

RECAP :

SET-COVER : Given $\langle X, \mathcal{S} \rangle$, $\mathcal{S} \subseteq \mathcal{P}(X)$
determine min-size covering $C^* \subseteq \mathcal{S}$:

$$X = \bigcup_{S \in C^*} S \quad (\text{size} = |C^*|).$$

APPROX-SET-COVER (X, \mathcal{S})

$\cup_{t \in X}; C \neq \emptyset$

while $(\cup \neq \emptyset) \rightarrow$

$S \leftarrow \arg\max \{ |\cap_{t \in S} t| : T \in \mathcal{S} \}$

$\cup \leftarrow \cup - S; C \leftarrow C \cup \{S\}$

return C

We prove a lower bound
on the covering rate:

Let $K = |C^*|$:

$$\frac{|U_C - U_{C^*}|}{|U_C|} \geq \frac{|U_C|(1/K)}{|U_C|} = \frac{1}{K}.$$

... whence $f(n) \leq \lceil \ln n \rceil \leq 1 + \ln n$

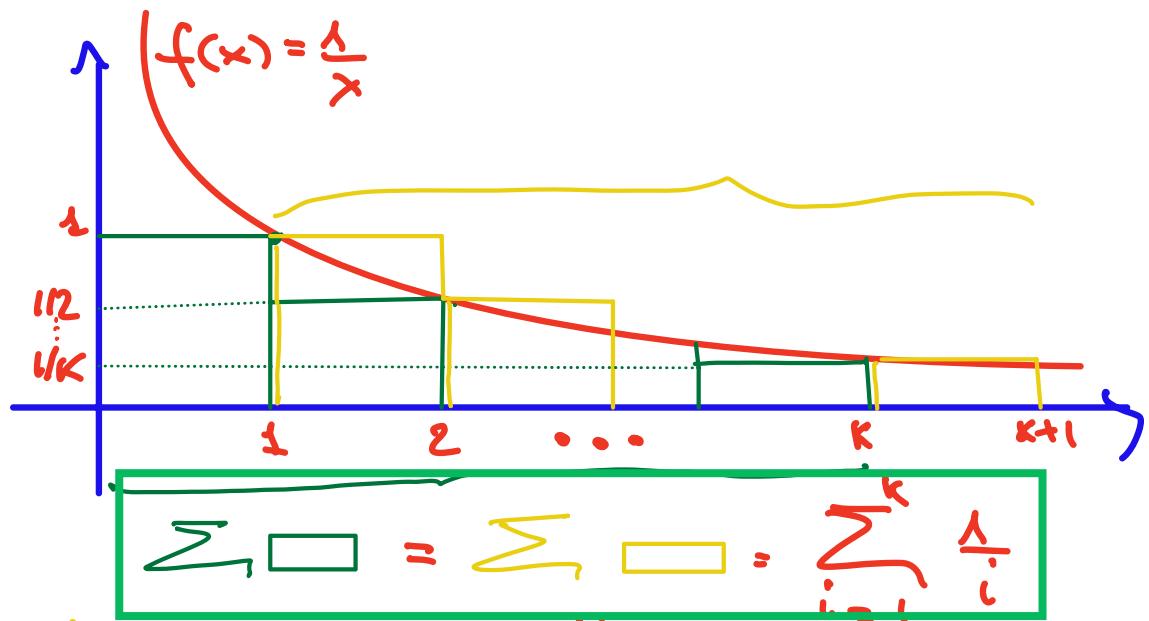
TIGHTER (MORE COMPLEX) ANALYSIS

DEF

Harmonic number $H(k)$:

$$H(k) = \sum_{i=1}^k \frac{1}{i}$$

We have:



Then

$$\int_1^{k+1} \frac{1}{x} dx < \sum_{i=1}^k \frac{1}{i} \leq 1 + \int_1^k \frac{1}{x} dx$$

Thus

$$\ln(k+1) < H(k) \leq 1 + \ln k$$

We will prove that given

$$|S_{\max}| = \max \{|S| : S \in \mathcal{G}\} \leq |x| = n$$

$$g \leq H(|S_{\max}|) \leq 1 + \ln |S_{\max}|$$

LEMMA We prove $P \leq 1 + \ln n$ ($P \leq \lceil \ln n \rceil$). In the worst case $|S_{\text{MAX}}| = \Theta(|X|) = \Theta(n)$. This bound is not better in the worst case but it can be much better for instances with "small" $|S_{\text{MAX}}|$.

