Căutăm un alg. de 2-aprox(=) satisface cel puțin $\frac{m}{2}$ clauze

m = nr. clauze

$\cancel{\text{Fie } \{x_1 = T \Rightarrow C_1 = \{x1\}}$

$\cancel{\{x_2 = F \Rightarrow C_2 = \{x_2\}}$

$\cancel{C = C_1 \cup C_2}$

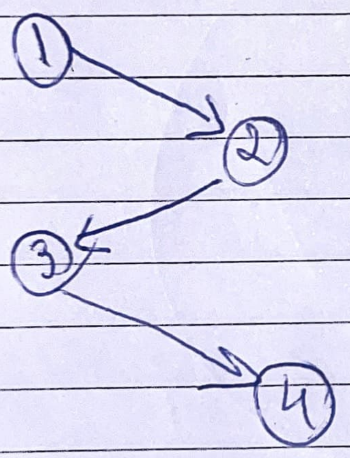
C = mț. tot. de clauze (muchii)

Cum $m = 5$, $OPT = 4$

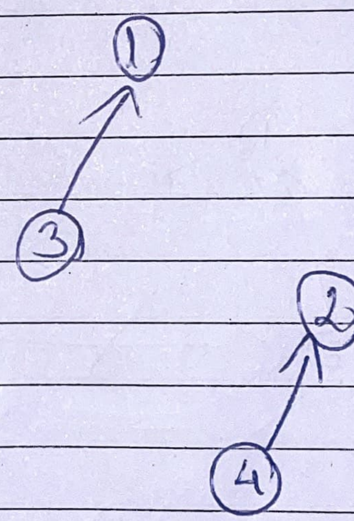
$$A(I) \geq \frac{OPT(I)}{c} = \frac{4}{c} = \frac{4}{5}$$

Soluția corectă:

G_1:



G_2:



$G = (V, E)$

$E_1 = \{(a,b) \in E \mid \overset{a<b}{\cancel{E_1 \subseteq E}}\}$

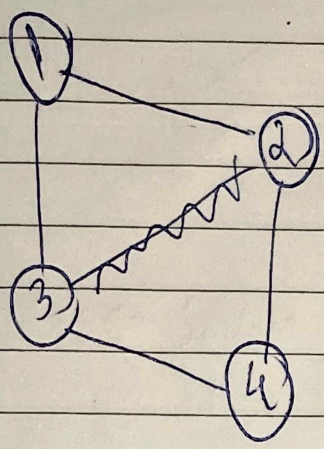
$E_2 = \{(a,b) \in E \mid a > b\}$

$E_1 \cap E_2 = \emptyset$ $\quad$ $|E| = |E_1| + |E_2|$

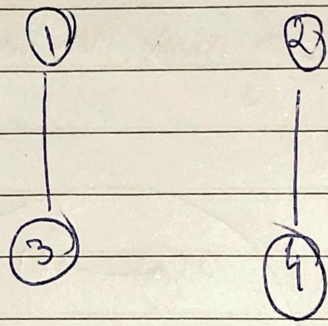
$E_1 \cup E_2 = E$

$$\max\{|E_1|, |E_2|\} \geq \frac{|E|}{2}$$

2) Să se găs. un cuplaj maximal de cardinalitate minimă. (2-aprox)
(⇔ dintre toate cuplajele maxim., îl vrem pe cel cu cele mai puține muchii)



→ cupl. max. (fără 2-3)



Cuplaj maximal ⇔ set de muchii care acoperă nodurii fără a lăsa alte muchii ~~dispo~~ disponibile pt. adăugare.

Cupl. max. de cardinalitate minimă ⇔ cupl. max. care are cel mai mic nr. posibil de muchii în comparație cu alte cupl. max.

2-aprox pt. ~~un cupl.~~ pb. dată ⇔ parcurgem graful și adăugăm muchii în cuplaj până când nu se mai pot adăuga.

Fie M cuplaj maximal

$$|M| \geq \frac{1}{2}|M_{OPT}|$$

Fiec. muchie acoperă 2 nod. dist.

$\begin{cases} \text{cupl. max.} \Leftrightarrow \text{nr. max muchii} \\ \text{fiecare muchie poate acoperi 2 nod. distincte} \end{cases}$

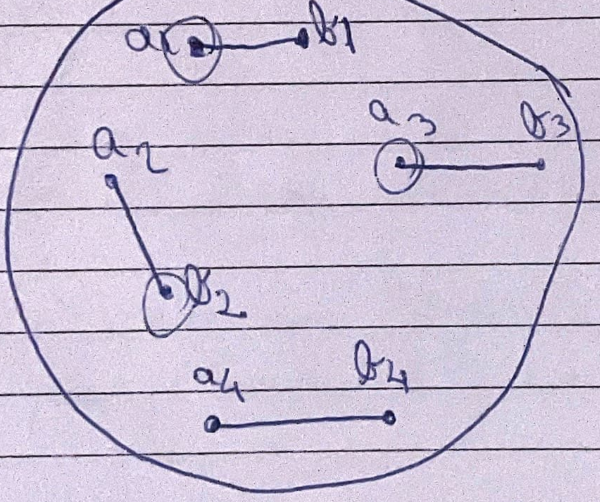
M maximal $\Rightarrow$ fiecare muchie $\in M_{OPT}$, $m \notin M$ $\Rightarrow$ cel puțin unul dintre nodurile sale este deja acoperit de o muchie din M.

$\Rightarrow$ 2 muchii $\in M_{OPT}$ ar putea coresp. unei sg. muchii $\in M$, worst case

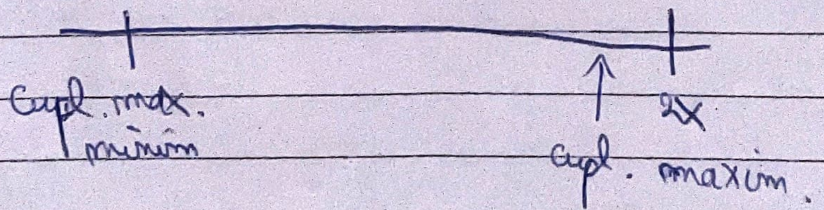
$\Rightarrow$ $|M| \geq \frac{1}{2} |M_{OPT}|$

<u>Soluția lui Popa</u>

Orice cuplaj max. va conține K noduri $\Rightarrow$ minim $\frac{K}{2}$ muchii



$\to$ K muchii pt. cupl. max.

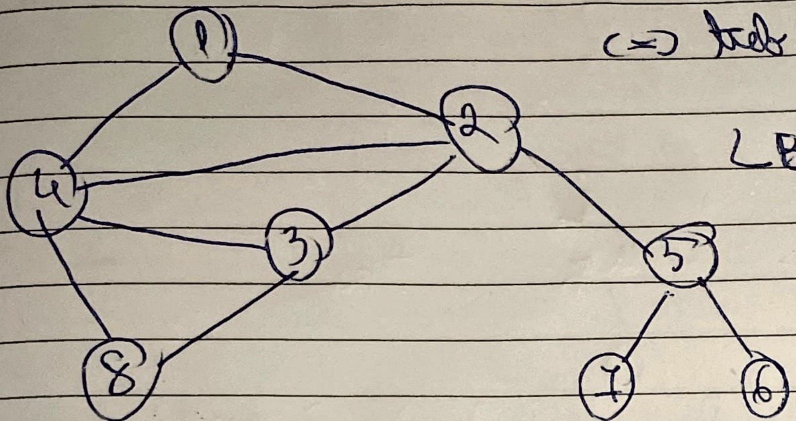


3] Alg. de 2 aprox. pt. Vertex Cover

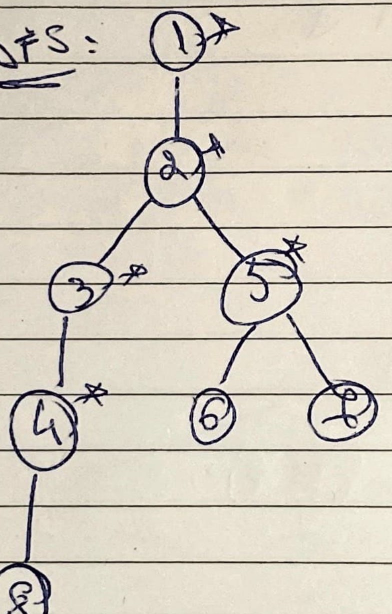
Realizăm o parcurgere DFS și adăugăm în VC toate nodurile din arborele DFS care nu sunt frunze.

