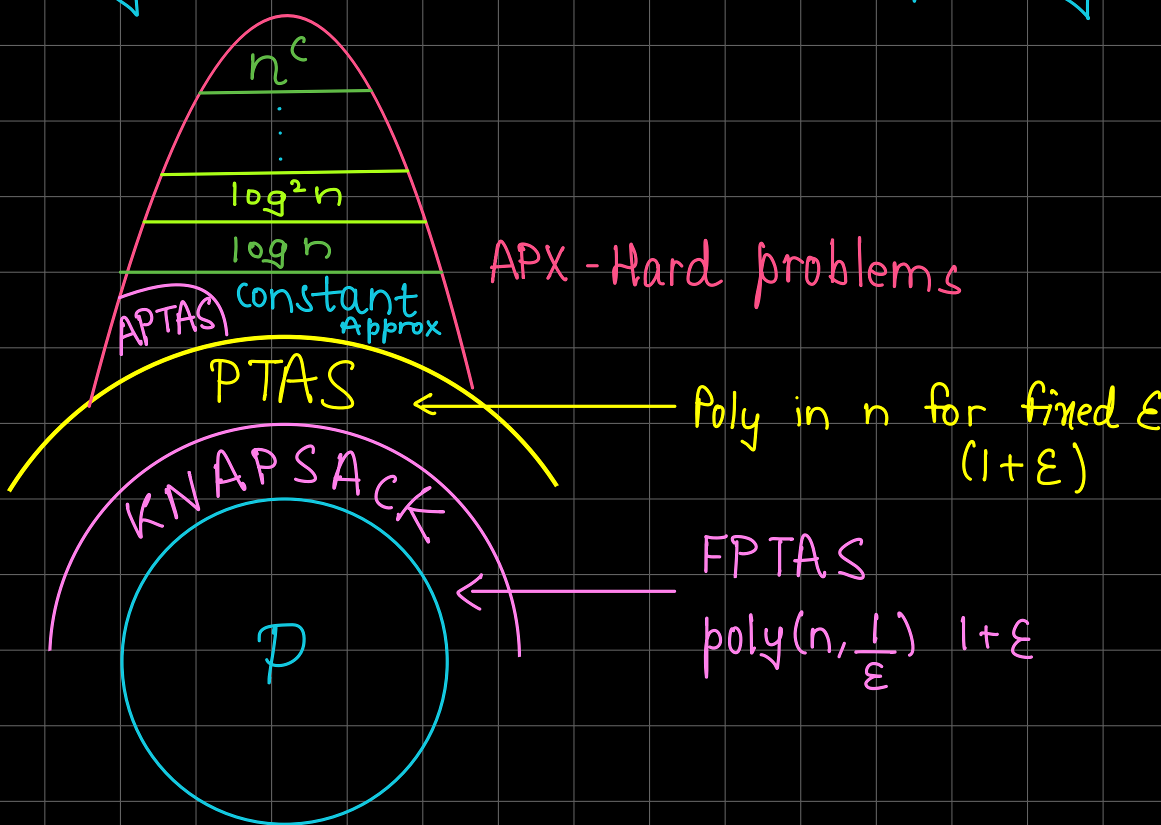
make 10 copies

10   10   . . .   10   A gives 153

by pigeonhole there is atleast one w/ 15

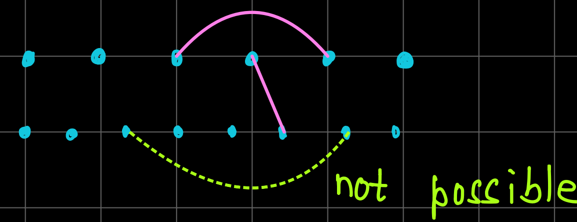
so we get 1.5 OPT algo

But this scaling doesn't work for bin packing



1) OPT = 1   min V.C.

2)   max I.S.



$|VC| = n - |IS|$

opt V.C = $n/2 - 10$ my VC = $n - 20$ (2-approx)

opt I.S. = $n/2 + 10$ my IS = 20

LP - duality

objective f^n

max $3x_1 + 4x_2 + 5x_3$

Constraints

$$
\begin{aligned}
2x_1 + 3x_2 + 5x_3 &\le 10 \\
3x_1 - 4x_2 - 6x_3 &\le 15 \\
7x_1 - 2x_2 + 10x_3 &\le 20 \\
x_1, x_2, x_3 &\ge 0
\end{aligned}
$$

canonical form

variables

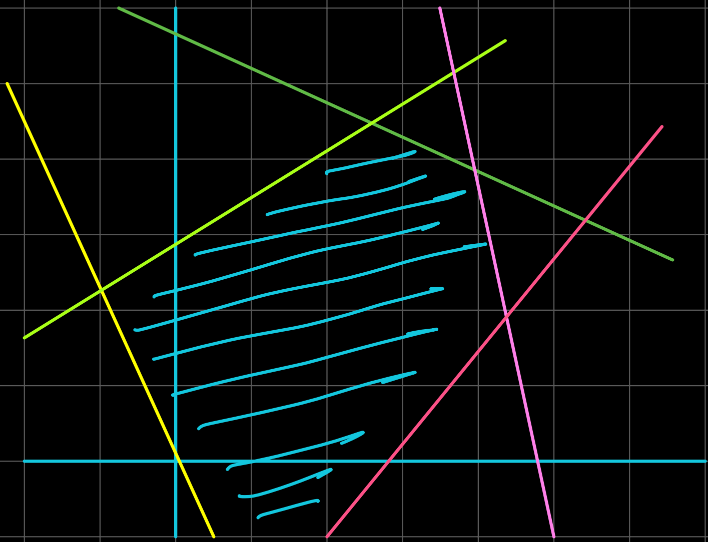
\# can't have strict inequalities

\# for unconstrained variables $x = a - b$, $a \geq 0$, $b \geq 0$

Feasible Solution - satisfies all constraints

Convex Set : line joining any 2 points in a convex set lies entirely in the convex set

\# Intersection of convex sets is also a convex set



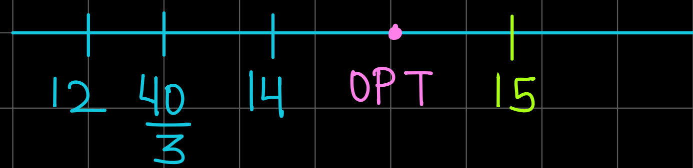
polytope - intersection of finite # of convex sets

interior point methods can compute optimum in polynomial time

$(1,1,1) \longrightarrow 12$

$(0, \frac{10}{3}, 0) \longrightarrow 40/3$

$(2, 2, 0) \longrightarrow 14$



$\frac{3}{2}(2x_1 + 3x_2 + 5x_3 \leq 10)$

$= \quad 3x_1 + 4.5x_2 + 7.5x_3 \leq 15$

$\checkmark \leftarrow$ All $x_i \geq 0$

$3x_1 + 4x_2 + 5x_3$

$(2x_1 + 3x_2 + 5x_3 \leq 10)\, y_1$

$(3x_1 - 4x_2 - 6x_3 \leq 15)\, y_2$

$(7x_1 - 2x_2 + 10x_3 \leq 20)\, y_3$

$$3x_1 + 4x_2 + 5x_3$$
$$\wedge\!\!/\!/$$
$$(2y_1 + 3y_2 + 7y_3)x_1 + (3y_1 - 4y_2 - 2y_3)x_2$$
$$+ (5y_1 - 6y_2 + 10y_3)x_3 \le \underbrace{10y_1 + 15y_2 + 20y_3}$$

objective is to minimise this

$$\boxed{\begin{aligned} &y_1, y_2, y_3 \geq 0 \\ &2y_1 + 3y_2 + 7y_3 \geq 3 \\ &3y_1 - 4y_2 - 2y_3 \geq 4 \\ &5y_1 - 6y_2 + 10y_3 \geq 5 \end{aligned}}$$

Dual Linear Program

Strong Duality Theorem: OPT primal = OPT dual
(not true if the linear program is infeasible)

Primal Feasible Solution	Dual Feasible Solution

OPT Primal = OPT Dual

Weak Duality Theorem:
Any feasible solution to primal $\le$ any feasible sol^n to dual

x_1^*, x_2^*, x_3^* - Optimal Solution to Primal
y_1^*, y_2^*, y_3^* - Optimal Solution to Dual

$$3x_1^* + 4x_2^* + 5x_3^*$$

$$\leq$$

$$(2y_1^* + 3y_2^* + 7y_3^*)x_1^* + (3y_1^* - 4y_2^* - 2y_3^*)x_2^*$$
$$+ (5y_1^* - 6y_2^* + 10y_3^*)x_3^* \leq 10y_1^* + 15y_2^* + 20y_3^*$$

$y_1, y_2, y_3 \geq 0$

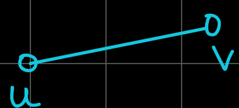
* At optimality, 1 of the 2 in each pair must be an equality

$$\begin{array}{l|l}
2y_1 + 3y_2 + 7y_3 \geq 3 & x_1 \geq 0 \\
3y_1 - 4y_2 - 2y_3 \geq 4 & x_2 \geq 0 \\
5y_1 - 6y_2 + 10y_3 \geq 5 & x_3 \geq 0
\end{array}$$

Complimentary Slackness : In every complementary pair atleast 1 inequality is tight.

$$x_u = \begin{cases} 1 & \text{if } u \text{ in the V.C.} \\ 0 & \text{o.w.} \end{cases}$$

atleast 1 endpoint in the V.C.



$$x_u + x_v \geq 1 \quad \forall (u,v) \in E$$
$$x_v \in \{0, 1\} \quad \forall v \in V$$

integer program

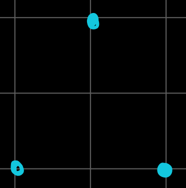
Objective : $\min \sum_{u \in V} x_u$

LP-relaxation : (relax integrality)

$0 \leq x_v \cancel{\leq 1}$

Not required, optimal would not have $x_v > 1$

$$x_u + x_v \geq 1 \quad \forall (u,v) \in E$$
$$x_v \geq 0 \quad \forall v \in V$$



optimal solution : 1, 1, 0 $\leftarrow$ integer prog.

$\frac{1}{2}, \frac{1}{2}, \frac{1}{2}$ $\leftarrow$ linear prog

\# optimal of linear program $\leq$ optimal of int program

$\hookrightarrow$ Since we have relaxed some constraints

Linear Program for Vertex Cover

$S \subseteq V$ s.t. $\forall\, e \in E$ atleast 1 endpoint of e in S

$$\forall\, v \in V,\; x_v = \begin{cases} 1, & \text{if } v \text{ is in the min-VC} \\ 0, & \text{otherwise} \end{cases}$$

objective : $\min \sum_{v \in V} x_v$

$(x_u + x_v \geq 1 \quad \forall\, (u,v) \in E)\; y_e$

$\cancel{x_u \in \{0,1\}} \quad \forall\, u \in V$

$x_u \geq 0 \leftarrow$ Linear Program

$$\sum_{e \in \delta(u)} y_e \leq 1, \quad u \in V$$

$$y_e \geq 0 \quad \forall\, e \in E$$

$$\max \sum_{e \in E} y_e$$

Finding the dual L.P.

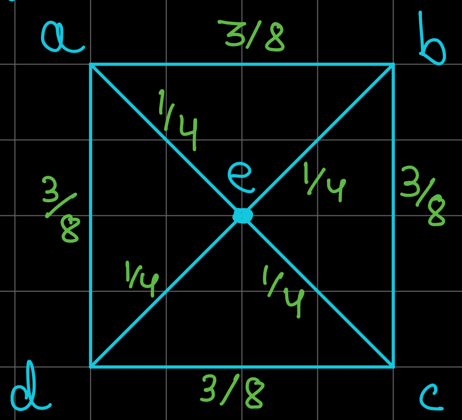
$$\sum_{e \in E} (x_u + x_v)\, y_e \geq \sum_{e \in E} y_e$$

$$\sum_{v \in V} x_v \geq \sum_{e \in E} (x_u + x_v) \cdot y_e$$

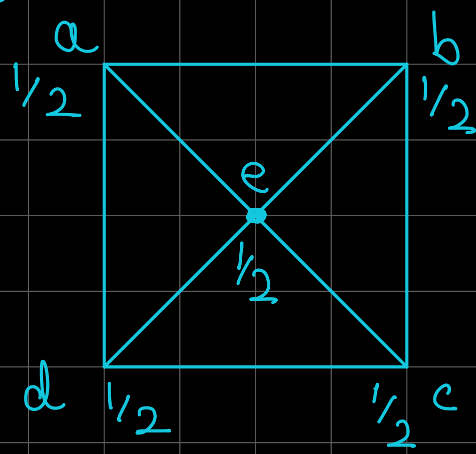
$$= \sum x_v \sum_{v \text{ in } e} y_e$$

$$\Rightarrow \sum_{v \text{ in } e} y_e \leq 1 \quad \forall\, v, \quad y_e \geq 0$$

Dual



Primal



How to tell if the solution is optimal?
- If there is a dual solution with the same value

maximisation   minimization

optimal dual integer solution = MAX MATCHING	optimal sol^n for primal and dual	optimal primal integer sol^n = VERTEX COVER

min-VC
1) solve LP, let x^* be optimum solution
2) $S = \{ v \mid x^*_v \geq 1/2 \}$
3) Output S

→ S is a v.c. ∵ ∀ edge (u,v) atleast one of $x_u^*, x_v^* \geq 1/2$

→ $|S| \leq 2 \sum x_v^* = 2 * \text{opt. solution to VC-LP}$
$\leq 2 \text{ min-VC}$

(Deterministic Rounding)

weighted vertex-cover

$w : V \to \mathbb{R}^+$

Find a V.C. of minimum weight

Integer Linear Program

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

$$\sum_{v \in S} w_v \leq 2 \sum_{v \in S} w_v x_v^*$$

Same rounding as for unweighted

$$\leq 2 \sum_{v \in V} w_v x_v^*$$

$$= 2 \text{ opt sol}^n \text{ to VC-LP}$$

$$\leq 2 \text{ min wt. VC}$$

Set Cover

$U = \{e_1, e_2, \ldots e_m\}$

$S_i \subseteq U \, , \; 1 \leq i \leq n$

$\mathcal{L}$ is a set cover if $\bigcup_{S_i \in \mathcal{L}} S_i = U$

x_i is a 0/1 variable corresponding to set S_i

$$x_i = \begin{cases} 1, & \text{if } S_i \in \mathcal{L} \\ 0, & \text{otherwise} \end{cases}$$

$\left(\forall e_j \in U, \sum_{i: e_j \in S_i} x_i \geq 1\right) y_j$ # sum x_is over all sets containg any x_i Atleast 1 chosen

Objective : $\min \sum_{i \in [n]} x_i$

$\forall i \in [n]$

$\cancel{x_i \in \{0, 1\}}$ $x_i \geq 0$

Dual Linear Program :

$$\max \sum_{j \in [m]} y_j$$

$$\sum_{j: e_j \in S_i} y_j \leq 1 \quad \forall i \in [n]$$

$$y_j \geq 0 \quad \forall j \in [m]$$

Finding the dual LP

$$\sum_{i \in [n]} y_i \left(\sum_{j: e_i \in S_j} x_j\right) \geq \sum_{i \in [n]} y_i$$

$$\sum x_i \geq \sum_{j \in [m]} x_j \left(\sum_{i: j \in S_i} y_i\right) \geq \sum y_i$$

$$\therefore \sum_{i: i \in S_j} y_i \leq 1 \quad \forall j \in [m]$$

Dual | Primal

Any feasible Dual sol^n (integer L.P.) | Opt sol^n to dual/primal (frac.) | Feasible Primal Sol^n (integer L.P.)

Randomized Rounding

1) Solve LP. Let x^* be an optimum sol^n.
2) Pick set S_i with probability x_i^*
   Let S be the collection of sets picked.

$$\text{Prob } e_j \text{ is covered} = 1 - \text{Prob. } e_j \text{ not covered}$$
$$= 1 - \prod_{i : e_j \in S_i} (1 - x_i^*)$$

$$e^{-x} \approx 1 - x + \underbrace{\left[\frac{x^2}{2!} - \frac{x^3}{3!} + \ldots\right.}_{+ve}$$

if $x < 1$:
$$e^{-x} > 1 - x$$

$$e^{-\sum x_i} > \prod (1 - x_i), \quad i : e_j \in S_i$$

$$\text{Prob } e_j \text{ is covered} > 1 - \prod (1 - x_i^*)$$
$$> 1 - \prod e^{-x_i^*}$$
$$= 1 - e^{-\sum x_i^*}, \quad i : e_j \in S_i$$
$$\geq 1 - e^{-1} \approx 0.62$$

Lets increase the probability

$$x_i^* \rightarrow x_i^* \, 2 \ln(m)$$

Prob e_j is covered $> 1 - \prod (1 - x_i^* \cdot 2\ln(m))$

$$> 1 - \prod e^{-x_i^* \cdot 2\ln(m)}$$

$$= 1 - e^{-\sum x_i^* \cdot 2\ln(m)}, \quad i : e_j \in S_i$$

$$\geq 1 - e^{-2\ln(m)}$$

$$= 1 - \frac{1}{m^2} \quad \Big\} \text{ v. high prob.}$$

we'll get a set cover in a few trials!

Expected size of set cover $= 2\ln(m) \sum_{i \in [n]} x_i^*$

$= 2\ln(m) \cdot$ opt frac sol^n to LP

$\leq 2\ln(m) \cdot$ opt set cover

- could not use deterministic rounding because values could have been very small

- Suppose set system is such that each element appears in atmost f-sets.

Pick all sets w/ $x_i^* > 1/f$ → This is an f-approximation

Randomised Rounding for Set Cover

1. Solve the LP & let x^* be the optimal solution
2. Pick set S_i with prob $x^* \ln(m)$
   // $\mathcal{S}_1$ is collection of sets picked
3. If element j remains uncovered then pick the cheapest set containing element j.
   // $\mathcal{S}_2$ is collection of sets picked.
4. Output $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$

Claim: $\mathcal{S}$ is a set cover

Claim: $\mathbb{E}[c(\mathcal{S})] \leq c \ln(m) . OPT_{LP}$

$$= \mathbb{E}[c(\mathcal{S}_1) + c(\mathcal{S}_2)]$$
$$= \mathbb{E}[c(\mathcal{S}_1)] + \mathbb{E}[c(\mathcal{S}_2)] \leq (1+\ln(m)) OPT_{LP}$$

$$\mathbb{E}[c(\mathcal{S}_1)] = \sum_{i \in [n]} Pr(S_i \text{ is picked}) . c(S_i)$$
$$= \sum_{i \in [n]} x_i^* \ln(m) . c(S_i)$$
$$= \ln(m) \sum_{i \in [n]} x_i^* . c(S_i) = \ln(m) . OPT_{LP}$$

$$\mathbb{E}[c(\mathcal{S}_2)] \leq \sum_{j \in [m]} \left(\min_{i : j \in S_i} c(S_i) \right) \times P\left[\begin{matrix} j \text{ is uncovered} \\ \text{after step 2} \end{matrix} \right]$$
$$\leq \sum_{j \in [m]} \frac{1}{m} \times OPT_{LP} = OPT_{LP}$$

$\forall$ element j,
min cost set containing $j \leq OPT_{LP}$

Markov's Inequality

X is a non-negative r.v.

$$Pr[X > \mu . \delta] \leq \frac{1}{\delta} = \frac{\mu}{\delta\mu} = \frac{E[X]}{\delta\mu}$$, $\mu = E[X]$

$$Pr[X > \delta] \leq E[X]/\delta$$

If $\geq \delta$ then its a contradiction

Chernoff Bounds

$X_1, X_2, \ldots X_n$ are independent r.v.

$$X_i = \begin{cases} a_i & \text{with prob } p_i \\ 0 & \text{o.w.} \end{cases}$$ $\mathbb{E}[X_i] = p_i . a_i$

$0 \leq a_i \leq 1$

$$X = X_1 + X_2 + \ldots + X_n$$

$$\mathbb{E}[X] = \sum_{i=1}^{n} p_i . a_i = \mu$$

$$Pr[X \geq (1+\delta)\mu] = Pr[e^{tX} \geq e^{t(1+\delta)\mu}] \quad \forall t > 0$$

$$\leq \frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}} \quad - \text{①}$$

$$\mathbb{E}[e^{tX}] = \mathbb{E}\left[e^{t\sum_{i=1}^{n} X_i}\right] = \mathbb{E}\left[\prod_{i=1}^{n} e^{tX_i}\right]$$

$$= \prod_{i=1}^{n} \mathbb{E}[e^{tX_i}]$$

$$\mathbb{E}[e^{tX_i}] = p_i e^{ta_i} + (1-p_i)\cdot 1$$

$$= 1 + p_i(\underbrace{e^{ta_i} - 1}) \qquad e^{ta_i} - 1 \geq a_i(e^t - 1)$$

$$= \prod_{i=1}^{n} \left(1 + p_i(e^{ta_i} - 1)\right)$$

$$= \prod_{i=1}^{n} \left(1 + p_i a_i (e^t - 1)\right)$$

$$\leq \prod_{i=1}^{n} e^{p_i a_i (e^t - 1)} = e^{\mu(e^t - 1)}$$

continuing ①

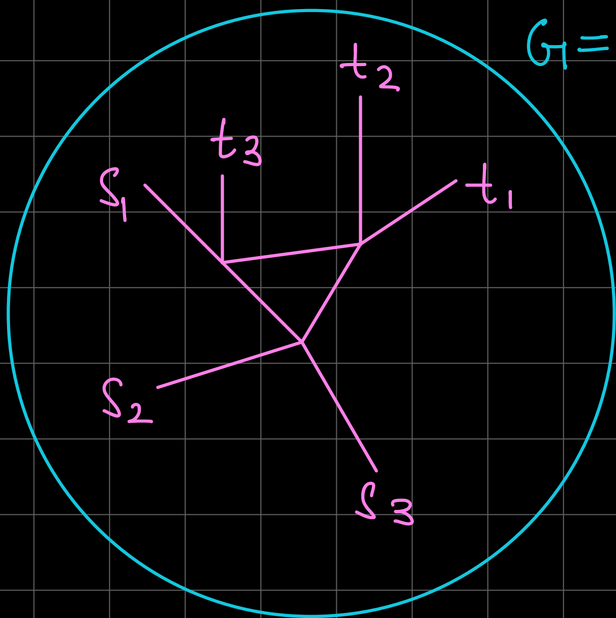
$$\frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}} \leq \frac{e^{\mu(e^t - 1)}}{e^{t(1+\delta)\mu}} = \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}, \quad e^t = 1+\delta$$

$$\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \leq e^{-\delta^2/3}$$

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}$$

$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2}$$

Raghavan & Thompson



$G = (V, E)$

Edge - disjoint paths problem
Given a graph $G = (V, E)$ & pairs of vertices. Given (s_i, t_i), $i \in [k]$. find paths b/w s_i & corresponding t_i which are edge-disjoint.

