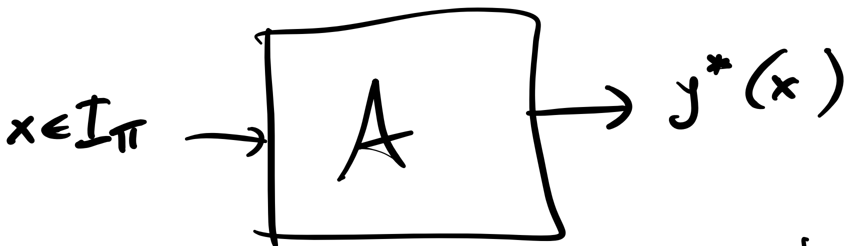
NPO - COMPLETI

NPOC è la classe dei problemi di ottimizzazione Π t.c. $\Pi \in NPO$ e $\hat{\Pi} \in NPC$

P.es. MaxSat $\in NPOC$

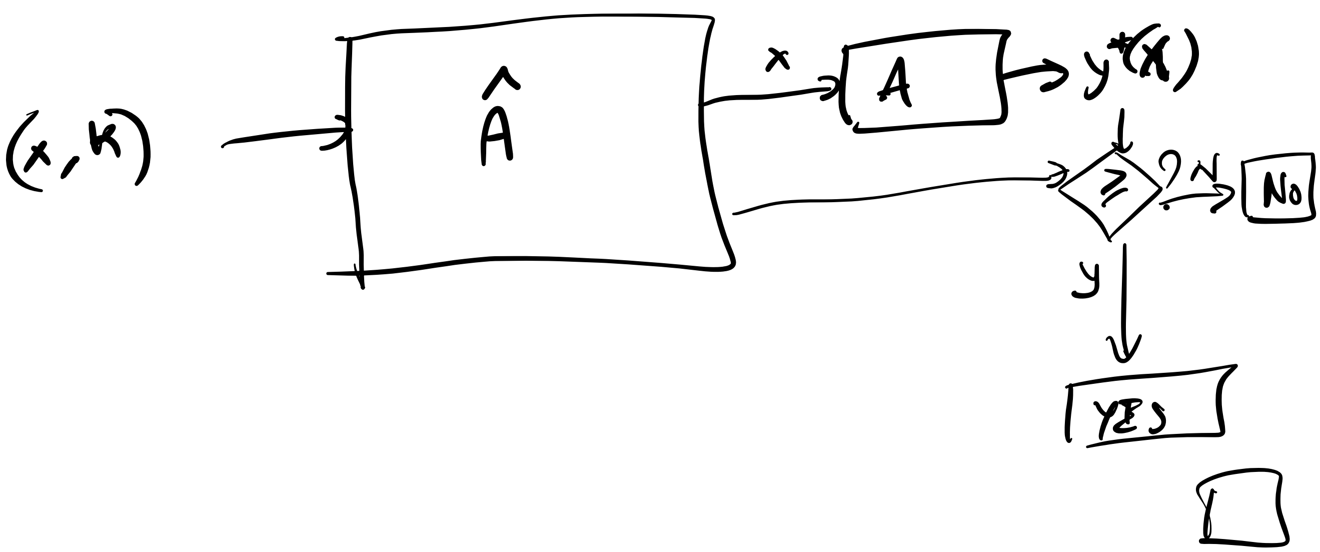
__Teorema__: Se Π è NPOC allora $\Pi \notin PO$ (a meno che $P = NP$)