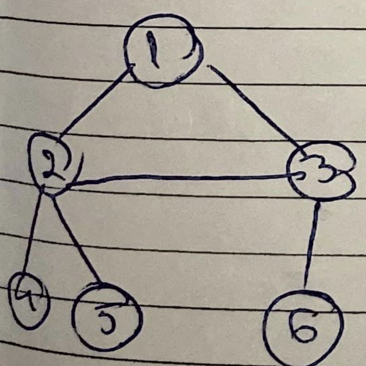
→ Când căutăm un alg. 2-apr. pt VC ⟺ trebuie să găsim un LB

$LB = \frac{m}{2}$ → nr. muchii

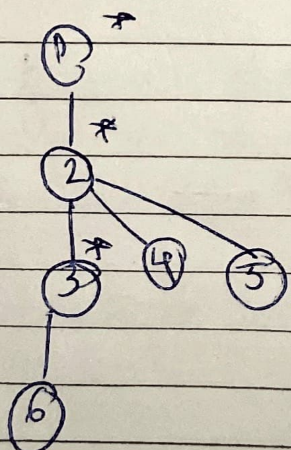


DFS:

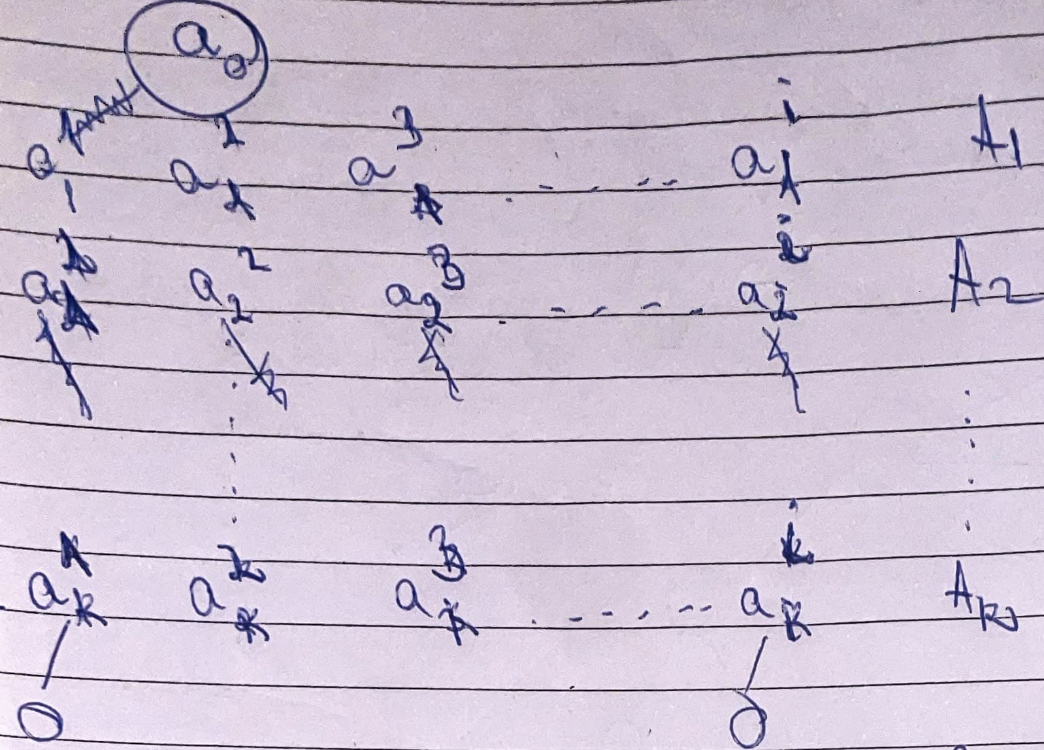




DFS:



BFS:



Vrem să ar. că avem cuplaj cu nodurile impare / pare.

$M_1 = \{a_0, a_2, ...\}$

$M_2 = \{a_1, a_3, ...\}$

Cursul 6
Prog. alg. ef.

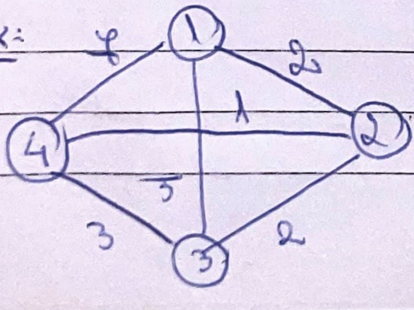
1. Traveling Salesman Problem (TSP)
2. Steiner Tree

1. Def TSP

Se dă un graf neorientat complet în care fiecare muchie are un cost pozitiv.

Vrem să facem un tur ~~care~~ de cost minim a.î. fiecare nod să fie vizitat exact o dată.

Ex:



Cost (1 2 3 4 1) = 14

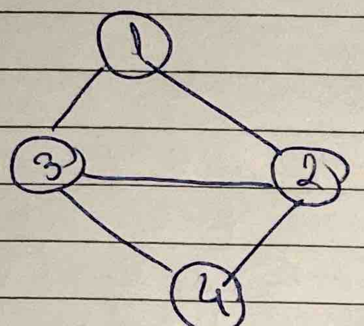
Cost (1 3 4 2 1) = 11 - optim

Teoremă: Pt orice funcție $f(n)$ calculabilă în timp polin. pb. TSP nu se poate aproxima la un factor de $f(n)$ presupunând că $P \neq NP$.

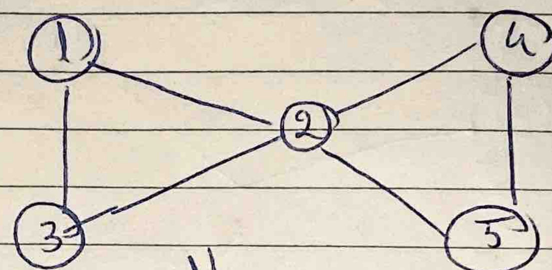
Dem: Reducție de la pb. Ciclului Hamiltonian.

Def: Se dă un graf neorientat $G = (V, E)$. Să se decidă dc $\exists$ un circuit care vizitează fiecare nod exact o dată.

Ex:

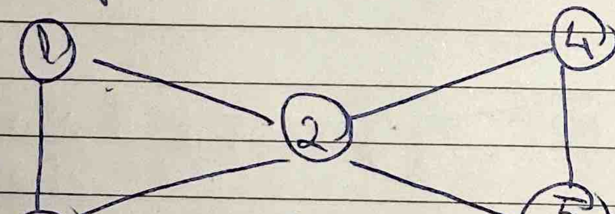


Da.

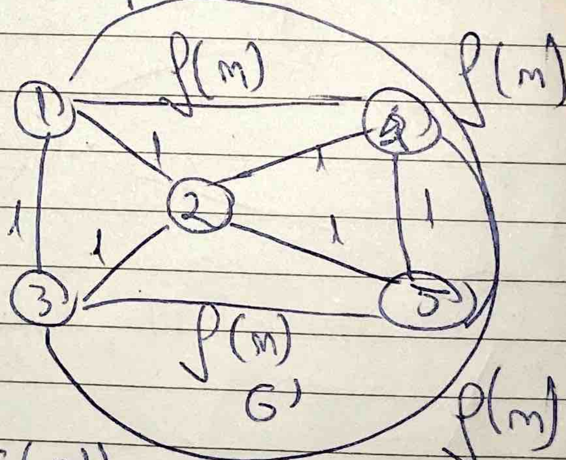


Nu.