NP - Hard

Integer Multi Commodity Flow:
Given $G = (V, E)$, $c: E \to \mathbb{R}^+$ & vertices (s_i, t_i) $i \in [k]$ & demands $d_i \in \mathbb{Z}^+$

- send d_i units of flow from s_i to t_i
  In a multicommodity flow, we associate a commodity with each source / sink pair

- capacity constraint: total flow of all commodities through an edge is atmost the capacity of the edge

· conservation: every commodity is conserved at each node other than source / sink

$f^i(e)$ is flow of commodity i in edge e

$\forall e \in E, \sum_{i \in [k]} f^i(e) \leq c(e) \leftarrow$ CAPACITY Constraint

$\sum_{e \in \delta in(v)} f^i(e) = \sum_{e \in \delta out(v)} f^i(e) \quad \forall\ v \neq s_i, t_i \ \forall\ i$

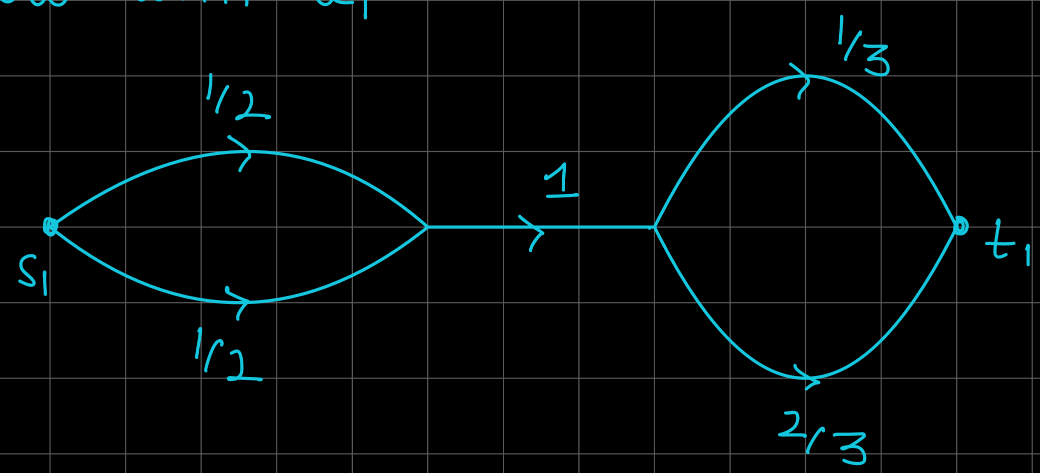
↖ CONSERVATION Constraint

$\sum_{e \in \delta out(s_i)} f^i(e) = d_i$ (at source)

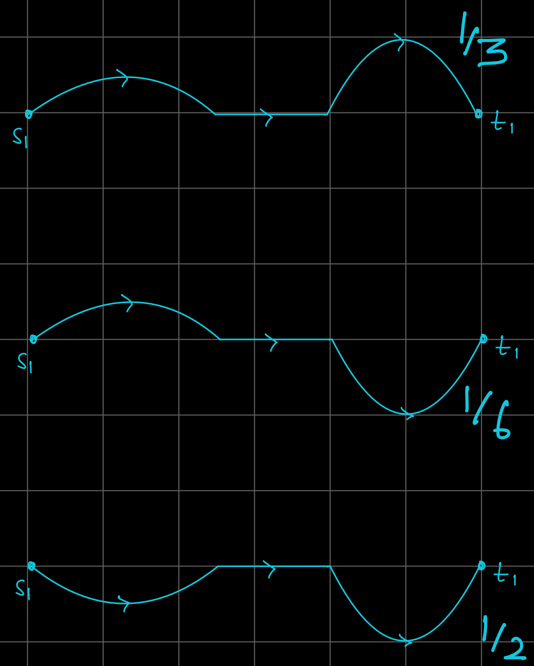
$f^i(e) \geq 0 \quad \forall\ i \in [k],\ e \in E$

$f^i(e) \in \mathbb{Z}^+$ integer multi commodity flow

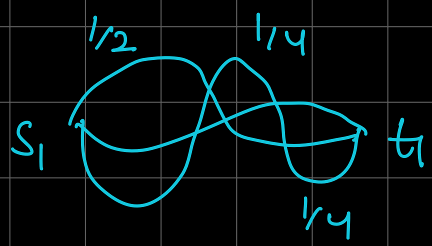
1. Solve the L.P. for integer multicommodity flow with $d_i = 1$

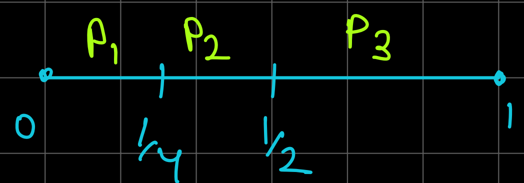


Flow Decomposition



pick paths with prob proportional to flow





Ch 5 of book

Edge-disjoint Path

↳ special case of integer commodity flow

$G=(V,E)$, $C: E \to \mathbb{R}^+$ $\quad (s_i, t_i) \quad 1 \le i \le K$

d_i

$$\mathcal{P} = \bigcup_{i=1}^{K} \mathcal{P}_i$$

$\mathcal{P}_i$ is the set of paths from s_i to t_i in G (can be exponential)

$$x_p = \begin{cases} 1 & \text{if we pick the path } p \\ 0 & \text{o.w.} \end{cases}$$

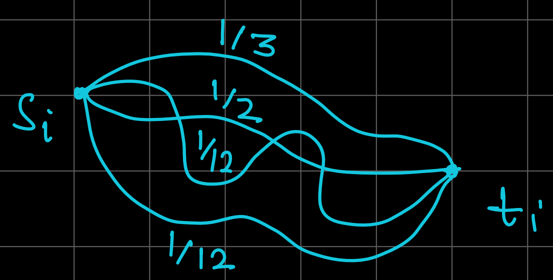
$$\forall i \quad \sum_{p \in \mathcal{P}_i} x_p = 1$$ ← Exactly 1 path b/w each s_i - t_i pair

$$\forall e \in E \quad \sum_{\mathcal{P}_i : e \in p} x_p \le 1$$ ← For each edge, only 1 path through it taken

$$0 \le x_p \cancel{\le 1}$$ not required $\quad \therefore \; 0 \le x_p$

LP1

x^* is an optimum solution for LP1



Chernoff Bounds $\quad X = X_1 + X_2 + \dots + X_n$

$$\Pr\left[X \ge (1+\delta)\mu\right] \le \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right] \underset{0 \le \delta \le 1}{\le} e^{-\mu\delta^2/3}$$

$$\Pr[X \geq (1+\delta)U] \leq e^{-U\delta^2/3}$$

$X_e^i = 1$ if the path chosen for pair i uses e
$\quad\;\; 0$ otherwise

$X_e = X_e^1 + X_e^2 + \dots + X_e^k$ is a r.v. which is the congestion on edge e

$$\mathbb{E}[X_e] = \sum_{i=1}^{k} \mathbb{E}[X_e^i]$$

$$\leq 1$$

$\mathbb{E}[X_e^i]$ is sum of probs of paths b/w s_i & t_i through it

$$\Pr[X_e \geq (1+1)4\ln(m)] \leq e^{-\frac{4\ln(m)}{3}}$$

$\delta = 1, \quad U = 4\ln(m)$

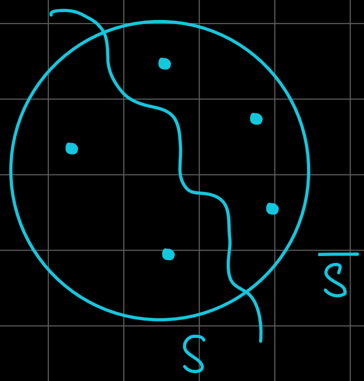
$$\Pr[X_e > 8\ln(m)] \leq \frac{1}{m^{4/3}}$$

$$\Pr[\max_e X_e > 8\ln(m)] \leq \sum \Pr[X_e > 8\ln(m)] \leq \frac{m}{m^{4/3}} = \frac{1}{m^{1/3}}$$

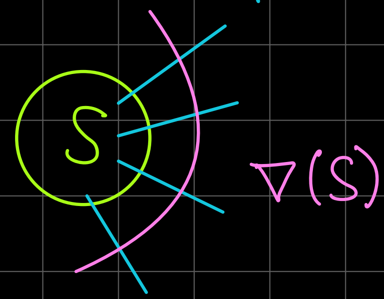
Steiner Tree Problem

Graph $G = (V, E)$, $l : E \to \mathbb{R}^+$, $T \subseteq V$

Find a connected subgraph in G which connects all vertices in T



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



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

subject to

to ensure the subgraph connects all vertices in T $\left\{ \sum_{e \in \nabla(S)} x_e \geq 1, \quad \forall S \subseteq V, \ S \cap T \neq \phi \ \ \& \ \bar{S} \cap T \neq \phi \right.$

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

$0 \leq x_e \cancel{\leq 1}$

↑ not reqd. as $x_e \leq 1$ for the optimal solution

Ellipsoid Method

$\max \ C^T x$

subject to

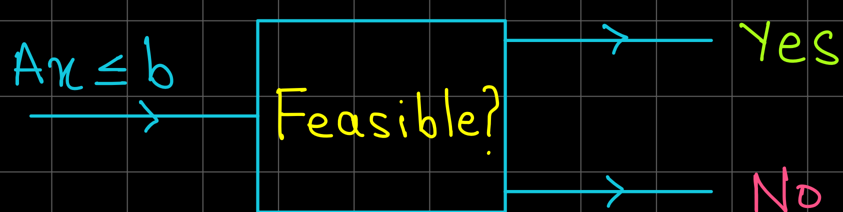
$Ax \leq b$

$x \geq 0$

Optimisation ≡ Feasibility

$Ax \leq b$

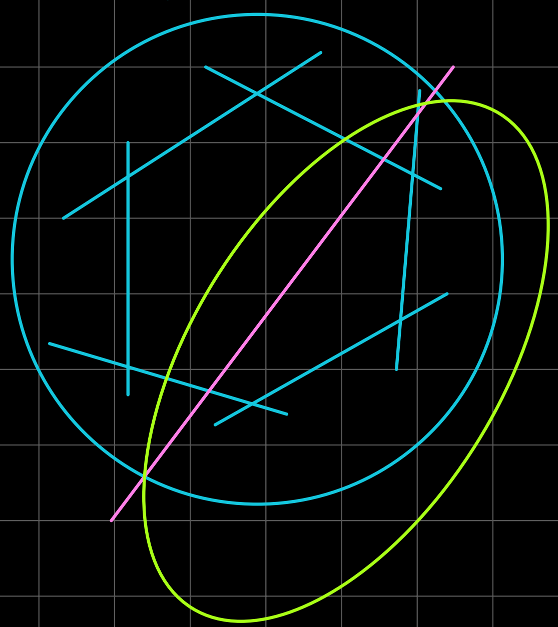
$x \geq 0$



How to do optimisation?

Add $c^T x \le d$ constraint and check feasibility
Do binary search on d.

$Ax \le b$



Optimisation ≡ Feasibility ≡ Separation

Given a solution x, declare x as feasible or provide an inequality violated by x

The ellipsoid method allows us to find an optimal solution to the linear program in time polynomial in n (the no. of variables) and ϕ (bound on the no. of bits needed to encode any inequality $Ax \le b$), given a polynomial time separation oracle.

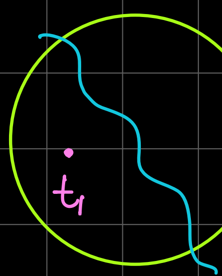
Separation Oracle: Takes as input a supposedly feasible solution x to the linear program, and either verifies that x is indeed a feasible solution or, if it is infeasible, produces a constraint violated by x.

To solve an LP we need a separation oracle

coming back to Steiner Tree,

Separation Oracle : Take a solution x. For every pair of terminals, find the min-cut with one of them on one side and the other is in the complement. If every such min-cut has value ≥ 1 it is feasible, otherwise we have a violated constraint.

can fix t_1



\# In every cut t_1 is on one side, taking every pair is redundant

\# of variables : polynomial
\# of constraints : exponential

$$\max \sum_{i=1}^{K} z_i$$

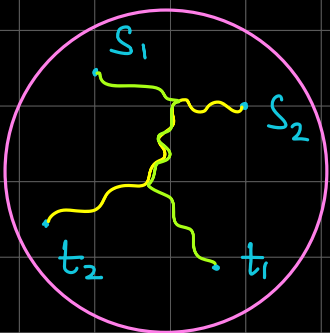
$$\sum_{e \in E} l_e \leq 1$$

$$z_i \leq \sum_{e \in P} l_e \quad \forall\, P \in \mathcal{P}_i$$

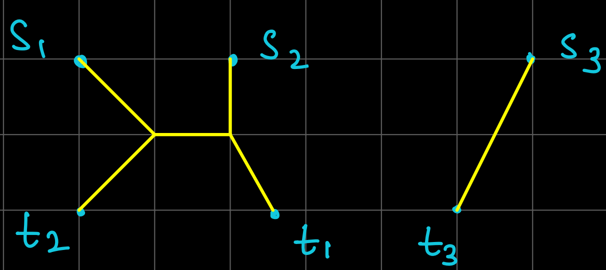
For separation I want to find out the shortest path between s_i & t_i

Steiner Forest (Generalised Steiner Tree)

Undirected Graph $G=(V,E)$, costs $c_e : E \to \mathbb{R}^+$
$(s_i, t_i) \quad 1 \le i \le K$.



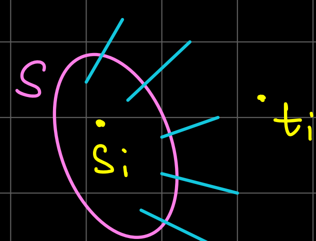
Find a minimum-cost subset of edges $F \subseteq E$ such that every $s_i - t_i$ pair is connected in the set of selected edges



Find such a subgraph of minimum cost

Observation: If I can solve Steiner Forest then I can solve Steiner Tree (take all pairs in T as $s_i - t_i$ pairs)

$$x_e = \begin{cases} 1, & \text{if } e \in F \text{ (the Steiner Forest)} \\ 0, & \text{otherwise} \end{cases}$$



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

subject to $\displaystyle\sum_{e \in \nabla(S)} x_e \ge 1 \quad \forall\, S \subseteq V$ s.t. $s_i \in S$ & $t_i \notin S$ for some i

$\cancel{x_e \in \{0,1\}}$ $\quad x_e \ge 0$

$\mathcal{S} \subseteq 2^V$ is the collection of all sets that separate some (s_i, t_i) pair

Primal-Dual Algorithm

Dual L-P

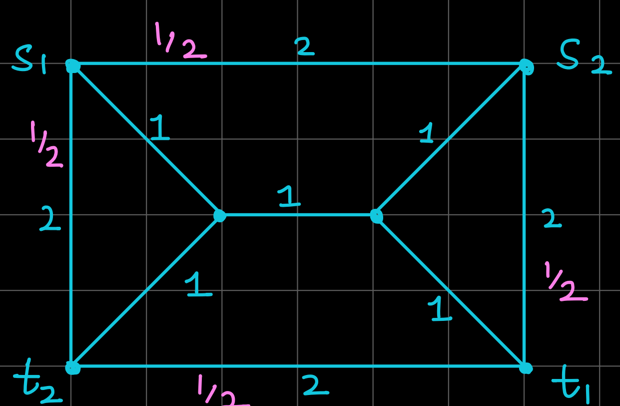
$$\sum_{\substack{S \subseteq V,\\ \exists i \text{ s.t. } s_i \in S, t_i \notin S \\ \text{or vice-versa}}} y_S (1) \le \sum_{S \subseteq V} y_S \left(\sum_{e \in \nabla(S)} x_e \right)$$

$$= \sum_{e \in E} x_e \sum_{e \in \nabla(S)} y_S \le \sum_{e \in E} x_e c_e$$

$$\max \sum_{\substack{S \subseteq V,\ \exists i \text{ s.t. } s_i \in S, t_i \notin S \\ \text{or vice-versa}}} (y_S)$$

$$\text{subject to } \sum_{S : e \in \nabla(S)} y_S \le c_e \quad \forall\, e \in E$$

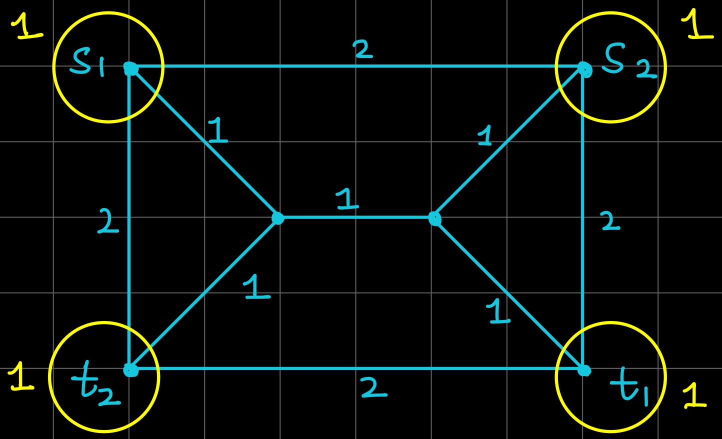
$$y_S \ge 0 \quad , \exists i \text{ s.t. } s_i \in S, t_i \notin S \text{ or vice-versa}$$



Optimal Primal $= \frac{1}{2} \cdot 2 \ldots 4$ times

$= 4$

To argue that this is optimal, we show the existence of a dual solution with the same value

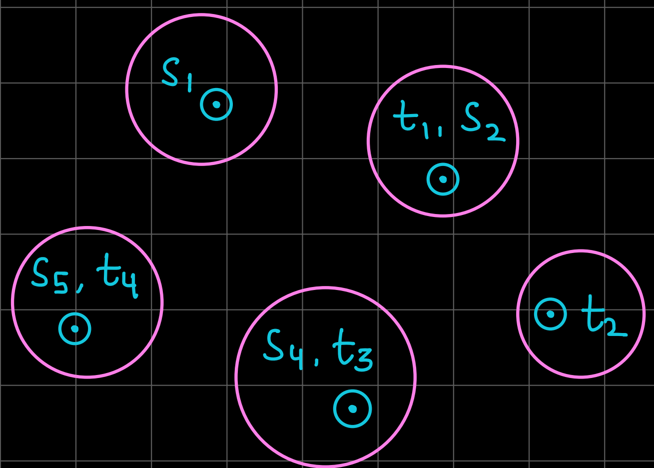


All constraints are met
∴ value of dual = 4

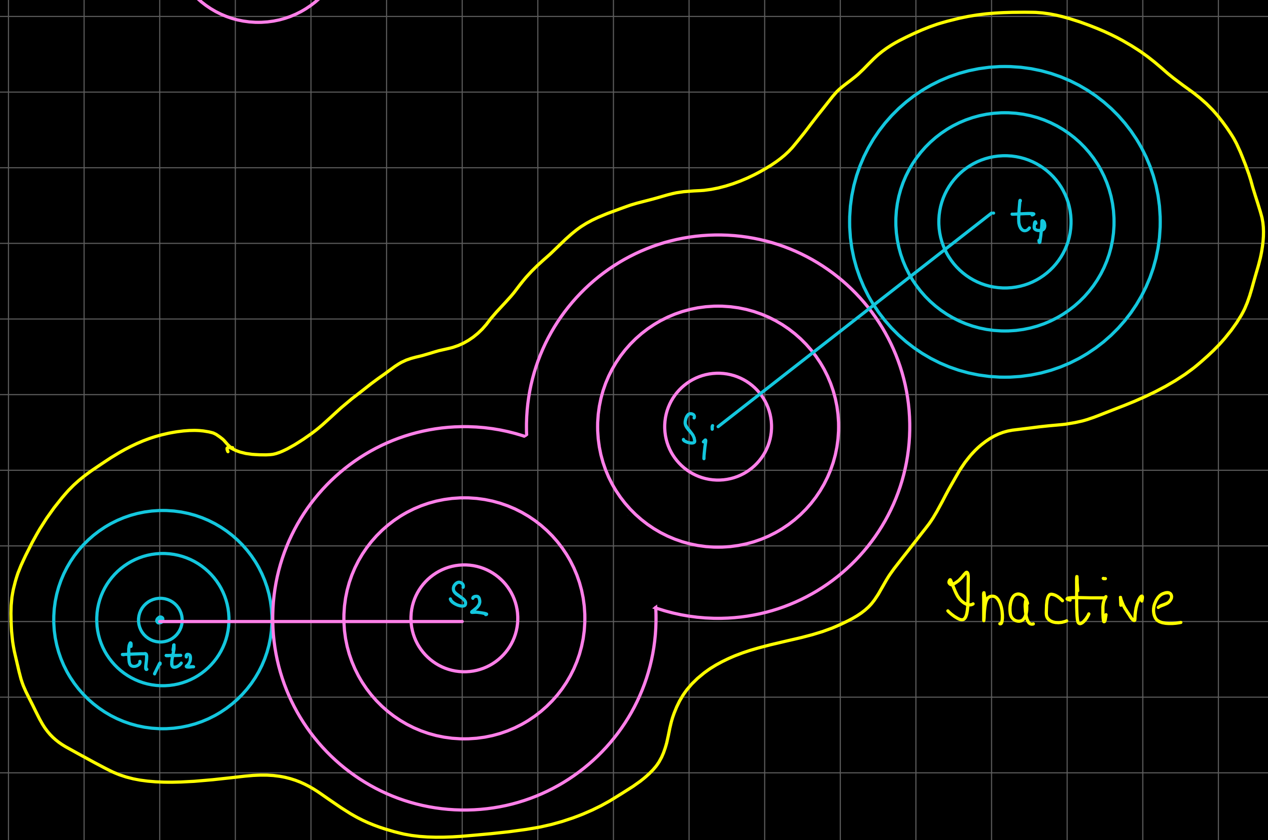
= optimal primal

Hence, 4 is the optimum

A : the collection of active sets at any point in the algorithm



> Raise dual variables of all sets in A till some edge say e goes tight



Inactive

Dual $\longleftarrow$ 2 $\longrightarrow$ Primal

Feasible Dual — OPT Frac Primal = OPT Frac Dual — Best Steiner Forest — Feasible Primal

goes tight

$e = (u, v)$

sets that separate any $s_i - t_i$ pair

- $u \in S_1 \in A$, $v \in S_2 \in A$ then
  if $S_1 \cup S_2 \in \mathcal{J}$: $A \leftarrow A + S_1 \cup S_2 - S_1 - S_2$
  else $I \leftarrow I + \{S_1 \cup S_2\}$,
  $A \leftarrow A - S_1 - S_2$

A : active sets
I : inactive sets

- $u \in S_1 \in A$, $v \in S_2 \in I$ then
  $A \leftarrow A + S_1 \cup S_2 - S_1$
  $I \leftarrow I - S_2$

initially $F = \phi$
if e is tight then we add e to F

Claim: F is a forest at any stage of the algorithm

$G = (V, E)$, (s_i, t_i) , $i \in [K]$

e is tight if $\sum_{S : e \in \nabla(S)} y_S = c_e$

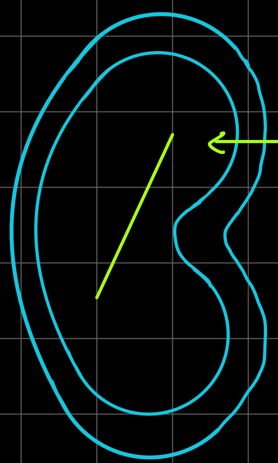
Growth Phase :

$F \leftarrow \phi$ // set of tight edges, forms a forest
$I \leftarrow$ connected components of F which are inactive
$A \leftarrow$ " " " " active

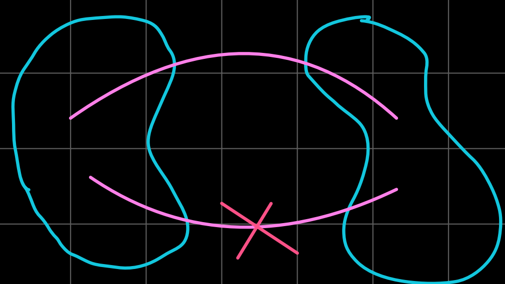
a set is active iff $\exists\, i$ s.t. $|S \cap \{s_i, t_i\}| = 1$

Increase dual variables of all sets in A, simultaneously till some edge e goes tight.