TECHNIQUE: CHARGING ARGUMENT

Each $x \in X$ is assigned a weight c_x . The weight depends on $\text{ASC}(x, S)$ and is "small" if x is covered together with many other elements.

We will relate $|P^*|$ to the weights determined by $\text{ASC}(x, S)$ to show that the "covering rate" of the algorithm is good.

RECALL THAT:

U_t : set of elements still to be covered at the beginning of iteration $t \geq 1$

$S_t = \arg\max \{ I \cap U_t : I \in S \}$ selected subset during iteration t

We assign weight c_x to $x \in X$ when x gets covered for the first time at iteration t:

DEF: Let $x \in S_t \cap U_t$ (all elements covered together at iteration t):

Then

$$c_x = \frac{1}{|S_t \cap U_t|}$$

REMARK 1: If iteration t covers many elements ($|S_t \cap U_t|$ large) then c_x is small.

REMARK 2: Since A-S-C covers all elements $x \in X$, each x is assigned a given weight c_x , eventually.

PROPERTY 1:

$$\sum_{x \in S_t \cap U_t} c_x = \sum_{x \in S_t \cap U_t} \frac{1}{|S_t \cap U_t|} = \frac{|S_t \cap U_t|}{|S_t \cap U_t|} = 1$$

We assign unit weight at each iteration. Since there are $\lceil \ell \rceil$ iterations (where ℓ is the cover returned by A-S-C):

$$\sum_{x \in X} c_x = \sum_{t=1}^{|C|} \left(\sum_{\substack{x \in S \cap U_t \\ \text{if } t=1}} c_x \right) = |C|$$

We are ready to relate C^* and C using the weights.

LEMMA

$$\sum_{S \in C^*} \sum_{x \in S} c_x > \sum_{x \in X} c_x = |C|$$

optimal

PROOF: We know that $\underline{C^*}$ is a cover. Thus $\forall x \in X$, c_x must be present as a summand in one of the sums on the L.H.S. !

ARGUMENT IDEA: We will prove that

$$\forall S \in \mathcal{G} : \sum_{x \in S} c_x \leq H(\text{ISI})$$

Therefore, from the lemma:

$$|C| \leq \sum_{S \in C^*} H(\text{ISI}).$$

Let $S_{\max} = \arg \max \{ |S| : S \in \mathcal{F} \}$

We have :

$$|E| \leq \sum_{S \in \mathcal{F}^*} H(|S|) \leq |E^*| H(|S_{\max}|)$$

whence :

$$\rho = \frac{|E|}{|E^*|} \leq H(|S_{\max}|)$$

The only thing to prove to complete the analysis is :

LEMMA

$$\forall S \in \mathcal{F} : \sum_{x \in S} c_x \leq H(|S|)$$

(Intuition: the sum is not too large because A-S-C covers many elements together at each iteration)

PROOF: Let $S \in \mathcal{F}$.

Recall that $S_1, S_2, \dots, S_{|E|}$ are the subsets selected by A-S-C such that U_t is the set of uncovered elements before iteration $t \geq 1$

For $1 \leq t \leq |U|$ define

$$M_t = |S \cap U_{t+1}|$$

The value M_t is the number of elements of S still uncovered after iteration $t \geq 1$

Let $M_0 = |S|$. We have :

$$|S| = M_0 \geq M_1 \geq \dots \geq M_t \geq \dots \geq M_{|U|} = 0$$

(Observe that it might be $M_{t-1} = M_t$)