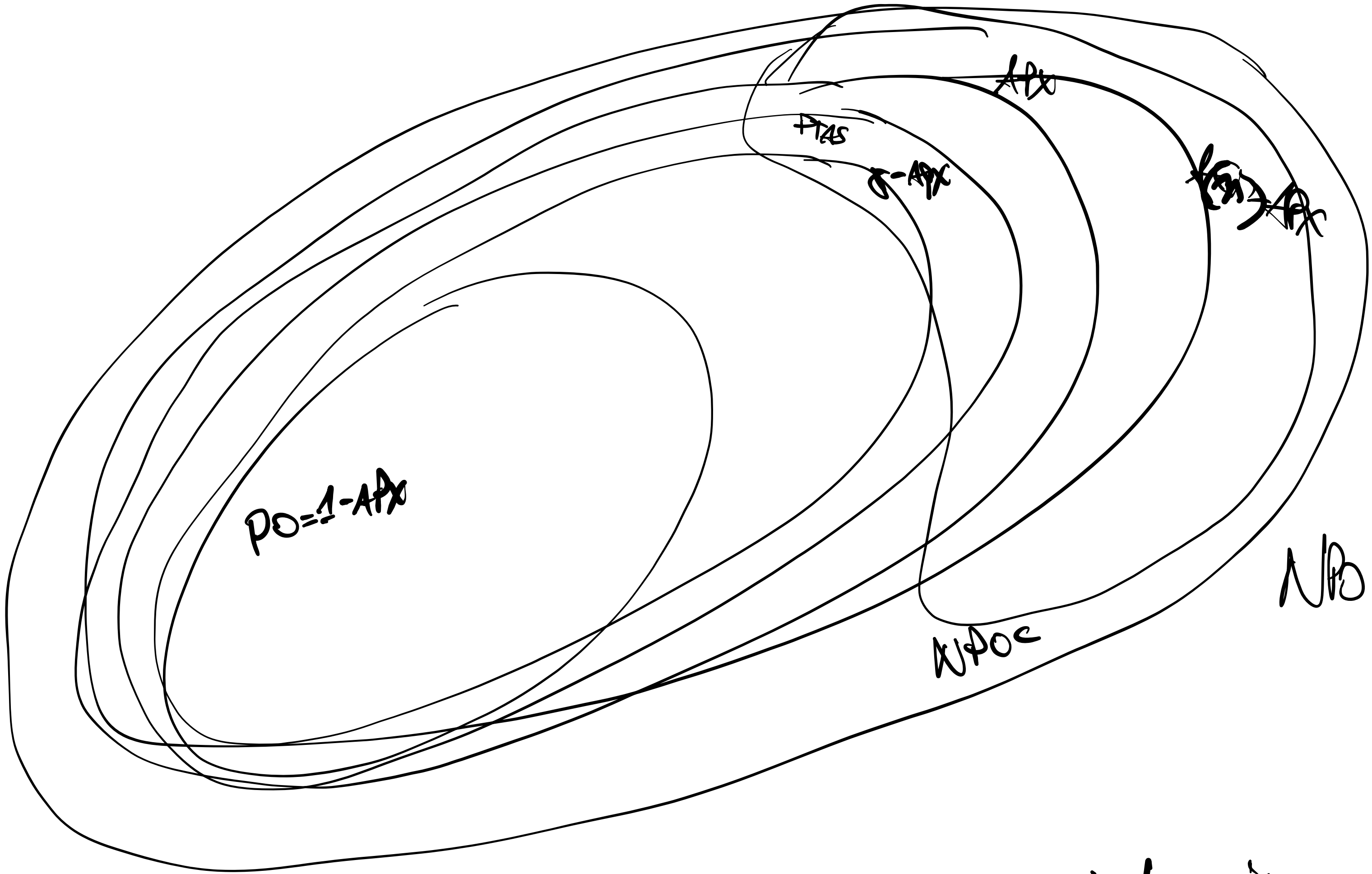
__Dim__: Per assurdo $\Pi \in PO$ $(tipo_\Pi = Max)$



$\hat{\Pi}$ $\quad (x, k) \in YES_{\hat{\Pi}}$ __sse__ $obj_\Pi(x, y^*(x)) \geq k$

$\Pi \in NPOC \Rightarrow \hat{\Pi} \in NPC$





$\forall \gamma \geq 1$ γ-APX = Problemi risolv. in tempo polinomiale con tasso $\leq \gamma$

