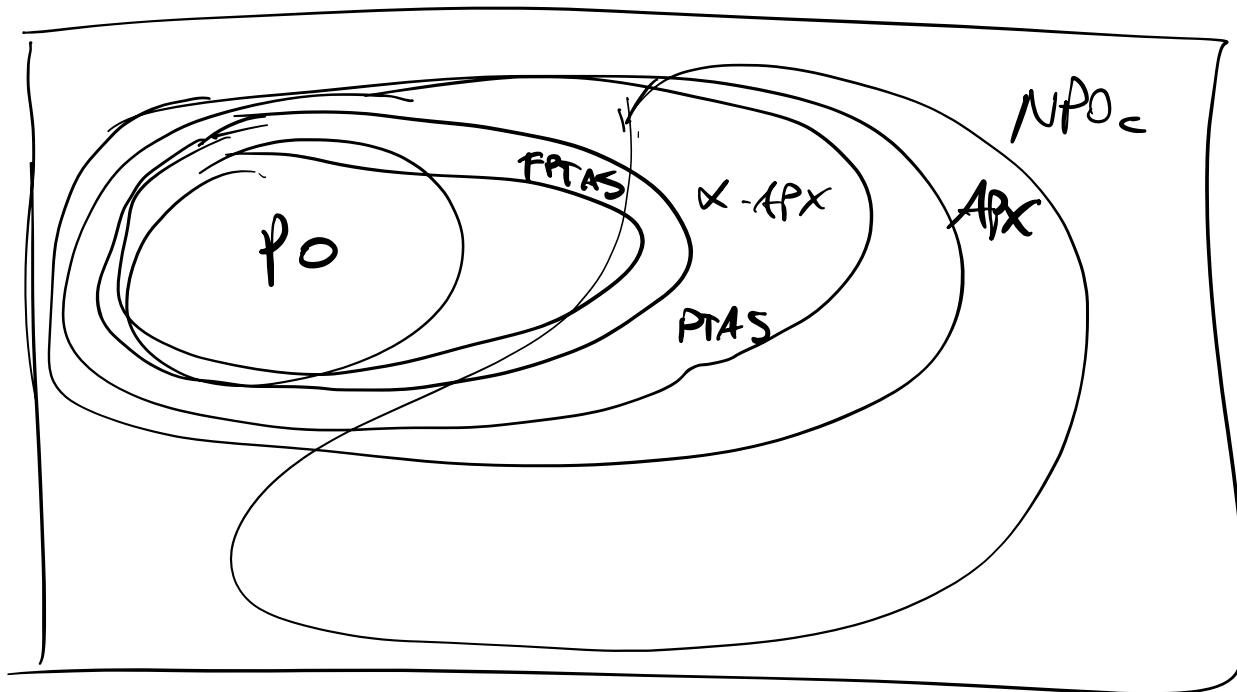


PROBL.
OPTIM.



$$APX = \bigcup_{\alpha \geq 1} \alpha\text{-APX}$$

PTAS = Polynomial-Time Approx. Scheme
 π $(x \in I_\pi, \epsilon \geq 1)$

PROBLEMA DEL MAX MATCHING

MAX B. MATCHING

INPUT: Grato non
orientato $G = (V, E)$
bipartito

SOL. AMMISSIBILI:

$$M \subseteq E$$

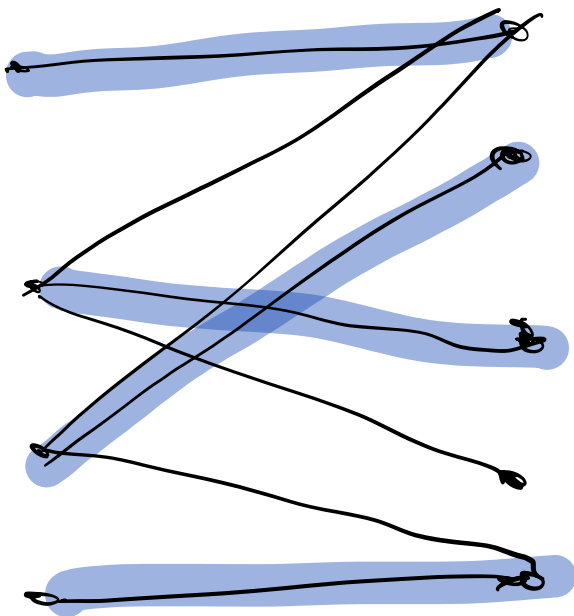
t.c. $\forall x \in V$
al massimo un lab. di M
è incidente su x