Pp. că dc. avem un Alg. de $f(n)$-aproximare pt. TSP, atunci vom putea rezolva pb. ciclului Hamiltonian în timp polin. Dar, cum pb. Ciclului Hamiltonian este NP-completă, un asemenea alg. ar implica $P = NP$.



G



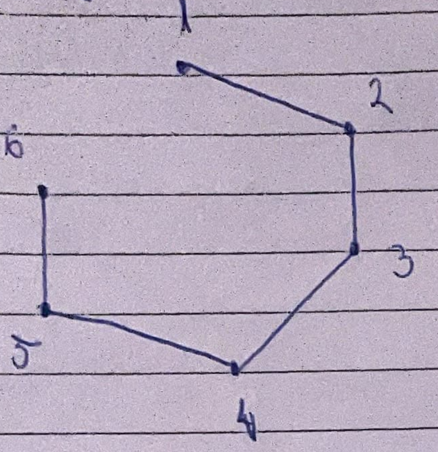
G are ~~ciclu~~ ciclu Hamiltonian $\Rightarrow OPT(G') = n$

G nu are ciclu Hamiltonian $\Rightarrow OPT(G') = f(n) + n - 1$

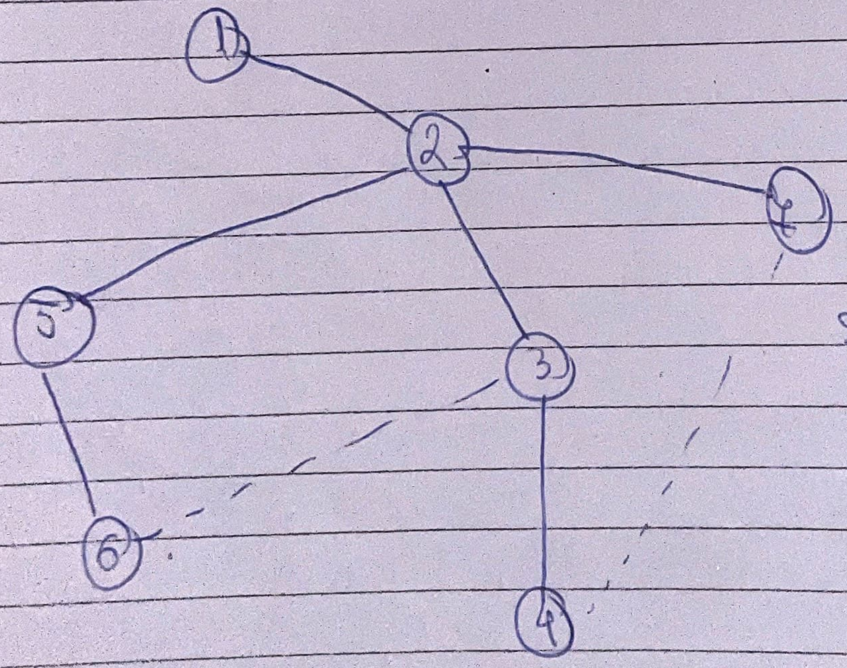
O metrică pe o mț. X este o funcție $d: X, X \to \mathbb{R}_+$

1) $d(x,x) = 0 \ \forall x \in X$
2) $d(x,y) = d(y,x) \ \forall x,y \in X$
3) $d(x,y) + d(y,z) = d(x,z), \forall x,y,z \in Y$

Un alg. de 2-aproximare pt. TSP pe spații metric.



În orice graf complet în care funcția $d : V \times V \to \mathbb{R}$ este o metrică avem Minimum Spaning Tree (arborele parțial de cost minim) în $G \leq$ soluția optimă a problemei!



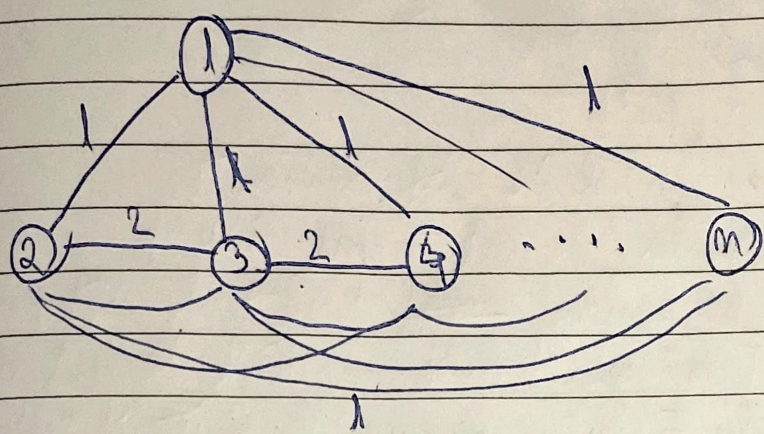
Tur Euler pe arborele din stânga 1256523432721

Suma muchiilor dintr-un Tur Euler $= 2 \times MST \leq 2 \times OPT$

Turul Euler scurtcircuitat are cost $\leq$ decât Euler inițial pt. că distanța este o metrică.

Deci costul algoritmului $\leq 2 \times MST \leq 2 \times OPT$.

$ALG \Rightarrow 2n$
$OPT \Rightarrow n+1$

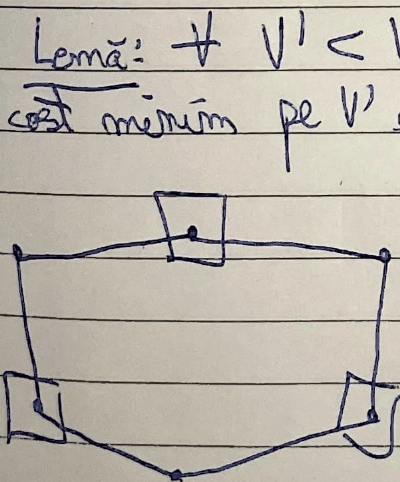


$OPT = n$
$ALG = 2(n-2)+2 = 2n-4+2 = 2n-2$

$n \to \infty \qquad \frac{ALG}{OPT} = 2$

ALG $\frac{3}{2}$ aprox. pt. TSP < metric

Lemă: $\forall V' \subset V$ un cuplaj perfect, $|V'|$ este pară, de cost minim pe V' are cost $\leq \frac{OPT}{2}$



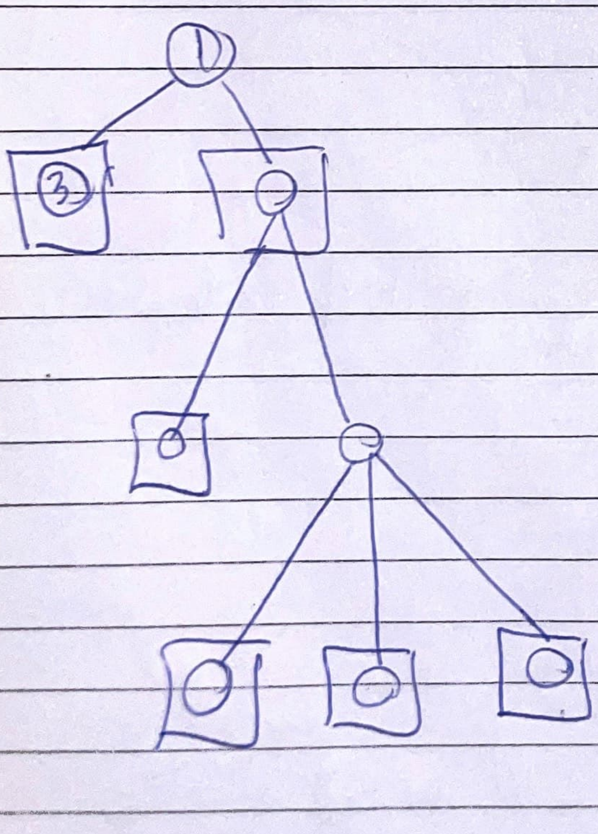
Deoarece $V' \subset V$ un tur optim pe V' are cost $\leq$ un tur

optim pe V pt. că distanța este metrică.
Deoarece V' are un nr. par de noduri ~~pt.~~ putem construi două cuplaje perfecte M_1 și M_2 a.î. cost M_1 + cost M_2 = costul turului optim pe V: $\Rightarrow$ cuplaj perfect de cost minim $\le$ cost (M_1) și cost $(M_2) \le \frac{OPT}{2}$ ~~costul~~ turului optim pe V' și implicit pe $\frac{V}{2}$

ALG ($\frac{3}{2}$ - aprox)

1) Găsim un arbore parțial de cost minim T.
2) Fie V mulț. nodurilor de grad impar în MST.
3) Pe V' calculăm un cuplaj ~~perfect~~ perfect de cost minim M.
4) În graful $M \cup T$ vom avea ~~toate~~ nodurile de grad par
5) Calculăm un ciclu Eulerian pe care îl scurtcircuităm.