Now, in S $M_{t-1} - M_t$ elements are covered at iteration t . The cost C_x of each of these elements is $\frac{1}{|S \cap U_t|}$

therefore :

$$\sum_{x \in S} C_x = \sum_{t=1}^{|U|} (M_{t-1} - M_t) \cdot \frac{1}{|S \cap U_t|} \quad (4)$$

Now, recall that s_t is the greedy choice at iteration t , thus

$$|S \cap U_t| \geq |T \cap U_t| \forall t \in S$$

Therefore:

$$|S_{E \cap U_t}| \geq |S \cap U_t| = n_{t-1}$$

Substituting in (1):

$$\sum_{x \in S} c_x \leq \sum_{t=1}^{|E|} \frac{n_{t-1} - n_t}{n_{t-1}}$$

From now on, simple manipulations:

$$\sum_{t=1}^{|E|} \frac{n_{t-1} - n_t}{n_{t-1}} = \sum_{t=1}^{|E|} \sum_{j=n_t+1}^{n_{t-1}} \frac{1}{n_{t-1}}$$

CHECK
AT
HOME!

$$\leq \sum_{t=1}^{|E|} \sum_{j=n_t+1}^{n_{t-1}} \frac{1}{j}.$$

$$\leq \sum_{t=1}^{|E|} \left(\sum_{j=1}^{n_{t-1}} \frac{1}{j} - \sum_{j=1}^{n_t} \frac{1}{j} \right) =$$

$$= \sum_{t=1}^{|E|} (H(n_{t-1}) - H(n_t)) =$$

$$-H(u_0) - H(u_1) + H(u_1) - H(u_2) + \\ \dots + H(u_{|E|-1}) - H(u_{|E|})$$

(telescoping sum)

$$= H(\bar{u}_0) - H(\bar{u}_{\text{ISI}}) =$$

$$= H(\text{ISI}) - H(0) = H(\text{ISI})!$$

OBSERVATIONS

The bound is much better when $|S_{\max}|$ is small.

APPLICATION: MINIMUM VERTEX COVER
restriction: graphs of degree ≤ 3 :
still hard!

Using the $VC \Leftrightarrow SC$ reduction
 (which preserves the approximation)
 I obtain

$$\rho_{3-\Delta-VC} = H(3) = 1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6} < 2$$

An FPTAS for Subset Sum (SS)

Optimization version of SS:

$$\begin{cases} \mathcal{I} = \langle S, t \rangle \mid S \subseteq \mathbb{N}^+ \text{ finite; } t \in \mathbb{N} \\ \mathcal{S} = \{T \subseteq \mathbb{N}^+ : T \text{ finite}\} \end{cases}$$

Given $i = \langle S, t \rangle$: $\mathcal{I}(i) = \{S' \subseteq S : c(S') = \sum_{s \in S'} s \leq t\}$

Type: max

- Determine the subset of S of maximum sum among all those (admissible) whose sum is not larger than target +
- "Best lower approximation" of t

APPLICATION: $S = \{s_1, \dots, s_m\}$ are weights and t is the capacity of the container.

Aim: pack the container as much as possible (hypotheses: ① no volume constraints; ② all items have the same unit value)

HARDNESS: Decision version:

$$\begin{cases} \text{SS}' \\ \mathcal{I} : \langle S, t, k \rangle \\ \mathcal{Q} : \exists S' \subseteq S : k \leq \sum_{s \in S'} s \leq t ? \end{cases}$$

lower bound

Clearly $SS \subset SS'$ (generalization)

$$f(\langle S, t \rangle) = \langle S, t, t \rangle$$

The question for SS' under f becomes:

$$\exists S' \subseteq S : t \leq \sum_{\substack{S \subseteq S' \\ S \in SS'}} s \leq t$$
$$\Leftrightarrow \exists S' \subseteq S : \sum_{S \in SS'} s = t ?$$

Same question as SS .

