

PARTE IV: | Aproximación Algoritmos Paralelos

Aproximación

① Recuerdos de Teoría de la Computación

NP

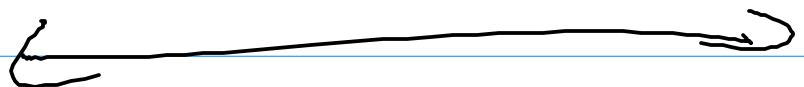
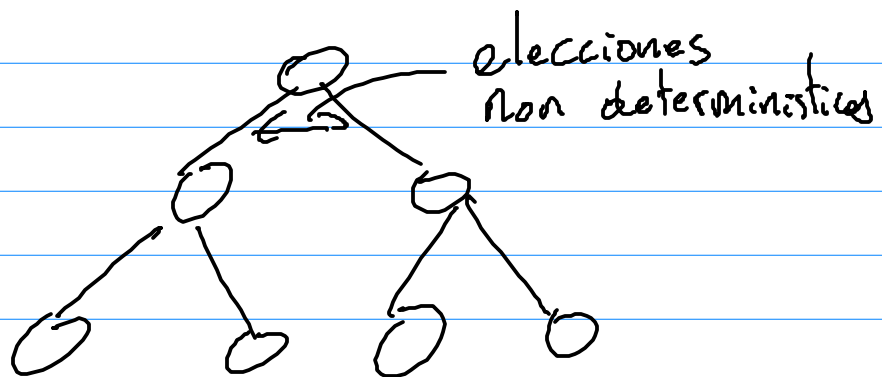
- ? Familia de problemas con ?
soluciones "exponenciales".

- Problemas que se pueden | verificar
certificar
en tiempo polinomial

- $P \subset NP$

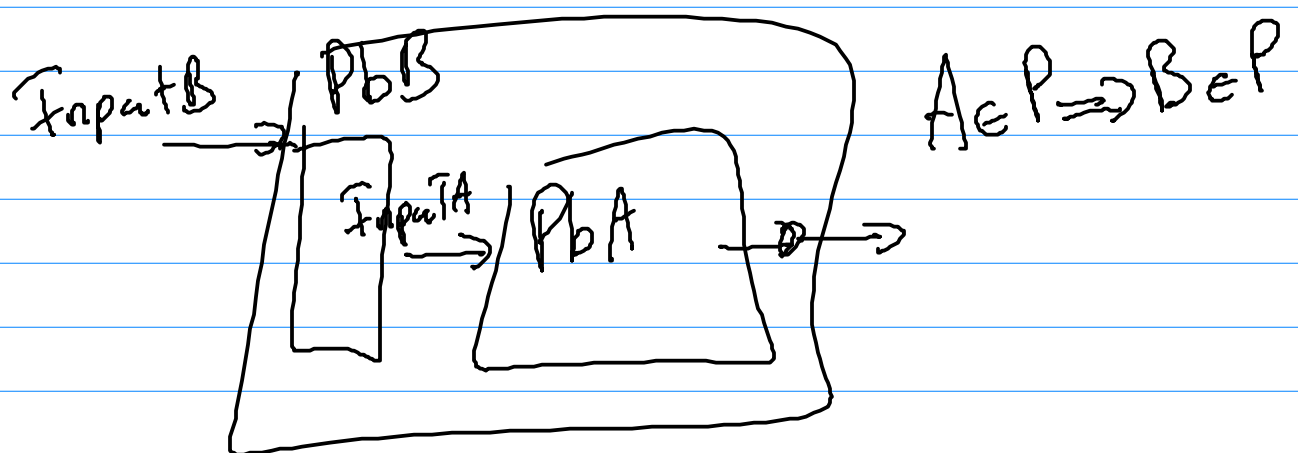
- No se sabe si

$P \neq NP$



NP-difícil $P_b \in NP$

"reducción Polinomial"



$B \in NP_{difícil} \Rightarrow A \in NP_{difícil}$

"A se reduce a B"

$NP_{Completo} \subset NP_{difícil}$

$\forall A \in NP_{Completo}, \forall B \in NP_{difícil}$
B se reduce a A

② Aproximación

- No heurísticas
- Algoritmos
 - polinomiales (en tiempo)
 - para problemas difíciles (en general NP difíciles, pero no necesariamente)

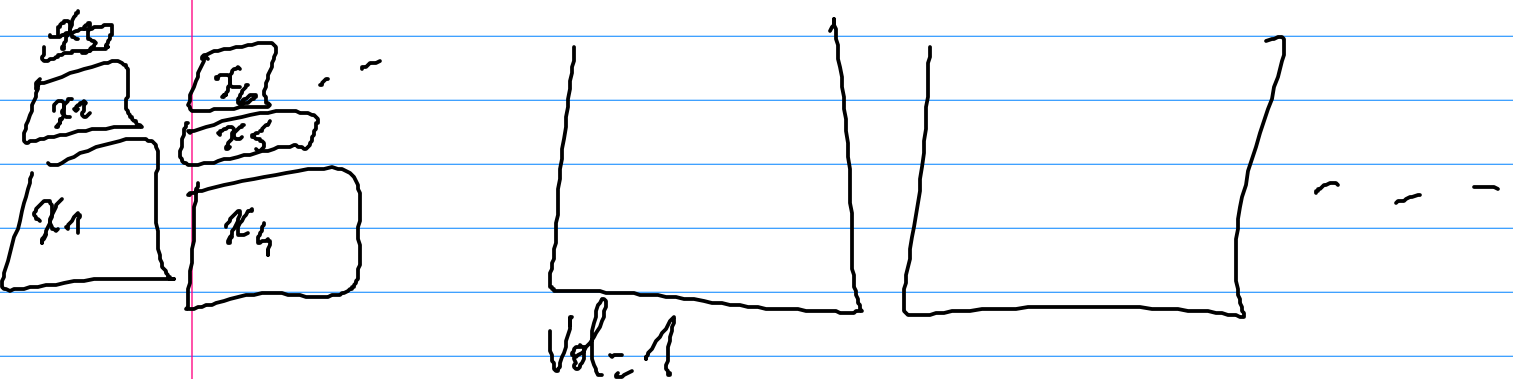
- Como medir la calidad de la aproximación?

$C_A(I)$ = costo de solución del algo A

$OPT(I)$ = costo de la solución óptima
(e.g. en tiempo exponencial)

$$\left\{ \begin{array}{l} C_A(I) \\ \hline OPT(I) \\ OPT(I) - C_A(I) \end{array} \right. \text{ para un Pb de minimización}$$

Caso de Estudio: Bin Packing



Algoritmo Greedy

- tomar cada objeto x_i en orden
- si hay espacio en la ultima caja, agregarlo
- Sino
cerrar la caja
iniciar una nueva

Theorem

Este algoritmo llena el maximo

$\frac{1}{2} \text{ COPT}$ Cajas

Proof Peor caso