FUNZ. OBIETTIVO:

$$|M|$$

TIPO: MAX

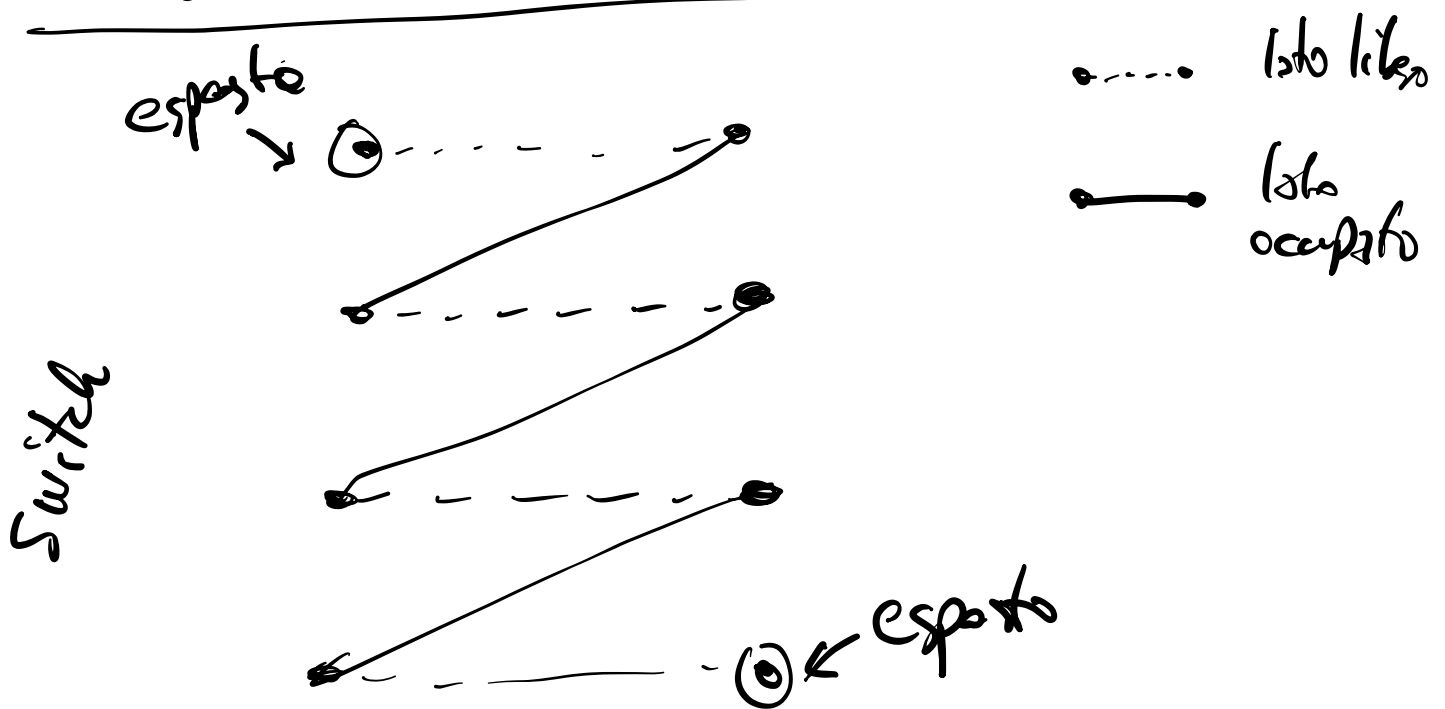
Teorema: $(B.) \text{Max Matching} \in PO$