ALG $\le$ ~~cel~~ Ciclul Eulerian va avea cost = cost(M) + cost(T) $\le \frac{OPT}{2} + OPT \le \frac{3}{2}$ OPT

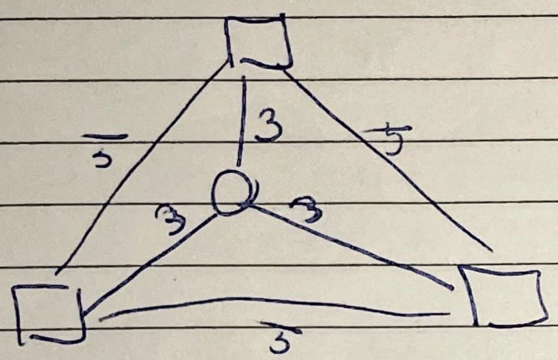


2. ~~Har~~ Minimum Steiner Tree

Se dă un graf neorientat $G = (V, E)$, $d : V \times V \to \mathbb{R}_+$ în care fiecare muchie are asociat un cost. Mai mult, nodurile sunt partiționate în:

- R – required (obligatorii)
- S – Steiner (opționale)
- $V = R \cup S$

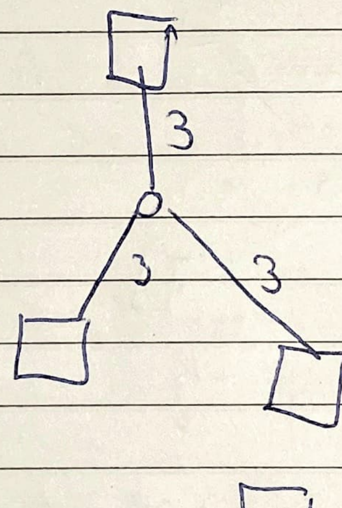
Vrem să găsim un arbore parțial de cost minim care include toate nodurile din R și posibil un subset din S.



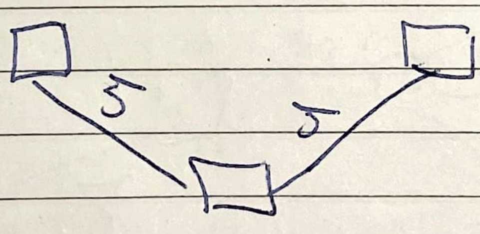
□ ← required

○ ← Steiner

OPT = 9



MST pe □ – 10

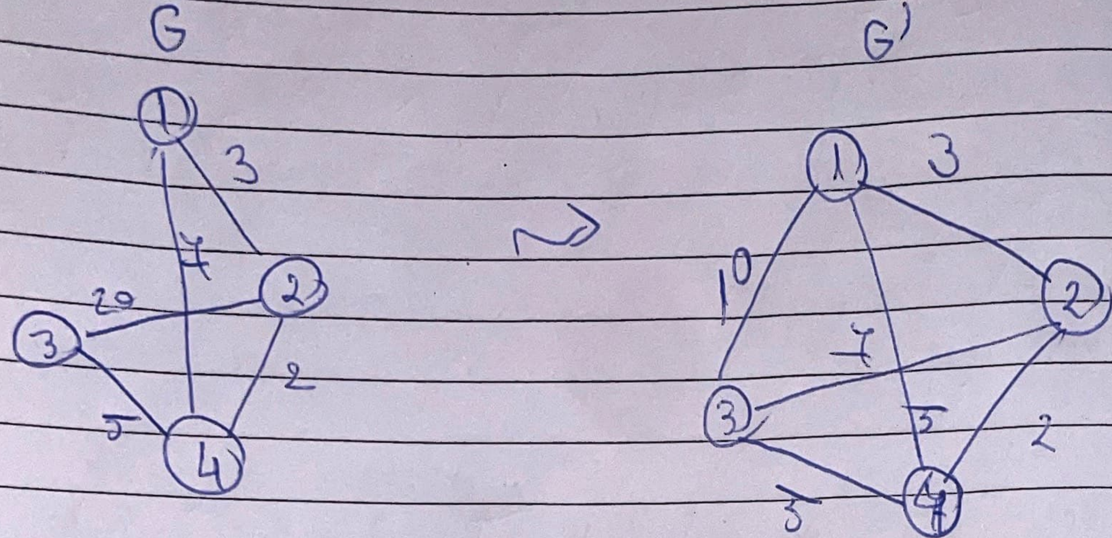


Lemă: Fie $G=(V,E)$ un graf neorientat și $d : V \times V \to \mathbb{R}_+$. ~~R~~ Putem construi în timp polin. un $G' = (V', E)'$ și o metrică d' pe V a.î. OPT Steiner Tree pe G = OPT Steiner Tree pe G'.

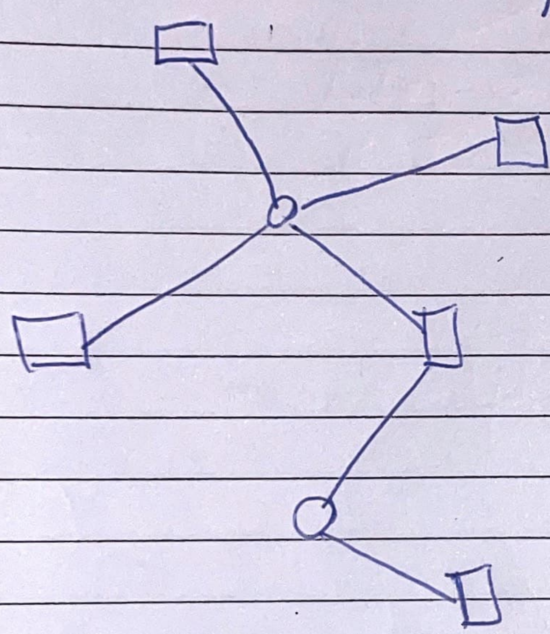
G

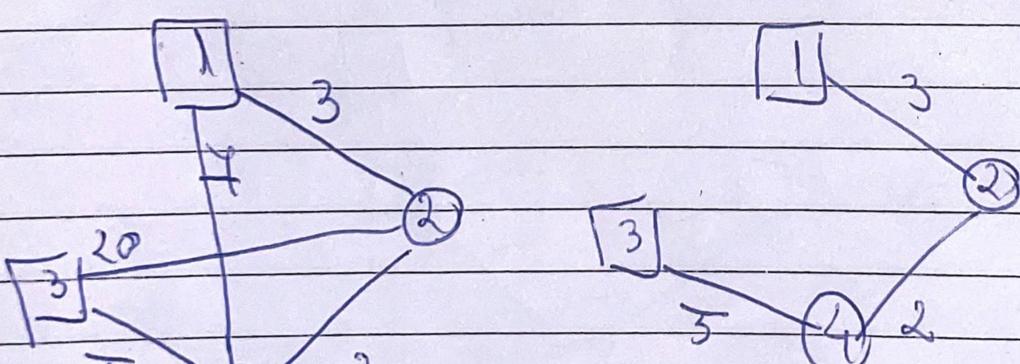
G'

(u, v) în G' va avea cost cel mai scurt drum de la u la v în G.



