Sheet 2

1. Efficient alg. for unweighted VC on trees.

   Repeat choosing a parent of a leaf and deleting all incident edges.
   This greedy choice is always a safe choice.

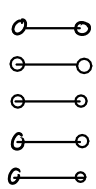
2. a) Opt. alg. for VC $\rightarrow$ opt. alg. for IS

   V VC $\Leftrightarrow$ $\overline{V}$ IS :
   Each edge has at least one endpoint in V $\Leftrightarrow$
   No edge has two endpoints in $\overline{V}$.

   Hence, V min VC $\Leftrightarrow$ $\overline{V}$ max IS

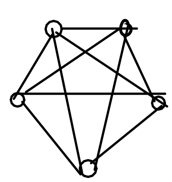
   b) Approx. alg. for VC $\rightarrow$ approx. alg. for IS ?

   No. Ex:



factor 2 $\Rightarrow$ factor 0

K_n:



factor $> \frac{n}{n-1}$ $\Rightarrow$ factor 0

Section 1.6: A Greedy Algorithm

A natural greedy choice would be to „pay" as little as possible for each additional covered element:

Alg 1.2 for Set Cover: Greedy

$I \leftarrow \emptyset$

For $j \leftarrow 1$ to m

  $\hat{S}_j \leftarrow S_j$ (uncovered part of S_j)

While $\{S_j \mid j \in I\}$ is not a set cover

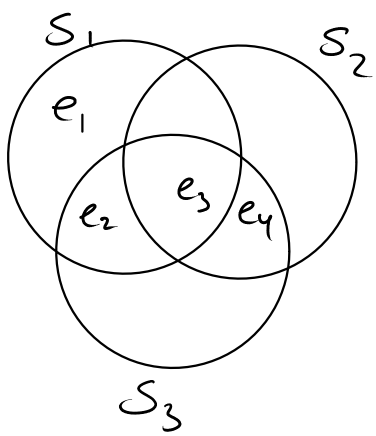
  $\ell \leftarrow \underset{j:\, \hat{S}_j \neq \emptyset}{\arg\min} \dfrac{w_j}{|\hat{S}_j|}$ (S_ℓ: set with smallest cost per uncovered element)

  $I \leftarrow I \cup \{\ell\}$

  For $j \leftarrow 1$ to m

    $\hat{S}_j \leftarrow \hat{S}_j - S_\ell$

Ex:



$w_1 = 12$

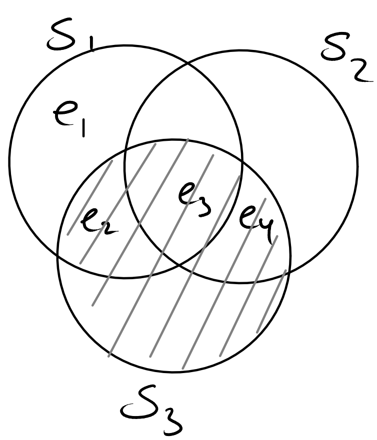
$w_2 = 8$

$w_3 = 9$

$\frac{w_1}{|S_1|} = \frac{12}{3} = 4, \quad \frac{w_2}{|S_2|} = \frac{8}{2} = 4, \quad \frac{w_3}{|S_3|} = \frac{9}{3} = 3$

Pick S_3

price per element in first iteration



$\frac{w_1}{|\hat{S}_1|} = \frac{12}{1} = 12$

Pick S_1

price per element in second iteration

Done!

The greedy alg. is an H_n-approx. alg

Recall: $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \approx \ln(n)$

It is „likely" that no approx. ratio of $\frac{\ln n}{c}$ can be obtained for any $c > 1$.