$\text{Lati} \begin{cases} \text{liberi} \\ \text{occupati} \end{cases}$

Vertice = esposto se
 su di esso
 incidono solo
 lati liberi

CAMMINO AUMENTANTE



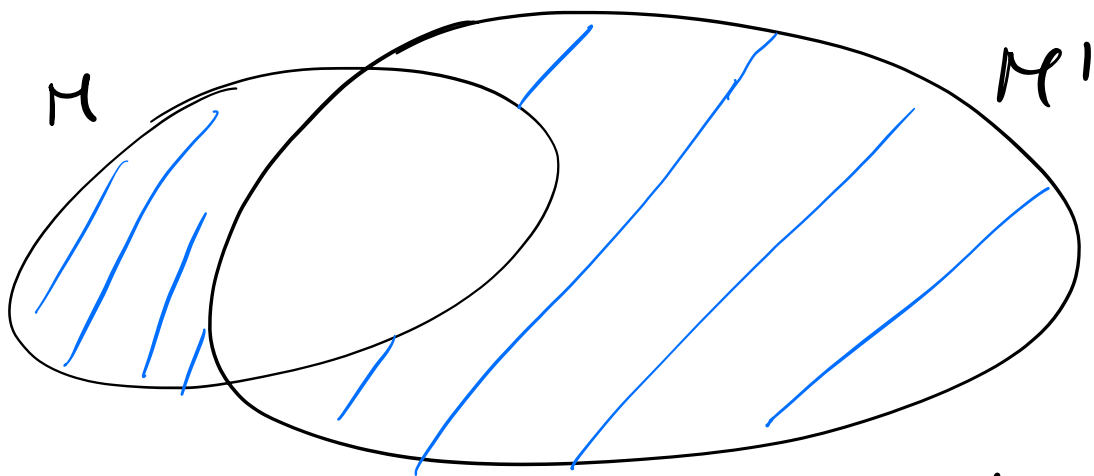
Proprietà ①: Se per M esiste
 un cammino aumentante,
 allora M non è massimo

Proprietà ②: Se M non è massimo,

Esiste un cammino semplice

Dim.: Sia M' un astico t.c.
 $|M'| > |M|$.

$$X = M \Delta M' = \\ = (M \setminus M') \cup (M' \setminus M)$$



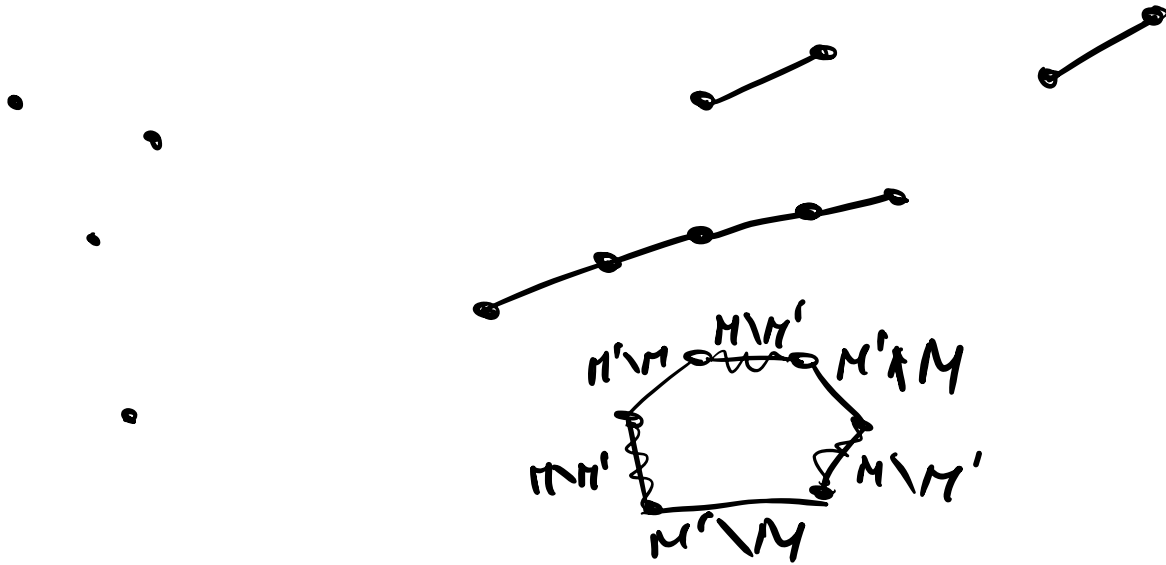
Oss. 1: X contiene più elementi
di M' che elementi di M

Oss. 2: Ogni vertice ha al
meno due lati di X
incidenti.



