Proiect. alg. ef.

Cursul 1

* dem. că anumite pb. sunt ușor de rez.

1. Cum dem. că o pb. este NP-completă (2-3 cursuri) → găsim un alg. polinomial
2. Alg. de aproximare (4-5 cursuri) "Approximation Algorithms"
3. Fixed parameter algorithms (3-4 cursuri)
4. Metode de programare liniară
5. Metaeuristici
6. Pb. de decizie (răsp. DA/NU)

Ex. de pb.:

1) Se dă un șir de cifre 0, 1. Este șirul de forma $0^k, 1^k$?

2) Se dau $x, y \in \mathbb{N}$, sunt prime între ele?

Problemele sunt repr. de limbajele (mulț. de șiruri)

$L_1 = \{0^k 1^k \mid k \geq 1\}$

$L_2 = \{p, q \mid p, q \in \mathbb{N}^*$ și p, q prime între ele$\}$

$L_2 = p, q \mid p, q$ sunt reprezentări în baza ale lui p, q.

Algoritmi → Mașini Turing

Ex. de pb. nedecizională: soft de testare: DA / NU / rulează la ∞

$A_{TM} = \{ \langle M, m \rangle \mid M$ acceptă inputul $m \}$

Pb. A_{TM} este nedecizională

• Soluție: pp. prin RA că A_{TM} decidabilă $\Leftrightarrow \exists$ o Mașină Turing care acceptă limbajul H care ia ca input M și m și acceptă m, altfel respinge

$H \langle M, m \rangle$ { acceptă, dc. M acceptă m ; nu acceptă, dc. M nu acceptă m }

Definesc o nouă MT D care are ca input o MT M

$D \langle M \rangle =$ { acceptă, dc. M respinge ; respinge, dc. M acceptă }

$D \langle H \rangle =$ { respinge, dc. H acc. ; acceptă, dc. H resp. }

$D<D> = \begin{cases} \text{acceptă, dc. respinge} \\ \text{respinge, dc. acceptă} \end{cases}$

$HALT_{TM} = \{ <M, w> \mid M \text{ se oprește pe inputul } w \}$

Teoremă: $HALT_{TM}$ nedecidabilă

timp optim – probl. decidabile

$L_1 = \{ 0^k 1^k \mid k \geq 1 \}$

0	0	$\cancel{0}$	0	$\cancel{1}$	1	1	1	1	<

blank

n = mărime input, $\sim n^2$ pași

$L_1 \in TIME(n^2)$

$0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ 0\ \cancel{0}\ \ 1\ \cancel{1}\ 1\ \cancel{1}\ 1\ \cancel{1}\ 1\ \cancel{1}$

- marcam 0-urile din 2 în 2
- la fel la 1

$L_1 \in TIME(n \log n)$

$\exists$ 2 clase mari de complexitate P NP P=NP

$P = \bigcup_{k \geq 1} TIME(n^k)$ → toate pb. care pot fi rez. în timp n^k

Programarea alg. eficienți

Seminar I

P1) Se dau x, y. Sunt prime între ele?
Este problema P_1 în clasa de complexitate P?

$10 \quad 8 \quad \to$ c.m.m.d.c
$2 \quad 8$
$2 \quad 6$
$2 \quad 4$
$2 \quad 2$
$\boxed{2} \quad 0$

$O(\max(x,y))$ — scăderi repetate $(=)$ exponențial

n = mărimea inputului $= \log_2 x + \log_2 y$

$x = 2^{\log_2 x}$

$10 : 8 =$ ~~…~~ — , 2

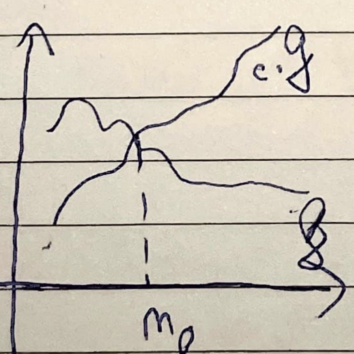
$10 \quad 8$
$2 \quad 8$
$2 \quad 0$

Alg. lui Euclid $\to O(\log_2 n)$

$$\left\{ 10 = a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 \right\}$$
$$1010$$

$O, o, \Omega, \omega, \Theta$

• O



$f \in O(g) \iff \exists\, n_0, c > 0$ a.î.
$\forall n \geq n_0$ avem
$$\boxed{f(n) \leq c \cdot g(n)}$$

Exemple: $\rightarrow n \in O(n^3)$

Putem alege: $c = n_0 = 1$

$n \le 1 \cdot n^2$

$\forall\, n \ge 1$

$\rightarrow 100n \in O(n)$

Alegem $c = 150$

$n_0 = 2$

$\rightarrow 2^{n+1} \in O(2^n)$

$c = 2$

$n_0 = 1$

$\rightarrow 2^{2n} \notin O(2^n)$

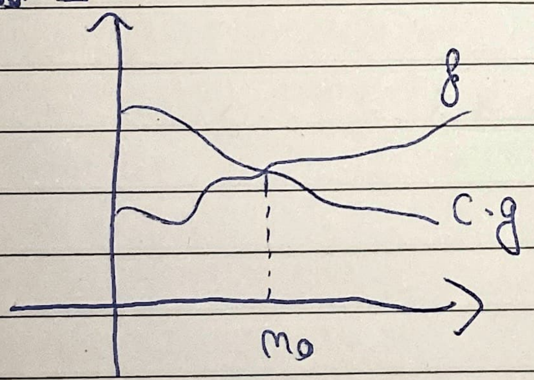
Pp. prin RA că $2^{2n} \in O(2^n)$

Fie c, n_0 constantele din def.

$\Rightarrow 2^{2n} \le c \cdot 2^n, \ \forall\, n \ge n_0$

$2^n \le c, \ \forall\, n \ge n_0 \Rightarrow$ do, $c = ct$, $2^n \to \infty$

Ω



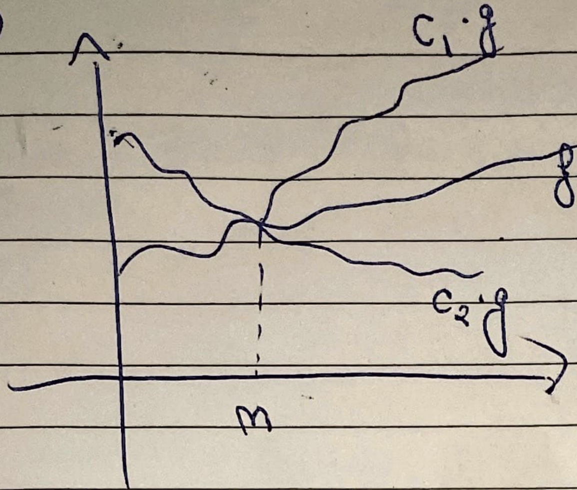
$f \in \Omega(g) \Leftrightarrow \exists\, n_0, c > 0$ a.î. $\forall\, n \ge n_0 \Rightarrow$

$\boxed{f(n) \ge c \cdot g(n)}$

Exemple: $\rightarrow n^3 \in \Omega(n^2)$

$\frac{n^3}{1000} \in \Omega(n^3)$

• Θ



$f \in \Theta(g) \Leftrightarrow \exists n_0, c_1, c_2 > 0$ a.î. $\forall n \geq n_0 \Rightarrow$
$\boxed{c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)}$

Exemple: $\Rightarrow n_0 = 1$
$c_2 = 1$ $\quad c_1 \cdot n^3 \leq \frac{n^3}{1000} \leq c_2 \cdot n^3$
$c_1 = \frac{1}{1000}$ $\quad \Leftrightarrow \frac{n^3}{1000} \leq \frac{n^3}{1000} \leq n^3$