Thm 1.13:

Approx. factor $\frac{\ln n}{c}$, $c > 1$, for unweighted Set Cover

$\Rightarrow \underbrace{n^{O(\log \log n)}}_{\sim k^{\log n}}$-approx alg. for NPC

Thm 1.11

Alg. 1.2 is an H_n-approx. alg. for Set Cover

<u>Proof</u>:

n_k: # uncovered elements at the beginning of the k'th iteration

In the ex. above:

$n = 5$

$n_1 = 5, \quad n_2 = 1, \quad n_3 = 0$

$n_1 - n_2 = 4, \quad n_3 - n_2 = 1$

Any algorithm, including OPT, has to cover these n_k elements using only sets in $\mathcal{S} - \{S_j \mid j \in I\}$, since none of them are contained in $\{S_j \mid j \in I\}$.

Hence, there must be at least one element with a price of at most OPT/n_k. Otherwise, OPT would not be able to cover the n_k elements (and certainly not all n elements) at a cost of only OPT.

Hence, the $n_k - n_{k+1}$ elements covered in iteration k cost at most $(n_k - n_{k+1})\, OPT/n_k$ in total.

Thus, the cost of the set cover produced by the greedy alg. is

$$\sum_{j \in I} w_j \leq \sum_{k=1}^{\ell} \frac{n_k - n_{k+1}}{n_k} \, OPT$$

\# iterations

$$= OPT \sum_{k=1}^{\ell} \Big(\underbrace{\frac{1}{n_k} + \frac{1}{n_k} + \dots + \frac{1}{n_k}}_{n_k - n_{k+1} \text{ terms}} \Big)$$

$$\leq OPT \sum_{k=1}^{\ell} \left(\frac{1}{n_k} + \frac{1}{n_k - 1} + \dots + \frac{1}{n_{k+1}+1} \right)$$

$$= OPT \sum_{s=1}^{n} \frac{1}{s}$$

$$= OPT \cdot H_n \qquad \square$$

Ex from before:

$$OPT = w_1 + w_2 = 12 + 8 = 20$$

The cost of the greedy alg is

$$\begin{aligned} w_3 + w_1 &= 9 + 12 \\ &= (3+3+3) + 12 \\ &\leq \left(\frac{20}{4} + \frac{20}{4} + \frac{20}{4}\right) + \frac{20}{1} \\ &\leq \left(\frac{20}{4} + \frac{20}{3} + \frac{20}{2}\right) + \frac{20}{1} \\ &= 20\left(\frac{1}{4} + \frac{1}{3} + \frac{1}{2} + 1\right) \end{aligned}$$

Let $q = \max\{|S_j| \mid S_j \in \mathcal{S}\}$.

Thm 1.12

Alg. 1.2 is an H_q-approx. alg. for Set Cover

<u>Proof</u>: By dual fitting.

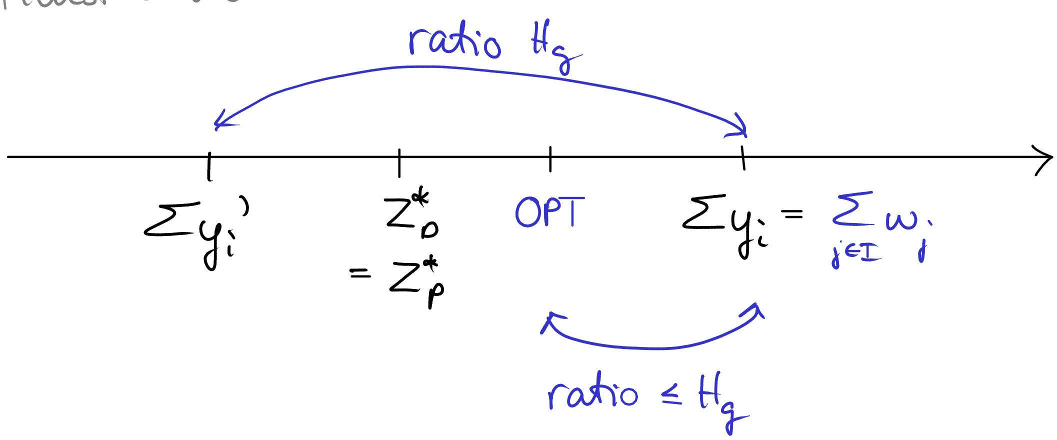
Consider the dual D of the LP for Set Cover.
We will construct an infeasible solution $\vec{y}$ to D s.t.

