

Aproximación

Un **algoritmo** de una O_f $p(n)$ -aproximación

Si el coste de su output siempre está dentro de un factor de $p(n)$ del óptimo

E.g. 2-aproximación $\rightarrow$ Bin Packing

Σ -aproximación

$\sqrt{n}$ -aproximación

② Otro ejemplo: "Vertex Cover"
Recubrimiento de Vertices.

Input - Grafo $G = (V, E)$

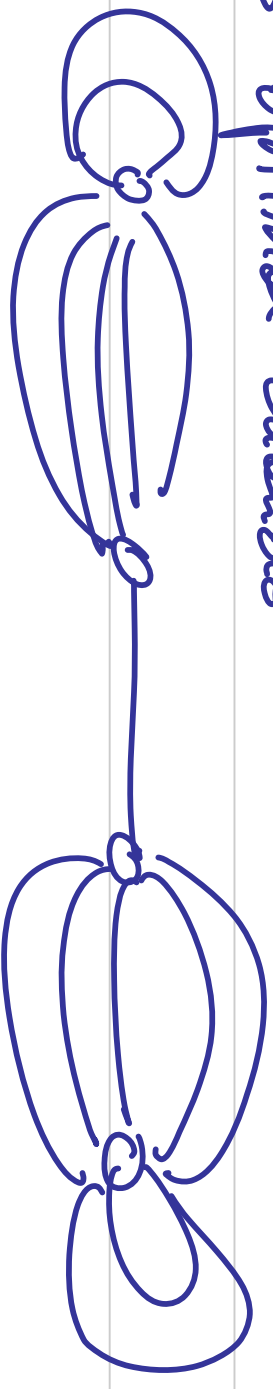
- Verles un **Vertex Cover**
 $Ve = (u, v) \in E \quad u \in V' \quad v \in V'$