$\frac{n^3}{1000} \in \Theta(n^3)$

Teoremă: $\left.\begin{array}{l} f \in O(g) \\ f \in \Omega(g) \end{array}\right| \Leftrightarrow f \in \Theta(g)$

$f(n) \in O(g(n)) \Leftrightarrow f(n) \leq c \cdot g(n)$
$f(n) \in \Omega(g(n)) \Leftrightarrow f(n) \geq c \cdot g(n)$
$f(n) \in \Theta(g(n)) \Leftrightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

(1) Dem. $\Rightarrow$
$\exists n_{01}, c_1 > 0$ a.î. $\forall n \geq n_{01} \Rightarrow f(n) \leq c_1 \cdot g(n)$
$\exists n_{02}, c_2 > 0$ a.î. $\forall n \geq n_{02} \Rightarrow f(n) \geq c_2 \cdot g(n)$ $\Big/ \Rightarrow$
$n_0 = \max(n_{01}, n_{02})$

$\Rightarrow \exists n_0, c_1, c_2 > 0$ a.î. $\forall n \geq n_0 \Rightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

(2) Dem. $\Leftarrow$
$\exists n_0, c_1, c_2 > 0$ a.î. $\forall n \geq 0 \Rightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$n_0 = \max(n_{01}, n_{02}) \Rightarrow$ invers

Din (1), (2) $\Rightarrow$ ipoteza

• o

Exemple: $n^2 \in o(n^2 \log n)$

$n \in o(n^2)$

$n \notin o(100n)$

$n^2 \in \omega(n)$

$n^2 \notin \omega\left(\frac{n^2}{10}\right)$

$f \in o(g) \Leftrightarrow \forall c > 0\ \exists n_0 > 0$ a.î. $\forall n \geq n_0$

$\Rightarrow \boxed{f(n) < c \cdot g(n)}$

• ω

$f \in \omega(g) \Leftrightarrow \forall c > 0,\ \exists n_0 > 0$ a.î. $\forall n \geq n_0 \Rightarrow$

$\boxed{f(n) > c \cdot g(n)}$

Exemple:

$n \in o(n^2)$ $\qquad n < c \cdot n^2$

Fie $c > 0$ fixat

Alegem $n_0 = \left\lfloor \frac{1}{c} \right\rfloor + 1$ $\quad \forall n \geq n_0$

Pp. că $n \in o(100n)$

Fie c d. din def. $\Rightarrow \exists n_0 > 0$ a.î. $\forall n \geq n_0 \Rightarrow$

$\Rightarrow n < c \cdot 100n$

Dacă $c < \frac{1}{100} \Rightarrow \forall n,\ n < n \Rightarrow$ de, $\forall n \in \mathbb{N}$

1) $n! \in O(n^n)$

2) $\log_2 n! \in \Theta(n \log n)$ (A.)

$\log_2 n! \le c \cdot n \log n$ (A.) $\Rightarrow \log_2 n! \in O(n \log n)$

$c_1 \cdot n \log_2 n \le \log_2 n! \le c_2 \cdot n \log_2 n$

Programarea alg. ef.
Cursul 2

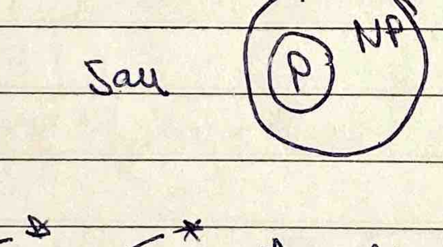
P – clasa pb. care pot fi rezolvate în timp polin.
NP – clasa pb. care pot fi verif. în timp polin.

A verifica: un verificator pt. un limbaj A. Pb. este un alg. a.î. A conține: $A = \{ w \mid \langle w, c \rangle$ sunt acceptate de verificator unde c este un șir de mărime polin $\}$

P = NP sau P NP => nu știm

Def: $f : \Sigma^* \to \Sigma^*$ este polin-calculabilă dc. $\exists$ o mașină Turing care rulează în timp polin. și se oprește cu $f(w)$ pe bandă atunci când are w ca început.

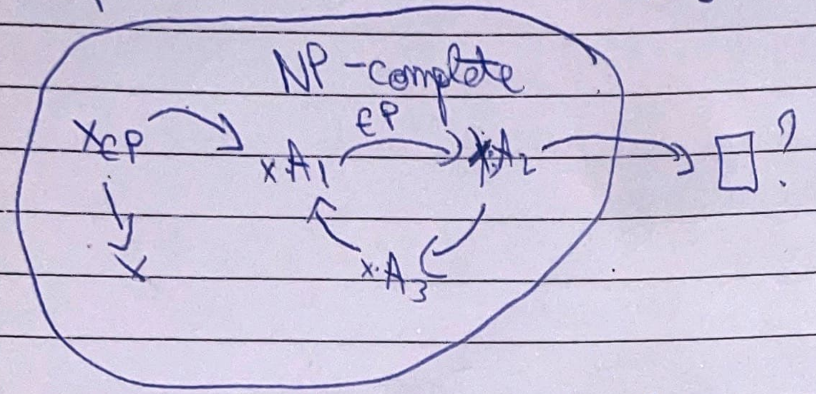
Def: O pb. A este polin. reductibilă la o pb. B dc. $\exists f : \Sigma^* \to \Sigma^*$ polin. calculabilă a.î.: $w \in A \Leftrightarrow f(w) \in B$



Def: O pb. A este NP-completă dc:

1) $A \in NP$

2) Orice problemă B din clasa NP se poate reduce în timp polin. la A $\Leftrightarrow B \le_p A$



Def: Pb. SAT (satisfiable)

Se dă o formula booleană $\Phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, unde $C_i = x_{1j} \vee x_{2j} \vee \ldots \vee x_{ij}$

C_i - clauze

x_{ij} - variabile booleene

Să se decidă dc. $\exists$ o asignare a variabilelor a.î. Φ să fie True

Ex: $\Phi = \underbrace{(x_1 \vee \bar{x}_2)}_{C_1} \wedge \underbrace{(x_1 \vee x_3 \vee \bar{x}_1)}_{C_2} \wedge \underbrace{(\bar{x}_3 \vee x_4)}_{C_3} \wedge \underbrace{\bar{x}_1}_{C_4} \wedge \underbrace{(x_2 \vee \bar{x}_3 \vee \bar{x}_4)}_{C_5}$

$\left.\begin{array}{l} x_1 = F \\ x_2 = F \\ x_3 = 1 \\ x_4 = F \end{array}\right\} \Rightarrow$ Da.

- Lemă: SAT este NP-completă

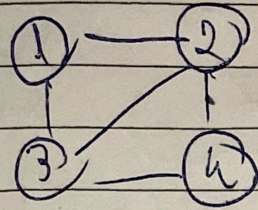
$SAT \le_p 3\text{-}SAT$

3-SAT: $\Phi = C_1 \wedge C_2 \ldots \wedge C_m$

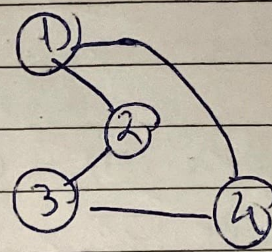
$C_i = (x_{1j} \vee x_{2j} \vee \ldots \vee x_{ij})$

Teoremă: Pb. ~~Clique~~ Clică este NP-completă (Maximum Clique)

Def: Se dă un graf neorientat $G=(V,E)$ și un nr. k. Să se decidă dc. $\exists\ V' \subseteq V$ cu $|V'| \geq k$ a.î. $\forall a,b \in V'$, $a \neq b$, $(a,b) \in E$



$K=3$ Da.



