• Online Set Cover :

Given :

- $U := \{1, 2, \dots, n\}$: Ground set of n elements.
- $\mathcal{F} := \{S_1, S_2, \dots, S_m\}$: Family of subsets of U.
- A set cover $\mathcal{F}' \subseteq \mathcal{F}$ is a subcollection of sets from $\mathcal{F}$ such that their union is U.

  i.e. $\bigcup_{S \in \mathcal{F}'} S = U$.

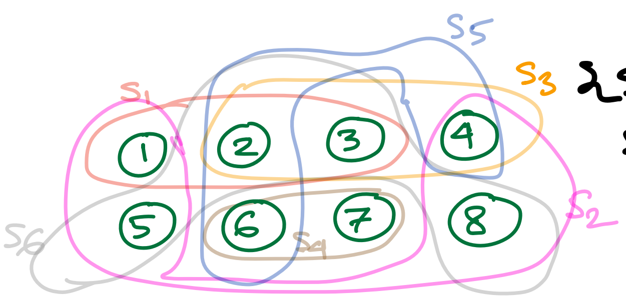
- Each set $S \in \mathcal{F}$ has a nonnegative $c(S)$ associated with it.

Goal :

Find a set cover of minimum cost.

<u>Example</u>: $U = \{1, 2, \dots, 8\}$; $n = 8$, $m = 6$,

$S_1 = \{1, 2, 3\}$, $S_2 = \{1, 4, 5, 8\}$, $S_3 = \{2, 3, 4\}$,
$S_4 = \{6, 7\}$, $S_5 = \{2, 4, 6\}$, $S_6 = \{2, 3, 5, 8\}$.



$\{S_1, S_2, S_4, S_6\}$ is a set cover

$\{S_2, S_3, S_4\}$ is a set cover.

$\{S_1, S_3, S_6\}$ is NOT!

– Offline set cover is well-studied in approximation algorithms. known to have $\Theta(\log n)$- approx (Assuming P != NP).

- Set cover LP :

min cost collection

$$\min \sum_{S \in \mathcal{F}} c(S) \cdot x_S$$

(P)

$$\text{s.t.} \sum_{S : e \in S} x_S \geq 1 , \quad \forall e \in U$$

each item is covered

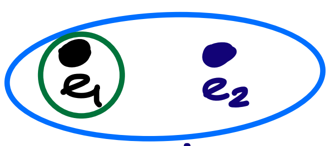
$$x_S \in \{0, 1\}$$

$x_S \geq 0$

0: the set is not selected

1: the set is selected

Toy example 1 : $U = \{e_1, e_2\}$.



$S_1 = \{e_1\}$, $S_2 = \{e_1, e_2\}$.

$c(S_1) = c(S_2) = 1$.

IP. $\min \ x_{S_1} + x_{S_2}$

s.t. $x_{S_2} \geq 1$, $x_{S_1} + x_{S_2} \geq 1$, $x_{S_1}, x_{S_2} \in \{0, 1\}$

OPT: $\{S_2\}$, obj = 1.

Note ≤ 1 is redundant

LP. $\min \ x_{S_1} + x_{S_2}$

s.t. $x_{S_2} \geq 1$, $x_{S_1} + x_{S_2} \geq 1$, $x_{S_1}, x_{S_2} \geq 0$.

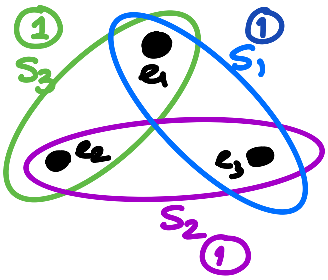
LP obj $= x_{S_1} + x_{S_2} \geq 1 = OPT_{obj}$.

Now LP obj $\leq OPT_{obj}$ as it is a relaxation.

$\Rightarrow LP_{obj} = OPT_{obj}$

→ LP obj = OPT obj.

Toy example 2:



$$\min \; x_{S_1} + x_{S_2} + x_{S_3}$$

$$\text{s.t.} \quad x_{S_1} + x_{S_2} \geq 1$$
$$x_{S_2} + x_{S_3} \geq 1$$
$$x_{S_3} + x_{S_1} \geq 1$$

$x_{S_1}, x_{S_2}, x_{S_3} \in \{0,1\}$ → IP

≥ 0 → LP

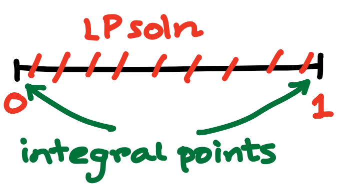
OPT = 2, as $\geq$ two sets are needed to cover all.

