Set Cover

Input: Universe $U = \{e_1, e_2 .. e_n\}$

Family F of Subsets $S_1, S_2, .. S_m \subseteq U$ & $\bigcup_{i \in [m]} S_i = U$

Costs $c_1, c_2, \quad c_m$

Goal: Find a set $I \subseteq \{1, 2, .. m\}$

that minimizes $\sum_{i \in I} c_i$ such that $\bigcup_{i \in I} S_i = U$

Ex: $U = \{1, 2, 3, 4, 5, 6\}$

$S_1 = \{1, 3, 5\}$, $S_2 = \{2, 3, 4\}$, $S_3 = \{2, 4, 6\}$,

$c_1 = 1$, $c_2 = 1$, $c_3 = 1$

$k = 2$,

then $S_1 \cup S_3 = U$, hence it is a YES instance.

total cost $= 2$

Remark : Set cover is NP-Complete. [All sets are of equal cost]

[Vertex cover is a special case of Set Cover]

Vertex Cover $\leq_p$ Set Cover

Given an instance (G, k) of vertex cover.

We will construct an instance of the set cover problem.

let $U = E(G)$ and let S_i be the set of edges that incident to vertex i.

Clearly $S_i \subseteq U$ for all i.

[For full Proof Refer to Lecture notes of NP-Completeness]

The greedy algorithm

Algorithm 2.2 (Greedy set cover algorithm)

1. $C \leftarrow \emptyset$
2. While $C \neq U$ do
   Find the set whose cost-effectiveness is smallest, say S.
   Let $\alpha = \frac{c(S)}{|S-C|}$, i.e., the cost-effectiveness of S.
   Pick S, and for each $e \in S - C$, set $\mathrm{price}(e) = \alpha$.
   $C \leftarrow C \cup S$.
3. Output the picked sets.

Number the elements of U in the order in which they were covered by the algorithm, resolving ties arbitrarily. Let $e_1, \ldots, e_n$ be this numbering.

Example

$X = \{a_1, a_2, a_3, a_4, a_5\}$

$Y = \{a_3, a_4, a_5, a_6, a_7\}$

$Z = \{a_1, a_2, a_8, a_9, a_{10}, a_{11}, a_{12}\}$

$c(X) = 6, \quad c(Y) = 15, \quad c(Z) = 7$

Choose Z: $\alpha_Z = \frac{c(Z)}{|S-C|} = \frac{7}{7} = 1$

Choose X: $\alpha_X = \frac{c(X)}{|S-C|} = \frac{6}{3} = 2$

Choose Y: $\alpha_Y = \frac{c(Y)}{|S-C|} = \frac{15}{2} = 7.5$

Total Cost $= 6 + 15 + 7 = 28$

The greedy algorithm is not optimal.

An optimal solution would have chosen Y and Z for a cost of 22.

Lemma: For each $k \in \{1, 2 .. n\}$, $Price(e_k) \leq \frac{OPT}{n-k+1}$

Proof: let the optimal set $O_1, O_2, .. O_p$

$$\therefore \; OPT = \sum_{i=1}^{p} c(O_i) \quad \text{—} ①$$

Now, assume that the greedy algorithm has covered the elements in C, so far.

The the uncovered elements are $U - C$ &

we have

$$|U-C| \leq |O_1 \cap (U-C)| + |O_2 \cap (U-C)| + \cdots |O_p \cap (U-C)| \quad \text{—} ②$$

In the greedy algorithm we select the set with cost effectiveness α, where

$$\alpha \leq \frac{c(O_i)}{|O_i \cap (U-C)|}, \quad i = 1, 2 .. p. \quad \text{—} ③$$

We know this because the greedy algorithm will always choose the set with the smallest cost effectiveness, which will either be smaller than or equal to a set that the optimal algorithm chooses.

$$\therefore \quad c(O_i) \geq \alpha \, |O_i \cap (U-C)|$$ from ③

$$OPT = \sum_i c(O_i) \geq \alpha \sum_i |O_i \cap (U-C)|$$

$$\geq \alpha \cdot |U-C|$$ from ①, ②, and ③

$$\alpha \leq \frac{OPT}{|U-C|}$$

$\therefore$ The price of the k^{th} element is

$$\alpha \leq \frac{OPT}{n-(k-1)} = \frac{OPT}{n-k+1}$$

=x=x=

$\therefore$ Since the cost of each set picked is distributed among the new elements covered, the total cost of the set cover picked is

$$= \sum_{k=1}^{n} \text{Price}(e_k)$$

$$= \sum_{k=1}^{n} \frac{OPT}{n-k+1}$$

$$= \left(1 + \frac{1}{2} + \cdots + \frac{1}{n}\right) OPT$$

$$= H_n \cdot OPT.$$

where

$H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n}$