- $\sum_{j \in I} w_j = \sum_{i=1}^{n} y_i$
- $y_i' = \frac{y_i}{H_q}$, $1 \le i \le n$, is a feasible sol. to D

This will imply that

$$\sum_{j \in I} w_j = \sum y_i = H_q \sum y_i' = H_q Z_D^* = H_q Z_P^* \le H_q \cdot OPT$$

Illustration:



For each i, $1 \leq i \leq n$, we let

$y_i = \text{price}(e_i)$

Then

- $\sum_{j \in I} w_j = \sum_{i=1}^{n} y_i$

Hence, we just need to show that

- $\vec{y'}$ is feasible,

i.e., show that

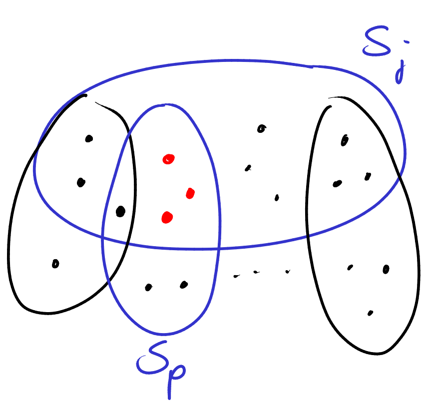
$$\sum_{e_i \in S_j} y_i' \leq w_j, \text{ for each set } S_j \in \mathcal{S}:$$

a_k: #uncovered elements in S_j at the beginning of the k'th iteration.

Then $a_k - a_{k+1}$ elements in S_j are covered in the k'th iteration.

S_p: the set chosen in the k'th iteration.

S_p covers $a_{k+1} - a_k$ previously uncovered elements in S_j



$a_{k+1} - a_k$ elements

The price per element covered in the k'th it. is at most w_j/a_k:

$\frac{w_p}{|\hat{S}_p|} \leq \frac{w_j}{a_k}$, since otherwise S_j would be a more greedy choice than S_p.

Thus,

$$\sum_{i:\, e_i \in S_j} y_i \leq \sum_{k=1}^{\ell} (a_k - a_{k+1}) \frac{w_j}{a_k}$$

$$\leq \sum_{i=1}^{|S_j|} \frac{w_j}{i}$$, by the same arguments as in the proof of Thm 1.12

$$\leq w_j \sum_{i=1}^{g} \frac{1}{i}$$

$$= w_j \cdot H_g$$

Hence,

$$\sum_{e_i \in S_j} y_i' = \frac{1}{H_g} \sum_{e_i \in S_j} y_i \leq w_j$$

□

Compare the proof of Thm 1.12 (dual fitting) to the proof of Thm 1.11:

- Simpler: Compare prices to w_j instead of OPT
- Stronger result: H_g instead of H_n
  (could also have been obtained with the technique of the proof of Thm 1.11)

Ex from before:

$y_2 = y_3 = y_4 = 3$

$y_1 = 12$

$H_3 = 1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$

$y_2' = y_3' = y_4' = \frac{1}{H_3} \cdot 3 = \frac{6}{11} \cdot 3 = \frac{18}{11} < 2$

$y_1' = \frac{6}{11} \cdot 12 = \frac{72}{11} < 7$

$\vec{y'}$ is feasible:

$$
\begin{aligned}
y_1' + y_2' + y_3' &< 7 + 2 + 2 < 12 \\
y_3' + y_4' &< 2 + 2 < 8 \\
y_2' + y_3' + y_4' &< 2 + 2 + 2 < 9
\end{aligned}
$$