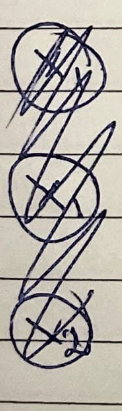
$k=3$ Nu.

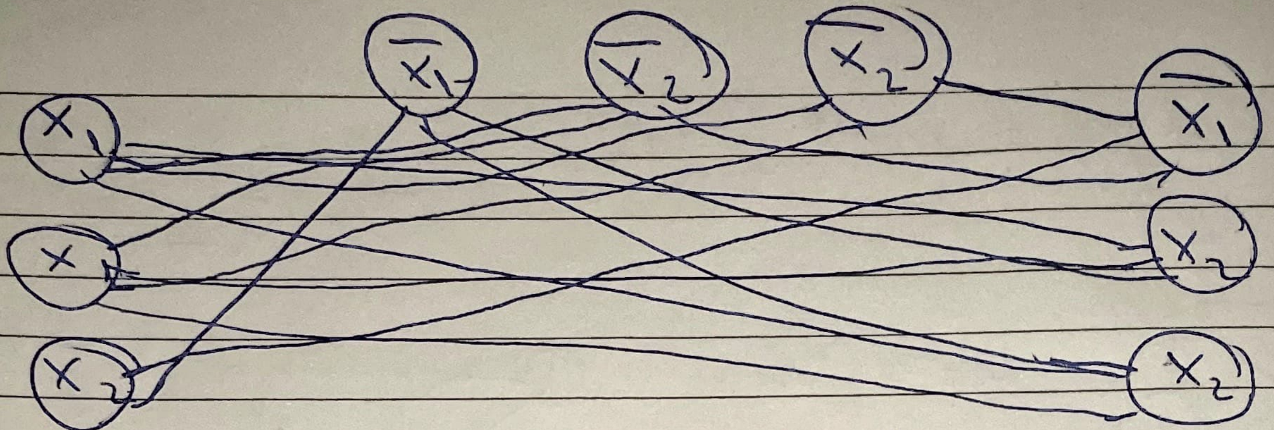
Se dă o formulă Φ, instanță a 3-SAT. Construim o ct. m. Pb. Maximum Clique, adică $\langle G, K \rangle$ unde G este un graf și un $K \in \mathbb{Z}$ a.î.:

- G va avea $3m$ noduri, câte 3 noduri pe clauză
- fiecare nod va avea eticheta variabilei din clauza respectivă
- $\forall a,b$, vom avea o muchie cu excepția cazurilor:

1) a,b sunt noduri din ac. grup de noduri
2) a are eticheta x și $b \to \overline{x}$ (k = nr. clauze = 3)

$$(x_1 \vee x_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2 \vee x_2)$$





$K=3$
$x_1 = F$
$x_2 = 1$

ϕ satisfiabilă $\Leftrightarrow$ $\exists$ o clică de[(1)] mărime $= K$ în G | $\exists$ o clică $\geq K$ în G

Dc. în clauza C_i avem literalul x_{ij} = True, at. punem nodul coresp. în clică

(1) Aceste noduri $(x_2, \overline{x_1}, x_2)$ formează o clică de mărime K pt. că sunt din grămezi dif. și nu putem avea x T și F simultan

(2) Fie nodurile x_1, x_2 din clică, dc. știm că x_i sunt din grămezi dif.
Dc. un nod cu eticheta x este în soluție, at. setez variabila $x = T$, altfel $x = F$.

2-SAT $\in P$
Max 2-SAT $\in$ NP-complete

~~Prog. alg. ef.~~
~~Seminarul 13~~

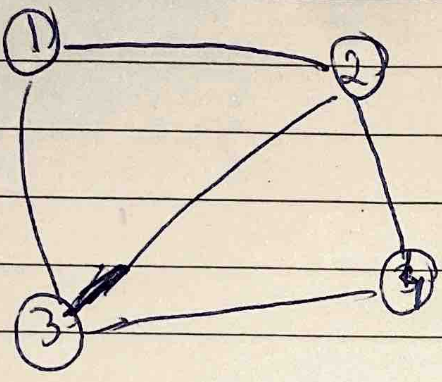
4/10

Proiectare alg. ef.

Seminarul 2

· Pb. Independent Set

Se dă un graf neorientat $G = (V, E)$ și un nr. k. Să se decidă dc. $\exists$ o submt. de noduri $I \subseteq V$, $|I| \geq k$ a.î. $\forall a, b \in I$, $(a, b) \notin E$.

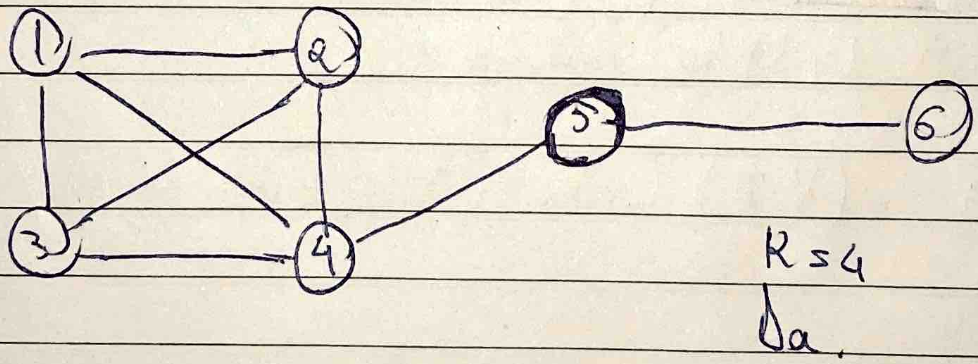


$k = 2$ Da.

$k = 3$ Nu.

· Pb. Clique

Se dă un graf neorientat $G = (V, E)$ și un nr. k. Să se det. de. $\exists$ o submt. de noduri $V' \subseteq V$, $|V'| \geq k$ a.î. $\forall a, b \in V'$, $(a, b) \in V'$

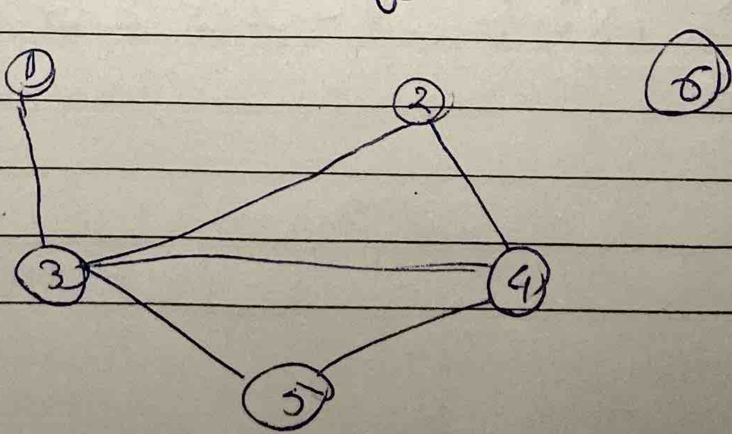


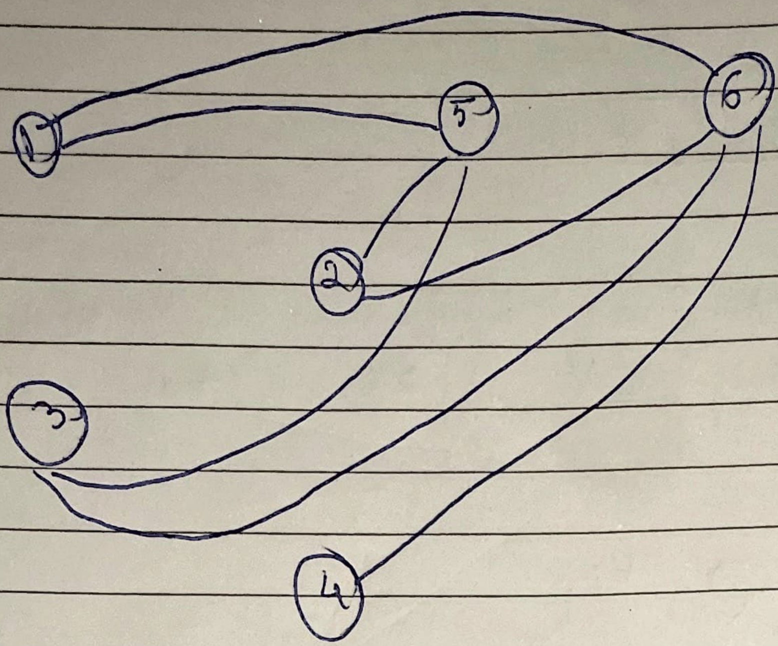
$k = 4$ Da.