$OPT(G) = OPT(G')$





Seminarul 4
Prog. alg. ef.

Set Cover

1) Se dau m mulțimi $S_1, S_2, \dots, S_m \in f$

$S_i \subseteq U$, $|U| = n$. Să se găs. un nr. minim de mț. din cele m a.î. reuniunea lor să fie U.

Ex: $U = \{1, 2, 3, 4, 5, 6\}$
$S_1 = \{1, 2, 6\}$
$S_2 = \{3, 4, 5\}$
$S_3 = \{6, 4\}$
$S_4 = \{1, 5\}$
$\Rightarrow \boxed{S_1 \cup S_2 = U}$

→ fiecare elem. din U apare în cel mult f mulțimi
pt. ex. de mai sus $\Rightarrow f = 2$ (1 apare în 2 mț)

a) Formulați pb. ca un indice Linear Program (ILP)
b) Fol. relaxarea LP pt. a obține o f - aproximare

Var. mea
a) variabile $x_i \in \{1, 2, 3, 4, 5, 6\}$, $x_i \in U$
~~x_i~~
$S_i = \{x_i \mid x_i \in U\}$
$\min S_1 \cup S_2 \cup \dots \cup S_i = U$

Var. ~~corecta~~ corectă:
a) Variabila x_s pt. fiecare mț. $S_i = \begin{cases} 1, & \text{dc. } S_i \text{ este selectată în sol.} \\ 0, & \text{altfel} \end{cases}$

Vrem să minimizăm suma de variabile

$\min \sum_{S \in \mathcal{Y}} x_S$ a.î. să acoperim toate muchiile din $\mathcal{U}$

Vrem pt. $\forall e \in \mathcal{U}$ ca suma val. care-l conțin:

$$\sum_{S \ni e} x_S, \forall e \in \mathcal{U}$$

$\Rightarrow$ 4 variabile

$\Rightarrow \min x_1 + x_2 + x_3 + x_4$

a.î. $x_1 + x_4 \geq 1$ elem 1

$x_1 \geq 1$ elem 2

$x_2 \geq 1$ elem 3

$x_2 + x_3 \geq 1$ ⋮

$x_2 + x_4 \geq 1$

$x_1 + x_2 \geq 1$ $x_1, x_2, x_3, x_4 \in \{0, 1\}$

b) Din rez. a) relaxăm LP $\Rightarrow x_S'$ a.î. $\sum_{S \in \mathcal{Y}} x_S' \leq OPT_{SC}$

$\Rightarrow x_S^* = \begin{cases} 1, & dc\ x_S' \geq \frac{1}{f} \\ 0, & altfel \end{cases} \Rightarrow x_S^* \leq f \cdot x_S'$

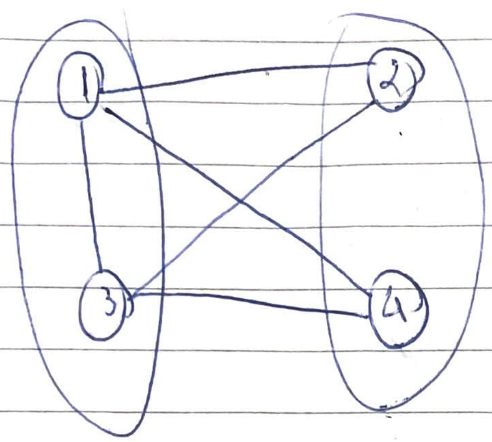
$$\sum_{S \in \mathcal{Y}} x_S^* \leq f \cdot \sum_{S \in \mathcal{Y}} x_S'$$

Vrem să ar. că $\sum_{S \ni e} x_S^* \geq 1$ $\forall e \in \mathcal{U} \Rightarrow$ adevărat pt. că $\exists$ $e \in S$ a.î. $x_S' \geq 1$

MAX-CUT

1) Se dă un graf neorientat $G = (V, E)$. Să se partiționeze V în V_1 și V_2 a.î. nr. muchiilor $(a, b) \in E$ cu $\begin{cases} a \in V_1 \\ b \in V_2 \end{cases}$ să fie maxim.

Ex:



Găsiți un alg. de 2-aprox + dem!

Alg. de 2 aproximare

~~Val. optimă~~

$V_1 = \{1\}$
$V_2 = \{2\}$

La fiecare pas pt. V_3+ mă uit care pune mai multe muchii în tăietură

$\Rightarrow V_1 = \{1, 3\}$, $V_2 = \{2\}$ $\Rightarrow V_1 = \{1, 3\}$, $V_2 = \{2, 4\}$

A = alg. este 2-aprox pt. P ⇔ pt. dată ⇒

Val. mea

$\Leftrightarrow A(I) \geq \dfrac{OPT(I)}{c}$

cnt. de muchii/clasa

$c = 5$

Pp. că $A(I) \leq \dfrac{OPT(I)}{c} \Leftrightarrow A(I) \leq \dfrac{4}{5}$

Cum avem nr. maxim de muchii = 4

$\Rightarrow A(I) \geq \dfrac{4}{5} \Rightarrow$ A este 2-aprox.

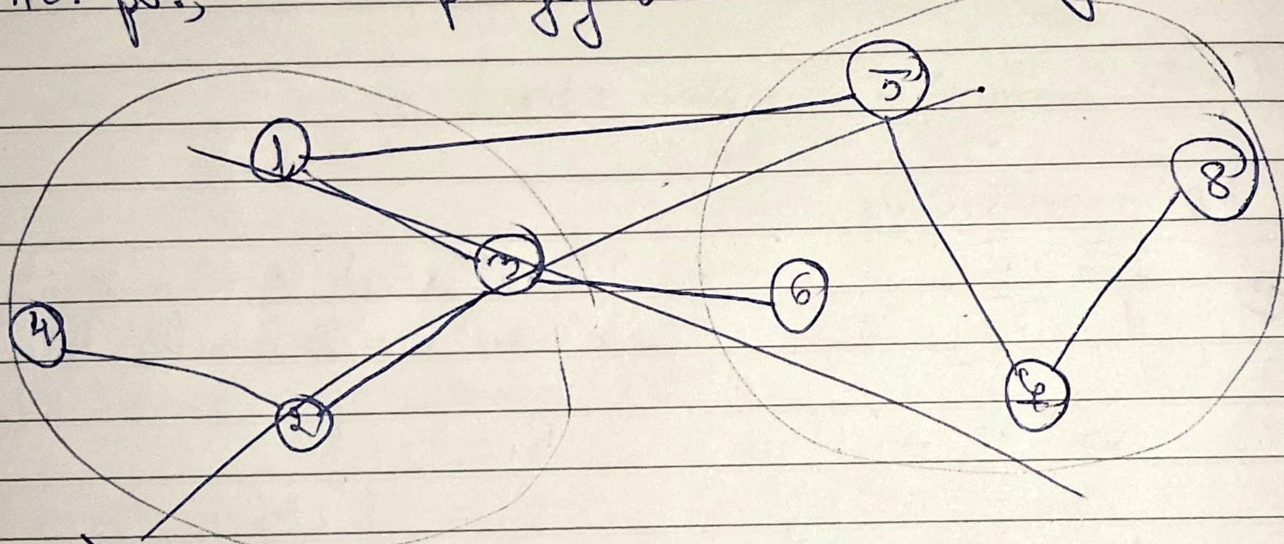
Var. corectă

$A(v)$ = muchiile adăugate în tăietură când îl procesăm pe v
$B(v)$ = muchiile care nu sunt adăugate

$$\sum_{\forall v \in V} A(v) + B(v) = |E|$$

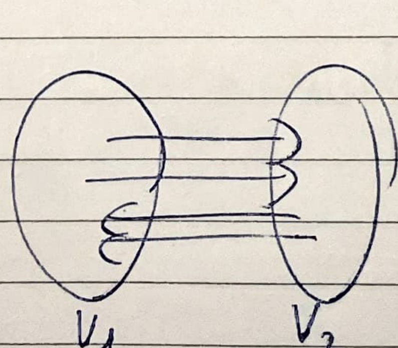
$$\Rightarrow ALG \geq \sum_{v \in V} A(v) \geq \frac{|E|}{2} \qquad \cancel{\max\{A(v), B(v), \forall v \in V\} \geq \frac{|E|}{2}}$$

2) Ac. pb., doar că pt. graf orientat. Vrem să găsim o 4-aprox



~~$V_1 = \{1\}$~~
~~$V_2 = \{2\}$~~
~~$V_3 = \{3\}$~~
~~$V_4 = \{4\}$~~

Ac. alg, dar cu arce orientate $\Rightarrow$



dar știm că $\sum$ arcelor $V_1 \to V_2 \geq$ jum. din arce
$\Rightarrow$ nr. de arce pt. max-cut $= \frac{1}{4} \Rightarrow$ 4-aprox.