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

PTAS = Polynomial Time Approximation Scheme $\forall \varepsilon > 1$

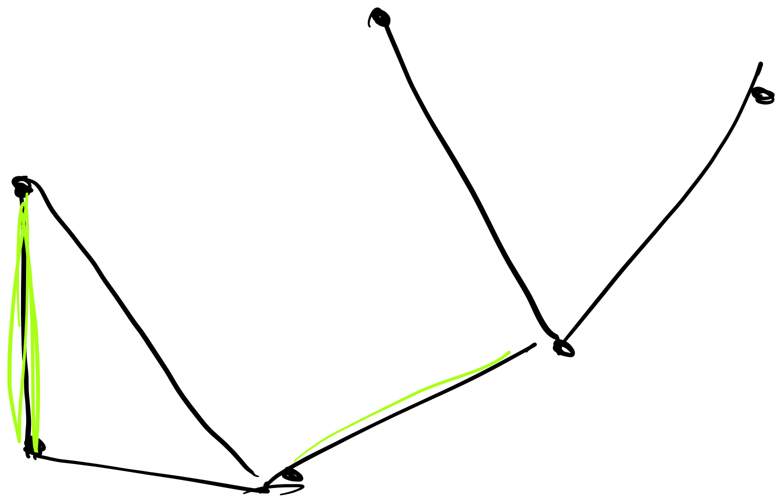
$(\underline{x}, \varepsilon) \longrightarrow$ soluzione con tasso $\leq \varepsilon$