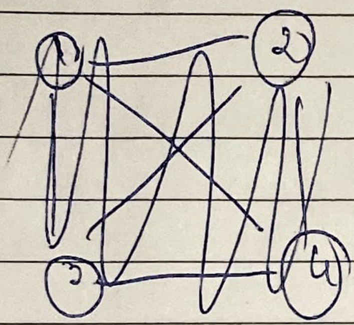
$k = 5$ Nu.

Teorema 1: Clique $\leq_p$ Independent Set

Teorema 2: Independent Set $\leq_p$ Clique







Dându-se un graf $G = (V, E)$ și un nr. K (instanță a Clique) construim ~~..~~ un graf $G' = (V', E')$ și un nr. K' a.î. G are o ~~..~~ clică de mărime $K \Leftrightarrow G'$ are un independent set (IS) de mărime K.

$G' = \overline{G} = (V, E)$ unde $(a,b) \in \overline{E} \Leftrightarrow (a,b) \notin E$

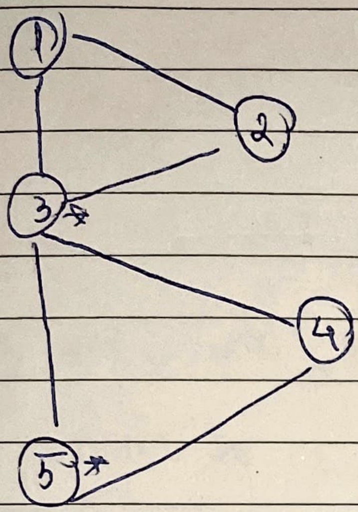
$K' = K$

„$\Rightarrow$" Fie V o clică în G. Ar. că V este IS în G' $\forall a, b \in V$ avem că $(a,b) \notin \overline{E}$ pt. că $(a,b) \in E$, deoarece V - clică.

„$\Leftarrow$" Fie V ~~un~~ altun IS în G'. V - clică în G, deoarece $\forall a, b \in V$ avem $(a,b) \in E$

Def: Vertex Cover (VC)

Se dă un graf G neorientat $G = (V, E)$ și un nr. K. Să se decidă dc. $\exists$ o mț. $V' \subseteq V$, $|V'| \leq K$ a.î. $\forall (a, b) \in E \Rightarrow a \in V'$ sau $b \in V'$.



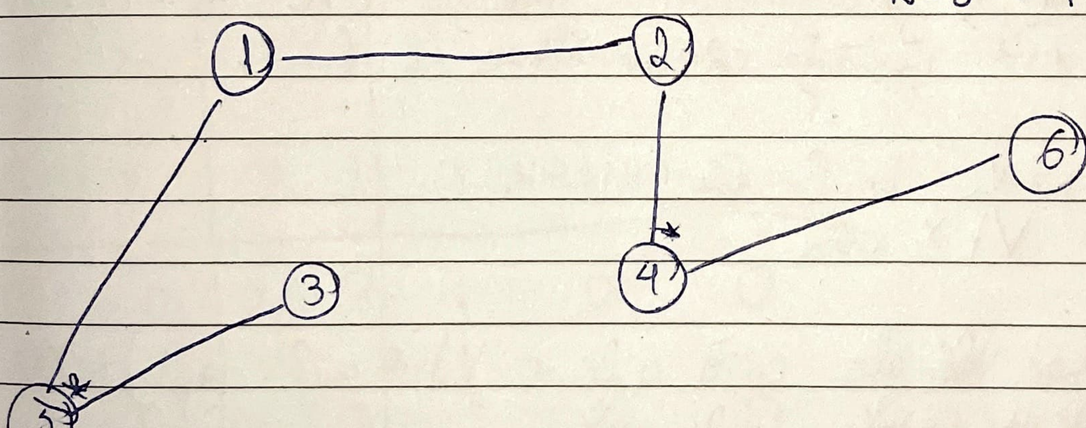
$K = 2$ — Nu.

$K = 3$ — Da.

Teorema 4: $VC \leq_p IS$

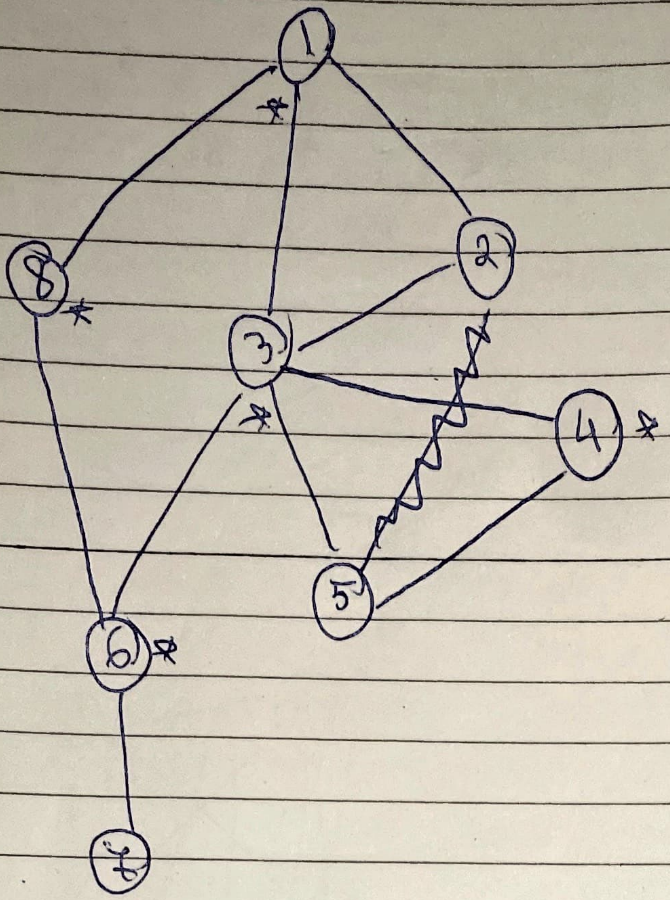
$K = 2 \rightarrow$ IS ✓, VC ✗

$K = 3 \rightarrow$ IS , VC ✓



$(G, k) \Rightarrow (G, n - k)$

n = nr. de noduri ale lui G



Muchiile fără * = IS
cu * = VC

Fie $G = (V, E)$ și un nr. k o instanță a VC $G = (V, E)$ și $k' = n - k$ este instanța corespunzătoare a IS.

„$\Rightarrow$" Fie X un VC în G cu $|X| \leq k$.
Ar. că $V \setminus X$ este un IS.

Pp. prin red. la abs. că $\exists\, a, b \in V \setminus X$ a.î. $(a, b) \in E \Leftrightarrow$ $a \in X$ sau $b \in X$ $\Rightarrow$ ↯ $a, b \in V \setminus X$
(def. lui VC)

„$\Leftarrow$" Fie X un IS în G. Ar. că $V \setminus X$ este un VC în G.