$F \leftarrow F \cup \{e\}$
until there are no active components



such an edge won't become tight



if 2 edges become tight, only choose 1

At most (n-1) iterations

Reverse - Delete Phase

Let $f_1, f_2 \ldots f_k$ be the edges in F in the order in which they were added.

$F \leftarrow F \setminus \{f_i\}$ if $F \setminus \{f_i\}$ is feasible

The final solution obtained earlier in the growth phase is feasible because no active components are remaining. $\therefore$ every $\{s_i, t_i\}$ pair is in the same connected component

Let $\bar{F}$ be the final solution

$$\left.\sum_{e \in F} c_e \leq 2 \sum_{S \in \mathcal{S}} y_S \right\} \text{to show}$$

$$\mathcal{S} = \{ S \subseteq V : \exists\, i : |S \cap \{s_i, t_i\}| = 1 \}$$

Primal	Dual
$\min \sum_{e \in E} c_e x_e$	$\max \sum_{S \in \mathcal{S}} y_S$
subject to	subject to
$\forall S \in \mathcal{S},\ \sum_{e \in \nabla(S)} x_e \geq 1$	sum of all sets that edge crosses through $\longrightarrow$ $\forall e \in E,\ \sum_{S \in \mathcal{S},\, e \in \nabla(S)} y_S \leq c_e$
$\forall e \in E,\ x_e \geq 0$	$\forall S \in \mathcal{S},\ y_S \geq 0$

To show: $\sum_{e \in F} c_e \leq 2 \sum_{S \in \mathcal{S}} y_S$

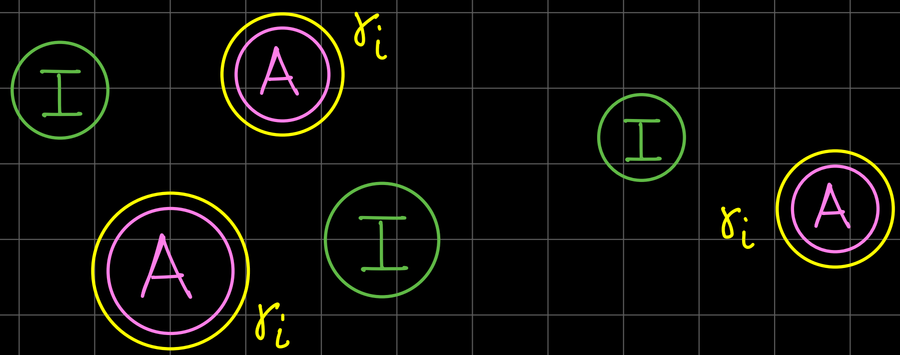
$\forall e \in \bar{F},\quad c_e = \sum_{S : e \in \nabla(S)} y_S \leftarrow$ $\because$ we only add an edge to F when it becomes tight

using this,

To show: $\sum_{e\in \bar{F}} \sum_{S:\, e\in \delta(S)} y_S \le 2 \sum_{S\in \mathcal{L}} y_S$

$$\equiv \sum_{S} y_S \deg_{\bar{F}}(S) \le 2 \sum_{S\in\mathcal{L}} y_S$$

Let γ_i = increase in the dual of active sets in the ith iteration



In an iteration

Increase in LHS $= \sum_{S\in A_i} \gamma_i \deg_{\bar{F}}(S)$

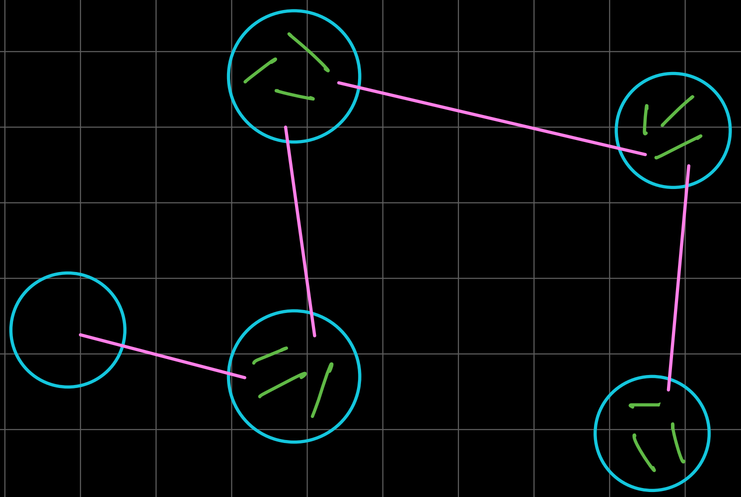
Increase in RHS $= 2\gamma_i |A_i|$

we show that increase in LHS ≤ increase in RHS in each iteration

We will show that $\forall i$

$$\sum_{S\in A_i} \cancel{\gamma_i} \deg_{\bar{F}}(S) \le 2 \cancel{\gamma_i} |A_i|$$

To show: $\sum_{S\in A_i} \deg_{\bar{F}}(S) \le 2|A_i|$ (using the result that $\bar{F}$ is a forest)



superimpose edges of $\overline{F}$ on the i^{th} iteration

In the i^{th} iteration,

forest

$$\sum_{S \in A_i \cup I_i} \deg_{\overline{F}}(S) = 2 * \text{no. of edges in } \overline{F} \text{ across } A_i \cup I_i \leq 2|A_i \cup I_i|$$

$$\sum_{S \in A_i} \deg_{\overline{F}}(S) + \sum_{S \in I_i} \deg_{\overline{F}}(S) \leq 2|A_i| + 2|I_i|$$

we need $\sum_{S \in I_i} \deg_{\overline{F}}(S) \geq 2|I_i|$

to get $\sum_{S \in A_i} \deg_{\overline{F}}(S) \leq 2|A_i|$

Every inactive component has $\deg_{\overline{F}}(S) \geq 2$

$\because$ If there was an inactive leaf, we would have deleted it in the reverse delete phase

(remove inactive components of degree 0)

If I_i was inactive at iteration i and only has 1 edge coming out of it in F then adding that edge did not help us (no new $s_i - t_i$ pairs connected)

General Version of Steiner Forest:

$\forall S \subseteq V$, $r(S)$ is a 0/1 requirement f^n
We want to pick a set of edges
F s.t. $\forall S$: $|F \cap \nabla(S)| \geq r(S)$

For Steiner Forest: $\forall S \in \mathcal{S}$, $r(S) = 1$
to ensure connectivity of terminals

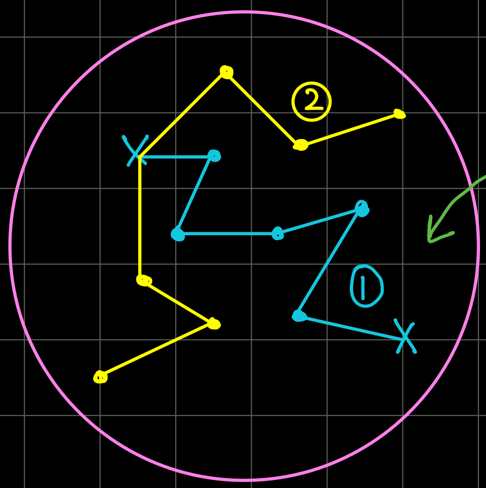
can show if r is a proper function then the primal dual algorithm is a 2-approximation

r is a proper function iff
1) $r(V) = 0$
2) $r(S) = r(\bar{S})$
3) $r(A \cup B) \leq r(A) + r(B)$

T-join

$G = (V, E)$, $\omega: E \rightarrow \mathbb{R}^+$, $T \subseteq V$
A T-join is a set of edges E' s.t.
$\forall v \in T$, $\deg_{E'}(v)$ is odd
$\forall v \notin T$, $\deg_{E'}(v)$ is even

T-join as a general version of Steiner Forest
$r(S) = 1$ iff $|S \cap T|$ is odd

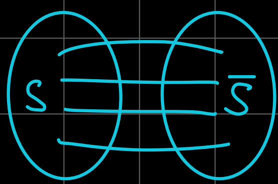


› Find path b/w 2 vertices in T
Remove the path now
(the property still holds except that the matched vertices in T have even degree in the remaining edges)

Multi-Cut:

A cut is a partition of the vertex set into 2 pieces

$\text{cost}(S, \bar{S})$ = total cost of edges in $\nabla(S)$



Minimum Cut in a graph: $G = (V, E)$, $c: E \to \mathbb{R}^+$
↳ A cut of minimum cost

fix a vertex u

$\forall v \in V \setminus \{u\}$ do:
  find min u-v cut using Ford-Fulkerson
↳ repeat over all $\cancel{\binom{n}{2}}$ pairs
$n-1$ ($\because$ we fixed u)

Multicut: Given $G = (V, E)$, $c: E \to \mathbb{R}^+$ & (s_i, t_i), $i \in [K]$
a multicut is a set of edges $E' \subseteq E$ whose removal disconnects every s_i-t_i pair

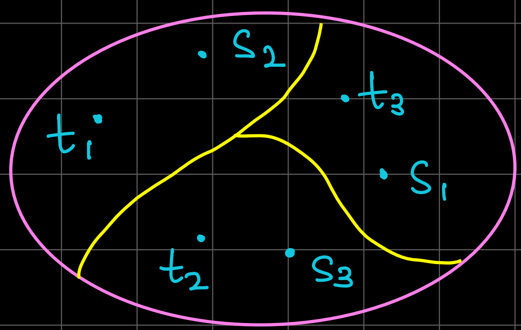
Example:

min multicut = 6

For $k = 1$, max s-t flow = min s-t cut

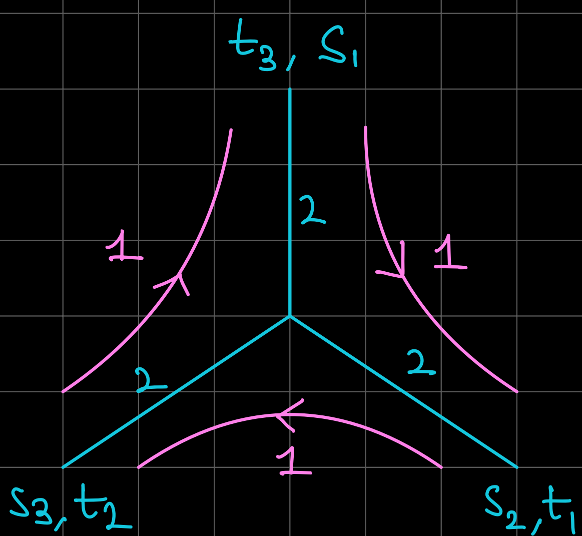
Maximum Multicommodity Flow:

Send flow of commodity i from s_i to t_i subject to conservation & capacity constraints such that the total flow (over all commodities) is maximised



maximum multicommodity $\leq$ minimum multicut

Equal for 1 commodity
What about > 1?



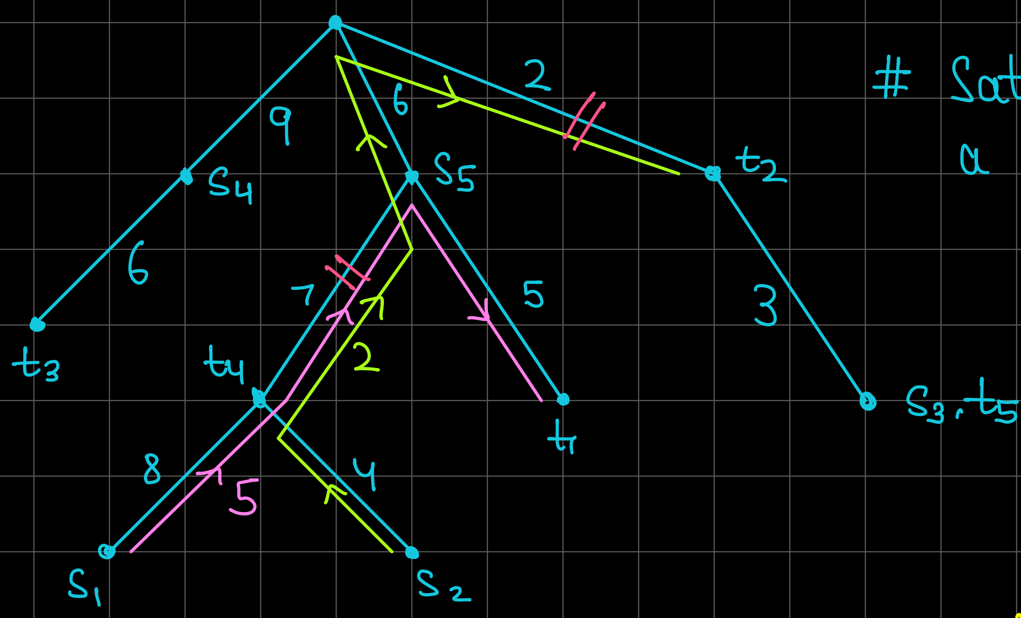
max multicommodity = 3
min multicut = 4
any 2 edges

Clearly max multicommodity $\neq$ min multicut

If G is a tree, $T = (V, E)$, $c: E \to \mathbb{R}^+$, (s_i, t_i), $1 \leq i \leq k$ then

$$\frac{\text{min multicut}}{2} \leq \text{max multicommodity flow} \leq \text{min multicut}$$

Approximate Max flow - Min Cut Theorem

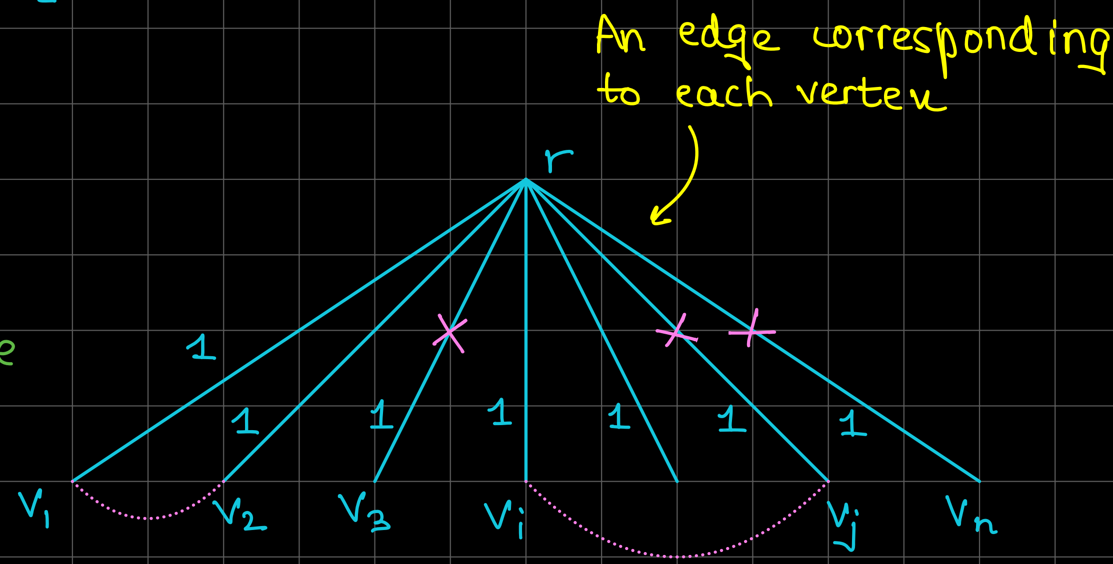


\# Saturated edges form a multicut

$G = (V, E)$

$(v_i, v_j) \in E$

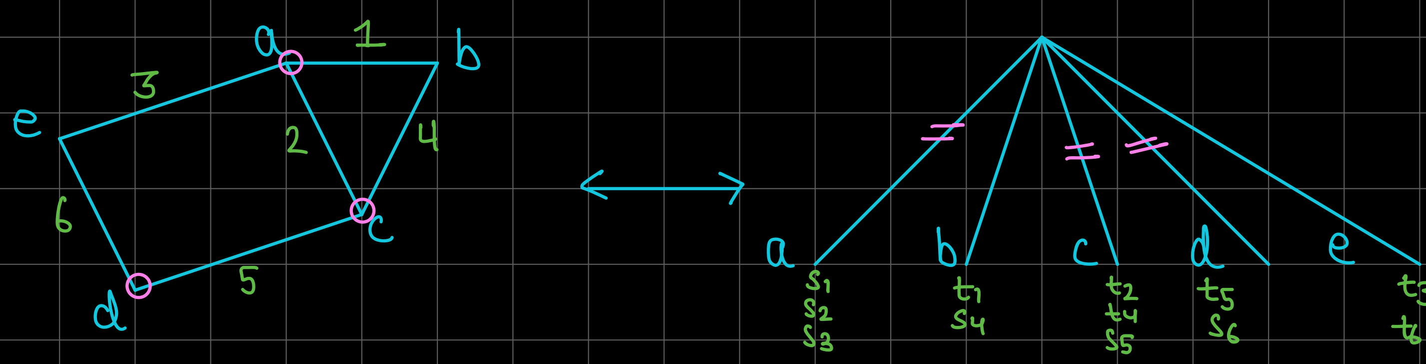
edges in G are s-t pairs in T



\# There is a 1-1 correspondence b/w a vertex cover in G and a multicut in T.

set of vertices that cover every edge

\> If I can get a $2-\epsilon$ approximation for a multicut in T then I can get a $2-\epsilon$ approximation for a vertex cover in G (not known)



Max Multicommodity Flow :

P_i : set of all paths b/w s_i & t_i

$\mathcal{P} = \bigcup_i P_i$

f_P is the flow along path $P \in \mathcal{P}$

$$\max \sum_{P \in \mathcal{P}} f_P \quad \text{subject to: } \forall e \in E, \sum_{P \in \mathcal{P}, e \in P} f_P \leq c_e$$

required to ensure +ve flows only $\quad \{ \forall P \in \mathcal{P}, f_P \geq 0$

Dual LP

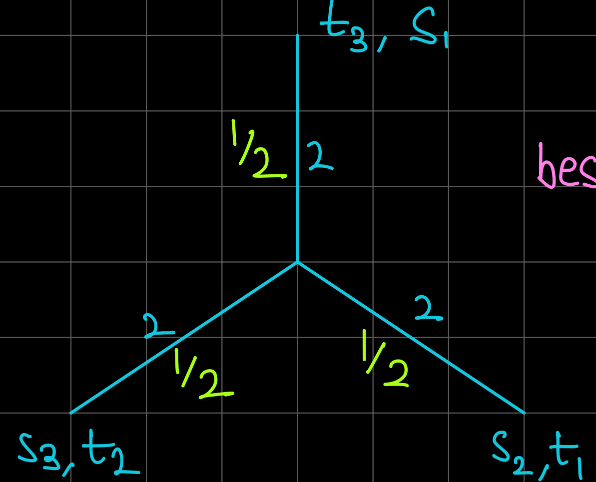
$$\sum_{e \in E} l_e \sum_{P \in \mathcal{P}, e \in P} f_P \leq \sum_{e \in E} l_e c_e$$

$$\sum_{P \in \mathcal{P}} f_P \leq \sum_{P \in \mathcal{P}} f_P \sum_{e \in P} l_e \leq \sum_{e \in E} l_e c_e$$

Shortest $s_i - t_i$ path is 1

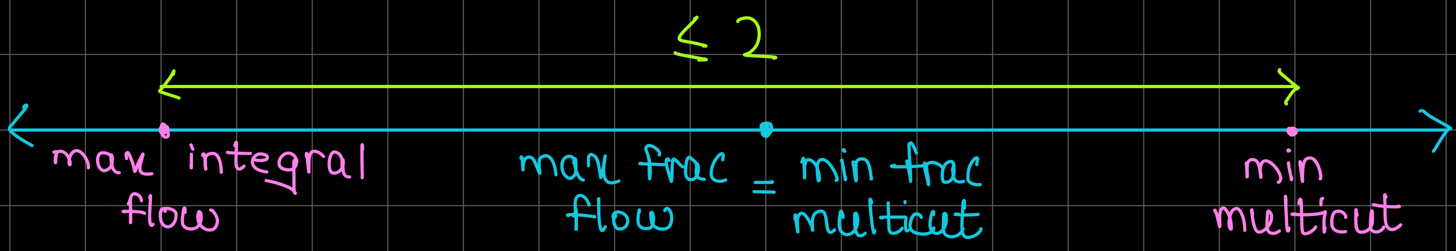
$$\min \sum_{e \in E} l_e c_e \quad \text{subject to: } \forall P \in \mathcal{P}, \sum_{e \in P} l_e \geq 1$$