$\Rightarrow$ Oss. 3: Nel graf astratto di X

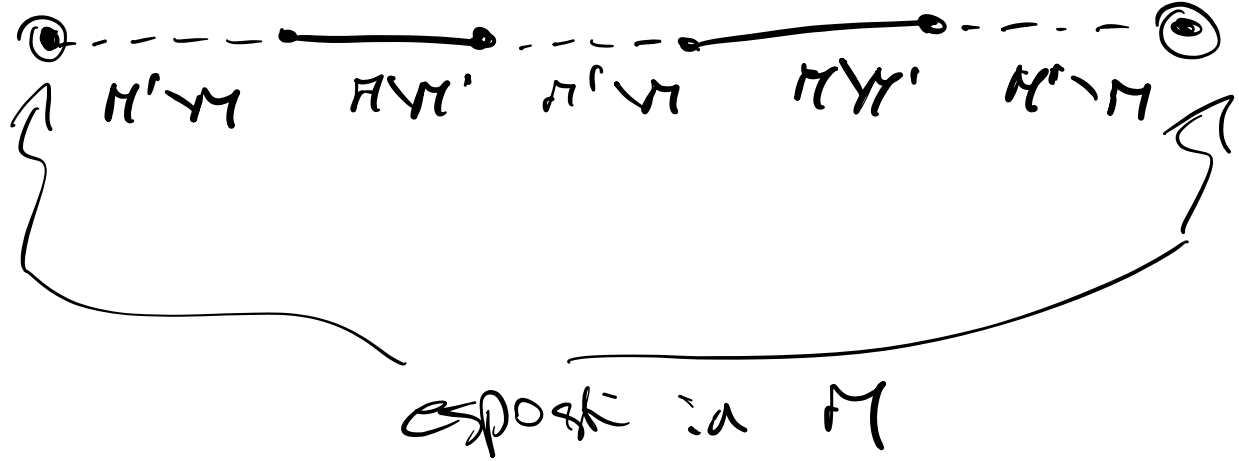
tutti i vertici hanno grado
0, 1, 2.



Oss. 4: Non possono essere solo
cicli, perché ogni ciclo
contiene lo stesso
numero di $l_i \in M \setminus M'$
e $l_i \in M' \setminus M$

$\Rightarrow$ Oss. 5: Ci deve essere
almeno un cammino
con più l_i di $M' \setminus M$
che l_i di $M \setminus M'$.

U D U D U D



$\Rightarrow$ CAMMINO AUMENTANTE
PER M . □

ALGORITMO PER B. MAX MATCHING

INPUT $\Leftarrow G = (V_1 \cup V_2, E)$

$M \Leftarrow \emptyset$

forever

$\pi \leftarrow \text{FIND AUGMENTING}(M)$

if $\pi = \perp$

output M

stop

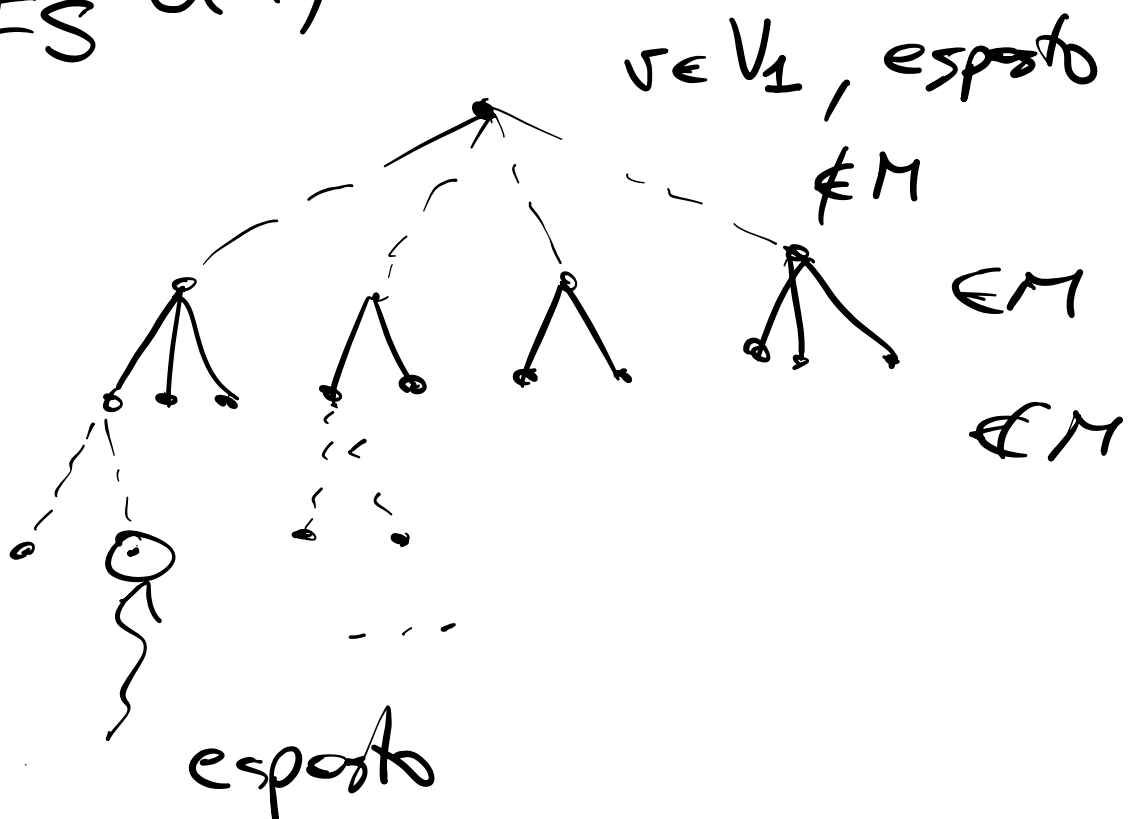
else

$M \leftarrow \text{switch}(M, \pi)$

$O(n)$

FIND AUGMENTING (M)