$\forall (a, b) \in E \Rightarrow a \in V \setminus X$ sau $b \in V \setminus X$

Pp. prin red. la abs. că $a \notin V \setminus X$ și $b \notin V \setminus X \Leftrightarrow a \in X$ și $b \in X$ $\Rightarrow X$ nu e IS $\Leftrightarrow$ ↯

· Pb. Subset Sum (SS)

Se dă o mulț. de nr. $S = \{s_1, s_2, \dots, s_m\}$ și un nr. k. Să se det. dc. ∃ o submulț. a lui S a cărei sumă este k.

Ex: $S = \{2, 4, 3, 7\}$, $k = 5 \Rightarrow$ Da

~~1825~~ $k = 8 \Rightarrow$ Nu

Teoremă: 3-SAT $\leq_p$ Subset Sum

$$\Phi = \underbrace{(x_1 \vee x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_2})}_{C_2} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee x_2)}_{C_3}$$

$x_1 = F$, $x_2 = T$

	Variabile		Clauze C_1	C_2	C_3
x_1	1	0	1	0	0
$\overline{x_1}$	1	0	0	1	1
x_2	0	1	1	0	1
$\overline{x_2}$	0	1	0	1	0
c_1	0	0	1	0	0
c_1'	0	0	1	0	0
c_2	0	0	0	1	0
c_2'	0	0	0	1	0
c_3	0	0	0	0	1
c_3'	0	0	0	0	1
K	1	1	3	3	3

cue ∃ macar la col. variabile

Φ Sat $\Rightarrow$ ∃ submulț. cu sumă K

⇐

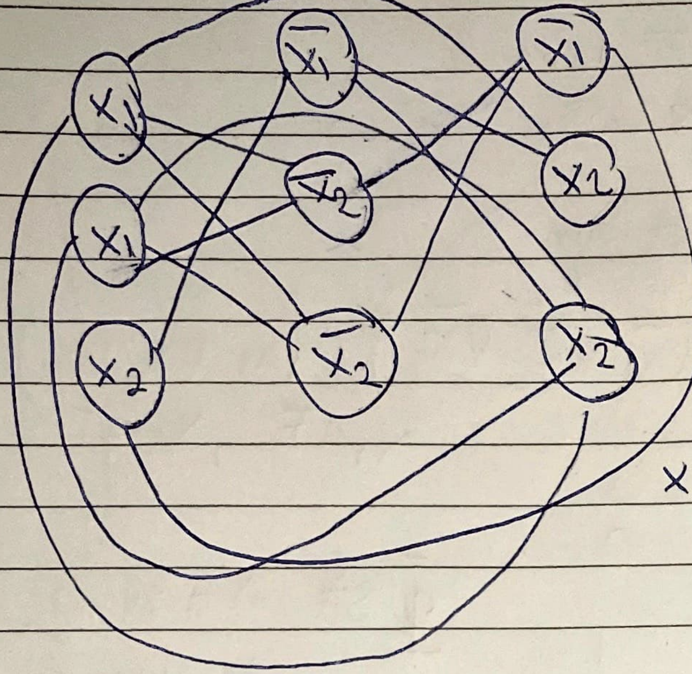
$\overline{x_1} + x_2 \Rightarrow 1\ 1\ |\ 1\ 1\ 2$

⋮

Dc. val. e false ⇒ alegem $\overline{m}$. Altfel, alegem m. Căutăm până la suma 3 3 3

Prog. alg. ef.
Cursul 3

$\underline{\Phi} = \underbrace{(x_1 \vee x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_2})}_{C_2} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee x_2)}_{C_3}$



1. $\underline{\Phi}$ satisfiabilă $\Leftrightarrow \exists$ o clică de mărime $\geq k$ în G

2. $\exists$ o clică $\geq k$ în G $\Rightarrow$ $\underline{\Phi}$ satisfiabilă $k = m$ (nr. de clauze) ≤ 3

$x_1 = F$, $x_2 = T$ $\checkmark$ $\Rightarrow$ clică (*)

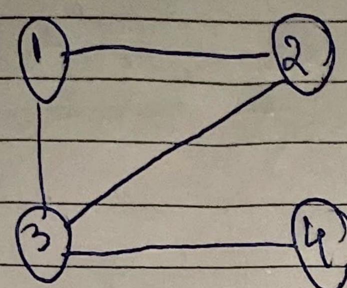
$x_2 \overline{x_1} x_2$

1. Dc. în clauza C_i avem literalul $x_{ij} = T$ at. punem nodul coresp. în clică
Aceste noduri (*) form. o clică de mărime k pt. că sunt din grupuri dif. și nu putem avea x_i = true și false în ac. timp (reg. 1, 2)

2. Fie $x_1 \ldots x_k$ din clică, știm că x_i sunt grupuri dif. Dc. x_i a fost un nod cu eticheta x este în soluție at. setez validă x cu true, ~~a. î.~~ astfel setez x cu false

Pb. Vertex Cover

Se dă un graf neorientat $G = (V, E)$ și un nr. k. Să se decidă dc. $\exists$ cel mult. de noduri $\exists$ $V' \subseteq V$, $|V'| \leq k$ a.î. $(a, b) \in E$ avem $a \in V'$ sau $b \in V'$ (acoperim toate muchiile grafului)

Ex: 

$K=1$ Nu

$K=2$ Da $\to \{3,1\}$, etc

Teoremă: Vertex Cover este NP-completă

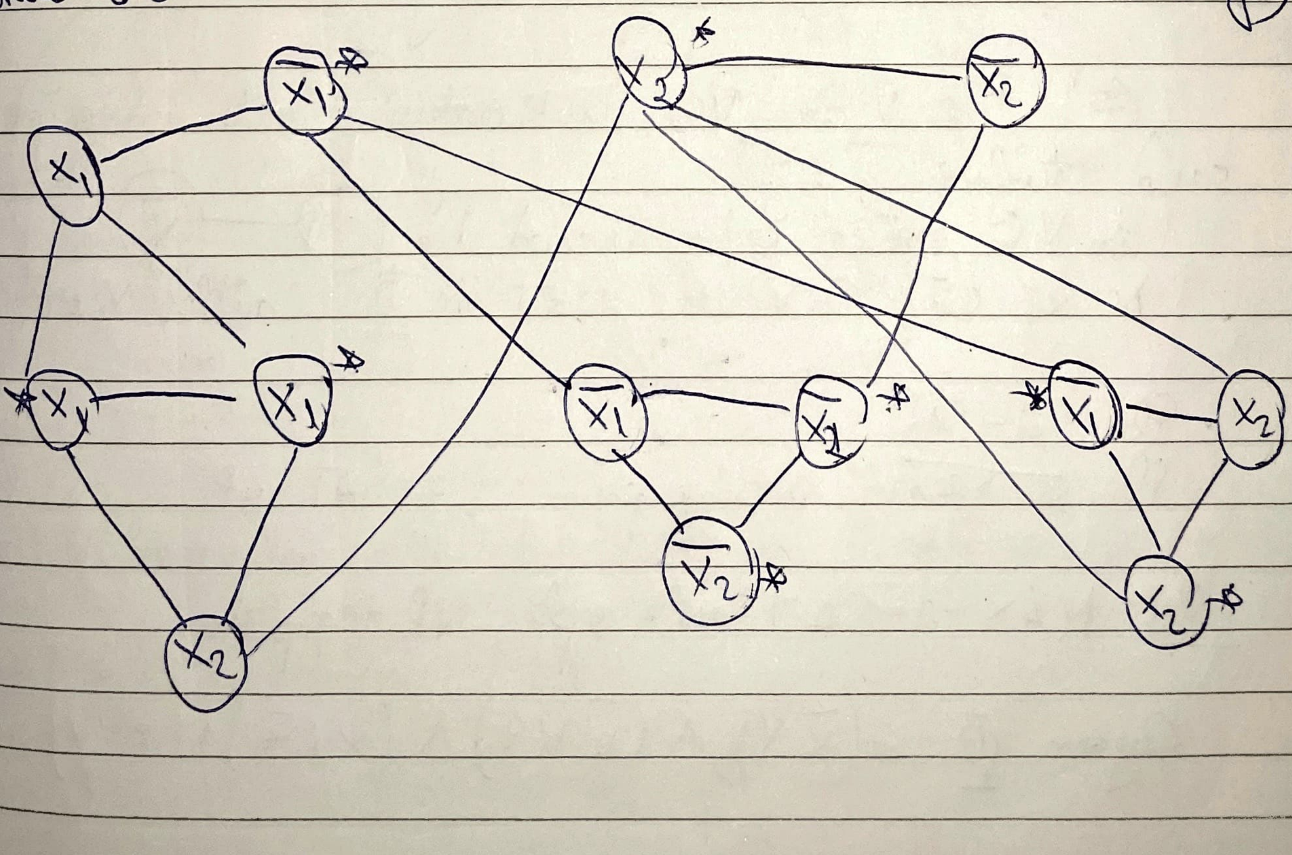
Dem: 3SAT $\leq_P$ Vertex Cover (3-SAT se reduce în timp polin. la VC)