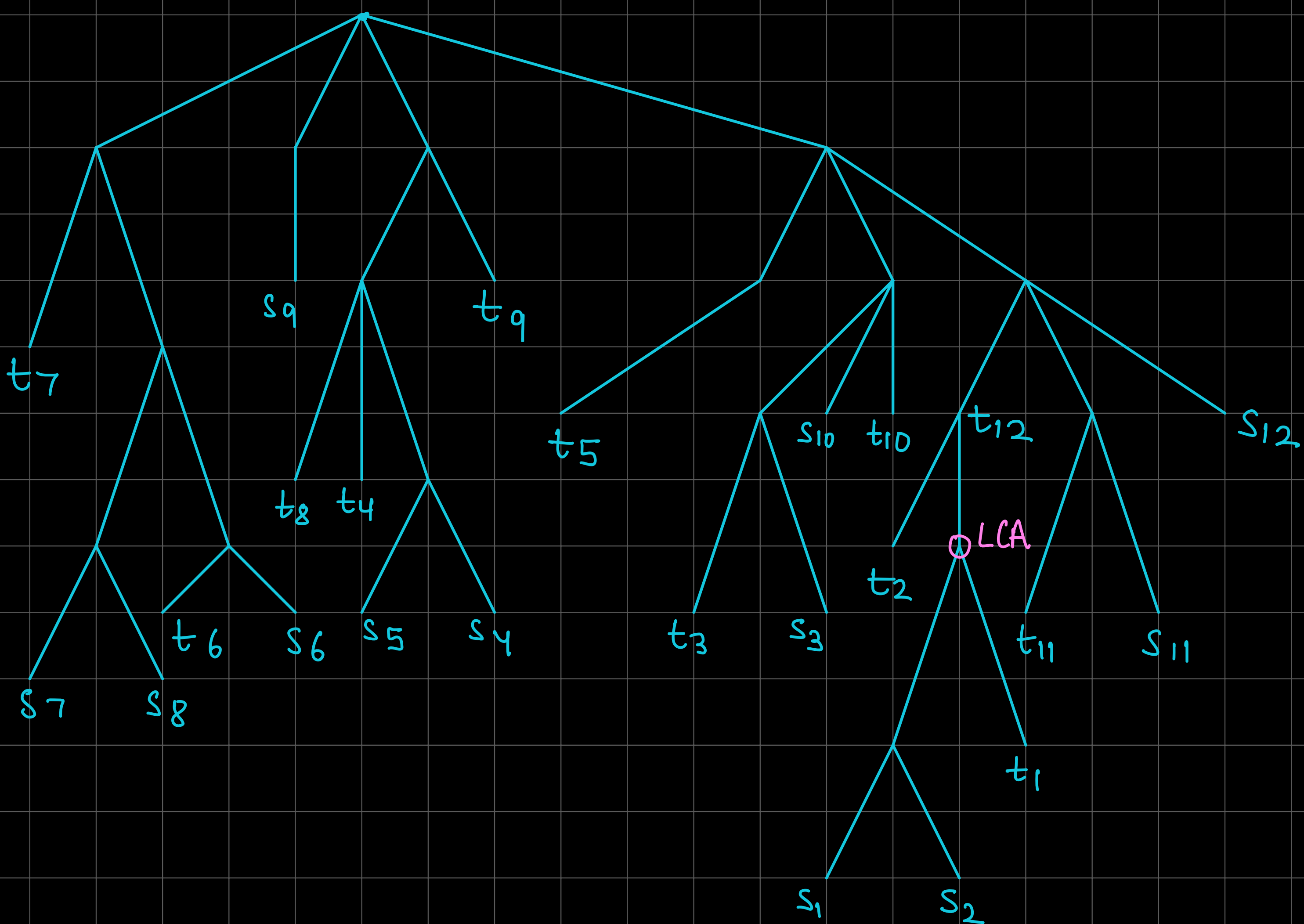
$$\forall e \in E, \quad l_e \geq 0$$



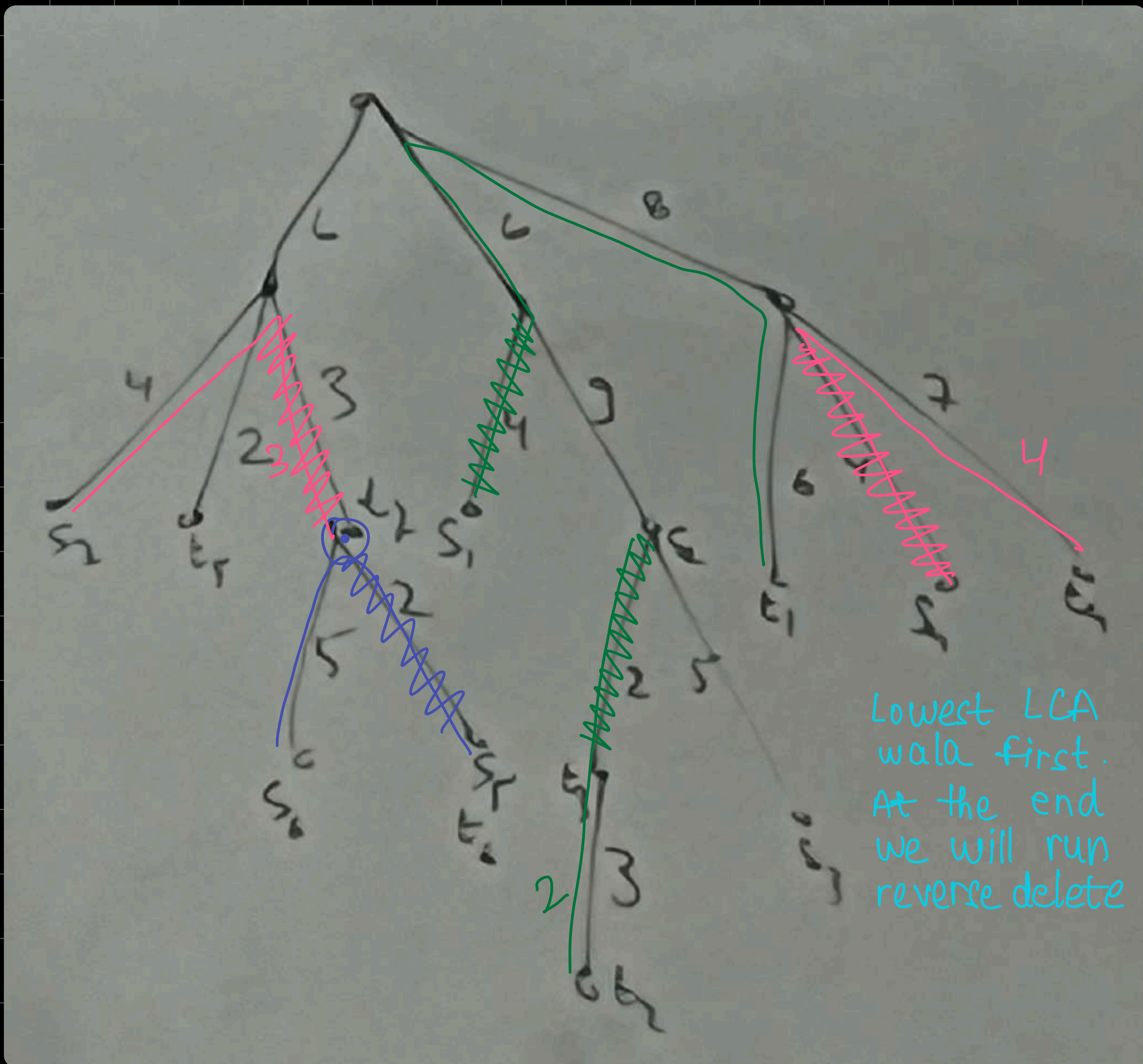
best solution to dual = 3

Consider an integer solution to this LP, $\ell_e \in \{0, 1\}$,
edges w/ $\ell_e = 1$ form a multicut





> Reverse delete



after reverse delete

$$\text{multicut}(F') \leq \sum_{e \in F'} c_e \leq 2\,(\text{flow computed})$$

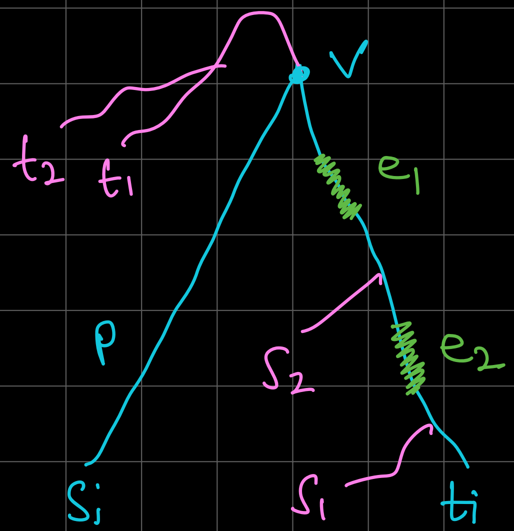
$$c_e = \sum_{p : e \in \mathcal{P}} f_p$$

$$\sum_{e \in F'} \sum_{p : e \in \mathcal{P}} f_p \leq 2 \sum_{p \in \mathcal{P}} f_p$$

$$\sum_{p \in \mathcal{P}} f_p * |F' \cap p| \leq 2 \sum_{p \in \mathcal{P}} f_p$$

$\Uparrow$ trivially $\qquad$ $\cancel{\Uparrow}$ not necessary

$$\forall p : f_p > 0 \quad |F' \cap p| \leq 2$$



e_2 was added to F before e_1 (necessary)

e_1 is separating some path that e_2 isn't

$P \in \mathcal{P}_i$ is a path s.t. $f_p > 0$

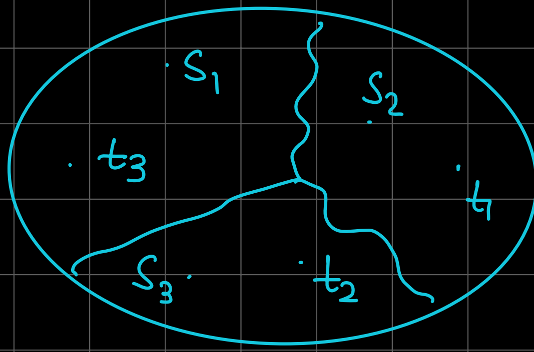
The edges e_1, e_2 became tight due to some s_i-t_i pair after P $\therefore$ The LCA must have been more above v hence having e_1 is sufficient

only then we could have sent any flow

Multicut

$G = (V, E)$, $c : E \to \mathbb{R}^+$

(s_i, t_i) , $i \in [k]$



min-multicut

max multicommodity flow

trees

$$\frac{\text{min multicut}}{2} \leq \begin{matrix}\text{max}\\ \text{multicommodity}\\ \text{flow}\end{matrix} \leq \text{min multicut}$$

graphs

$$\frac{\text{min multicut}}{O(\log k)} \leq \begin{matrix}\text{max}\\ \text{multicommodity}\\ \text{flow}\end{matrix} \leq \text{min multicut}$$

$\mathcal{P}_i$: set of paths from s_i to t_i

$\mathcal{P}$: $\bigcup_i \mathcal{P}_i$

f_p : flow along path p

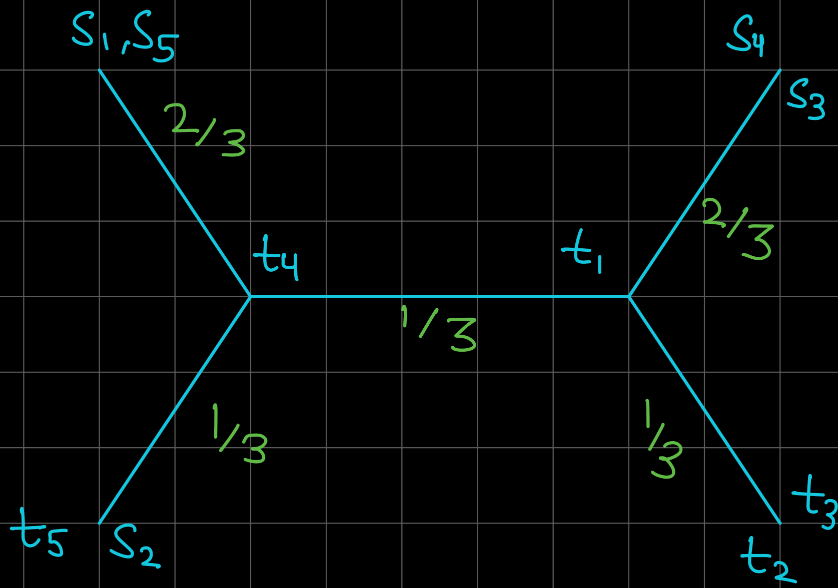
$$\max \sum_{p \in \mathcal{P}} f_p$$

$$\text{s.t.} \quad \sum_{p : e \in p} f_p \leq c_e \ , \quad f_p \geq 0$$

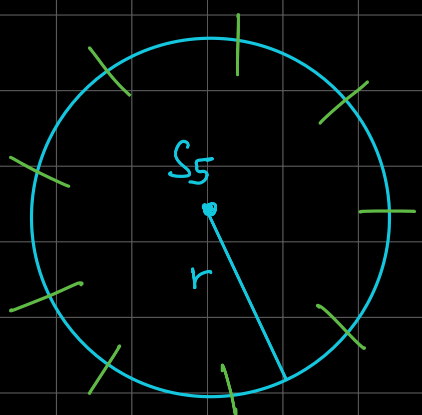
$$\min \sum_{e \in E} l_e c_e$$

$$\text{s.t. } \forall P : \sum_{e \in P} l_e \geq 1 \qquad \forall i \; l(s_i, t_i) \geq 1$$

$$l_e \geq 0$$



Region Growing Algorithm



$B(s_5, r)$

$\cdot\ s_1$

$\cdot\ t_5$

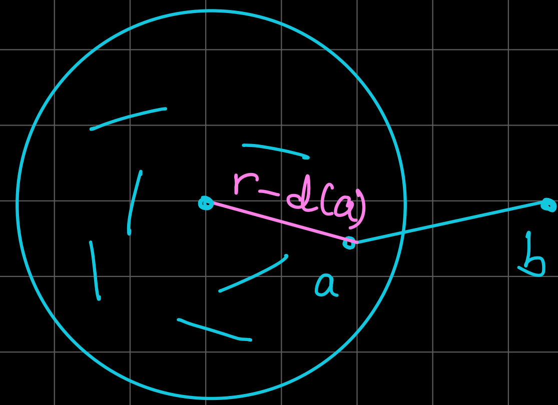
$\cdot\ t_2$

$r_i < 1/2$

because every pair is atleast 1 distance apart, so this ensures no s_i, t_i pair in the ball

Volume of the ball

$$V(u,r) = \sum_{e \in B(u,r)} \bar{l}_e c_e$$

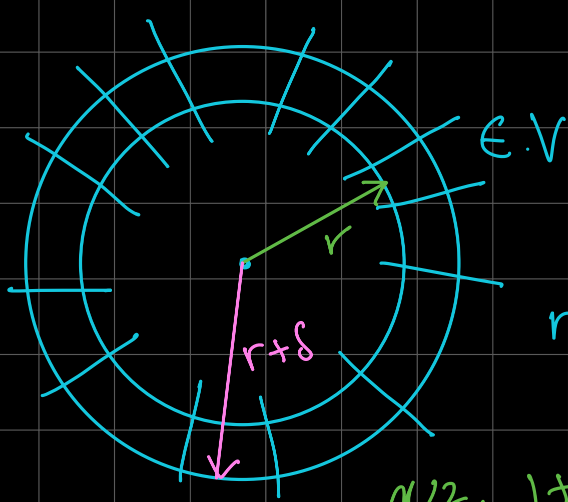


- keep growing ball while cut > ε. volume
- Stop when cut corresponding to ball ≤ volume of the ball.

cost of multicut $\leq \varepsilon \cdot$ total volume

$$\leq \varepsilon \cdot 2 \sum_{e \in P} l_e c_e$$

$= 2\varepsilon \cdot$ max multicommodity flow



$8 \ln(k+1)$ max flow

new volume = old volume + δ.cut

$$\geq V + \delta.\epsilon.V$$

$$= V(\underbrace{1+\delta.\epsilon}_{\epsilon.\delta})$$

$$\geq V.\, e^{\epsilon\delta}$$

$e^{x/2} \leq 1+x$

final volume $\geq V_0 e^{\epsilon r}$

$$V(u,r) = \sum_{e \in B(u,r)} \bar{l}_e c_e + F/2$$

$$F + \frac{F}{K} \geq \text{Final volume} \geq V_0 \cdot e^{\epsilon r}$$

$$= \frac{F}{K} \cdot e^{\epsilon r}$$

$$e^{\epsilon r} \leq K+1$$

$$\epsilon r \leq \ln(K+1)$$

$$r \leq \ln(K+1) \cdot \frac{1}{\epsilon}$$

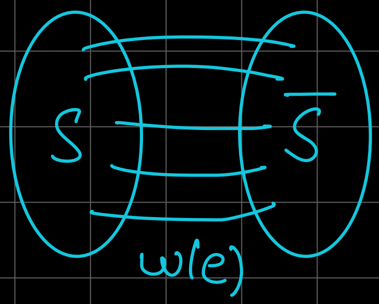
$$\epsilon = 2 \ln(K+1)$$

$$\Rightarrow \quad r \leq 1/2$$

Max Cut Problem

$G = (V, E)$, $\omega : E \to \mathbb{R}^+$

cut $(S, \underbrace{V \setminus S}_{\overline{S}})$ which has maximum cut



$$wt(S, \overline{S}) = \sum_{e \in \nabla(S)} w(e)$$

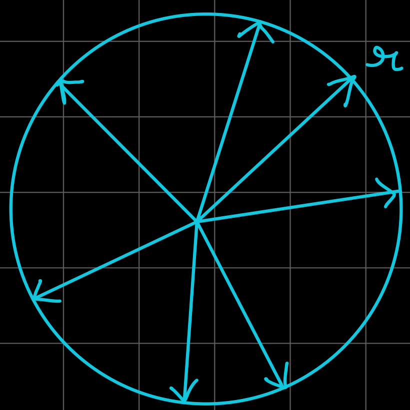
X is a positive semidefinite (p.s.d.) then $\forall y,\ y^T X y \geq 0$
$X = A^T A,\ \exists A$

infinitely many vectors

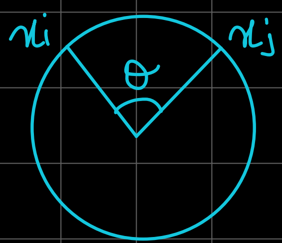
- Solve SDP
- Pick a random hyperplane through the origin
- Let $(S, \bar{S})$ be the cut obtained.

$v_i \overset{w_{ij}}{\text{———}} v_j$

OPT $\leq$ SDP value



$\frac{w_{ij}}{2}(1 - x_i \cdot x_j)$ is the contribution of edge (v_1, v_2) to SDP



> expected contribution of this edge to the random cut $\frac{\theta}{\pi} w_{ij}$

Prob (v_i, v_j are separated by random cut) $= \frac{\theta}{\pi}$

Determine largest α s.t. $\frac{\theta}{\pi} w_{ij} \geq \alpha \cdot \frac{w_{ij}}{2}(1 - \cos\theta)$

expected weight of cut $\geq \alpha \cdot$ sdp value $\geq \alpha \cdot$ OPT

$\geq 0.878 \cdot$ OPT

$$\alpha \leq \frac{2\theta}{(1-\cos\theta)\pi} \quad \forall\, \theta$$

$$\alpha = \min_{\theta} \frac{2\theta}{(1-\cos\theta)\pi} \, , \qquad \alpha = 0.878$$

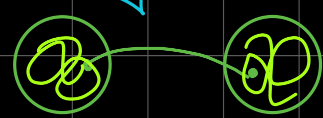
Sparsest Cut

sparsity of a cut S

$$S \subseteq V \qquad \text{sparsity}(S) = \frac{c(S,\bar{S})}{\min(|S|,|\bar{S}|)}$$

› sparsest cut is the cut of minimum sparsity

- Sparsity of a graph also called conductance
- also a (inverse) measure of the convergence time of a random walk



sp↓

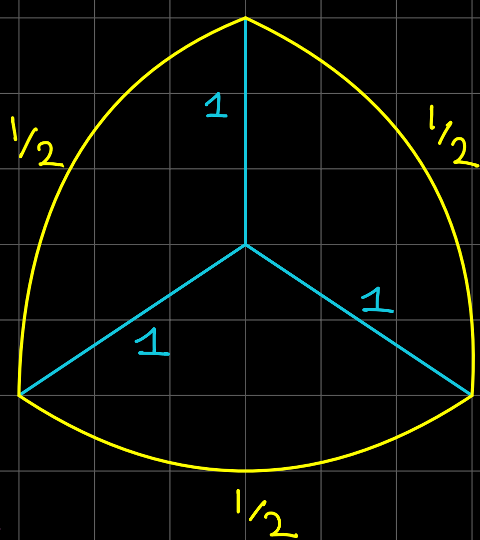
random walk will take a long time to escape to the other clique

$$\frac{\text{sparsity}}{n} \leq \min_{S \subseteq V} \frac{c(S,\bar{S})}{|S|\,|\bar{S}|} \leq \frac{\text{sparsity}}{n/2}$$

Generalised Sparsest Cut:

$d : F \to \mathbb{R}^+$ demand

$c : E \to \mathbb{R}^+$ supply



$$\text{generalised sparsity} = \frac{\text{capacity}}{\text{demand}}$$

$c(S,\bar{S}) = \sum_{e \in \nabla(S)} c(e)$, $\quad d(S,\bar{S}) = \sum_{e \in \nabla(S)} d(e)$

if $d_{ij} = 1 \ \forall i,j \in V$ then $\dfrac{c(S,\bar{S})}{d(S,\bar{S})} = \dfrac{c(S,\bar{S})}{|S|\,|\bar{S}|}$

capacity/demand relate to multi-commodity flow later

multicommodity flow

maximum multicommodity flow

$G = (V,E)$, $c: E \to \mathbb{R}^+$, (s_i, t_i), $i \in [K]$

- maximise total flow routed

$$\frac{\text{min multicut}}{4 \log_2 K} \leq \text{max m.c.f.} \leq \text{min multicut}$$

multicommodity flow with demands

$G = (V, E \cup F)$, $c: E \to \mathbb{R}^+$, $d: F \to \mathbb{R}^+$

Q. Can all the demand be routed concurrently?

$\forall S \subseteq V \quad c(S,\bar{S}) \geq d(S,\bar{S})$ cut condition is necessary but not sufficient multicommodity flow

- cut condition is necessary & sufficient when
  - (V,E) is a tree
  - (V,E) is planar & all the edges of F have endpoints in the outer face [Okamura-Seymour]
  - $(V, E\cup F)$ is planar [Seymour]

If cut condition is met, what fraction of flow can be routed?

