Greedy Set Cover

$I \leftarrow \emptyset$

$R \leftarrow U$

while $R \neq \emptyset$

- choose $\hat{i}$ minimizing $\frac{w_i}{|S_i \cap R|}$
- $I \leftarrow I \cup \{\hat{i}\}$
- $R \leftarrow R \setminus S_{\hat{i}}$

$\forall s \in U$

$c_s := \frac{w_i}{|S_i \cap R|}$

Lemma 1:

$$\sum_{s \in U} c_s = W$$

Dim: Ovvio (per la def. c_s).

□

Lemma 2: Per ogni k

$$\sum_{s \in S_k} c_s \leq H(|S_k|) \cdot w_k$$

Dim: Sia

$S_k = \{s_1, s_2, \ldots, s_d\}$ ← numerati in ordine di copertura

Per ogni j, consideriamo

S_j sta per essere coperto.

$$R \supseteq \{S_j, \dots, S_d\}$$

Quindi

$$|S_k \cap R| \geq d - j + 1$$

Quindi

$$c_{S_j} = \frac{w_i}{|S_i \cap R|} \leq \frac{w_k}{|S_k \cap R|} \leq$$

(*) $$\leq \frac{w_k}{d - j + 1}$$

Ma

$$\sum_{S \in S_k} c_S = c_{S_1} + c_{S_2} + \dots + c_{S_d} =$$

$$= \sum_{j=1}^{d} \frac{w_k}{d - j + 1} = w_k \sum_{j=1}^{d} \frac{1}{d - j + 1} =$$

$$= \sum_{j=1} c_{s_j} \leq \sum_{j=1} \frac{w_k}{d-j+1} =$$

(*)

$$= w_k \left(\frac{1}{d} + \frac{1}{d-1} + \dots + \frac{1}{1} \right) =$$

$$= w_k \; H(d) = w_k H(|S_k|)$$

□

Sia $M = \max_i |S_i|$.

Teorema: GREEDY SET COVER è una $H(M)$-approssimazione per SET COVER.

Dim: Sia I^* una soluzione

ottimo e sia

(A) $$w^* = \sum_{k \in I^*} w_k$$

Per il Lemma 2

$$\sum_{s \in S_k} c_s \leq w_k \, H(|S_k|)$$

(B) $$w_k \geq \frac{\sum_{s \in S_k} c_s}{H(|S_k|)} \geq \frac{\sum_{s \in S_k} c_s}{H(M)}$$

I^* è una soluzione ammissibile, quindi

$$\bigcup_{i \in I^*} S_i = U$$

(C) $$\underbrace{\sum_{i \in I^*} \sum_{s \in S_i}}_{\text{include ogni elemento dell'universo}} c_s \geq \sum_{s \in U} c_s = w$$

Lemma 1

Quindi

$$W^* = \sum_{k \in I^*} w_k \geq \sum_{k \in I^*} \frac{\sum_{s \in S_k} c_s}{H(M)} \geq \frac{W}{H(M)}$$

(A) (B) (C)