$LP \leq 3/2$, as $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is a feasible solution.

So, $OPT / LP \geq 4/3$.

In fact, one can show integrality gap of set cover LP is $\Omega(\log n)$.

- **Rounding**: How to get $\{0,1\}$ from $[0,1]$?



idea of ceiling or floor?

- Simplest approach: Based on threshold (θ).

Say, $x_i \in [0,1]$, if $x_i \geq \theta \rightarrow \tilde{x}_i = 1$

$x_i < \theta \rightarrow \tilde{x}_i = 0$.

Online version:

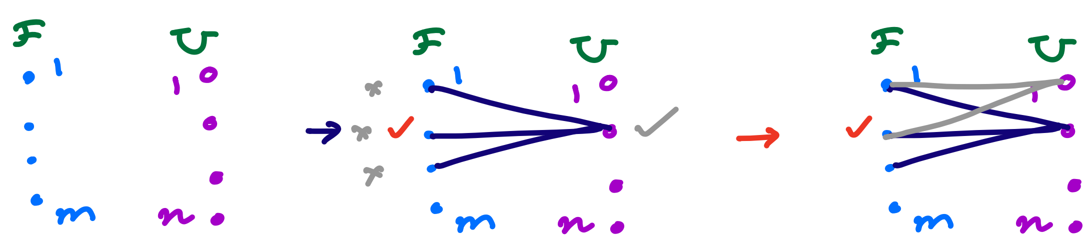
→ Elements of $\mathcal{U}$ & members of $\mathcal{F}$ is known.

→ Elements of $\mathcal{U}$ appear one-by-one (one at each time step)

→ Also only a subset of elements $\mathcal{U}' \subseteq \mathcal{U}$ may arrive. Neither the order of arrival nor $\mathcal{U}'$ is known to the algorithm.

→ Once a new element e appears, we get to know the sets covering the new element.

The algorithm (we call it $\mathcal{A}$) must cover e by some set of $\mathcal{F}$ containing e.



→ For $\mathcal{U}'$ with some arrival order σ, let $\mathcal{F}'(\mathcal{U}', \sigma)$ be the cover produced by $\mathcal{A}$ & $\mathcal{F}'_{OPT}(\mathcal{U}')$ be the optimal (offline) cover.

Then competitive ratio $(\mathcal{A})$

$$= \max_{\mathcal{U}', \sigma} \frac{\mathcal{F}'(\mathcal{U}', \sigma)}{\mathcal{F}'_{OPT}(\mathcal{U}')}.$$

We want to design $\mathcal{A}$ to minimize this.

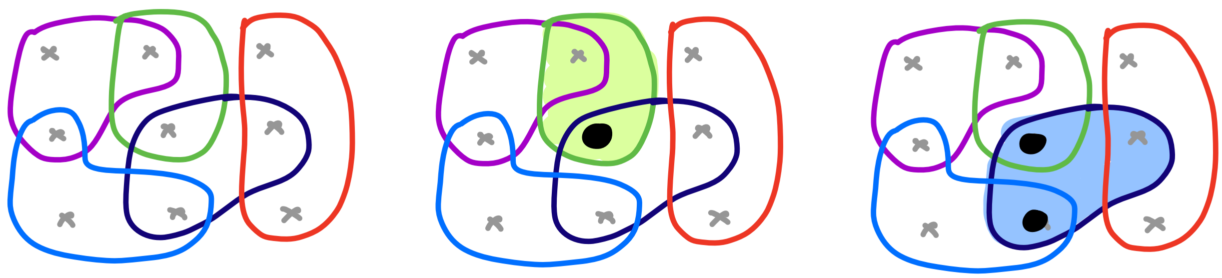
- Similarly, CR can be defined for fractional set cover.

• Application: Network service providers.

Clients : Elements.
Servers : Sets.

A server can provide service to a subset of clients.

- Clients arrive one-by-one.
- There is a set-up cost / activation cost for each server.
- We know all potential clients, but not the clients who will request service.



→ In online version, at each iteration one item & thus one constraint appears.

Q. Can we obtain $O(\log m)$-competitive algo for fractional set cover?

→ Monotonicity.

• A candidate algorithm for fractional set cover.

- Maintain a weight $w_S > 0 \ \forall S \in \mathcal{F}$.
- Define weight of an element j:

$$w_j = \sum_{S \in \mathcal{F}_j} w_S ,$$

where $\mathcal{F}_j$ are collection of sets in $\mathcal{F}$ that contains j.