LP to check if a multicommodity flow instance with demands is feasible:

P_i : set of paths from s_i to t_i $\qquad \mathcal{P} = \bigcup_i P_i$

f_p : flow on path P

max 0 → feasibility question

$$\forall i \quad \Big(-\sum_{p\in P_i} f_p \le -d_i\Big)\ z_i,$$

$$\forall e \quad \Big(\sum_{p:e\in p} f_p \le c_e\Big)\ l_e, \quad f_p \ge 0 \quad \forall p\in\mathcal{P}$$

Dual

$$\min \sum l_e c_e - \sum d_i z_i$$

l_e: length of an edge

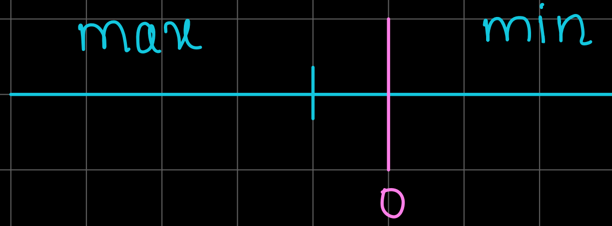
$$p\in P_i \qquad -z_i + \sum_{e\in p} l_e \ge 0 \qquad \Bigg\| \qquad z_i \le \sum_{e\in p} l_e,\ p\in P_i$$

$$l_e \ge 0,\ z_i \ge 0 \qquad \Bigg\| \qquad z_i = sp_l(s_i, t_i)$$

shortest path

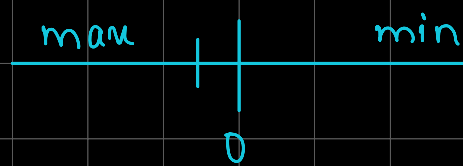
We don't know if the Primal is feasible, but we know if both are feasible then Primal = Dual.
If Primal is feasible then it equals 0

infeasible if dual < 0



$$\sum l_e c_e < \sum d_i \, sp_l(s_i, t_i)$$

$$\sum l_e c_e \leq \sum d_i \, sp_l(s_i, t_i)$$



$$\max \lambda$$

$$\forall i \quad \Big(-\sum_{p \in P_i} f_p + \lambda d_i \leq 0 \Big) \, z_i$$

$$\forall e \quad \Big(\sum_{p : e \in p} f_p \leq c_e \Big) \, l_e$$

$$f_p \geq 0 \quad \forall \, p \in P$$

$$\min \sum_e l_e c_e$$

s.t.

$$p \in P_i \qquad -z_i + \sum_{e \in P} l_e \geq 0$$

$$\sum_{i=1}^{K} z_i d_i \leq 1$$

$$z_i \leq \sum_{e \in P} l_e, \quad p \in P_i$$

$$z_i = sp_l(s_i, t_i)$$

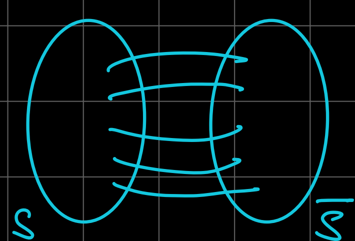
$$\sum_{i=1}^{K} d_i \, sp_l(s_i, t_i) \geq 1$$

$$\frac{\min \sum_{e \in E} l_e c_e}{\sum_{i=1}^{K} d_i \, sp_l(s_i, t_i)} = \frac{\sum_{e \in E} l_e c_e}{\sum_{e \in F} l_e d_e}$$

$$\min \frac{\sum_e l_e c_e}{\sum_e l_e d_e}, \quad l \text{ is a metric}$$

Generalised sparsity of set $(S, \bar{S}) = \dfrac{c(S, \bar{S})}{d(S, \bar{S})}$

generalizes sparsity since if $\forall\, i, j \;\; d(i,j) = 1$



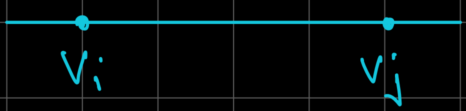
min gen sparsity of $(S,\bar{S}) = \dfrac{c(S,\bar{S})}{|S||\bar{S}|} =$ sparsity of $(S,\bar{S})$

find a set S that minimizes $\dfrac{c(S,\bar{S})}{d(S,\bar{S})}$

$$\min \frac{\sum_{e\in E} \ell_e c_e}{\sum_{e\in E} \ell_e d_e}$$ where ℓ is a ~~cut~~ relax metric

$\leq$ g-sparsest cut in G

We solve the LP to obtain l which is a metric.
Suppose l is L_1 embeddable in 1 dimension?



$l(v_i, v_j)$ = distance b/w v_i & v_j on the line

l can be expressed as a linear combination of cut metrics.

specified as distance b/w points ($^nC_2$)

- l is the given metric
- c_i is a cut metric

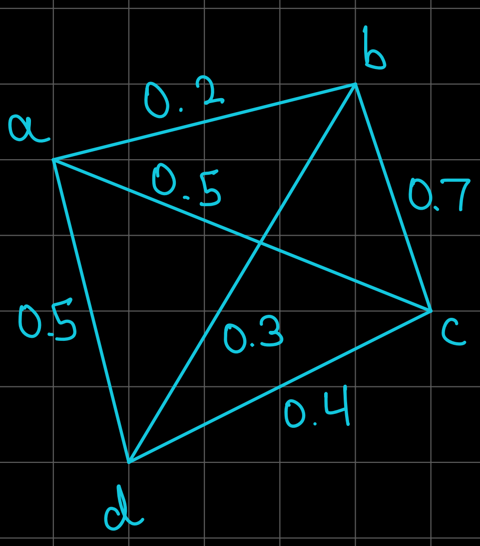
$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

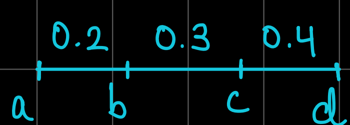
$$l = \sum \lambda_i c_i, \quad \lambda_i \geq 0$$

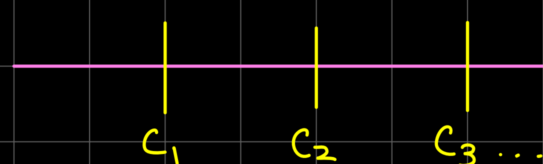
$$\frac{\sum_{e \in E} l_e c_e}{\sum_{e \in E} l_e d_e} = \sum_{e \in E} c_e \left(\sum \lambda_i c_i\right)$$

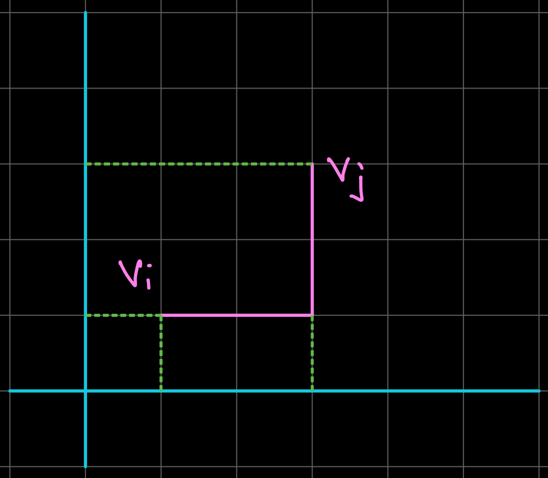
$$= \frac{\sum \lambda_i \, c(S_i, \overline{S_i})}{\sum \lambda_i \, d(S_i, \overline{S_i})}$$ where S_i is the i^{th} cut

$$\geq \min_{S_i} \frac{c(S_i, \overline{S_i})}{d(S_i, \overline{S_i})}$$









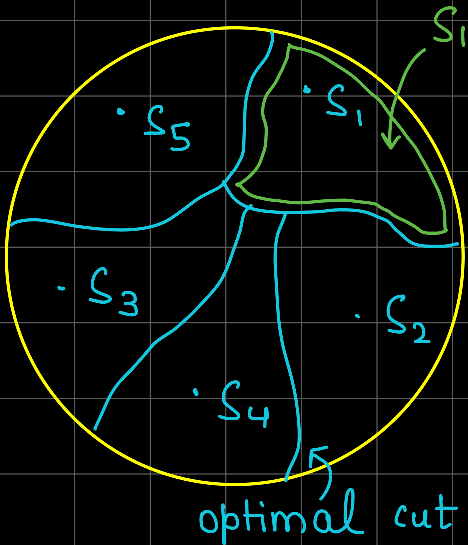
l is L_1 embeddable in $\mathbb{R}^2$

manhattan distance

Every n-point metric is l_1-embeddable in $\mathbb{R}^n$ with distortion $O(\log_2 n)$ [Burgain]

Given a metric l on $V \times V$, $\phi : V \to \mathbb{R}^d$ is an embedding

Multiway Cut Problem



$G = (V, E)$, $c : E \to \mathbb{R}^+$
$T \subseteq V$, $T = \{s_1, s_2, \ldots, s_k\}$

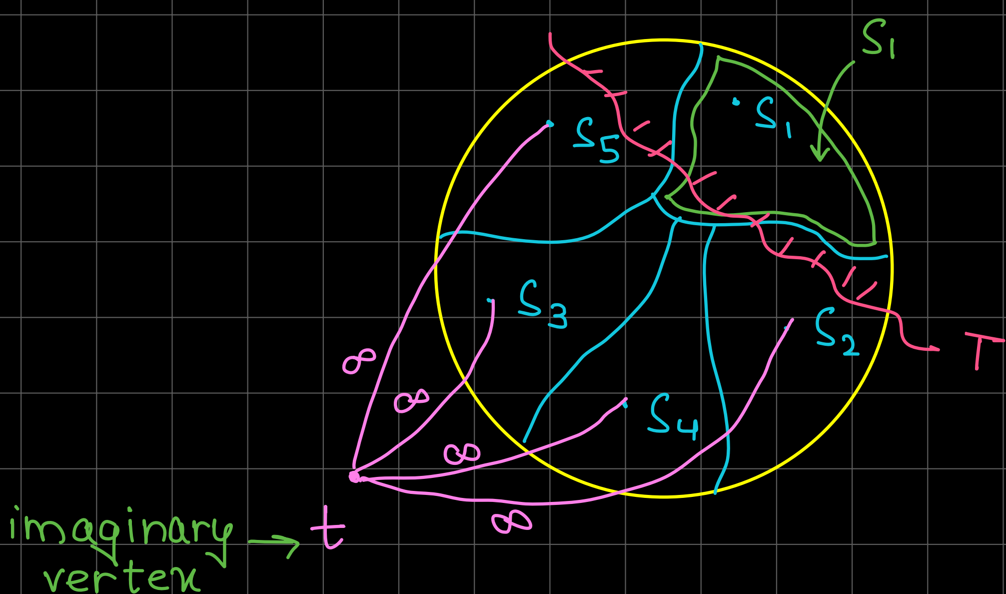
Find $E' \subseteq E$ s.t. there is no path between any pair of terminals in $E \setminus E'$

Find a multiway-cut of minimum capacity
NP-hard for $k \geq 3$

$$2\,OPT = \sum c(\nabla(S_i))$$

S_i = set containing s_i in the optimal multiway cut

T_i = set containing s_i in the minimum cut separating s_i from $T \setminus \{s_i\}$

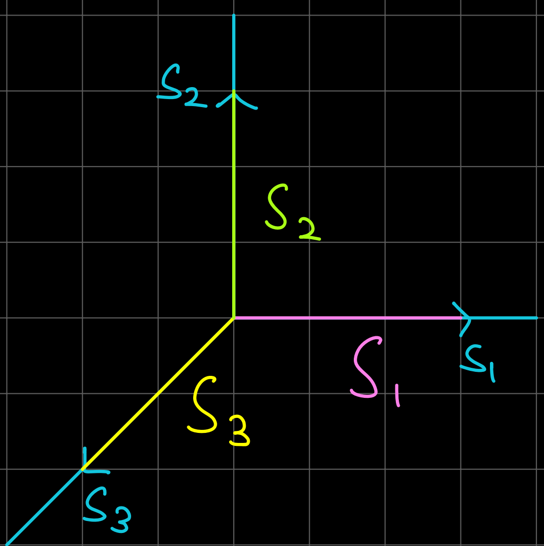


$c(T_i) \leq c(\nabla(S_i))$ since T_i is the min-cut separating s_i & $T \setminus \{s_i\}$

$$c\left(\bigcup_{i=1}^{k} \nabla(T_i)\right) \leq \sum_{i=1}^{k} c(\nabla T_i) \leq \sum_{i=1}^{k} c(\nabla(S_i))$$

$$= 2 \cdot OPT$$

For all



$K = 3$

$s_1 = (1, 0, 0)$

$s_2 = (0, 1, 0)$

$s_3 = (0, 0, 1)$

$s_i = e_i \leftarrow$ unit vector in ith direction

$x_i \in \mathbb{R}^k$ is a vector associated with $v_i \in V$

$$\|x_i\|_1 = 1 \quad \forall v_i \in V \quad , \quad x_{s_i} = e_i$$

$$\sum_{(i,j) \in E} \frac{1}{2} c_{ij} \|x_i - x_j\|_1$$

Linear Program:

$\forall\, v_i \in V, \qquad \sum_{j=1}^{[K]} x_i^j = 1$

$\forall\, v_i \in V, j \in [K] \qquad x_i^j \geq 0$

$j \in [K] \qquad x_{s_i}^i = 1$

$x_i^\ell - x_j^\ell \leq y_{ij}^\ell$

$x_j^\ell - x_i^\ell \leq y_{ij}^\ell$

objective: $\min \frac{1}{2} \sum_{(i,j) \in E} c_{ij} \sum_{\ell=1}^{k} y_{ij}^\ell$

Take a random permutation of $S_1, \dots S_k$

Take a random $r \in (0,1)$

r.v. $X_{uv} = \begin{cases} 1, & \text{if } (u,v) \text{ is cut} \\ 0, & \text{o.w.} \end{cases}$

$\text{Prob}[(u,v) \text{ is cut}] \leq \frac{3}{4} \| x_u - x_v \|$

$$E[\text{capacity of cut}] = E\left[\sum_{u,v} c_{uv} \cdot X_{uv}\right]$$

$$= \sum_{u,v} c_{uv} E[X_{uv}]$$

$$= \sum_{u,v} c_{uv} P(X_{uv} = 1)$$

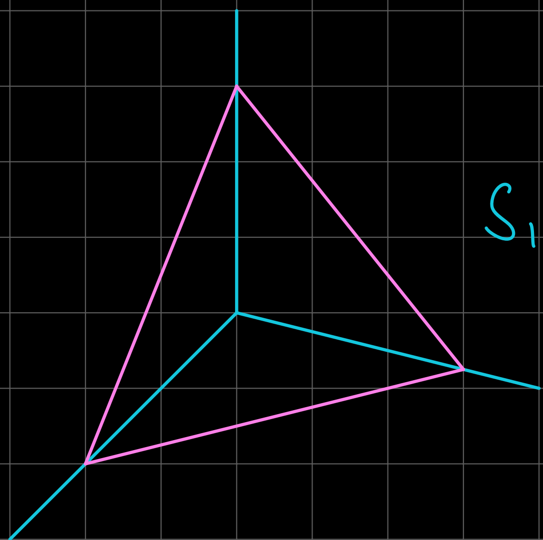
$$= \sum_{u,v} c_{uv} \cdot \frac{3}{4} \| x_u - x_v \|_1$$

$$E[\text{capacity of cut}] = \frac{3}{2}\sum_{u,v} \frac{1}{2} c_{uv} \|x_u - x_v\|_1$$

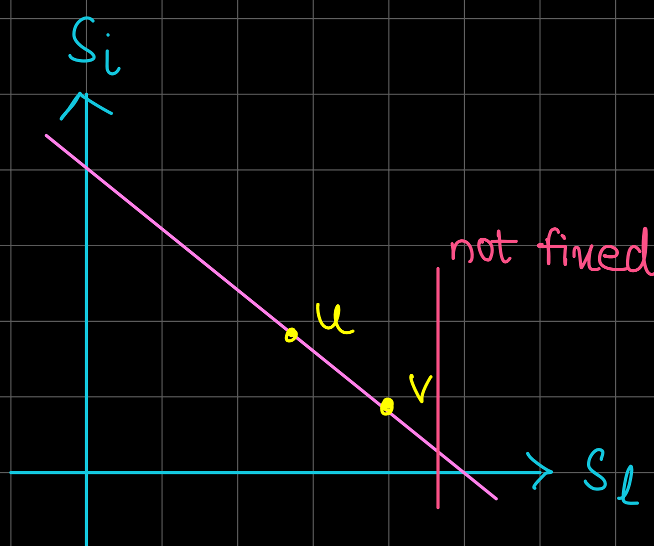
$$\leq \frac{3}{2} \cdot OPT$$

For edge (u,v) let ℓ be the index s.t. $\min(1-x_u^{\ell},\ 1-x_v^{\ell})$ is smallest

think of S_ℓ as being closest to edge (u,v)



$S_{10}\ S_j \ldots S_\ell \ldots S_i$



\# an edge gets fixed if atleast one of its endpoints gets included in $B(S_i, r)$

$\uparrow$ ball

$j > \ell$ will not cut (u,v) since ℓ is the closest

$$P[(u,v) \text{ is cut}] \leq P[(u,v) \text{ is cut by } S_\ell]$$

$$+ \sum_{j=1, j\neq \ell}^{K} \text{Prob}[(u,v) \text{ is cut by } S_j \,/\, S_j \text{ occurs before } S_i \text{ in the permutation}] \times P[S_j \text{ occurs before } S_\ell]$$

cond. prob.

$$= |x_u^\ell - x_v^\ell| + \frac{1}{2} \sum_{j=1, j\neq \ell}^{K} |x_u^j - x_v^j|$$

$$= \frac{1}{2} |x_u^\ell - x_v^\ell| + \frac{1}{2} \sum_{j=1}^{K} |x_u^j - x_v^j|$$

$$= \frac{1}{2} |x_u^\ell - x_v^\ell| + \frac{1}{2} \|x_u - x_v\|_1$$

$$\leq \frac{3}{4} \|x_u - x_v\|_1 \qquad \left[\text{using } |x_u^\ell - x_v^\ell| < \frac{1}{2} \|x_u^j - x_v^j\| \right]$$

Approximating metrics with tree metrics

$d : V \times V \to \mathbb{R}^+$

embed d into $\mathbb{R}^k$ s.t. $\phi : V \to \mathbb{R}^k$

distortion

$\forall u, v \in V \qquad d(u,v) \leq \| \phi(u) - \phi(v) \|_1 \leq K \cdot d(u,v)$

Tree Metric

- A metric supported by a tree

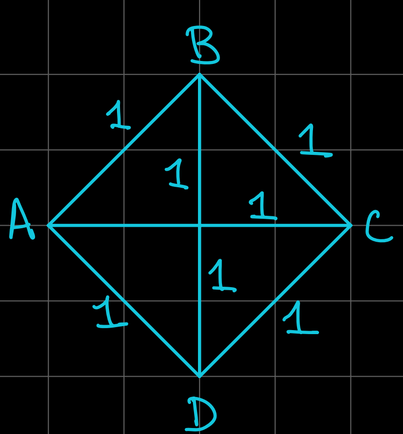
\# A tree metric (V', T) for a set of vertices V is a tree T defined on some set of vertices $V' \supseteq V$, together with non-negative lengths on each edge of T.

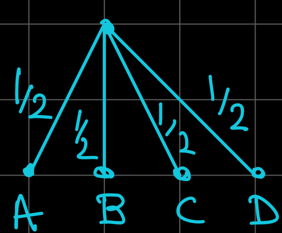
$u, v \in V'$, T_{uv}: length of the unique shortest path between u & v in T

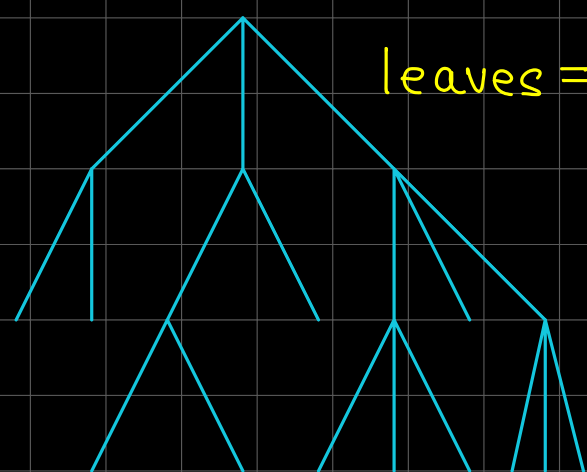
We want

distortion

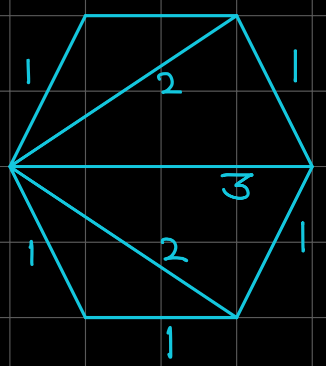
$$d_{u,v} \leq T_{u,v} \leq K \cdot d_{u,v} \quad \forall u, v \in V$$







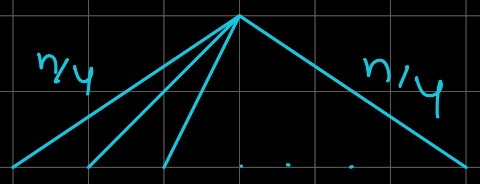
leaves = V



cycle with n vertices

cannot be approximated with a tree metric with distortion $o(n)$

little-o



dist. b/w any $2 = \frac{n}{2}$

If we try to approximate a cycle metric with a star then we get a distortion of $n/2$

We give a randomized algorithm for producing a tree T such that for each $u, v \in V$, $d_{uv} \le T_{uv}$ and $E[T_{uv}] \le \underbrace{O(\log n)}_{\text{Expected Distortion}} d_{uv}$

d

$T_1 \quad T_2 \quad \ldots \quad T_\ell$

$P_1 \quad P_2 \quad \quad \quad P_\ell$

Probability Dist.