Clearly,

$$\langle S, t \rangle \in SS \Leftrightarrow f(\langle S, t \rangle) = \langle S, t, t \rangle \in SS'$$

We will develop a FPTAS for (the optimization version of) $SS : \text{APPROX-}SS(\langle S, t \rangle, \varepsilon)$

$$\forall 0 < \varepsilon \leq 1, \text{ if } S' = \text{APPROX-}SS(S, t, \varepsilon),$$

$$\frac{CC(S^*)}{CC(S')} : \frac{\sum_{S \in S^*} s}{\sum_{S \in S'} s} \leq (1 + \varepsilon)$$

APPROACH (paradigmatic): we start from an exhaustive approach which enumerates all costs of subsets in $\mathcal{J}(i)$. The exhaustive search is modified (according to ε) to yield the FPTAS. $\downarrow \varepsilon, \text{ and a eliminate}$

REMARK Exhaustive search may cost $\Theta(2^{|S|})$

LET $S = \{x_1, x_2, \dots, x_n\}$. W.l.o.g. $x_i \leq t$

DEFINE L_i as the ordered list without duplicates of ALL costs of feasible subsets of $S_i = \{x_1, x_2, \dots, x_i\}$, $1 \leq i \leq n$.
Also, define $L_0 = \langle \rangle = \langle \text{CC}(\emptyset) \rangle$

EXAMPLE For instance $\langle S, 12 \rangle$, with
 $S = \{x_1=3, x_2=4, x_3=7\}$

$L_0 = \langle \rangle$; $L_1 = \langle 0, 3 \rangle$; $L_2 = \langle 0, 3, 4, 7 \rangle$;
 $L_3 = \langle 0, 3, 4, 7, 10, 11 \rangle$. Note that
 $x_1 + x_2 + x_3 = 14 \notin L_3$

DEFINE the operation $L +_t x$ as the list obtained by summing x to all elements of L and removing the values $> t$

e.g.: $L_2 +_{12} 7 = \langle 0+7, 3+7, 4+7, 7+7 \rangle =$
 $= \langle 7, 10, 11 \rangle$

clearly: $T_{+_t}(1|L|) = \Theta(|L|)$

LET $\text{MERGE-WD}(A, B)$ a subroutine doing the merge of two ordered lists A and B which also eliminates duplicates in the resulting list. $T_{\text{MERGE-WD}}(1|A|, 1|B|) = \Theta(|A| + |B|)$

PROPERTY

For $1 \leq i \leq n$:

$$L_i = \text{MERGE-WD}(l_{i-1}, l_{i-1} +_t x_i)$$

PROOF

Let $S \subseteq S_i$. If $x_i \notin S$ then $S \subseteq S_{i-1}$ and

$$c(S) = \sum_{x_j \in S} \in L_{i-1}. \text{ Otherwise:}$$

$$c(S) = \sum_{x_j \in S - \{x_i\}} + x_i, \text{ and}$$

$$S - \{x_i\} \subseteq S_{i-1}. \text{ Therefore } c(S) \in l_{i-1} +_t x_i$$

The ordered list without duplicates L_i can thus be obtained as

$$L_i = \text{MERGE-WD}(l_{i-1}, l_{i-1} +_t x_i)$$

We are ready to see the exhaustive algorithm:

EXP-SS(S, t)

* Let $S = \{x_1, x_2, \dots, x_n\}$ *

$L_0 = \langle \rangle$

for $i \leftarrow 1$ to n do

$$L_i \leftarrow \text{MERGE-WD}(L_{i-1}, L_{i-1} +_t x_i)$$

return $\text{MAX}(L_n)$

CORRECTNESS: Straight from PROPERTY:

$\text{MAX}(L_n)$ is the maximum cost of a feasible subset

RUNNING TIME: In the worst case:

$$\begin{cases} |L_i| = 2 |L_{i-1}| \\ |L_0| = 1 \end{cases}$$

$$\Rightarrow |L_i| = 2^i$$

\Rightarrow The time of HEROES.VD(L_i, L_{i+1}, x_i) is $\Theta(2^i)$

$$\Rightarrow T_{\text{ESS}}(n) = \Theta(2^n) !$$

OBSERVATION: $2^{|S|}$ can be $\Theta(2^{\langle S \rangle})$!