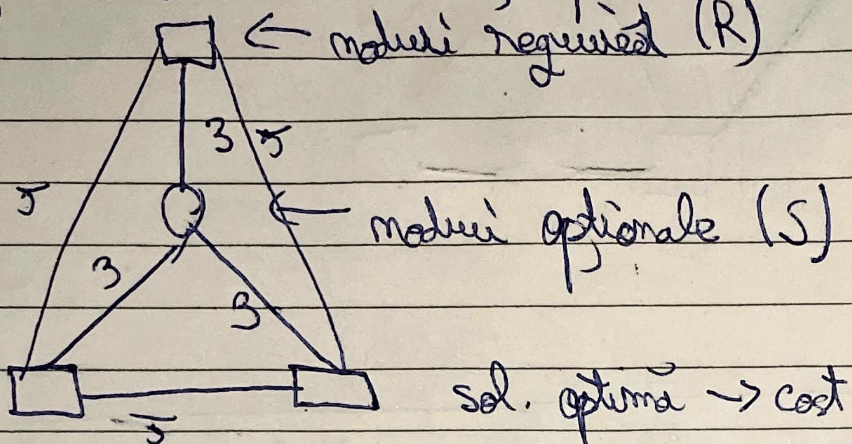
Cursul 7

Prog. alg. ef

Minimum Steiner Tree

Se dă un graf neorientat $G = (V, E)$ cu greutăți pe muchii și $V = R \cup S$, $R \cap S = \emptyset$. Să se găs. arborele de cost minim care conține nodurile din R și parțial unele noduri din S.

Ex:



sol. optimă → cost = 9

Alg. de 2 aproximare

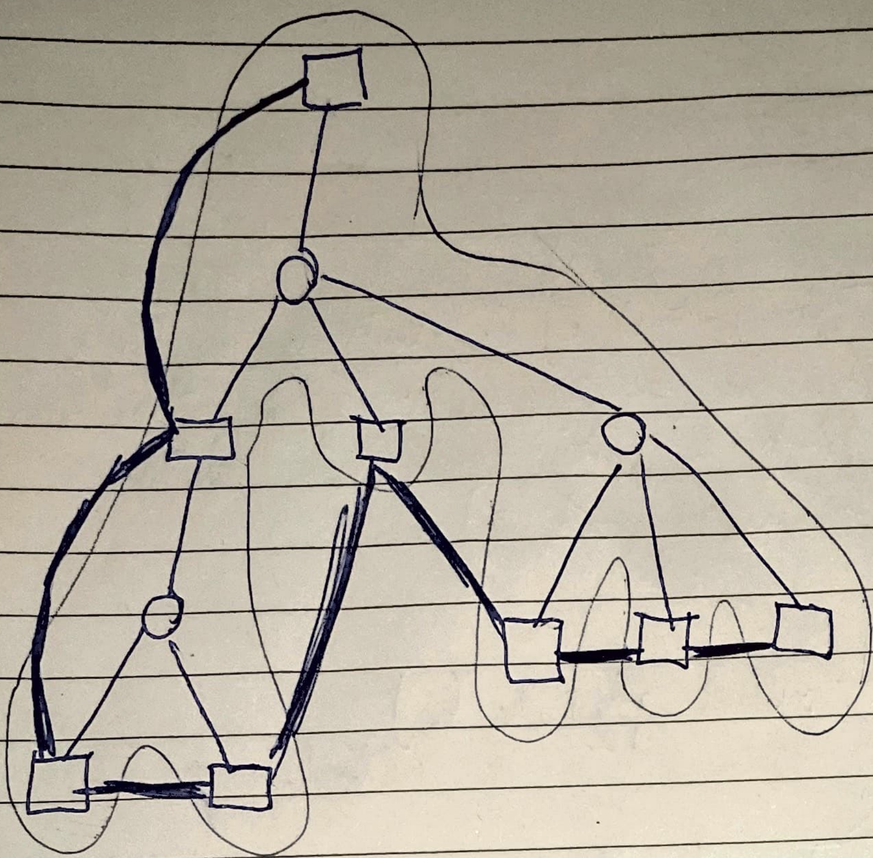
Calculam un MST (minimum spanning tree) pe mț. R.
Dc. în G distanța nu e metrică, înlocuim costul (a, b) cu "shortest path" de la a la b.

Dem:

Fie T un Steiner Tree optim.
~~Cost~~ Construiesc T' parcurgerea Euler a lui T ⇒ ~~Costul (T') = 2~~
$\Rightarrow cost(T') = 2 \cdot cost(T)$ (excluzând S)

Construiesc T'' prin scurtcircuitarea parcurgerii Euler T' ⇒
$\Rightarrow cost(T'') \leq cost(T') = 2 \cdot cost(T)$

T'' va fi pe un drum printre toate nodurile din mț. R, deoarece T'' arbore $\Rightarrow MST \text{ pe } R \leq cost(T'') \leq 2 \cdot \underbrace{cost(T)}_{\text{costul sol. optime}}$



$\boxed{\text{2-aprox. pt VC fol. Integer Linear Programming}}$

min $x - y$ a.î. $x \geq 3$
$x + y \geq 2$
$x, y \in \mathbb{R}$

– Linear Programming
(Pb. de prog. liniară)

$\boxed{\text{problemele de prog. lin. sunt rezolvabile în timp polin}}$

În general:

$$\min c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

$$a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \leq b_1$$

$$a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \leq b_2$$

$$\vdots$$

$$a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \leq b_m$$

$$x_i \in \mathbb{R}$$

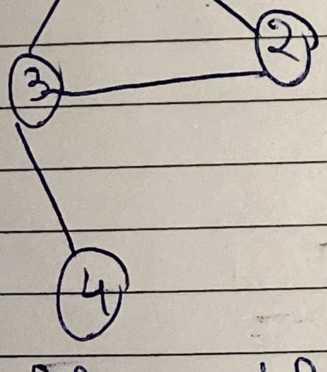
Pb. de progr. liniară în care $x_i \in \mathbb{Z}$ sunt NP-hard.
1. Vom exprima pb. VC ca un iLP (integer Linear ~~Problem~~ Program)
2. Vom folosi relaxarea iLP (var. pb. în care $x_i \in \mathbb{R}$) pt. a obț. un alg. de aprox.