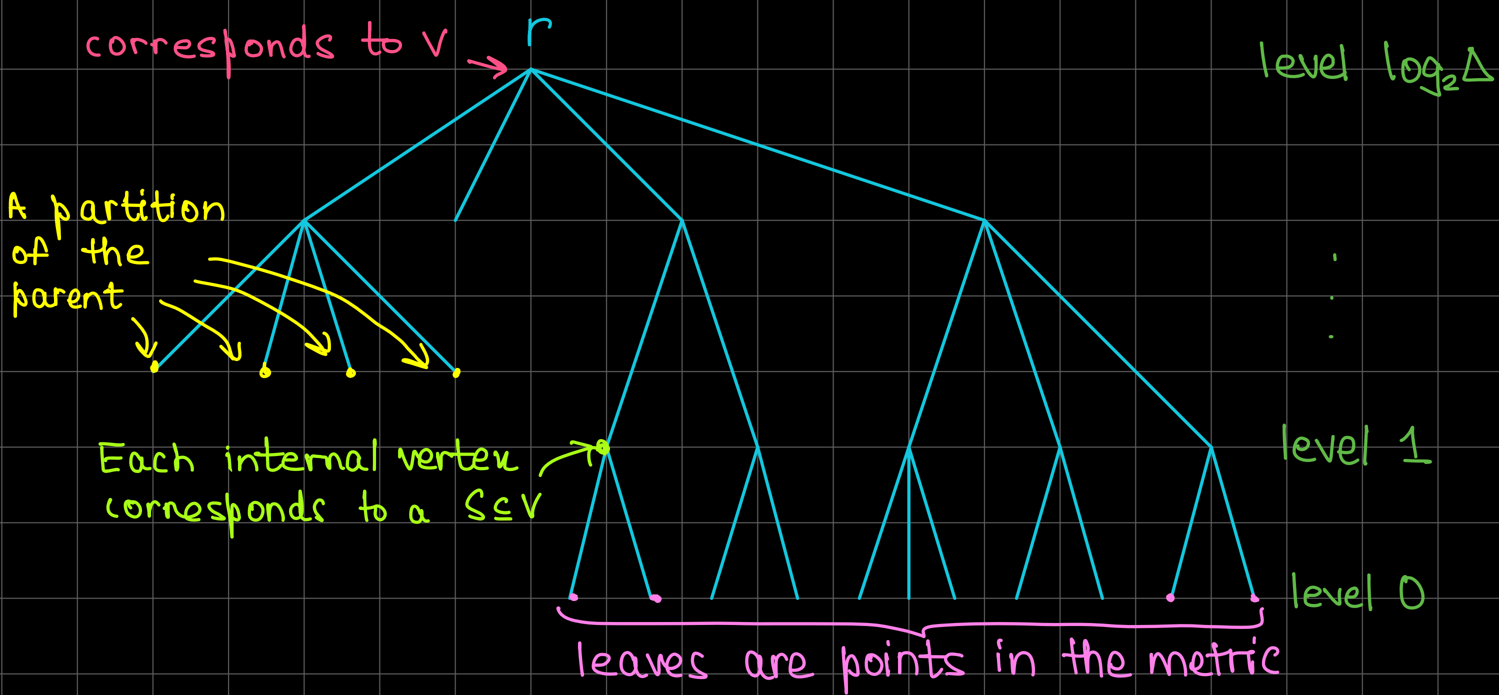
$l_i(u,v)$ is distance u,v in T_i

$l(u,v) = \sum_i P_i \, l_i(u,v)$

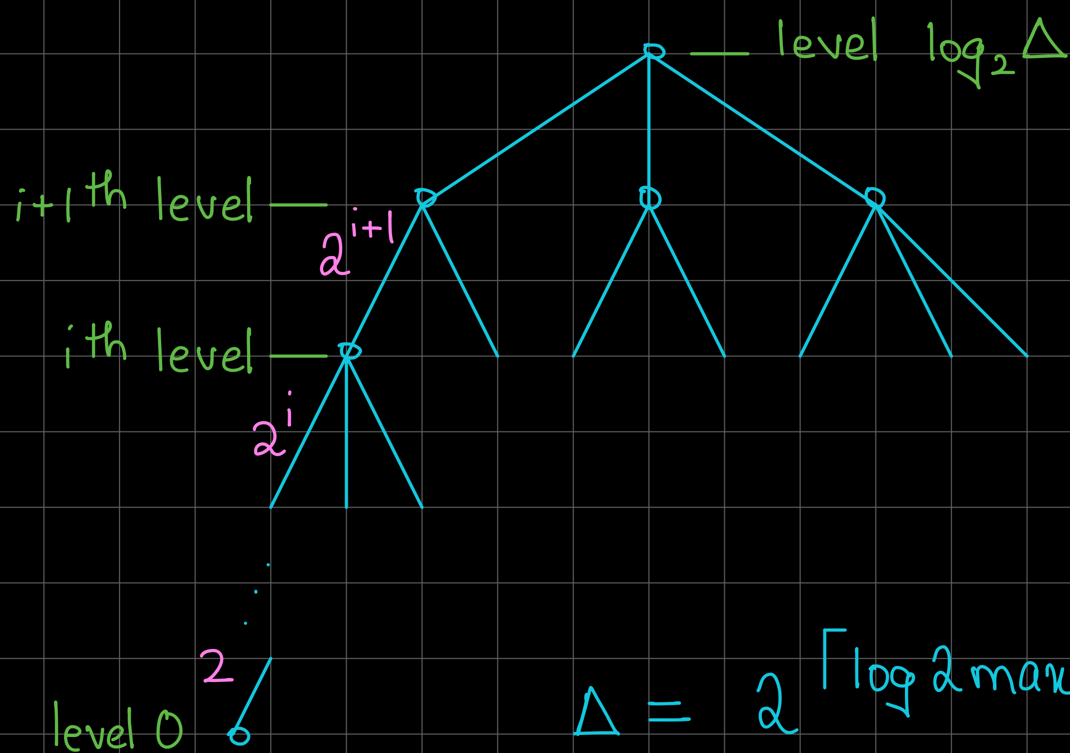
Δ : smallest power of 2 greater than $2 \max_{u,v} d_{uv}$

Hierarchical Cut Decomposition:

Rooted Tree with $\log_2 \Delta + 1$ levels



Scale distances to ensure $\min_{u,v} d(u,v) = 1$



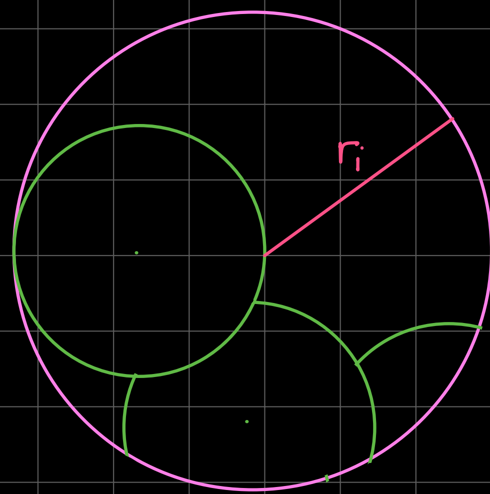
$$\Delta = 2^{\lceil \log 2 \max d(u,v) \rceil}$$

S_i is a set at level i

- each node at level i corresponds to a set considered in a ball of radius r_i (centred on some vertex)

$r_i = r \cdot 2^i, \; r \in [1/2, 1)$

$2^{i-1} \leq r_i < 2^i$



Each leaf is considered at level 0 which is in a ball of radius $< 2^0 = 1$ centred at some u. Hence ball contains only u as $\min_{u,v} d_{u,v} = 1$

V is contained in a ball of radius $\geq \Delta/2$. Hence all the vertices are contained in it since $\frac{\Delta}{2} \geq \max_{u,v} d_{u,v}$

$B(w, r_i)$: Ball of radius r_i drawn around w

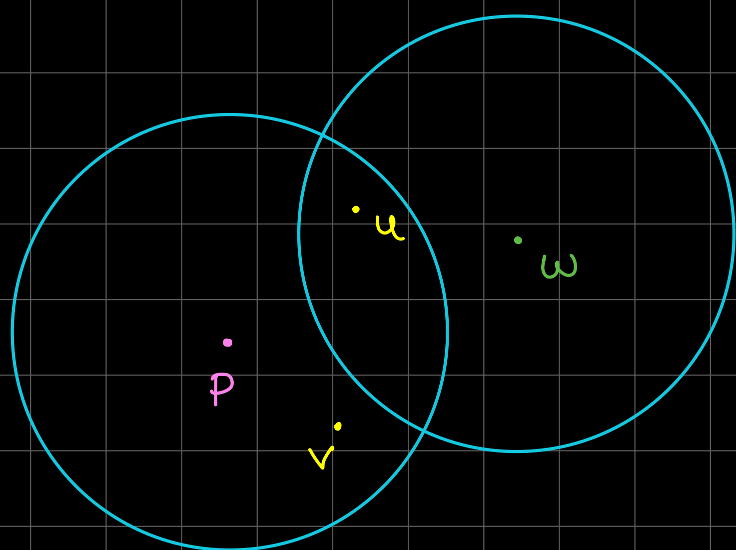
$$X_{iw} = \begin{cases} 1, & \text{if } B(w,r_i) \text{ separates } u,v \\ 0, & \text{otherwise} \end{cases} \Bigg\} \text{only depends on } r_i$$

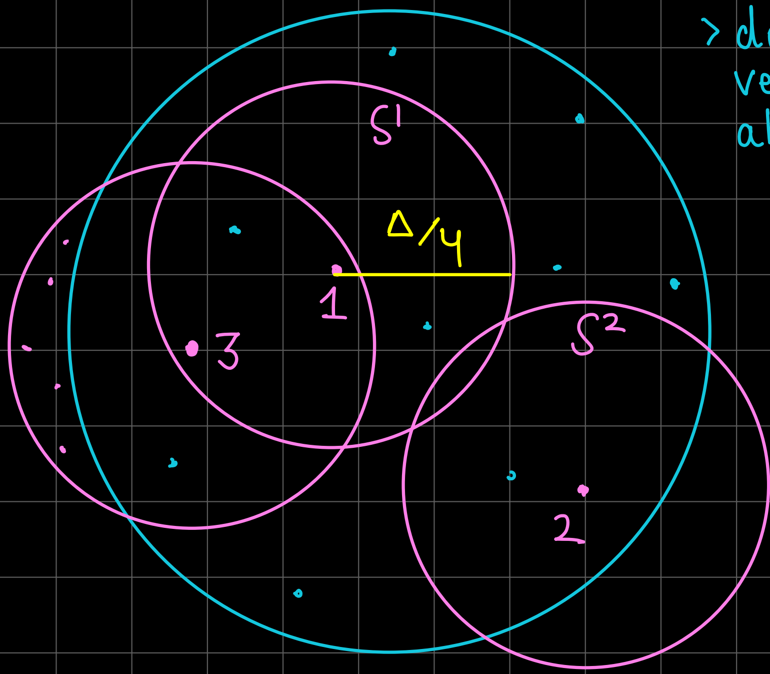
$$S_{iw} = \begin{cases} 1, & \text{if } B(w,r_i) \text{ includes atleast one of } u,v \text{ at level } i \\ 0, & \text{otherwise} \end{cases} \Bigg\} \text{depends on the permut}^n$$

- Consider a random permutation π of V
- Set $r_i = r \cdot 2^i$ where r is uniformly drawn in $[1/2, 1)$

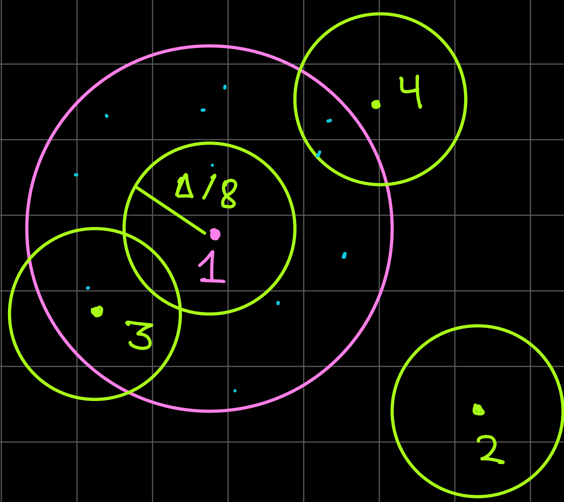
Pick a random permutation π of V
Set Δ to the smallest power of 2 greater than $2\max_{u,v} d_{uv}$
Pick $r_0 \in [1/2, 1)$ uniformly at random; set $r_i = 2^i r_0$ for all $i : 1 \leq i \leq \log_2 \Delta$
// $\mathcal{C}(i)$ will be the sets corresponding to the nodes at level i; the sets partition V
$\mathcal{C}(\log_2 \Delta) = \{V\}$
Create tree node corresponding to V
for $i \leftarrow \log_2 \Delta$ downto 1 **do**
  $\mathcal{C}(i-1) \leftarrow \emptyset$
  for all $C \in \mathcal{C}(i)$ **do**
    $S \leftarrow C$
    for $j \leftarrow 1$ to n **do**
      if $B(\pi(j), r_{i-1}) \cap S \neq \emptyset$ **then**
        Add $B(\pi(j), r_{i-1}) \cap S$ to $\mathcal{C}(i-1)$
        Remove $B(\pi(j), r_{i-1}) \cap S$ from S
    Create tree nodes corresponding to all sets in $\mathcal{C}(i-1)$ that are subsets of C
    Join these nodes to node corresponding to C by edge of length 2^i

If P comes before w, $S_{i,w} = 0$ since both u, v are in ball of P, but $X_{i,w} = 1$ still

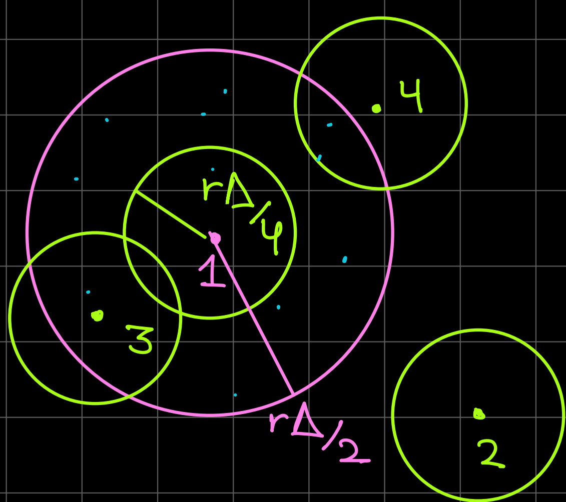




> do not assign vertices which have already been assigned



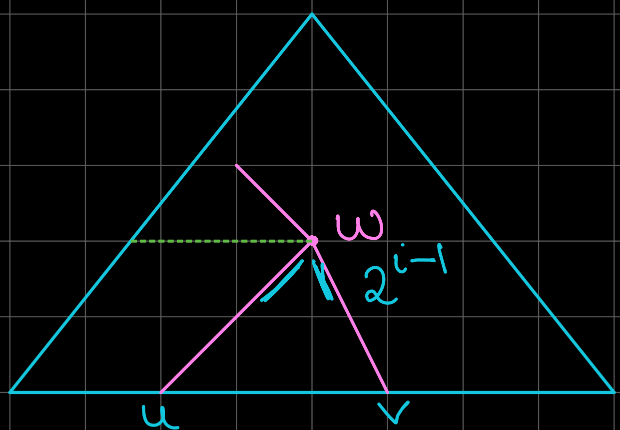
Pick a random $r \in [1/2, 1]$

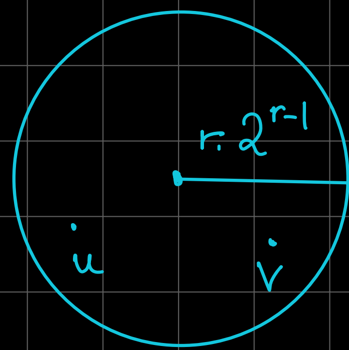


$l(u,v)$ is distance between (u,v) in the distribution over trees

$d(u,v) < l(u,v)$

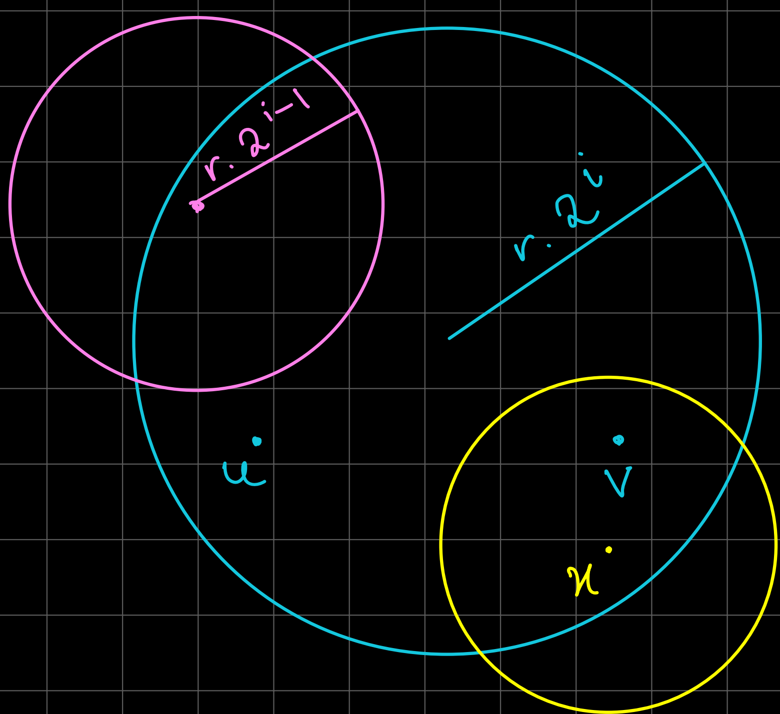
$d(u,v) \leq l(u,v) \leq \kappa \cdot d(u,v)$

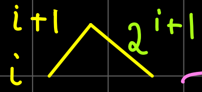




$T(u,v) = 2^{i+1} - 2$

$d(u,v) < r \cdot 2^{i} < 2^{i} < T(u,v)$





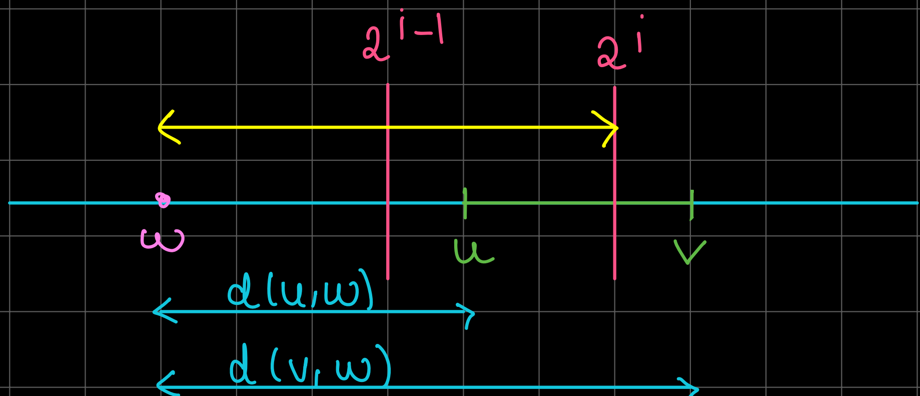
$$T_{u,v} \le \overbrace{\max_i \mathbb{1}(\exists\, w\in V : X_{iw}\cap S_{iw})}^{\text{highest level where } u,v \text{ in diff. branches}} \cdot \underbrace{2^{i+2}}_{=2\cdot 2^{i+1}}$$

$$T_{u,v} \le \sum_{i=0}^{\log_2\Delta} \mathbb{1}(\exists\, w\in V : X_{iw}\cap S_{iw}) \cdot 2^{i+2}$$

$$\le \sum_{i=0}^{\log\Delta}\sum_{w\in V} \mathbb{1}(X_{iw}\cap S_{iw}) \cdot 2^{i+2}$$

$$E(T_{u,v}) \le \sum_{i=0}^{\log\Delta}\sum_{w\in V} P(X_{iw}\cap S_{iw}) \cdot 2^{i+2}$$

$$\sum P(X_{iw}\cap S_{iw})\cdot 2^{i+2} = \sum P(S_{iw}\mid X_{iw})\cdot P(X_{iw})\cdot 2^{i+2}$$
$$= P(S_{iw}\mid X_{iw}) \sum_i P(X_{iw})\cdot 2^{i+2}$$