alegem $(x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2) = \Phi$

1) Pt. fiecare variabilă din formula Φ construim 2 noduri $x_i, \overline{x_i}$ legate printr-o muchie

Se dă o formulă Φ cu n variabile și m clauze. Pornind de la Φ construim (G, K)

2) Avem 2 variabile în Φ deci n noduri?. Pt. fiecare clauză $c_j = (x_{j1} \vee x_{j2} \vee x_{j3})$ constr. nodurile x_{j1}, x_{j2}, x_{j3} conectate între ele (triunghi)



3. Conectăm orice 2 noduri v_1, v_2 care au ac. etichetă, iar v_1 coresp. unei variabile și v_2 face parte dintr-un triunghi coresp. unei clauze

Alegem $K = n + 2m$

• Reducția : Vrem să dem. că Φ este satisfiabilă $\Leftrightarrow$ G are un VC cu cel mult $K = n + 2m$ noduri

„$\Rightarrow$" Φ este satisfiabilă dc. x_i este true pun în VC:
1) nodul coresp. din perechea x_i — $\overline{x_i}$ (cel true)
2) din clauza C_j pun în VC cele 2 noduri din triunghi care nu satisfac clauza

$\Rightarrow$ avem $V' = \{$* din desen$\}$

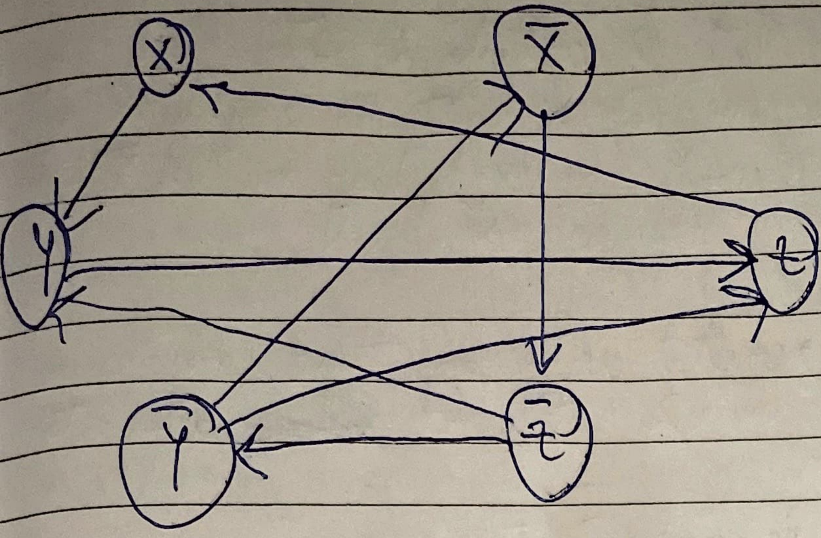
De ce e un VC valid?
- muchiile de sus sunt acoperite mereu (definiție)
- muchiile din triunghi sunt acoperite
- un nod neales de sus e F deci este conectat la un nod care nu satisface triunghiul

„$\Leftarrow$" Fie V' un VC cu k noduri. Constr. un assignment care satisface Φ astfel:
în VC avem cel puțin 1 nod din x_i — $\overline{x_i}$
Dc. x_i este în V' at. $x_i = T$ în Φ, altfel $x_i = F$

• Pb. 2-SAT
Pb. se rezolvă în timp polin., 2-SAT $\in P$

Obs: MAX-2-SAT este o pb. NP-complete

Alegem $\Phi = (\overline{x} \vee y) \wedge (\overline{y} \vee z) \wedge (x \vee \overline{z}) \wedge (z \vee y)$ 2-SAT

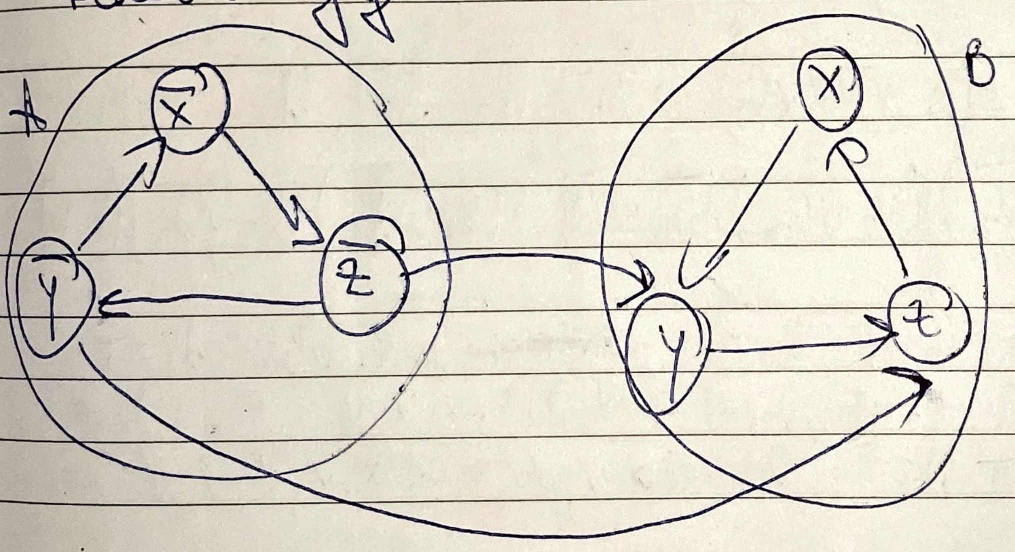


Def: $\neg a \vee b$ scriem: $a \to b$ (implică)
$\neg b \to \neg a$

Scriem implicații pt. clauze și trasăm arce în graf

Obs: dc. $x \in T \Rightarrow y \in T$
dc. $y \in T \Rightarrow z \in T$ ...