$$\Rightarrow \frac{W}{W^*} \leq H(M).$$

□

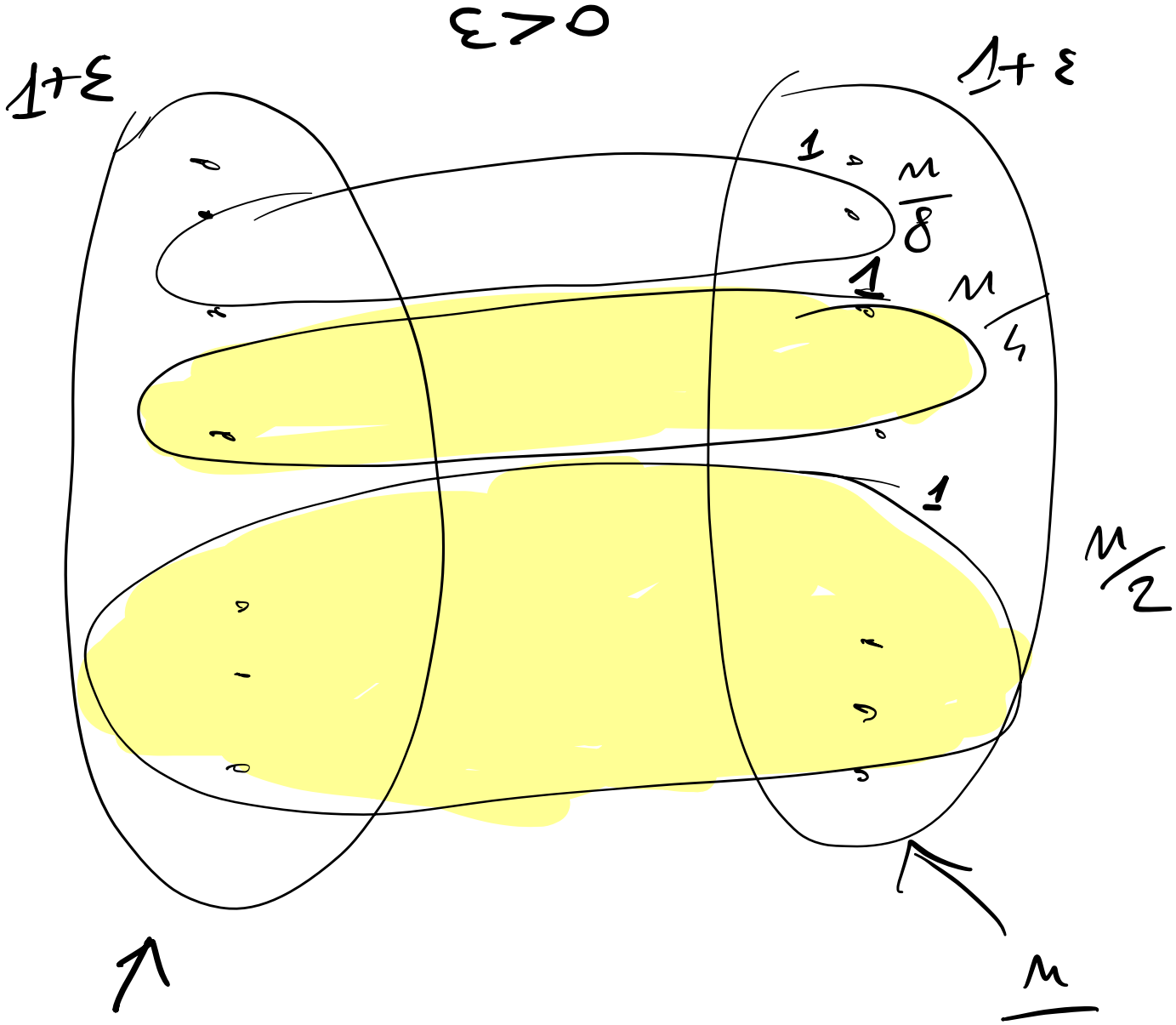
$$M = \max_i |S_i| \leq |U| = n$$

$$H(M) \leq H(n) = O(\log n)$$

L'algoritmo è una $O(\log n)$-approssimazione.

Teorema: Se $P \neq NP$, Set Cover non si può approssimare in tempo polinomiale meglio di $(1 - o(1)) \log n$

ANALISI DI TIGHTNESS



w $\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}$ $n/2$ $n/4$ $\left.\begin{matrix} \\ \\ \\ \end{matrix}\right\}$ $\log_2 n$ $2+\varepsilon$

$\Omega(\log n)$

PROBLEMA DISJOINT PATHS

INPUT : $G = (N, A)$ grafo orientato

$s_0, \dots, s_{k-1}, t_0, \dots, t_{k-1} \in N$

$c \in \mathbb{N}^+$ congestione

OUTPUT AMM. : $I \subseteq K$

e $\forall i \in I$ $\pi_i : s_i \rightsquigarrow t_i$ cammino

t.c. su ogni arco di G non passano più di c cammini

FUNZ. OBIETTIVO : $|I|$

TIPO : MAX

ALGORITMO BASATO SUL PRICING

(s_i, t_i) compete per essere collegata

$\ell : A \to \mathbb{R}^+$ evolve nel tempo

$\pi = \langle x_1, \dots, x_i \rangle$

$\ell(\pi) = \ell(x_1, x_2) + \ell(x_2, x_3) + \dots + \ell(x_{i-1}, x_i)$

GREEDY PRICING DISJOINT PATHS

INPUT
- $G = (N, A)$
- $(s_0, t_0), \dots, (s_{k-1}, t_{k-1})$
- $c \in \mathbb{N}^+$
- $B > 1$

(INPUT DEL PROBLEMA)

- $I \leftarrow \emptyset$
- $P \leftarrow \emptyset$
- $H_2 \leftarrow A$ $\quad \ell(2) \leftarrow 1$

- while True:
  - find the shortest path π_i connecting some pair (s_i, t_i) with $i \notin I$
  - if the path does not exist
    - BREAK
  - $I \leftarrow I \cup \{i\}$
  - $P \leftarrow P \cup \{\pi_i\}$
  - for $a \in \pi_i$
    - $\ell(a) \leftarrow \ell(a) * \beta$
    - if a has been used c times
      - delete a
- output I, P