$G = (V, E)$ instanța a VC
noduri v_i
Vrem o mț. de cardinalitate minimă.
fiecare muchie trebuie să fie acoperită

iLP
variabile $x_i \in \{0,1\}$

$$x_i = \begin{cases} 1, & \text{dc. nodul } v_i \text{ este în VC} \\ 0, & \cancel{\text{dc}} \text{ altfel } (v_i \notin VC) \end{cases}$$

$$\min \sum_{i=1}^{n} x_i \quad \text{a.î. } x_i + x_j \geq 1 \ \forall (x_i, y_i) \in E$$



$\min x_1 + x_2 + x_3 + x_4$ a.î.
$x_1 + x_2 \geq 1$
$x_1 + x_3 \geq 1$
$x_2 + x_3 \geq 1$
$x_3 + x_4 \geq 1$
$x_1, x_2, x_3, x_4 \in \{0,1\}$

(Formularea cu iLP)

Relaxarea LP

$\min x_1 + x_2 + x_3 + x_4$ a.î.
$x_1 + x_2 \geq 1$
$x_1 + x_3 \geq 1$
$x_2 + x_3 \geq 1$
$x_3 + x_4 \geq 1$
$x_1, x_2, x_3, x_4 \in \{0,1\}$

$ALG = \sum_{i=1}^{n} x_i^*$

OPT_{LP} $\quad OPT_{VC} = OPT_{iLP}$ $\quad \downarrow$ $\quad 2 \cdot OPT_{LP}$

$\leftarrow$ min

Vom rezolva (în timp polin) relaxarea problemei de progr. lin.
Fie x_i' ~~...~~ sol. opt.

$$\sum_{i=1}^{n} x_i' \le OPT_{ILP} = OPT_{VC}$$

Vrem să constr o sol. fezabilă (care satisface inegalitățile din ILP) x_i^* cu prop. că $\sum_{i=1}^{n} x_i^* \le 2 \cdot \sum_{i=1}^{n} x_i'$

$x_i^* \in \{0,1\}$

$$x_i^* = \begin{cases} 1, & dc.\ x_i' \ge \frac{1}{2} \\ 0, & dc.\ x_i' < \frac{1}{2} \end{cases} \Rightarrow x_i^* \le 2 \cdot x_i' \Rightarrow \sum_{i=1}^{n} x_i^* \le 2 \sum_{i=1}^{n} x_i'$$

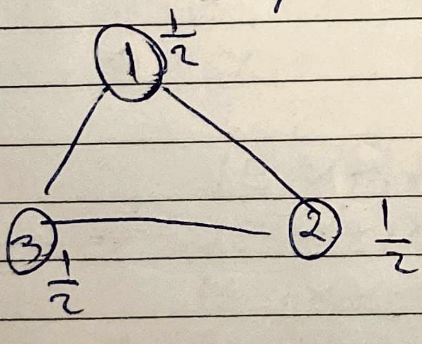
$$x_i^* + x_j^* \ge 1, \forall (x_i, y_i) \in E$$

Știm $x_i' + x_j' \ge 1 \Rightarrow x_i'$ sau $x_j' \ge \frac{1}{2} \Rightarrow x_i^*$ sau $x_j^* = 1$

$$\frac{ALG}{OPT_{LP}} \ge \frac{OPT_{ILP}}{OPT_{LP}} = \frac{2}{\frac{3}{2}} = \frac{4}{3}$$

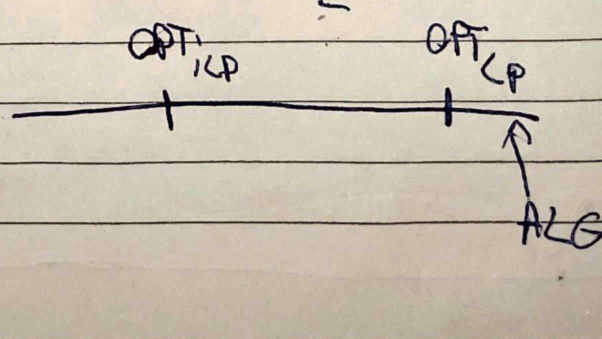
• integrality gap

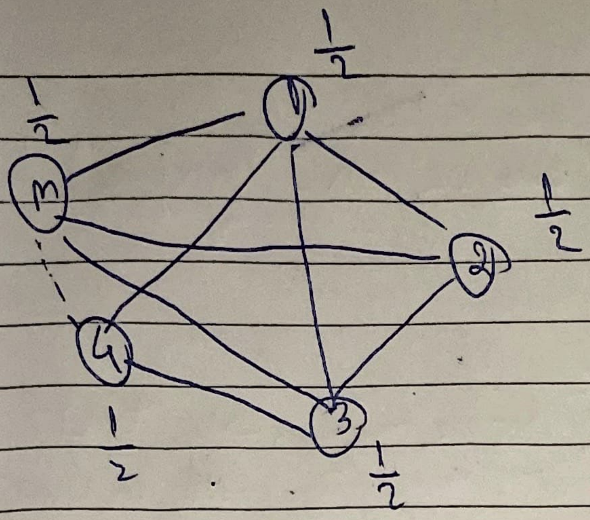
$$\max \frac{OPT_{ILP}(y)}{OPT_{LP}(y)}$$

Ex: 

$OPT_{ILP} = 2$

$OPT_{LP} = 1{,}5 = \frac{3}{2}$





K_n – graf complet

$OPT_{ILP} = n-1$

$OPT_{LP} = \frac{n}{2}$

$\frac{OPT_{ILP}}{OPT_{LP}} = \frac{n-1}{\frac{n}{2}} \xrightarrow[n\to\infty]{} 2$

Cursul 8
Prog. alg. ef.

Set Cover

Se dă o mt. de elemente U ← univers și o mt. $\mathcal{S}$ de submt. ale lui U. Varianta fără costuri ("unweighted") → toate submt. din $\mathcal{S}$ au cost 1.

Varianta cu costuri ("weighted") → pt. $S \in \mathcal{S}$ avem costul $c(S)$ (pozitiv). Vrem să găsim o colecție de mt. de cost minim din $\mathcal{S}$ a căror reuniune este U.

ex: $U = \{1,2,3,4,5\}$
$S_1 = \{1,2,3\}$ $S_2 = \{2,3\}$, $S_3 = \{3,4\}$ $S_4 = \{4,5\}$
$c(S_1) = c(S_2) = c(S_3) = c(S_4) = 1$ → sol. S_1 și S_4

alt ex: $U = \{1,2,3\}$
$S_1 = \{1,2\}$ $S_2 = \{2,3\}$ $S_3 = \{1,3\}$
$c(S_1) = 1$, $c(S_2) = 2$ $c(S_3) = 3$ ⇒ sol. optimă S_1 și S_2 $OPT = 3$
NU S_1 și S_3, unde cost

Algoritm $O(\log n)$ - aproximare
(adică o sol. care are max $O(\log n)$ sol)

Greedy

$C = \emptyset$

while $C \neq U$

Aleg $S \in \mathcal{S}$ a.î. $\frac{cost(S)}{|S \setminus C|}$ să fie minimum → raport preț / câte elem. acoperă

$C = C \cup \{S\}$

ex: pas 1: $S_1 : 1/2$, $S_2 : 2/2$, $S_3 : 3/2$ ⇒ alg S_1 → acoperă 1 și 2

pas 2: $S_2 : 2$ (are cost 2, acoperă un elem.), $S_3 : 3$ ⇒ aleg S_2 ⇒ acoperă 3

Dem:

Pt. un elem. $e \in U$ notăm cu $pret(e) = \frac{cost(S)}{|S \setminus C|}$ unde S este mulț. care acoperă e prima dată pe parcursul alg.

ex: $pret(1) = 1/2$, $pret(2) = 1/2$, $pret(3) = 2$ ⇒ $pret(1) + pret(2) + pret(3) = 3$

$\sum_{e \in U} pret(e) =$ costul soluției returnate de alg.

Fie $e_1, e_2, \dots, e_n$ ordinea în care sunt acoperite elem. de alg.