BFS $O(m)$



PROBLEMA DI DECISIONE

PROBLEMA : PERFECT MATCHING

- Data un grfo, determinare se esiste un matching che coinvolge tutti i vertici

Corollario : PERFECT MATCHING $\in P$

PROBLEMA DI LOAD BALANCING

INPUT: $m \in \mathbb{N}^+$ macchine
 $n \in \mathbb{N}^+$ task
 $t_0, t_1, \dots, t_{n-1} \in \mathbb{N}^+$
 t_i è la durata dell' i -esimo task

SOL. AMMISSIBILE:

$$\underbrace{A_0, A_1, \dots, A_{m-1}}_{\text{partizione}} \subseteq n$$

FUN. OBIETT: $L_i = \sum_{j \in A_i} t_j$

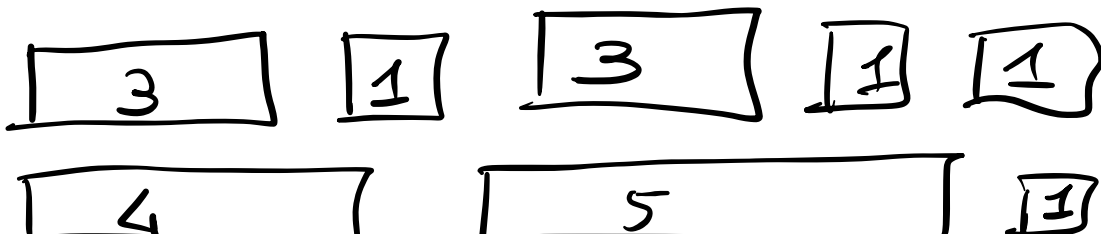
$$L = \max_{i \in m} L_i$$

span della soluzione

TIPO: MIN

ESEMPIO

$$m = 3$$



MACCHINA

0	5 1	$L_0 = 6$	
1	3 3 1	$L_1 = 7$	$L = 7$
2	1 4 1	$L_2 = 6$	

Theorem: LOAD BALANCING $\in \text{NPOc}$

GREEDY LOAD BALANCING

$$A_i \leftarrow \emptyset \quad \forall i \in m$$

$$L_i \leftarrow 0 \quad \forall i \in m$$

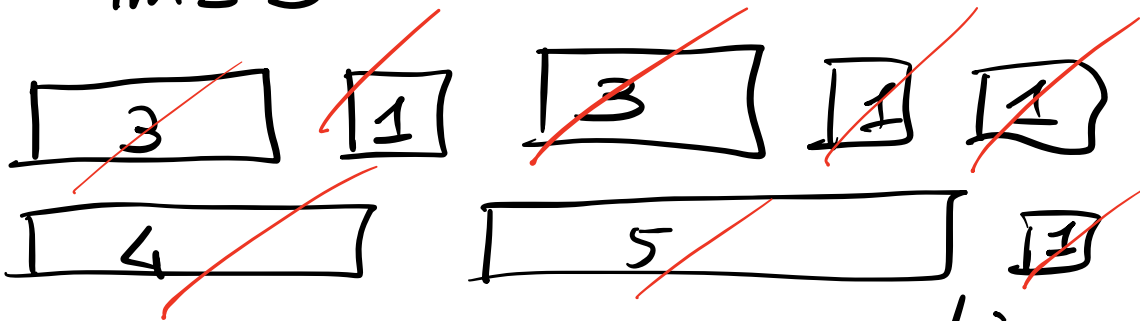
$$\text{for } j = 0, 1, 2, \dots, n-1$$

$\bar{i} \leftarrow \underset{i \in m}{\text{argmin}} L_i$

$$A_{\bar{i}} \leftarrow A_{\bar{i}} \cup \{j\}$$

$$L_{\bar{i}} \leftarrow L_{\bar{i}} + t_j$$

$$m = 3$$



0



1



2



L_i

7

8

4

$L = 8$

Theorem:

GREEDY LOAD BALANCING

2-approximate