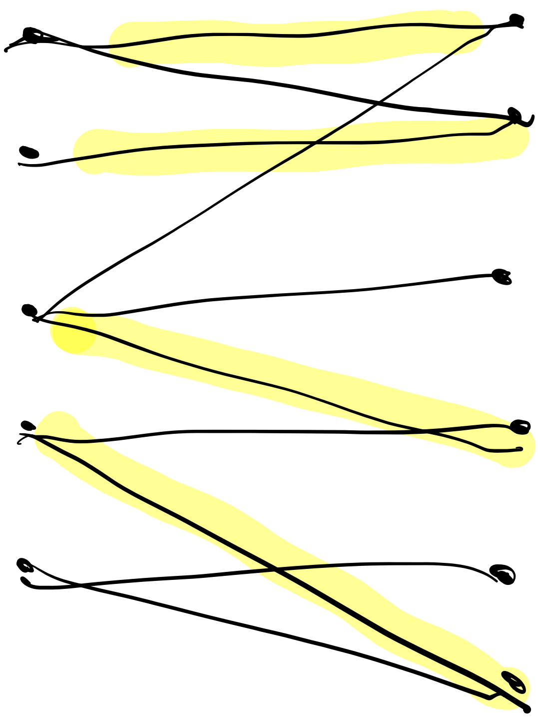
$\underline{x}$ ↓ Polinomiale

FPTAS = Fully PTAS

(x, ε) Polinomiale anche in ε

MAX MATCHING SU GRAFI BIPARTITI



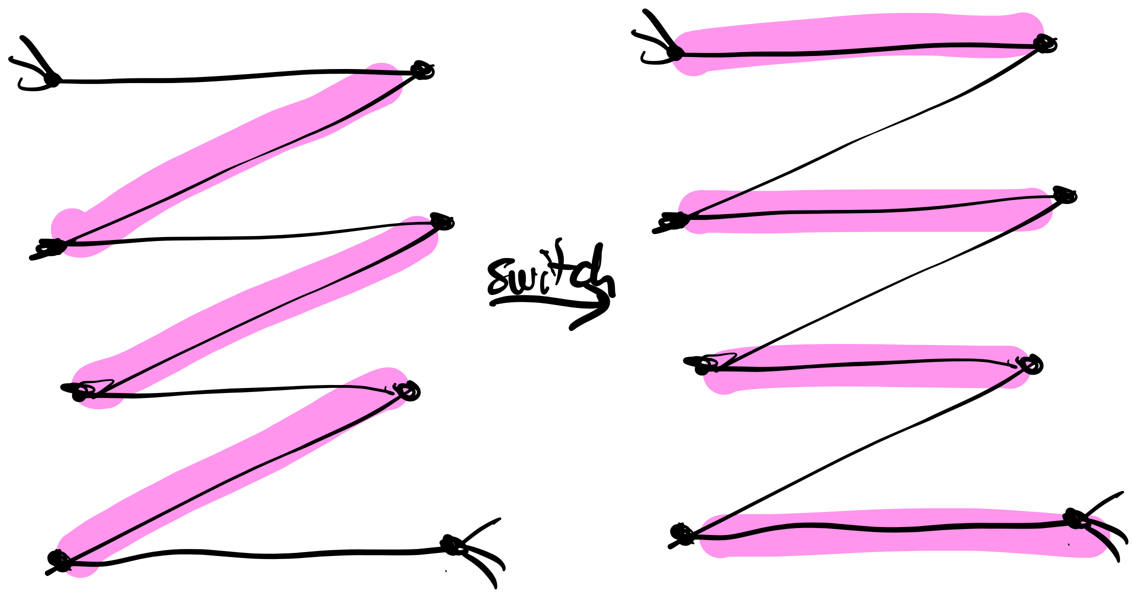


CAMMINO AUMENTANTE

LATI DEL GRAFO
- OCCUPATI $\in M$
- LIBERI $\notin M$

VERTICI ESPOSTI → su cui incidono solo lati liberi

Cammino aumentante:
cammino che alterna lati liberi e occupati e inizia e termina su un vertice esposto

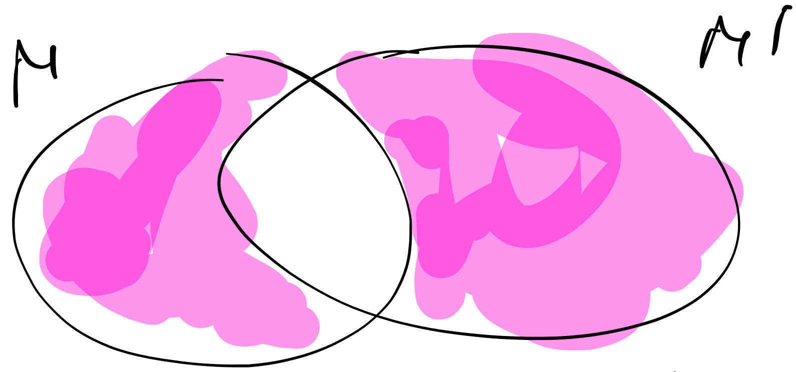


Lemma 1: Se un matching M ha un cammino aumentante, non è massimo.

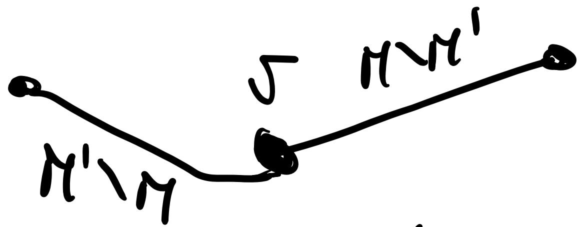
Lemma 2: Se M non è massimo allora esiste un cammino aumentante.

Dim: Se M non massimo, quindi esiste un matching M' $|M'| > |M|$.

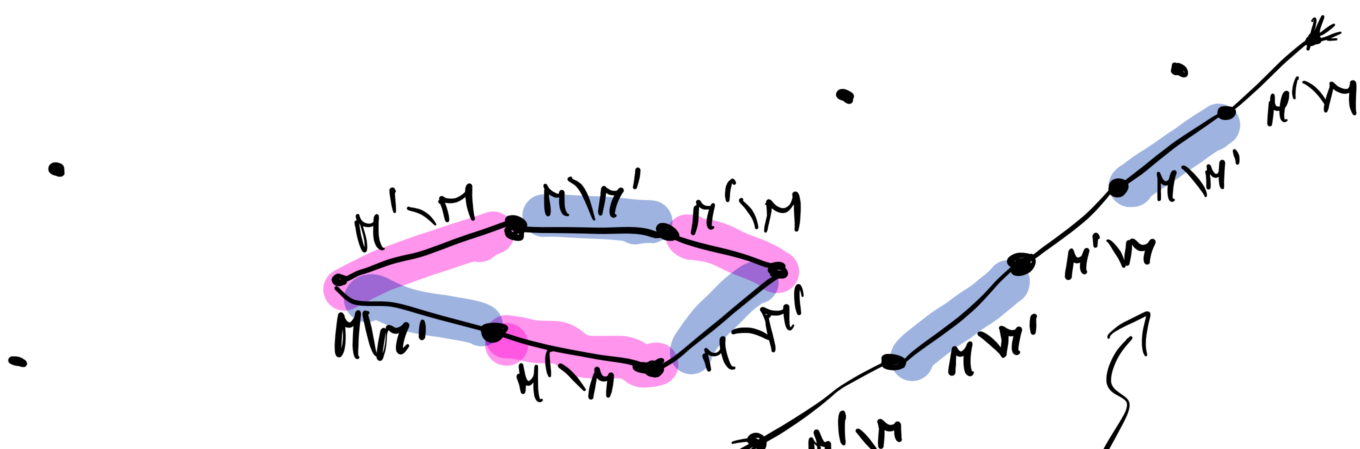
$$X = M \Delta M' = (M \setminus M') \cup (M' \setminus M)$$



1) Su ogni vertice incidono al massimo 2 lati di X.