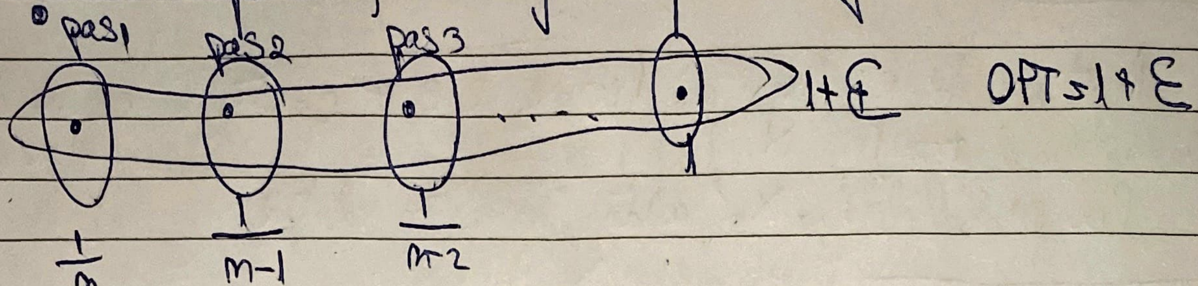
Ar. că $pret(e_i) \leq \frac{OPT}{n-i+1}$ (*)

$$(*) \Rightarrow \sum_{i=1}^{n} pret(e_i) = ALG \leq OPT \underbrace{\left(\frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2} + \dots + 1\right)}_{O(\ln n)}$$

$pret(e_1) \leq \frac{OPT}{n}$ $e_2, \dots, e_n$

ex: $OPT = 3$

! Nu se poate obț. un alg. de aprox. cu factor mai bun de $\ln n$.



$ALG = \frac{1}{n} + \frac{1}{n-1} + \dots + 1 \rightarrow$ este cu log

La Vertex Cover avem caz particular: o muchie are 2 vârfuri care să o acopere, de aceea acolo avem 2-aprox.

ex: $U = \{1, 2, 3, 4, 5, 6\}$
$S_1 = \{1, 2, 4\}$ dc. aleg S_1, x deoarece mă obligă să le iau pe toate după, când pot acoperi
$S_2 = \{1, 2, 3\}$
$S_3 = \{4, 5, 6\}$ toate nr. doar cu S_2 și S_3

Alg. cu factor $O(\log n)$ fol. programare liniară pe întregi (ILP).

Formularea ILP:

$x_S = \begin{cases} 1, & \text{dc. } S \text{ este în soluție} \\ 0, & \text{altfel} \end{cases}$

Vrem să minimizăm $\sum_{S \in \mathcal{S}} x_S \cdot c(S)$ a.î. $\sum_{S \ni e} x_S \geq 1 \ \forall e \in U$, $x_S \in \{0, 1\}$

Relaxarea pb: min $\sum_{S \in \mathcal{S}} x_S \cdot c(S)$ a.î. $\sum_{S \ni e} x_S \geq 1$, $x_S \in [0, 1]$

Rezolvăm relaxarea și obținem val. $x_S' \in [0,1]$

$$\sum_{S \in P} x_S' \cdot c(S) \leq OPT$$

<u>Algoritm:</u>

Repetăm de $c \cdot \log m$ ori (unde c e o constantă ce va fi aleasă mai tz)

Cu probabilitatea x_S' adăugăm S în soluție

1. Sol. returnată are cost $\leq O(\log m) \cdot OPT$
2. Sol. returnată este un set cover valid

Pr. ca un element $a \in U$ să nu fie acoperit la un pas (o iterație) $\leq$ Pr (să nu selectăm niciuna din mult. $S_1, S_2, \ldots, S_K$)

Pp. că $a \in S_1, S_2, \ldots, S_K$)

$$\prod_{i=1}^{K} (1 - x_{S_i}') \leq \left(1 - \frac{1}{K}\right)^K \leq \frac{1}{e}$$

Știm că $x'_{S_1} + x'_{S_2} + \ldots + x'_{S_K} \geq 1$

Pr (a să nu fie acoperit în toți cei $c \cdot \log m$ pași) $\leq \left(\frac{1}{e}\right)^{c \cdot \log m} \leq \frac{1}{4m}$

Alegem c a.î. $\left(\frac{1}{e}\right)^{c \cdot \log m} \leq \frac{1}{4m}$

Pr (să $\exists$ un elem. din U care să nu fie acoperit) $\leq$ Pr (a_1 să nu fie acoperit, a_2 ... în $c \cdot \log m$ pași) $\leq$

$$\leq \sum_{a \in U} Pr(a \text{ să nu fie acoperit în } c \cdot \log m \text{ pași}) \leq m \cdot \frac{1}{4m} \leq \frac{1}{4}$$

Am ar. că e un set cover valid cu prob. $3/4$.

Costul sol. mediu al $\leq (c \cdot \log m) \cdot \left(\sum_{S \in P} x_S' \right)$

$$Y_S = \begin{cases} 1, & \text{dc. } S \text{ este în soluție} \\ 0, & \text{altfel} \end{cases}$$

Costul mediu al sol $= \sum_{S \in P} \mathbb{E}[Y_S] = \sum_{S \in P} c \cdot \log m \cdot x_S'$

media lui Y_S

Costul mediu al sol. $\leq c \cdot \log m \cdot OPT$

$$\boxed{Pr(X \geq a) \leq \frac{E(X)}{a} \quad \text{(Markov)}}$$

$Pr(\text{costul mediu al sol.} \geq 4 c \log m \cdot OPT) \leq \frac{1}{4}$

Pr(alg. să returneze o sol. fezabilă și are costul $< 4 \cdot \log m \cdot OPT) \geq \frac{1}{2}$

Cursul 9

~~Seminarul 3~~

Prog. alg. ef.

Pb. rucsacului

Se dau n obiecte $a_1, a_2, \ldots, a_n$. Fiecare obiect are $size(a_i)$ și $profit(a_i)$ și se dă un sac de mărime B. Să se selecteze o mulț. de obiecte de profit maxim a.î. mărime totală să fie cel mult B.

Obs: Nu avem voie să alegem bucăți din obiecte.

ex: $n = 4$, $B = 8$

a_1: $size(a_1) = 2$, $profit(a_1) = 2$

a_2: $size(a_2) = 3$, $profit(a_2) = 4$

a_3: $size(a_3) = 7$, $profit(a_3) = 6$

a_4: $size(a_4) = 1$, $profit(a_4) = 2$

OPT = 8

Alegem a_3, a_4.

Notăm cu $P = \max_{i=1}^{m} profit(a_i)$

- Programare dinamică (Alg. exact)

$A(i, c) = \{$ greutatea minimă prin care pot obț. profit c cu primele i obiecte $\}$

Deci vom calcula $A(i, c)$ pt. $1 \le i \le m$, $0 \le c \le P$ | $\Rightarrow$ sol. optimă va fi cel mai mare c a.î. $A(m, c) \le B$

$A(1, profit(a_1)) = size(a_1)$, $A(1, x) = \infty$, $\forall x \neq profit(a_1)$

$A(i, c) = \min\{A(i-1, c), A(i-1, c - profit(a_i)) + size(a_i)\}$

	1	2	3	4	5	6	8	10	12
1	∞	2	∞	∞	∞	∞	∞	∞	∞
2		2		3		5	8		
3		2		3		5	9	10	12
4		1				4	6	10	11

Timp de rulare: $O(m^2 P)$

Def. $K = \frac{\varepsilon P}{m}$, pt $\varepsilon > 0$. Creez obiectele $a_1', a_2', \dots, a_m'$ cu $size(a_i') = size(a_i)$ și $profit(a_i') = \left\lfloor \frac{profit(a_i)}{K} \right\rfloor$.

Rulăm alg. de progr. dinamică. Timpul de rulare al alg. de aprox. va fi $O\left(m^2 \cdot \frac{P}{K}\right) = O\left(m^2 \cdot \frac{P}{\frac{\varepsilon P}{m}}\right) = O\left(m^3 \cdot \frac{1}{\varepsilon}\right)$

Vrem să ar. că sol. S returnată de alg. de aprox are

$\underbrace{K \cdot profit(S)}_{ALG} \ge (1 - \varepsilon) \underbrace{profit(O)}_{OPT}$, unde O = sol. opt.

Not. cu O' obiectele trunchiate din sol. optimă

$profit(a_i) - K \cdot profit(a_i') \le K$

$profit(O) - K \cdot profit(O') \le m \cdot K$

$profit(S) \ge profit(O')$

$K \cdot profit(S) \ge profit(O) - m \cdot K$

$profit(O) \ge P$

$\Rightarrow K \cdot profit(S) \ge P - \cancel{m} \cdot \frac{\varepsilon P}{\cancel{m}}$

$\ge P(1 - \varepsilon)$