- V' es optima si $|V'|$ es minima

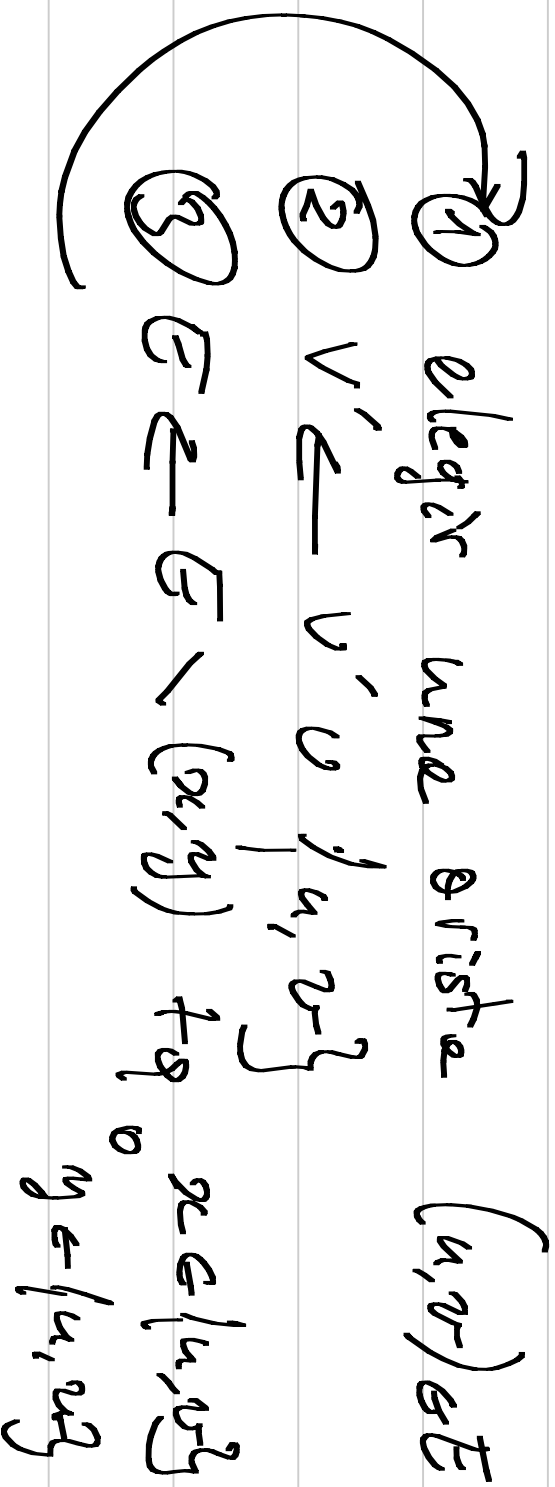
Algoritmo de aproximación

- ① elegir a cada paso el vértice con grado máximo
- ② es polinomial } Removerlo con sus aristas

③ no es optimal cuando



Algoritmo de Aproximación



Esto es una 2-aproximación.

③ Otros Ejemplos

— "Vertex Cover" con pesos 2-approximable

— Vendedor Viajero / Con inequality triangle 2-approximable

Sin

no es 2-approximable.

en tiempo polinomial.

Si $P \neq NP$

Thm Si la función de costo c no satisface la desigualdad triangular, el problema de encontrar **NO** es aproximable en tiempo polinomial (a menos que $P=NP$)

Proof

Supongamos que existe una $p(n)$ aproximación

Para $G=(V, E)$

Construimos $G'=(V, E')$ tal

$$E' = V^2$$

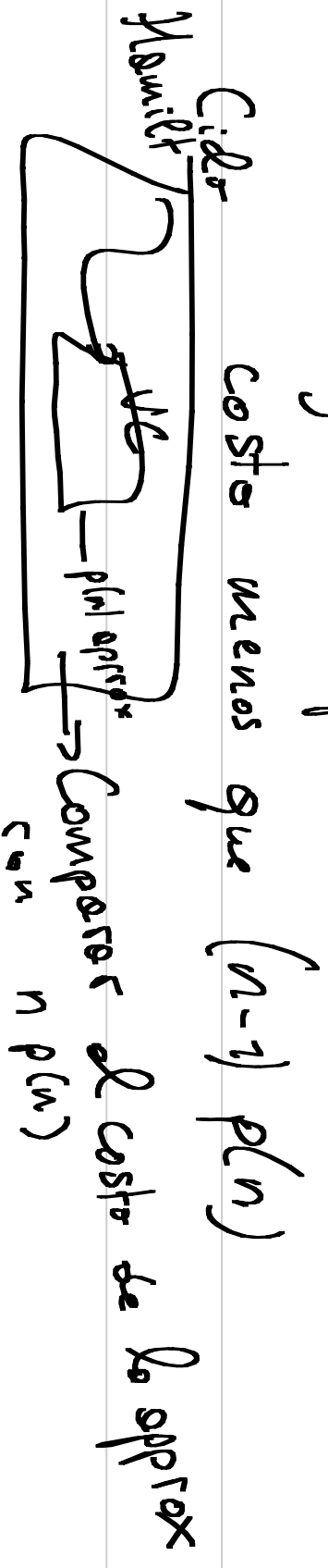
$$c(u, v) = \sum_{i=1}^k \lambda_i (u, v)^i \in E$$

a Si no hay un ciclo Hamiltoniano en E

- todas las soluciones usan al menos una arista no en E . (en $V^2 - E$)
- entonces costo $\geq n p(n)$

* Si hay un ciclo

$\Rightarrow$ hay una aproximación de



Q PTAS y FPTAS

el problema al revés:

Figar la calidad $\hat{\epsilon}$ de la aproximación

Estudiar el tiempo del algoritmo $f(n, \epsilon)$

→ Esquema de aproximación polinomial

→ polinomial en n

Esquema de aproximación completamente polinomial

→ polinomial en n y $\frac{1}{\epsilon}$