Single iteration of the algorithm:

→ Element j appears.

→ If $w_j \geq 1$, do nothing. already covered.

→ Else ($w_j < 1$), perform weight augmentation

(a) Let K be min integer s.t. $2^K \cdot w_j > 1$

($Note: \ 2^K \cdot w_j \leq 2$) monotonicity is maintained.

(b) For each set $S \in \mathcal{F}_j$, $w_S \leftarrow 2^K \cdot w_S$.

(So, after this w_j becomes > 1) $w_j \leftarrow 2^K \cdot w_j > 1$

Q. How do we initialize w_j's? Say, initially $w_S = \frac{1}{2\beta}$.

Lemma. # iterations where $w_j < 1$, is at most $|OPT| \cdot (\log \beta + 2)$.

→ Initially, $w_S = 1/2\beta$. Always $w_S \leq 2$. ensured by step a of Algo

→ Consider sets in OPT. In each augmentation, say C_j be one of the sets in OPT that covers j. Then w_{C_j} is at least doubled.

→ This can happen $|OPT| \cdot \log \left(\frac{2}{\frac{1}{2\beta}} \right)$ times.

So, we want β to be large, say $\beta \geq \frac{1}{m}$.

But large β is also problematic

Say, $\beta \geq \frac{\log^2 n}{m}$, then say element is covered by all sets. $OPT = 1$, $ALGO \geq \log^2 n$.

Now, we want to extend this to integral case.

- An $O(\log m \log n)$-competitive algorithm for unweighted case [$c(S) = 1 \ \forall S \in \mathcal{F}$].

[Alon, Awerbuch, Azar, Buchbinder, Naor, STOC'03]

Highlevel idea:

- Maintain a weight $w_S > 0 \ \forall S \in \mathcal{F}$.

(Intuitively, w_S can be thought of as a fraction that S is being selected. Our goal is to increase the weights over time so that we get a fractional set cover & convert the fractional soln to an integral soln)

- Initialize: $w_S = 1/2m \ \forall S \in \mathcal{F}$.

  $\mathcal{F}' = \phi$. (empty cover)

  $F = \phi$. (elements covered by $\mathcal{F}'$)

- Define weight of an element j:

$$w_j = \sum_{S \in \mathcal{F}_j} w_S,$$

where $\mathcal{F}_j$ are collection of sets in $\mathcal{F}$ that contains j.

(Intuitively, if $w_j \geq 1 \ \forall j \in X'$, we obtain a fractional cover. Else if $w_j < 1$, we want to increase weights of the sets in $\mathcal{F}_j$.).

Magic: Use of potential function to convert the fractional soln to an integral soln.

$$\boxed{\Phi = \sum_{j \notin F} n^{2w_j}}$$

Φ intuitively measures amount of "uncoveredness" & we want to decrease it to 0 over time.

Over iterations w_j grows, so we want to adjust F (by selecting more sets into cover) such that Φ dont increase.

Single iteration of the algorithm:

→ Element j appears.

→ If $w_j \geq 1$, do nothing.

Later part of the algorithm will ensure that this fractional cover imply integral cover

→ Else ($w_j < 1$), perform weight augmentation

(a) Let k be min integer s.t. $2^k \cdot w_j > 1$

(Note: $2^k \cdot w_j \leq 2$)

(b) For each set $S \in \mathcal{F}_j$, $w_S \leftarrow 2^k \cdot w_S$.

(So, after this w_j becomes > 1) $w_j \leftarrow 2^k \cdot w_j > 1$

(But ϕ also increases, so to decrease ϕ we need to increase cover by F)

(c) Choose from $\mathcal{F}_j$, at most $4 \log n$ sets & add them to $\mathcal{F}'$ s.t. $\Delta\Phi \leq 0$.

(We will see how to choose such sets).

Lemma 1. # iterations where $w_j < 1$, is at most $|OPT| \cdot (\log m + 2)$.

will help in bounding cost

weight augmentation iteration

will maintain a feasible soln.

Lemma 2. For iteration with $w_j < 1$, Let Φ_s & Φ_e be Φ before & after the iteration. Then $4 \log n$ sets can be chosen s.t. $\Delta\Phi = \Phi_e - \Phi_s \leq 0$.

- **Theorem:** $\mathcal{F}'$ is a feasible cover & $|\mathcal{F}'|$ is $O(|OPT| \log m \log n)$.