Redesenăm graful:



* Deci nu putem alege $\bar{x} = T$, deoarece $\bar{x} \to \bar{z} \to y \to z \to x = T$
$\Rightarrow$ ab

* A și B tare conexe $(A \to B)$

• Dându-se o formulă Φ instanță 2-SAT:

- Pt. fiecare variabilă x_i construim 2 noduri $x_i, \bar{x}_i$
- Pt. fiecare clauză $(x_i \vee x_j)$ adaug muchiile $\bar{x}_i \to x_j$ și $\bar{x}_j \to x_i$
- Dc. graful rezultat are o comp. tare conexă care conține $x, \bar{x} \Rightarrow \Phi$ nu este satisfiabil, altfel este

• Împărțim graful în componente tare conexe:

$(A) \to (B)$ noul graf. Sortăm componentele în ordine topologică (A, B) și procesăm nodurile în ordinea inversă (începem cu B) și asignăm variabilele coresp. cu T dc. nu au asignare

• Pb. MAX 2-SAT este NP-complete

Dem: reducere la 3-SAT

3-SAT = Φ cu m clauze $\Rightarrow \Phi$ 2-SAT cu $10m$ clauze, $k = 7m$ adică luăm fiecare clauză și o transform în 10 clauze

ex: $C_i = (l_1 \vee l_2 \vee l_3) \Rightarrow (l_1^*), (l_2^*), (l_3^*), (d_i)$ → variabilă

$(\bar{l}_1 \vee \bar{l}_2)(\bar{l}_2 \vee \bar{l}_3)(\bar{l}_3 \vee \bar{l}_1)(l_1 \vee \bar{d}_i)(l_2 \vee \bar{d}_i)(l_3 \vee \bar{d}_i)$

Pt. $l_1 = F, l_2 = F, l_3 = F$ ~~→~~ nu e satis. $d_i = T$

Pt. $l_1 = T, l_2 = F, l_3 = F \Rightarrow$ ✓, ✗, ✗, ✗, ✓, ✓, ✓, ✓, ✓ ✓ $d_i = F$

Pt. $l_1 = T, l_2 = T, l_3 = F \Rightarrow$ ✓, ✓, ✗, ✗, ✗, ✓, ✓, ✓, ✓ $k = 7m$

• Dându-se o formulă 3-SAT Φ cu m clauze, am construit o formulă 2-SAT Φ cu $10m$ clauze a.î. Φ este satisf. $\Leftrightarrow \Phi$ are un assignment care satisf. $7m$ clauze

Prog. alg. ef.
Cursul 4

1. Cum identif. ~~pb~~ pb. NP-complete
   (până acum)
   pb. de decizii (DA/NU)
   } primele 3 cursuri

2. Probleme de optimizare → min
   → max

La cursurile următoare → alg. de aproximare

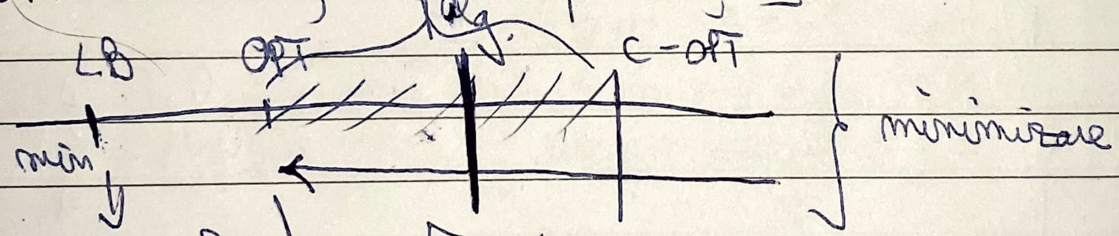
Un alg. A este o c-aproximare pt. pb. P dacă pe orice instanță I a pb. P avem:

! $A(I) \le c \cdot OPT(I)$, dc. P este o pb. de minimizare

$A(I) \ge \dfrac{OPT(I)}{c}$, dc. P este o pb. de maximizare

$A(I)$ = val. soluției returnată de algoritm pe instanța I

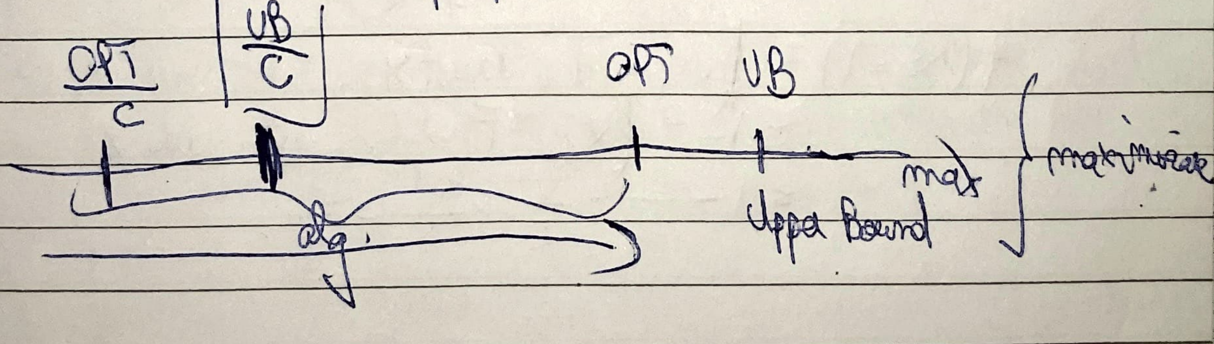
OPT = val. soluției optime pe instanța I



Lower Bound

$\boxed{c \cdot LB}$

calculabil în timp polinomial



MAX 2-SAT

Upper Bound al soluției optime: m = nr. de clauze
Căutăm un alg. de 2-aproximare, adică vrem un alg. care satisface cel puțin $\frac{m}{2}$ clauze

Alegem assignmentul care satisface cele mai multe clauze dintre $x_i = T$ și $x_i = F$, $\forall i$

C_1: mult. clauzelor satisfacute de $x_i = T$ $\forall i$
C_2: ——— " ——— $x_i = F$ $\forall i$

$C = C_1 \cup C_2$ = mult. totală de clauze

$|C_1| + |C_2| = |C| \Rightarrow \max\{|C_1| \cup |C_2|\} \geq \frac{|C|}{2}$

$\frac{7}{8}$ → aprox. pt. MAX-3-SAT

UB = nr. de clauze
Alegem $x_i = T$ cu prob: $\boxed{\frac{1}{2}}$

Alegem $x_i = F$ cu prob: $\boxed{\frac{1}{2}}$

$E = [\#\text{ clauze nesatis}] = \sum_{i=1}^{m} E[Y_i] = \sum_{i=1}^{m} P(Y_i = 1) = \boxed{\frac{7m}{8}}$

medie/expectation

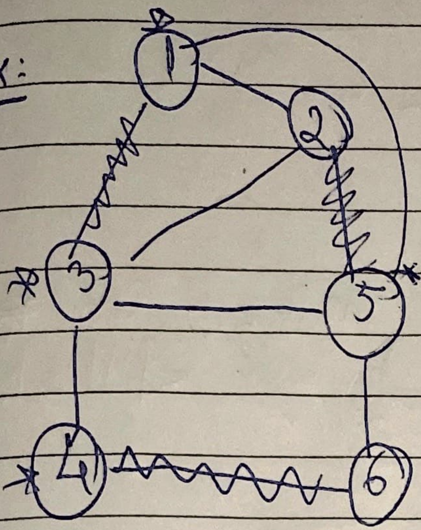
$Y_i = \begin{cases} 1 & \text{dc. } C_i \text{ este satisfiable} \\ 0 & \text{altfel} \end{cases}$

$P(Y_i = 1) = P(x_{i_1} = T \text{ sau } x_{i_2} = T \text{ sau } x_{i_3} = T)$
$= 1 - P(x_{i_1} = F \text{ și } x_{i_2} = F \text{ sau } x_{i_3} = F)$
$= 1 - \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{7}{8}$

Def: Vertex Cover

Se dă un graf neorientat $G=(V,E)$. Să se găsească o mț. $V' \subseteq V$ de cardinalitate minimă a.î. $\forall (a,b) \in E$, avem $a \in V'$ sau $b \in V'$ sau $(a,b) \in V'$