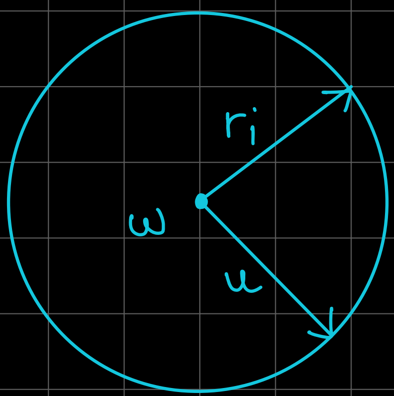


$$\sum_i P(X_{iw})\cdot 2^{i+2} = \sum_i \frac{[d(u,w), d(v,w)] \cap [2^{i-1}, 2^i]}{2^{i-1}} \cdot 2^{i+2}$$

$$= 8\,|d(v,w) - d(u,w)| \le 8\, d(u,v)$$

$$\sum P(X_{iw} \cap S_{iw}) \cdot 2^{i+2} \leq \sum_{w} P(S_{iw} \mid X_{iw}) \cdot 8 \cdot d(u,v)$$

$$= 8 \cdot d(u,v) \sum_{w} P(S_{iw} \mid X_{iw})$$



Order vertices by increasing distance from (u,v) : $\min\{d(u,w), d(v,w)\}$

$$q_1 \quad q_2 \quad \cdots \quad \underset{= w}{q_j} \quad q_n$$

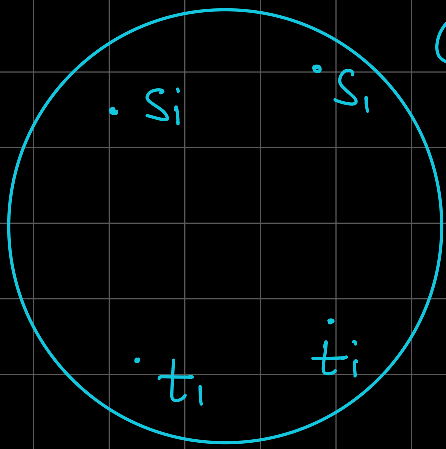
$$P(S_{iw} \mid X_{iw}) \leq 1/j$$

If w is the j^{th} closest vertex

$$\sum P(S_{iw} \mid X_{iw}) \leq \sum \frac{1}{j} = \log_2 n$$

$$\sum P(X_{iw} \cap S_{iw}) \cdot 2^{i+2} \leq 8 \cdot \log_2 n \cdot d(u,v)$$

Buy at Bulk Network Design



$G = (V, E)$

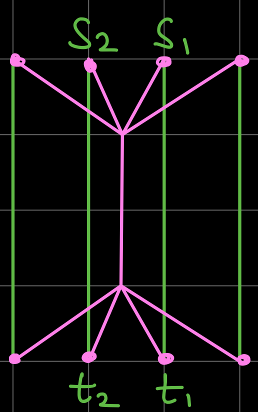
(s_i, t_i, d_i), $i \in [K]$

$l : E \to \mathbb{R}^+$

$c : \mathbb{R} \to \mathbb{R}$

$c(x)$ = cost of transporting x units

Find P_i, $i \in [K]$



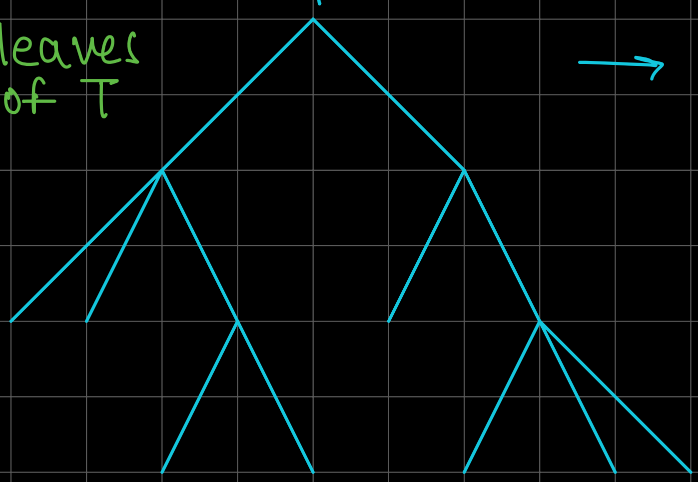
$$\sum_{e \in E} l_e \cdot c(d(e))$$

$$d(e) = \sum_{i : e \in P_i} d_i$$

V is vertex set of T

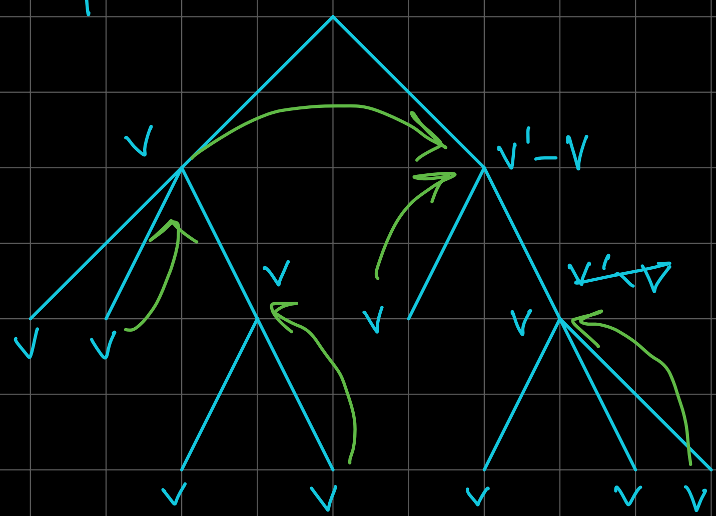
V = leaves of T

T

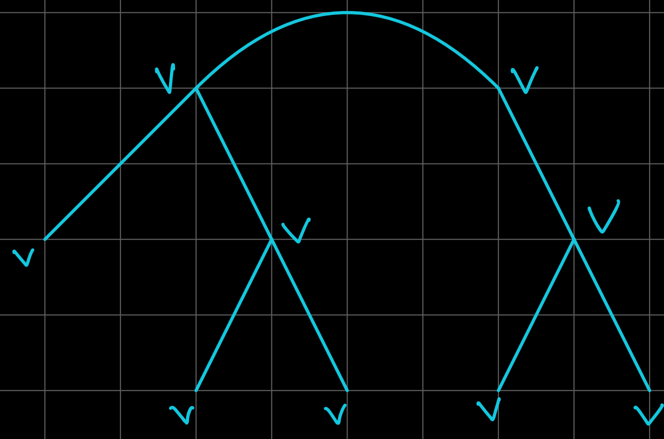


$\longrightarrow$

T'



after shrinking: T'



$$T'(u,v) \geq 2^i$$
$$\geq \frac{T(u,v)}{4}$$
$$T(u,v) = 4 \cdot 2^i$$

$$T(u,v) \leq T'(u,v) \leq 4\,T(u,v)$$

Bin packing problem

n_i objects of size s_i , $i \in [K]$

bin-size = 1 , $0 < s_i \leq 1$

Find min no. of bins needed to pack all $n = \sum n_i$ objects

$C = (c_1, c_2, \dots c_K)$ is a valid configuration if c_i objects of size s_i can fit in a bin

$$\sum_{i=1}^{K} c_i s_i \leq 1 , \quad c_i \leq n_i \ \forall i$$

$\mathcal{C}$ = set of valid configurations

x_c = no. of bins containing configuration c

$$\min \sum_{c \in \mathcal{C}} x_c \qquad |\mathcal{C}| = m$$

$$\sum_{c \in \mathcal{C}} x_c \cdot c_i \geq n_i \ \forall i$$

$$\cancel{x_c \in \mathbb{Z}^+} \quad x_c \geq 0 \ \forall c \in \mathcal{C}$$

OPT $\geq$ LP-value

At any vertex solution at most K variables are non-zero

Find an optimum solution. For each c,
let $y_c = \lceil x_c \rceil$

y is a feasible solution

$$\sum_{c \in \mathcal{C}} y_c - \sum_{c \in \mathcal{C}} x_c = K$$

$$\text{my sol}^n - \text{LP-sol}^n \leq K$$
$$\text{my sol}^n \leq \text{LP sol}^n + K$$
$$\leq \text{OPT} + K$$

Unrelated machine scheduling to minimize makespan

m non-identical machines
n jobs
P_{ij} is processing time of job j on m/c i
Assign jobs to m/c s so that maximum load on any m/c is minimised.

min T

$x_{ij} = 1$, if job j assigned to machine i
$\quad\;\; 0$, o.w.

$\sum_i x_{ij} = 1 \quad \forall j$, $\quad \cancel{x_{ij} \in \{0,1\}}$

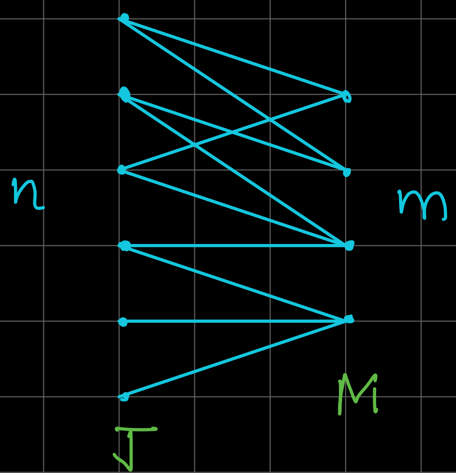
$\sum_j x_{ij} \cdot P_{ij} \leq T \quad \forall i$, $\quad x_{ij} \geq 0$

no. of variables = $mn + 1$

no. of constraints = $mn + m + n$

At a vertex solution, at most $m+n$ variables will take non-zero values.

At most $m+n+x_{ij}$ take non-zero value

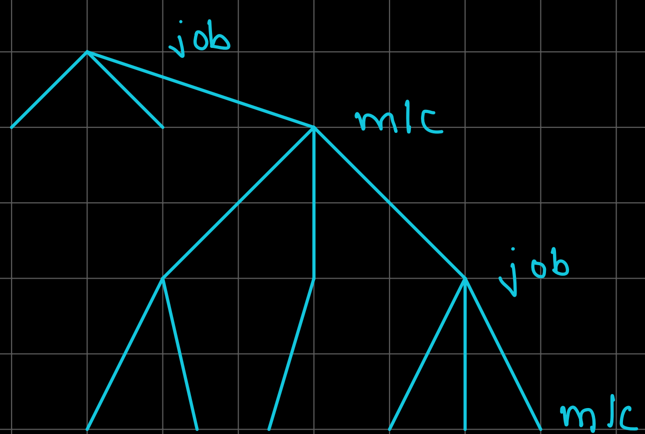


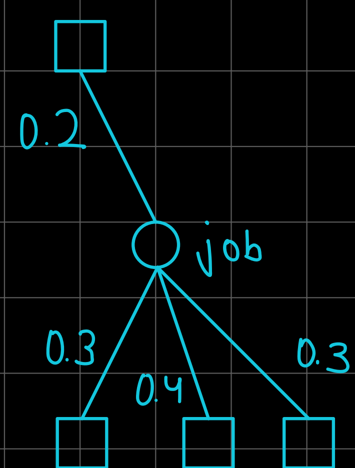
Let $E' = \{(i,j),\ x_{ij} > 0\}$

$V = J \cup M$

leaf can't be a job

Claim (V, E') is a tree





$T^* =$ opt value of LP $\leq$ OPT

In the solution x, load on each m/c $\leq T^*$

In solution y, load on a m/c

$$\leq T^* + \max_j P_{ij}$$

$P_{ij} \leq T^*$

$$\sum_i x_{ij} = 1 \quad \forall j$$

$$\sum_j x_{ij} \cdot P_j \leq T \quad \forall i$$

$$x_{ij} \geq 0$$

Run a binary search on T
if $P_{ij} > T$, set $P_{ij} = \infty$

This ensures $x_{ij} = 0$

Every job has a child
Assign job to one of the children
(root at a job)

$\therefore$ 2-approx

Unrelated machine scheduling

$x_{ij} = 1$, if job j is scheduled on machine i
$\quad\;\; 0$, otherwise

T^* is our current guess on the makespan

m m/cs
n jobs

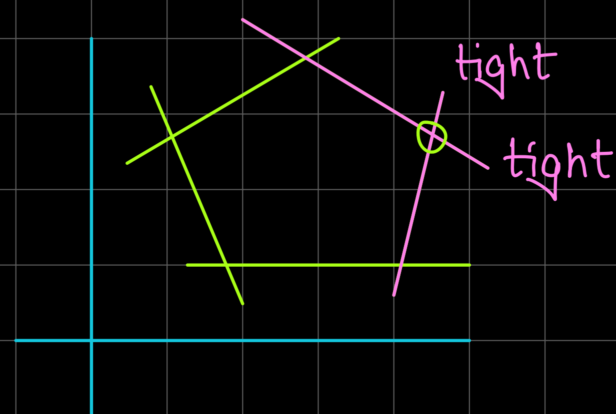
non-trivial
$$\left\{\begin{array}{ll} \forall\, j \in [n] & \sum_{i=1}^{m} x_{ij} = 1 \\ \forall\, i \in [m] & \sum_{j} p_{ij}\, x_{ij} \le T^* \end{array}\right.$$
$$\forall i\ \forall j \qquad x_{ij} \geq 0$$

L.P.

$m+n$ non-trivial constraints

can be computed in poly^n time

A vertex solution (in n-dimensions) is specified by n linearly independent tight constraints



Consider a vertex solution $\bar{x}$ which is the intersection of mn l.i. constraints.

At most $n+m$ constraints are non-trivial

$\Rightarrow$ In $\bar{x}$ atleast $mn-(m+n)$ variables are 0

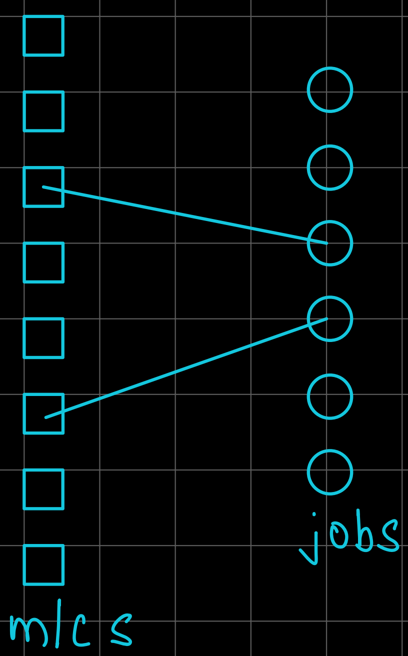
$\Rightarrow$ In $\bar{x}$ atmost $n+m$ variables are non zero

Construct a graph (M, J, E).
machines, jobs, E

$(i,j) \in E$ iff $\bar{x}_{ij} \geq 0$

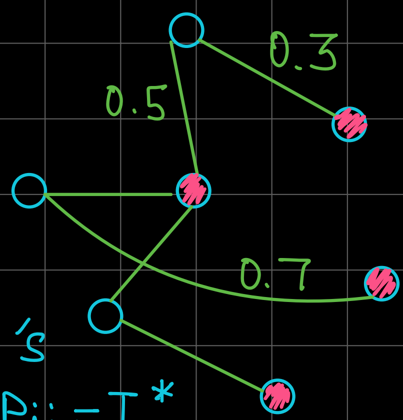
Consider a connected component on $G = (M, J, E)$

Since $\bar{x}$ is a vertex solution, # of non-trivial tight constraints = # of non-zero variables



$|E| \leq m+n$

connected component in G



m/c's

$\sum x_{ij} \cdot p_{ij} = T^*$

jobs

$\sum \overline{x_{ij}} = 1$

K vertices

$\geq$ K−1 edges

$\Rightarrow$ atleast K−1 constraints are tight ($\because$ connected)

atmost K constraints are tight

$\Leftarrow$

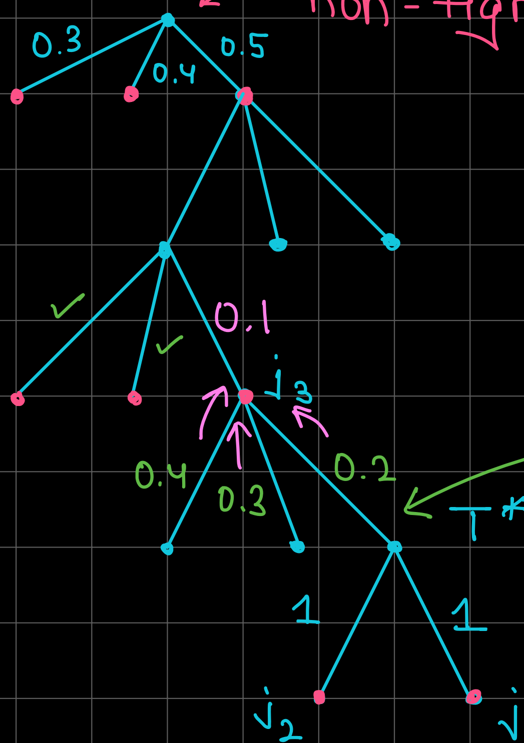
① either K−1 constraints are tight & K−1 variables are non-zero

OR

② K constraints are tight & K variables are non-zero

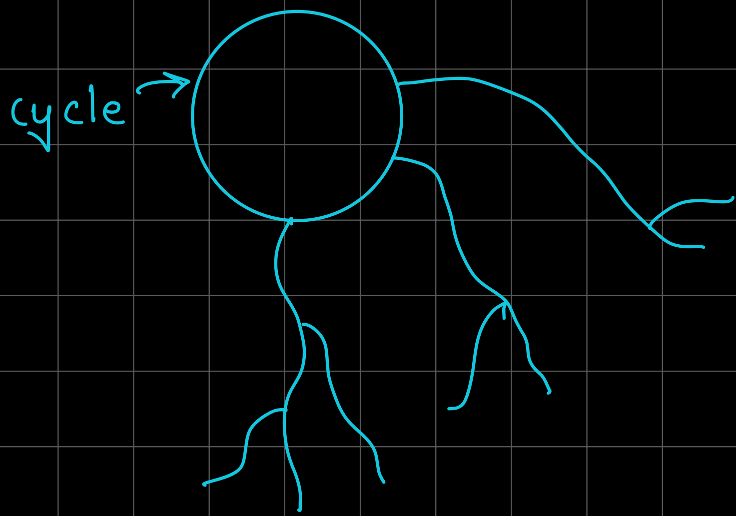
non-tight constraint

CASE I :

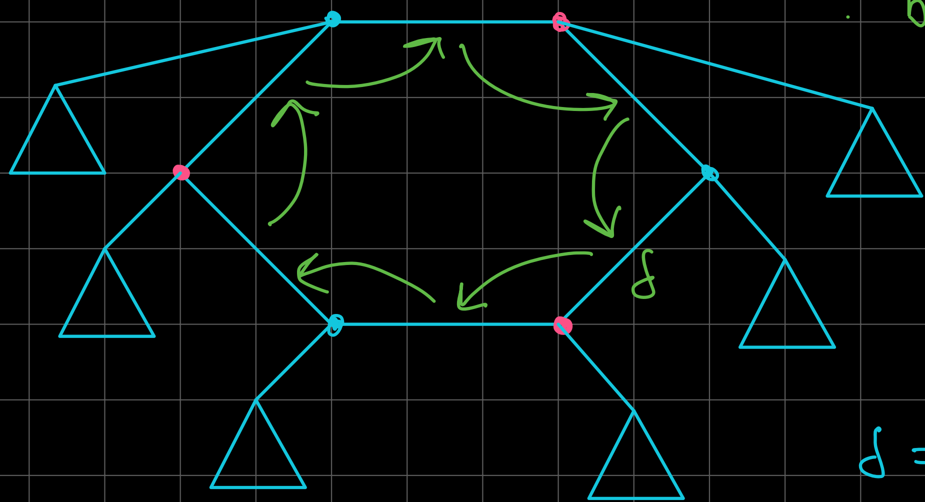


$T^* = p_{ij_1} \cdot 1 + p_{ij_2} \cdot 1 + p_{ij_3} \cdot x_{ij_3}$

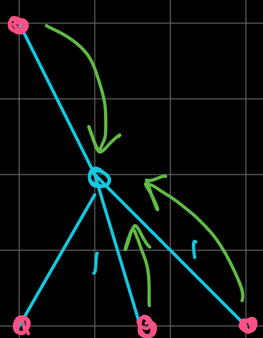
CASE 2 :



even length cycle
∵ bipartite



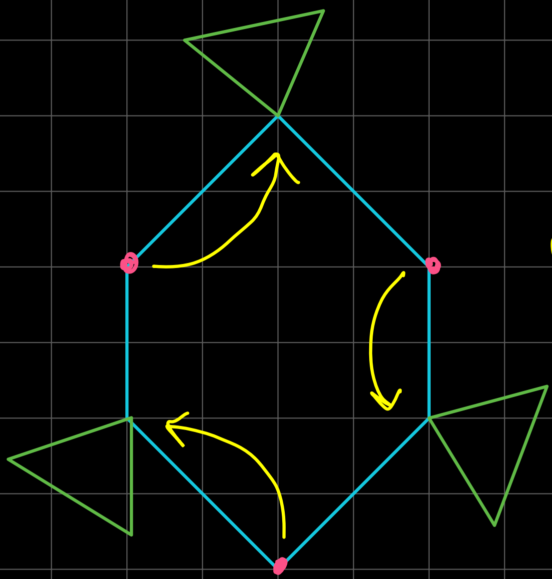
$\delta = f(\delta)$
solve to get δ



load on m/c i
= load from chidren $\leq T^*$
+ load from parent $\leq T^*$
$= p_{ij} \leq T^*$

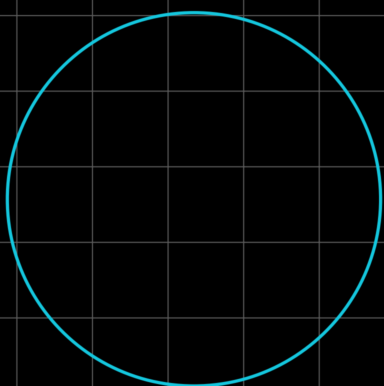
∵ we set $p_{ij} > T^*$ as ∞
hence $x_{ij} = 0$ for them

> If job has a child, assign the job to it



consistently do this in 1 direction

Bounded Degree MST



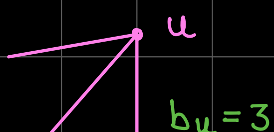
$G = (V, E)$

$c : E \to \mathbb{R}^+$

$b : \cancel{V \to \mathbb{Z}^+}$ $\quad W \to \mathbb{Z}^+, \ W \subseteq V$

bounds on some (or all) vertices

Find a spanning tree T of minimum cost s.t. $\underbrace{\forall\, v \in V \ \deg_T(v) \le b(v)}_{\text{degree bound}}$



$$\text{suppose } \forall v : \left.\begin{array}{l} b(v) = 2 \\ b(s) = 1 \\ b(t) = 1 \end{array}\right\} \begin{array}{l} \text{Hamiltonian Path} \\ \therefore \text{NP-Hard} \end{array}$$

$$x_e = \begin{cases} 1, & \text{if } e \in T \\ 0, & \text{otherwise} \end{cases}$$

← T is the spanning Tree

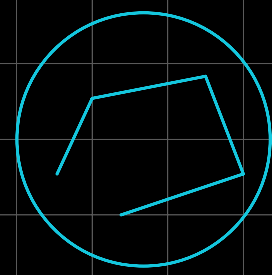
$E(S) = \{(u,v),\ u \in S, v \in S\}$ ← Edges induced by S (both endpoints in S)

$$\sum_{e \in E} x_e = |V| - 1 \quad \Big\}\ \text{Tree}$$

No set of vertices should have a cycle

$$\forall S \subseteq V,\ |S| \geq 2 \quad \sum_{e \in E(S)} x_e \leq |S| - 1$$

$S,\ |S| = K$ ..



$$\forall v \in W, \quad \sum_{e \in \delta(v)} x_e \leq b_v \quad \Big\}\ \text{Degree bounds}$$

$$\forall e \in E, \quad x_e = \cancel{\{0, 1\}} \qquad x_e \geq 0$$

$$\min \sum_{e \in E} c_e x_e \quad \Big\}\ \text{Objective}$$

Solve the LP to obtain a vertex solution x^*
- Remove all edges with $x^*_e = 0$
- Let E' be the remaining set of edges

Claim: In $V \cup E'$, either a vertex has degree = 1 or a vertex in W has atmost 3 edges incident on it.