2) $\Rightarrow$ Tutti i vertici (rispetto a X) hanno grado 0, 1, 2



∎

PER M

ALGORITMO

INPUT: $G = (V, E)$

$M \leftarrow \emptyset$

while True :

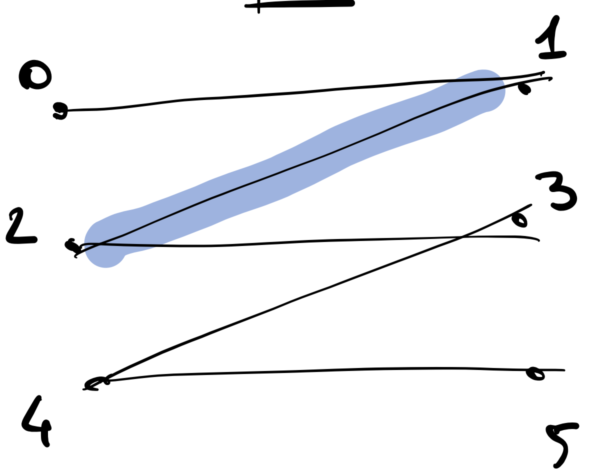
  π = FindAugmenting(G, M)

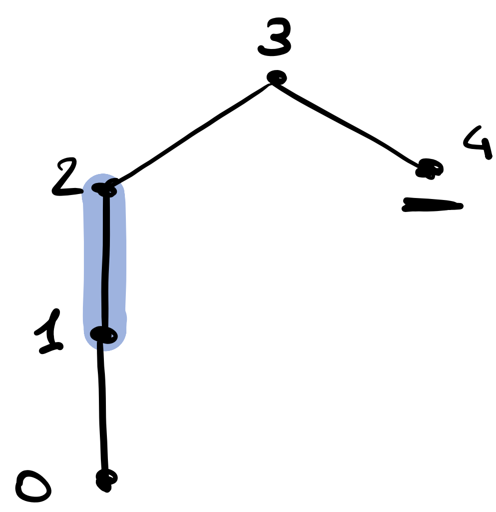
  if $\pi = \perp$

    return M

  else

    $M \leftarrow$ switch(M, π)





$$\left.\begin{aligned} &\text{BFS} \rightarrow O(m) \\ n\ &\text{BFS} \rightarrow O(n\,m) \end{aligned}\right] \text{FindAugmenting}$$

$$\downarrow$$

$$O(n\,m^2) = O(n^5)$$

VISITE DI GRAFI

(non orientato)

STATI DEI VERTICI:
- sconosciuto (BIANCO)
- FRONTIERA: conosciuto ma non visitato (GRIGIO)
- visitato (NERO)

VISITA GENERICA DA seed

```
for all v ∈ V:
    stato(v) = WHITE
stato(seed) = GREY
F = {seed}
while F ≠ ∅:
    v = extract(F)        ← estrae (e toglie) un elemento da F
    for all w ∈ neighbor(v):
        if stato(w) = WHITE:
            stato(w) = GREY
            F = F ∪ {w}
    visit(v)
    stato(v) = BLACK
```

COROLLARIO: MAX MATCHING $\in$ PO

COROLLARIO: PERFECT MATCHING $\in$ P

$\boxed{LP}$

$$\max \quad c_1 x_1 + \ldots + c_n x_n$$

$x_i \in \mathbb{R}$

$$\begin{cases} a_{11} x_1 + \ldots + a_{1n} x_n \geq / \leq b_1 \\ \vdots \\ a_{m1} x_1 + \ldots + a_{mn} x_n \geq / \leq b_m \end{cases}$$

$LP \in PO$