EX:



~~~ → cuplaje

$V' = \{1, 3, 4, 5\}$

2-aproximare

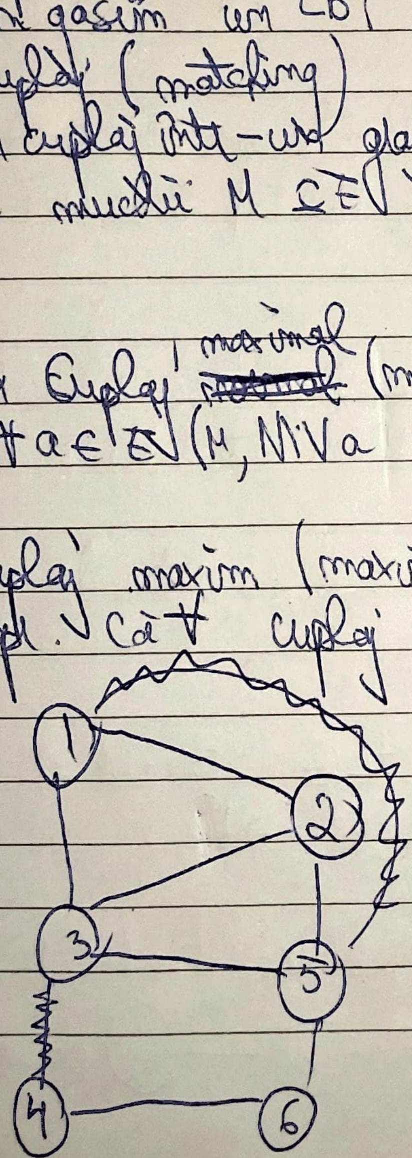
Cum găsim un LB?

Def: Cuplaj (matching)

Un cuplaj într-un graf neorientat $G=(V,E)$ este o mț. de muchii $M \subseteq E$ a.î. oricare 2 muchii nu se intersectează.

Def: Cuplaj maximal (maximum matching) este un cuplaj M a.î. $\forall a \in E \setminus M$, $M \cup a$ NU e cuplaj.

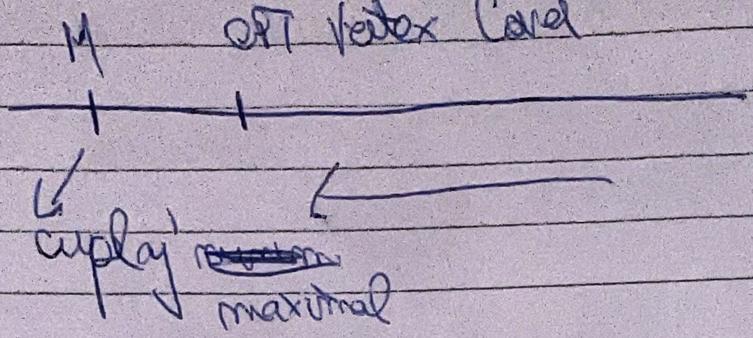
Def: Cuplaj maxim (maximum matching) este un cuplaj M cu propr. că $\forall$ cuplaj M' avem: $|M| \geq |M'|$



~~~ → maximal, dar NU maxim

Asta e pb. P

Teoremă: Dându-se un graf neorientat $G = (V, E)$, fie V' un vertex cover minim și M - cuplaj maximal. Atunci:
$|M| \le |V'|$



Dem: $\forall (a,b) \in M$, avem $a \in V'$ sau $b \in V'$. Cum ~~(a,b)~~ $\forall (a,b), (c,d) \in M$, $\{a,b\} \cap \{c,d\} = \emptyset \Rightarrow |M| \le |V'|$

- Alg. de aproximare:
  1. Găsesc 1 cuplaj maximal M
  2. Construiesc un vertex cover alg. care conține capetele tuturor muchiilor din M

Teoremă: $|ALG| \le 2 \cdot OPT$ și V' este un vertex cover

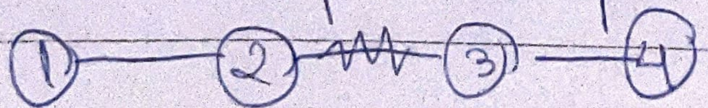
$$\left.\begin{array}{l} |M| \le OPT \\ |ALG| = 2 \cdot |M| \end{array}\right\} \Rightarrow |ALG| \le 2 \cdot OPT$$

Pp. prin RA că $\exists (a,b) \in E$ a.î. $a, b \notin ALG \Rightarrow$ $M \cup (a,b)$ este cuplaj $\Rightarrow M$ nu este maximal $\Rightarrow$ Ø

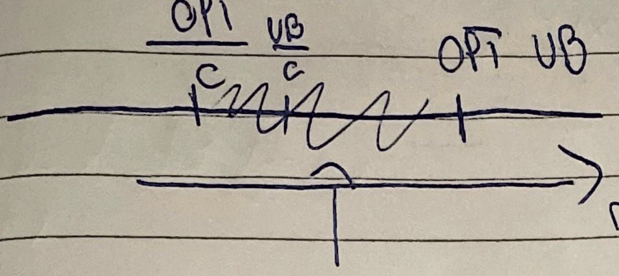
! Dc. M este maxim, am avea un factor de aproximare mai bun? Nu.

ALG → 2 noduri
OPT → 1

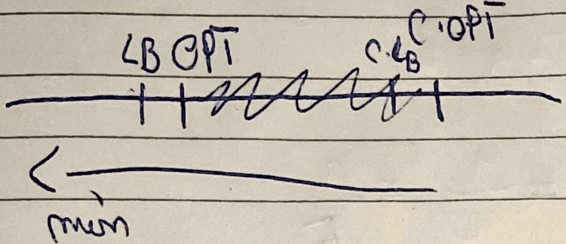
∃ un alg de 1,999 aproximare pt. VC? → pb. deschisă


⇐ sol. optimă

Seminarul 3
Prog. alg. ef.



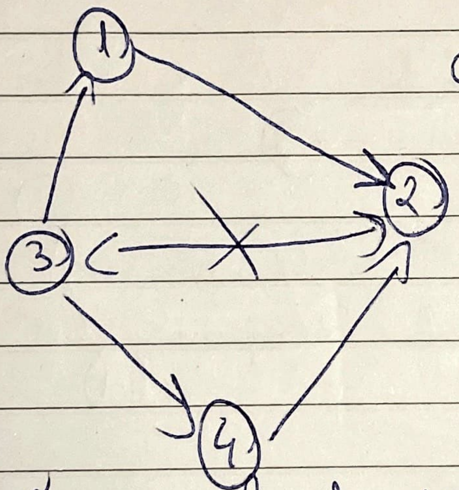
$\Rightarrow$ pb. de maximizare



$\Rightarrow$ pb. de minimizare

Pb: Se dă un graf orientat. Să se det. nr. maxim de muchii care poate fi păstrat a.î. graful să fie aciclic.

Exemplu:



OPT = 4

Vrem să găsim un alg. de aprox. pt pb. î

Indiciu: Vrem să găsim un alg. de 2-aprox.

($\Leftrightarrow$ găsiți un alg. care ret. jum. din nr. muchii)

Fie A alg. căutat

A este 2-aprox. pt P (pb. dată) $\Leftrightarrow \forall\, \overline{I}$ (instanță) avem:

$$\begin{cases} A(\overline{I}) \le c \cdot OPT(\overline{I}), & \text{dc. P e de minimizare} \\ A(\overline{I}) \ge \dfrac{OPT(\overline{I})}{c}, & \text{dc. P e de maximizare} \end{cases}$$

Cum P e de maximizare $\Rightarrow A(\overline{I}) \ge \dfrac{OPT(\overline{I})}{c}$

$\Rightarrow$ UB = nr. de muchii