Proof: Initially $\Phi = \sum_{j \in U} n^{2 w_j}$.

$w_j = \sum_{S \in \mathcal{F}_j} \frac{1}{2m} \leq \frac{1}{2}$.

$|U| = n$.

We cant have $|\mathcal{F}_j| = m, \forall j$

$$< \sum_{j \in U} n^{2 \cdot 1/2} \leq |U| \cdot n \leq n^2.$$

Lem 2 $\Rightarrow \Phi$ is non-increasing.

Thus if $w_j \geq 1$, then $j \in F$;

else $\Phi \geq n^{2 w_j} \geq n^2$.

Fractional cover imply integral cover!

So $\mathcal{F}'$ is a feasible cover.

$$|\mathcal{F}'| = |OPT| (\log m + 2) \cdot 4 \log n.$$

iteration where sets are added

max # sets added in an iteration. $\blacksquare$

Lemma 1. # iterations where $w_j < 1$, is at most $|OPT| \cdot (\log m + 2)$.

→ Initially, $w_S = 1/2m$. Always $w_S \leq 2$. ensured by step a of Algo

→ Consider sets in OPT. In each augmentation, say $C_j \in OPT$ covers element j. Then w_{C_j} is at least doubled.

→ This can happen $|OPT| \cdot \log\left(\frac{2}{\frac{1}{2m}}\right)$ times.

Note: Larger w_S is better. But we also need initial $\Phi \leq n^2$, so $w_S = 1/2m$ is chosen.

However, if each element appears in $\leq d$ sets, then $w_S = 1/2d$ will work & we'll get $O(\log d \log n)$ - approx.

Lemma 2. For iteration with $w_j < 1$, Let Φ_s & Φ_e be Φ before & after the iteration. Then $4 \log n$ sets can be chosen s.t. $\Delta\Phi = \Phi_e - \Phi_s \leq 0$.

Proof: For each set $S \in \mathcal{F}_j$, let w_S & $w_S + \delta_S$ be the weights before & after the iteration.

Let $\delta_j := \sum_{S \in \mathcal{F}_j} \delta_S$.

The algorithm maintains $w_j + \delta_j \leq 2$.

How sets are added:

Repeat $4 \log n$ times:

choose each set $S \in \mathcal{F}_j$ w.p. $\delta_S/2$.

[As $\delta_j \leq 2$, in expectation we select one set. In fact, we can select at most one set per iteration by choosing a number uniformly at random in [0, 1].]

$\Rightarrow$ $4 \log n$ sets are selected.

Claim: $\Delta\Phi \leq 0$.

Consider an element $j' \in U$ s.t. $j' \notin F$.

Its contribution to $\Phi_s = n^{2w_{j'}}$. $\Phi_s(j')$

" to $\Phi_e = 0$ if some set containing j' is chosen.

$\Phi_e(j')$ $= n^{2(w_{j'} + \delta_{j'})}$, otherwise.

For each of the $4\log n$ steps,

$\mathbb{P}$[Any set containing j' is not chosen]

$$\le 1 - \sum_{S \in \mathcal{F}_j} \mathbb{P}[S \text{ is chosen}] \le 1 - \sum_{S \in \mathcal{F}_j} \frac{\delta_S}{2} = 1 - \frac{\delta_{j'}}{2}.$$

$\mathbb{P}$[j' is not covered by $4\log n$ steps]

$$\le \left(1 - \frac{\delta_{j'}}{2}\right)^{4\log n} \le e^{-\delta_{j'}/2 \cdot 4\log n} \quad \left[\because 1-x \le e^{-x}\right]$$

$$\le n^{-2\delta_{j'}}$$

Hence, $\mathbb{E}[\Phi_e(j')]$

$$\le n^{-2\delta_{j'}} \cdot n^{2(w_{j'}+\delta_{j'})} + (1 - n^{-2\delta_{j'}}) \cdot 0$$

$$= n^{2w_{j'}} = \Phi_S(j').$$

From linearity of expectation,

$$\mathbb{E}[\Phi_e] = \mathbb{E}\Big[\sum_{j' \notin F} \Phi_e(j')\Big] = \sum_{j' \notin F} \mathbb{E}[\Phi_e(j')]$$

$$\le \sum_{j' \notin F} \Phi_S(j') = \Phi_S. \quad \blacksquare$$

Extensions:

- Can be extended to weighted case.
  Doubling trick + involved potential fn.
- Can be derandomized.
- Lower bound: $\Omega(\log n \log m / (\log\log m + \log\log n))$ for any online algorithm.