Let F be the solution set which is initially empty

CASE 1: If there is some vertex v in (V, E') with degree $= 1$.

- Add (u,v) to F
- Remove v from V
- Remove (u,v) from E' ← if $u \in W$ then decrease b_u by 1

We now solve the LP on the graph
$G = (V \setminus \{v\}, E' \setminus \{(u,v)\})$ with $b_u \leftarrow b_u - 1$ if $u \in W$
Repeat

CASE 2: There is some $v \in W$ such that there are atmost 3 edges of E' incident on v

- Remove v from W

Now solve the LP again, $E \leftarrow E(x)$
↑ remove $x_e = 0$ edges
Repeat

Claim: The cost of edges in F is atmost the value of the linear program.

$$\sum_{e \in F} c_e \leq \sum_{e \in E} c_e x_e$$

Proof: (by induction)

I.H.: Set of edges picked on reduced F' instance has cost atmost the optimal solution of the reduced instance.

Base Case: Graph has 2 vertices, algorithm returns a single edge e.

$x_e = 1$ in the LP. Value of LP $= c_e x_e = c_e$
$\therefore$ Base Case holds

Inductive Case: Suppose claim is true for any graph w/ K vertices. Consider a graph w/ $K+1$ vertices

CASE 1: If there is some vertex v in (V, E') with degree $= 1$.

We solve the LP to get a solution x, v has only 1 edge incident on it (u,v)

after removing $x_e = 0$

$V' = V \setminus \{v\}$, $E' = E \setminus \{(u,v)\}$

By I.H. we will find a spanning tree F' on (V', E') with cost atmost value of the LP on (V', E')

$F' \cup \{x_e\}$ is obviously a spanning tree of (V, E)
So the algorithm returns a spanning tree.

To show: x_e for $e \in E'$ is a feasible solution to the LP on (V', E')

From this it follows

$$\sum_{e\in F'} c_e + c_{e^*} \le \sum_{e\in F'} c_e \cdot x_e' + c_{e^*} x_e^* \le \sum_{e\in E'} c_e \cdot x_e + c_{e^*} x_e^*$$

$$= \sum_{e\in E} c_e \cdot x_e$$

All the constraints of the form $(\le |S|-1)$ are a subset of the constraints of the larger LP. (Since $V' \subseteq V$). Hence, they hold

Only need to check $\sum_{e\in E'} x_e = |V'|-1 = |V|-2$

$\leftarrow$ necessary o.w. violation

$$\sum_{e\in E} x_e = \sum_{e\in E'} x_e + \underbrace{x_e^*}_{=1} = |V|-1$$

$$\Rightarrow \sum_{e\in E'} x_e = |V|-2 = |V'|-1 \;\therefore \text{Feasible}$$

We also need to check for the new degree bound $b_u \leftarrow b_u - 1$.

Since x was feasible before, then

$$\sum_{\substack{e\in\delta(u)\\ e\in E}} x_e \le b_u$$

Hence this is also satisfied

$$\sum_{\substack{e\in\delta(u)\\ e\in E'}} x_e + 1 \le b_u \Rightarrow \sum_{\substack{e\in\delta(u)\\ e\in E'}} x_e \le b_u - 1$$

CASE 2: There is some $v \in W$ such that there are atmost 3 edges of E' incident on v

The algorithm removes v from W.
The process is repeated until we reach Case 1 for which the claim has been proved.

Claim: Degree of v in the solution is atmost $b_v + 2$.

1) Chose (u,v), $b_v \leftarrow b_v - 1$, Follows from I.H.

2) Chose (u,v), removed v. We must have had $b_v \geq 1$ for the solution to be feasible, we are removing v so the claim holds

3) Removed v from W. We do this when atmost 3 edges (after removing $x_e = 0$) are incident on v. $b_v \geq 1$ for there to be some edge incident on it. Hence $b_v + 2 \geq 3 \geq \deg_F(v)$

Note:

A set S is tight if $x(E(S)) = |S| - 1$

A vertex v is tight if $\sum_{e \in \delta(v)} x_e = b_v$

Laminar Sets: A collection of sets is laminar if no pair of sets $A, B \in \mathcal{S}$ is intersecting.

We will show that for any feasible solution x to the LP, there is a set $Z \subseteq W$ and a collection $\mathcal{L}$ of tight sets which form a laminar family.

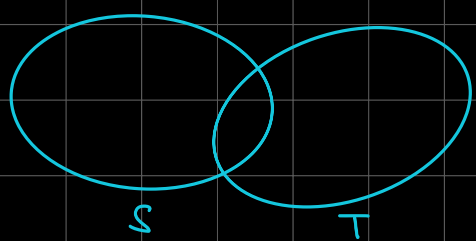
1) $Z = \{v \in W, \; b_v = \sum_{e \in \delta(v)} x_e^*\}$ ← tight vertices

2) $\forall S \in \mathcal{L}$, $|S| \geq 2$ and S is tight

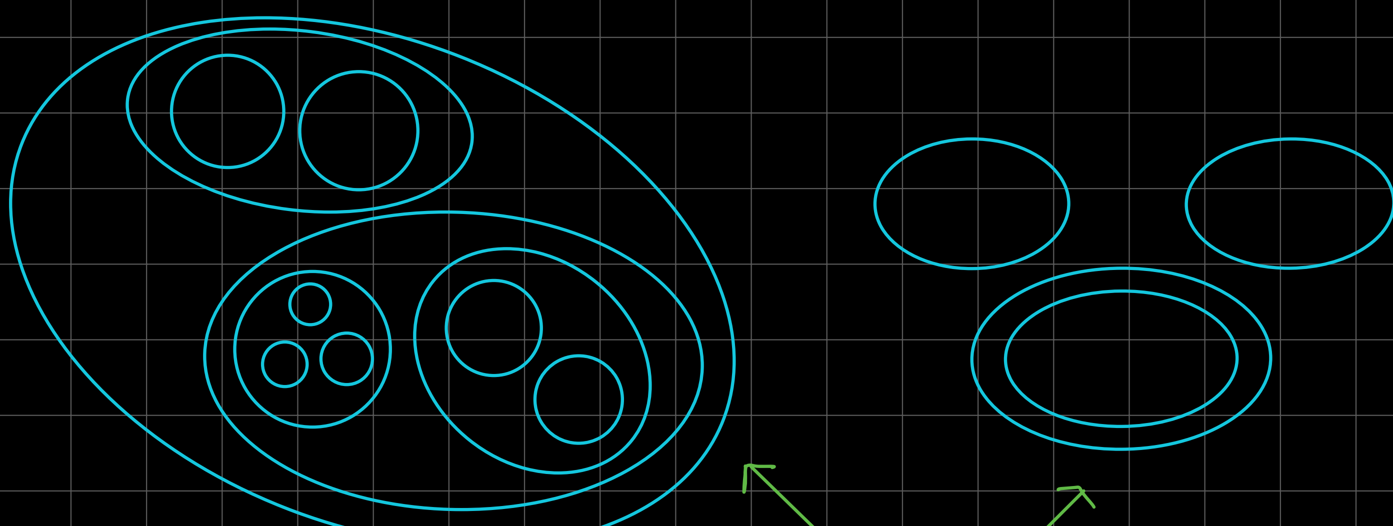
3) $|\mathcal{L}| + |Z| = |E'|$

4) $\mathcal{L}$ is laminar

S, T are intersecting



← Non Laminar



Laminar

Claim: For any solution to the LP, either $\exists$ a vertex $v \in V \setminus W$ s.t. $\deg_E(v) = 1$ or $\exists$ a vertex $v \in W$ s.t. $\deg_E(v) \le 3$

Proof: (by contradiction)
Suppose $\forall v \in V$, $\deg(v) \ge 2$ and $\forall v \in W$, $\deg(v) \ge 4$

$$|E(x)| \ge \frac{1}{2}\Big(2(n-|W|) + 4|W|\Big)$$

$$= n + |W| \ge n + |Z|$$

From our assumption $|\mathcal{L}| + |Z| = |E|$
Also $|\mathcal{L}| \le n-1$ (proved below)

$$\Rightarrow \quad |E| \le n + |Z| - 1$$

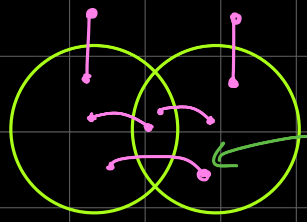
This contradicts the fact that $|E| \ge n + |Z|$

Note on Modularity:

$A, B \subseteq V$, $A \cap B = \phi$, $\overline{A \cup B} \neq \phi$

$f(A) + f(B) = f(A \cup B) + f(A \cap B)$: Modular f^n
no. of elements in a set

$f(A) + f(B) \ge f(A \cup B) + f(A \cap B)$: Submodular f^n
no. of edges going out of a set



counted twice on left
zero times on right

$f(A) + f(B) \le f(A \cup B) + f(A \cap B)$: Supermodular f^n

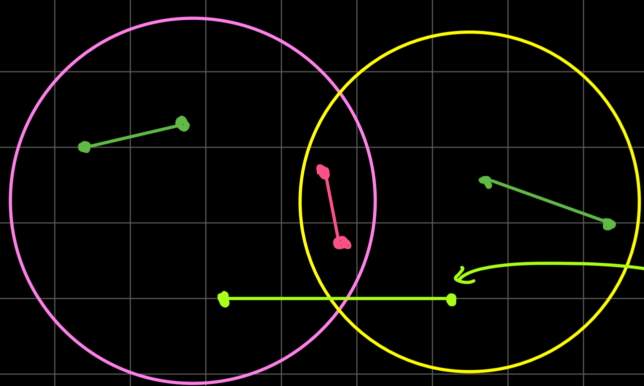
Weakly Supermodular F^n:

Either $f(A) + f(B) \le f(A \cup B) + f(A \cap B)$
or $f(A) + f(B) \le f(A \setminus B) + f(B \setminus A)$

Claim: If S, T are intersecting tight sets then $S \cap T$, $S \cup T$ are also tight.

We first show that $x(E(S))$ is supermodular

$$x(E(S)) + x(E(T)) \le x(E(S \cap T)) + x(E(S \cup T))$$



such edges only
occur in RHS
but not in LHS

S, T are intersecting, $S \cap T \neq \phi$. Thus by feasibility of x

$$\begin{aligned}(|S|-1) + (|T|-1) &= (|S \cap T| - 1) + (|S \cup T| - 1) \\ &\ge x(E(S \cap T)) + x(E(S \cup T))\end{aligned}$$

By Supermodularity

$$(|S|-1) + (|T|-1) \geq x(E(S\cap T)) + x(E(S\cup T))$$
$$\geq x(E(S)) + x(E(T))$$

Supermodular

Since S, T are tight sets

$$x(E(S)) + x(E(T)) = (|S|-1) + (|T|-1)$$

Thus all the inequalities are met w/ equalities
Hence $S\cup T$, $S\cap T$ are tight sets.

$$\chi_S + \chi_T = \chi_{S\cup T} + \chi_{S\cap T}$$

$$\Upsilon = \{ S \subseteq V : x(E(S)) = |S|-1, |S|\geq 2\}$$

collection of all tight sets

$$\text{span}(\Upsilon) = \text{span}\{\chi_{E(S)} : S \in \Upsilon\}$$

Claim: There exists a laminar collection $\mathcal{L}$ of tight sets such that $\text{span}(\mathcal{L}) \geq \text{span}(\Upsilon)$ and $\chi_{E(S)}$ for $S \in \mathcal{L}$ are linearly independent

Proof: Suppose $\mathcal{L}$ is a maximal such collection satisfying the given properties.

Suppose $\text{span}(\mathcal{L}) < \text{span}(\Upsilon)$ i.e. $\exists$ a tight set S, $|S|\geq 2$ such that $\chi_{E(S)} \in \text{span}(\Upsilon)$ and $\chi_{E(S)} \notin \text{span}(\mathcal{L})$

Choose S such that it intersects with least sets in $\mathcal{L}$
(intersects atleast 1 o.w. $\mathcal{L}$ not maximal)

Pick a set $T \in \mathcal{L}$ s.t. S & T are intersect.
$\Rightarrow \chi_{E(T)} \in \text{span}(\mathcal{L})$ (since $T \in \mathcal{L}$)

From above $\chi_S + \chi_T = \chi_{S\cup T} + \chi_{S\cap T}$
$\Rightarrow \chi_{E(S)} = \chi_{E(S\cup T)} + \chi_{E(S\cap T)} - \chi_{E(T)}$

Since $\chi_{E(T)} \in \text{span}(L)$ but $\chi_{E(S)} \notin \text{span}(L)$. Hence, atleast one of $\chi_{E(S\cup T)}$ or $\chi_{E(S\cap T)} \notin \text{span}(\mathcal{L})$

Claim: $|\mathcal{L}| \leq n-1$ where $\mathcal{L}$ is a laminar family of sets of size ≥ 2

Proof: Base Case: $n=2$. $\mathcal{L}$ contains only 1 set
Pick a min cardinality set $R \in \mathcal{L}$.
V': $V - \{$all vertices of R except $1\}$. $\mathcal{L}'$ are the sets of $\mathcal{L}$ restricted to $V' \setminus R$. $\mathcal{L}'$ fulfils conditions of the lemma, hence by I.H. $|\mathcal{L}'| \leq |V'|-1$
Since $|\mathcal{L}'| = |\mathcal{L}|-1$ and $|V'| \leq |V|-1$

$|\mathcal{L}|-1 \leq |V|-1-1 \Rightarrow |\mathcal{L}| \leq |V|-1$ $\therefore$ Proved

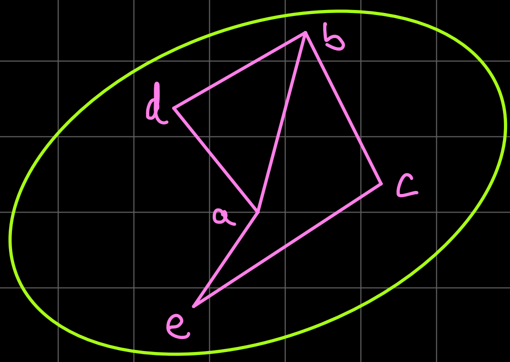
Steiner Network Design:

$G = (V, E)$
$c : E \to \mathbb{R}^+$

s_1 t_1 t_2 t_3

v_i, v_j, r_{ij}

$E' \subseteq E$. Find min cost set E' s.t. (V, E') s.t. there are r_{ij} disjoint paths from v_i to v_j $\forall i, j$



$r_{ab} = 3$, $r_{bc} = 2$

$r_{cd} = 1$, $r_{bd} = 2$

$$x_e = \begin{cases} 1, & \text{if } e \text{ is included} \\ 0, & \text{otherwise} \end{cases}$$

$$\min \sum_{e \in E} c_e x_e \quad \text{s.t.} \quad \forall\, S \subseteq V \quad \sum_{e \in \nabla(S)} x_e \geq \max_{\substack{i \in S \\ j \notin S}} r_{ij}$$

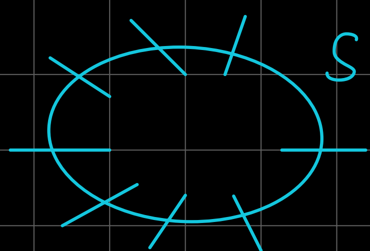
$$\cancel{x_e \in \{0,1\}} \qquad 1 \geq x_e \geq 0$$

to avoid setting an $x_e > 1$

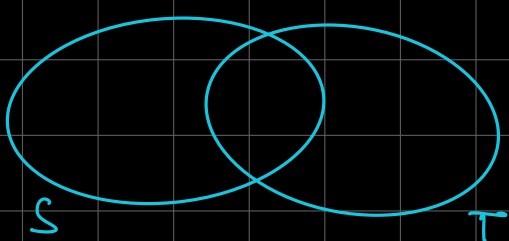
\# can find violating constraint by finding a max flow b/w a, b ($\forall\, a, b \in V$). If flow $< r_{ab}$ then there is a cut with value $< r_{ab}$ which is a violated constraint.

x is a vertex solution to the LP

tight sets



$$\sum_{e \in \nabla(S)} x_e = \max_{\substack{i \in S \\ j \notin S}} r_{ij} = r(S)$$



S, T are tight sets

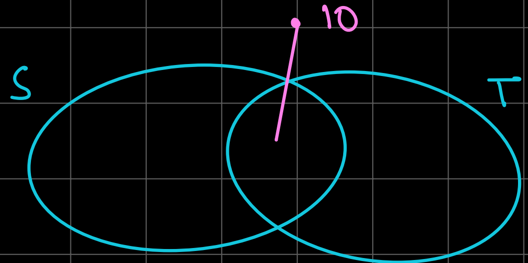
$\Rightarrow S \cap T,\ S \cup T$ are also tight

Weakly supermodular f^n :

Either $f(A)+f(B) \leq f(A \cup B) + f(A \cap B)$

or $f(A)+f(B) \leq f(A-B)+f(B-A)$

r is weakly submodular



$r(S) = r(\overline{S})$

$r(A \cup B) \leq \max(r(A), r(B))$ corollary

$r(A) \leq \max(r(A-B), r(A \cap B))$

$r(A) = r(V-A) \leq \max(r(B-A), r(A \cup B))$

$r(B) \leq \max(r(A-B), r(A \cap B))$

$r(B) \leq \max(r(A-B), r(A \cup B))$

$$r(A) + r(B) \leq r(A \cup B) + r(A \cap B)$$

$$x(\nabla(S)) = r(S), \quad x(\nabla(T)) = r(T)$$

$$r(S \cup T) + r(S \cap T) \leq x(\nabla(S \cup T)) + x(\nabla(S \cap T))$$
$$\leq x(\nabla(S) + x(\nabla v) = r(S) + r(T)$$

$$r(S-T) + r(T-S) \leq x(\nabla(S-T)) + x(\nabla(T-S))$$
$$\leq x(\nabla(S) + x(\nabla m) = r(S) + 8$$

drop all edges with $x_e = 0$, e is the new set of edges

$$\sum_{e \in \nabla(S)} x_e = \max_{i \in S} r_{ij} = r(S)$$

$\mathcal{L}$ is a a laminar family of tight sets

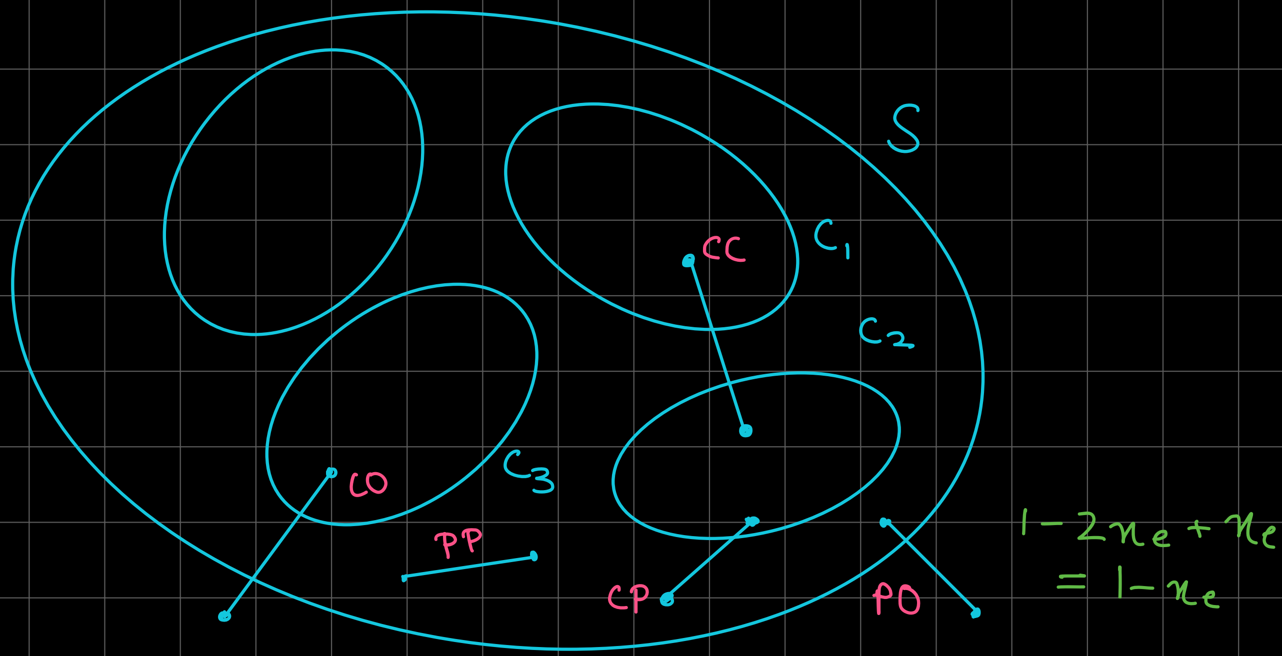
1) X_E, $S \in I$ is a linearly independent collection

2) $|I| = |E|$

$\exists$ an edge e s.t. $x_e \geq 1/2$

For contradiction assume $x_e < 1/2$. We give ₹1 to each judge

& assign $1-2x_e$ to the smallest set in $\mathcal{L}$ cont^n e
" " x_e to the smallest set cont^n each endpt. of e



$$\text{Total change on } S = |CC| - 2x(CC) + |CP| - x(CP) + x(PO)$$

$$|CC| + |CP| + r(S) - \sum r(C_i) =$$

\# All edges can't be CO (linear independence)

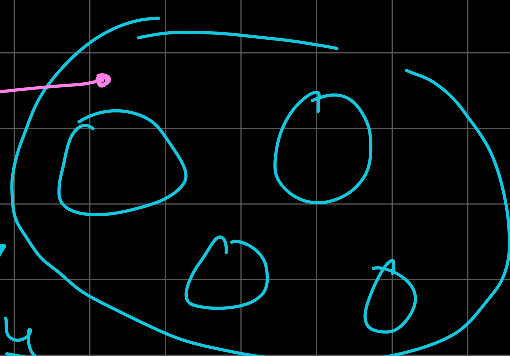
$$-\left(x(CO) + 2x(CC) + x(CP)\right) = \left(\sum r(C_i)\right)x - 1$$

$$x(PO) + x(CO) = r(S)$$

$$x(PO) - 2x(CC) - x(CP) = r(S) - \sum r(C_i)$$

tight but only $x(C)$ units used

highest set in my laminar family



integer ∴ atleast |E| charge but
we gave |E| charge some of which unused