Dual fitting for constrained set multicover

Constrained set multicover:

Input: $\mathcal{U} = \{1, \dots, n\}$, $\mathcal{S} = \{\underbrace{S_1, \dots, S_k}_{\text{subsets of } \mathcal{U}}\}$, $c(S) \to \mathbb{R}^+$.

A positive integer r_i for $i \in \mathcal{U}$.

Feasible sol$^{\underline{n}}$: A collection of subsets from $\mathcal{S}$ with the property that the element i appears in at least r_i sets in the collection. Each set is picked at most once.

Task: Find a feasible solution of minimum cost.

Algorithm: Generalization of the set-cover greedy

- At any step, an element i is "alive" if it appears in less than r_i sets that have been picked so far.
- For $S \in \mathcal{S}$, define its cost-effectiveness to be

$$\frac{c(S)}{\text{Number of alive elements in } S}.$$

- At each step, choose an unpicked set with minimum cost-effectiveness.
- Continue until you have a feasible solution.

for $i \in \mathcal{U}$, $j \in \{1, \dots, r_i\}$

price(i, j) := cost-effectiveness of the set that covers i for the j-th time.

Cost of the computed sol$|^n$ $= \sum_{i=1}^{n} \sum_{j=1}^{r_i} \text{price}(i,j)$. (verify)

$$\text{minimize} \sum_{S \in \mathcal{S}} c(S) \cdot x_S$$

Primal s.t. $\sum_{S : i \in S} x_S \geq r_i \quad \forall i \in \mathcal{U} \quad y_i$

L: $-x_S \geq -1 \quad \forall S \in \mathcal{S} \quad z_S$

$x_S \geq 0 \quad \forall S \in \mathcal{S}$

$x_S \in \mathbb{Z}$ relax: $\downarrow$ $x_S \in \mathbb{R}$

val$(L) \leq$ OPT.

Dual

L': maximize $\sum_{i=1}^{n} r_i y_i - \sum_{S \in \mathcal{S}} z_S$

s.t.

$$\sum_{i \in S} y_i - z_S \leq c(S)$$

$y_i, z_S \geq 0$

Setting of dual variables:

$$y_i := \text{Price}(i, r_i) \qquad \forall\, i \in \mathcal{U}$$

$$z_S := 0 \quad \text{if the algorithm does not pick } S$$

$$:= \sum_{\substack{i \text{ covered} \\ \text{by } S}} \left(\text{Price}(i, r_i) - \text{Price}(i, j_i) \right) \quad * \qquad \geq 0 \text{ by Claim 1 (later).}$$

$*$ $\quad j_i :=$ the copy i that S covers.

Dual objective under this setting:

$$\sum_{i=1}^{n} r_i \cdot \text{Price}(i, r_i) - \sum_{\substack{S \in \mathcal{S}: \\ S \text{ picked} \\ \text{by algo}}} \left(\sum_{\substack{i \text{ covered} \\ \text{by } S}} \left(\text{Price}(i, r_i) - \text{Price}(i, j_i) \right) \right)$$

$$= \sum_{i=1}^{n} r_i \cdot \text{Price}(i, r_i) - \sum_{i=1}^{n} r_i \cdot \text{Price}(i, r_i) + \sum_{i=1}^{n} \sum_{j=1}^{r_i} \text{Price}(i, j).$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{r_i} \text{price}(i, j) = \text{cost of the computed sol}^{n}.$$

Claim 1 $\forall i$,
$$\text{price}(i,1) \le \text{price}(i,2) \le \cdots \le \text{price}(i, r_i).$$

Proof: Pick any $j = 1, \ldots, r_i - 1$

Let S be the set that covers the $j+1^{st}$ copy of i.

S was available when the j-th copy of i was covered.

$$\text{Price}(i,j) \le \text{Price}(i, j+1).$$

LHS of the dual constraint for a set $S \in \mathcal{S}$

$$\sum_{i \in S} y_i - z_s.$$

Case 1: S is not picked by the algo.

$$LHS = \sum_{i \in S} \text{Price}(i, r_i).$$

Let $e_1, e_2, \ldots, e_k$ be the elements of set S in the order in which they cease to be alive.

$$LHS = \sum_{i=1}^{k} \text{Price}(e_i, r_{e_i})$$

Consider the step in which e_i ceases to be alive. In that step, S was available to the algorithm, with cost-effectiveness $\leq \frac{c(S)}{k-i+1}$ (since $e_{i+1}, \dots, e_k$ are alive).

$$\Rightarrow \text{price}(e_i, r_{e_i}) \leq \frac{c(S)}{k-i+1}$$

$$\text{LHS} \leq \sum_{i=1}^{k} \frac{c(S)}{k-i+1} = c(S) \cdot H_k .$$

<u>Case 2</u>: S is picked by the algorithm. Let $e_1, \dots, e_k$ be the elements of S. Let $e_1, \dots, e_{k'}$ be not alive when S is picked(*). Furthermore, assume that S covers the j_{e_i}th copy of the element e_i for $i = k'+1, \dots, k$.

(*) ceased to be alive in this order

$$\text{LHS} = \sum_{i=1}^{k} y_{e_i} - \bar{z}_S$$

$$= \sum_{i=1}^{k} \text{price}(e_i, r_{e_i}) - \sum_{i=k'+1}^{k} \left(\text{price}(e_i, r_{e_i}) - \text{price}(e_i, j_{e_i}) \right)$$

$$= \sum_{i=1}^{k'} \text{price}(e_i, r_{e_i}) + \underbrace{\sum_{i=k'+1}^{k} \text{price}(e_i, j_{e_i})}_{c(S)}$$

$$\sum_{i=1}^{k'} \text{Price}(e_i, r_{e_i})$$

The r_{e_i}-th copy of e_i was covered before S was picked. S was available in that iteration with cost effectiveness $\leq \dfrac{c(S)}{k-i+1}$ [since $e_i, e_{i+1}, \ldots, e_{k'}, e_{k'+1}, \ldots, e_k$ were alive]

$$\Rightarrow \text{price}(e_i, r_{e_i}) \leq \frac{c(S)}{k-i+1}$$

$$\sum_{i=1}^{k'} \text{Price}(e_i, r_{e_i}) \leq c(S) \cdot \left[\frac{1}{k} + \frac{1}{k-1} + \cdots + \frac{1}{k-k'+1}\right]$$

$$\text{LHS} \leq c(S) + c(S)\left[\frac{1}{k} + \frac{1}{k-1} + \cdots + \frac{1}{k-k'+1}\right]$$

$$\leq c(S) \cdot H_k$$

$y_i \leftarrow \dfrac{y_i}{H_n}$, $z_S \leftarrow \dfrac{z_S}{H_n}$ is a dual-feasible.

$\Rightarrow$ The algorithm is an H_n-factor approximation algorithm.