Part 5 Set Cover

(**Definition 1**) (**Set Cover**) Given a **set** U and a collection of n **subsets** of U: $S_1, S_2, \cdots, S_n$ such that $\bigcup_{i=1}^{n} S_i = U$. A **set cover** is a collection of their sets whose union is U. The goal of the **Set Cover** problem is to find a set cover with **smallest cardinality** (i.e., with **minimum total weight**). That is to find $I = \{1, 2, , \cdots, n\}$ such that $\bigcup_{i=1}^{n} S_i = U$ and $|I|$ is **minimum**.

(**Example 1**) Given a **set** U={C, C++, Ruby, Python, Java} as well as subsets S_1={C, C++}, S_2={C++, Java}, S_3={C++, Ruby, Python}, and S_4={C, Java}.
{S_3, S_4} is a feasible set cover.

(**Definition 2**) (**Weighted Set Cover**) Given a **set** U and n **subsets** of U: $S_1, S_2, \cdots, S_n$ such that $\bigcup_{i=1}^{n} S_i = U$. Each subset S_i has a **non-negative weight** w_i. The goal of the **Weighted Set Cover** problem is to find a **set cover** C with **minimum total weight** $\sum_{S_i \in C} w_i$.

Notes: The **Weighed Set Cover** problem is a **generalization** of the **Weighted Vertex Cover** problem. Consider a graph G=(V, E), where each vertex $v \in V$ has a non-negative weight w_v. One can treat U=E and associate each **vertex** $v \in V$ with a **subset** S_v, where S_v is the **set of edges incident to vertex** v.

Notes: **Weighted Vertex Cover** is a **special case** of **Weighted Set Cover**. In **Weighted Vertex Cover**, each edge $e \in E$ must have **two induced vertexes** (end points). While in **Weighted Set Cover**, each set item $u \in U$ can covered by <u>**multiple** (only one or more than two) **subsets**</u>.

(**Example 2**) One can formulate the **Weighted Set Cover** problem as an **Integer Linear Programming** (ILP) problem:

$$\begin{aligned} &\min \sum_{i=1}^{n} w_i x_i \\ &\text{s.t. } \sum_{u \in S_i} x_i \geq 1 \text{ for each } u \in U, \\ &\quad x_i \in \{0,1\} \text{ for } i \in \{1, 2, \cdots, n\} \end{aligned}$$

which corresponds to the following LP Relaxation:

$$\begin{aligned} &\min \sum_{i=1}^{n} w_i x_i \\ &\text{s.t. } \sum_{u \in S_i} x_i \geq 1 \text{ for each } u \in U. \\ &\quad x_i \geq 0 \text{ for } i \in \{1, 2, \cdots, n\} \end{aligned}$$

Since the Weighted Set Cover problem can be formulated as the equivalent ILP, we can use the **Deterministic Rounding Algorithm** (as for the **Weighted Vertex Cover** problem) to get the approximated solution of **Weighted Set Cover**.

Suppose all n sets contain an element u. One should set the threshold that determines whether to pick a subset to be $1/n$. However, such a strategy has the approximation ratio of n, which isn't good. In this case, the **Deterministic Rounding Algorithm** is not a good alternative for the **Weighted**

Vertex Cover problem.

(**Algorithm 1**) (**Randomized Rounding**) For the **ILP** of the original **Weighted Set Cover** problem, obtain the **optimal solution** $\mathbf{x}=[x_1,x_2,\cdots,x_n]$ to the corresponding **LP-Relaxation**.

Then, pick each **subset** S_i with the corresponding **probability** x_i.

Let C be the collection of the sets picked via **Algorithm 1**. The expected objective value of C is

$$E[cost(C)]=\sum_{i=1}^{n} P(S_i \text{ is picked})\cdot w_i = \sum_{i=1}^{n} w_i x_i = Opt_{LP}(I) \le Opt_{WSC}(I),$$

where $Opt_{LP}(I)$ is the optimal value of the **LP-Relaxation**, while $Opt_{WSC}(I)$ denotes the optimal value of **Weighted Set Cover**. Note that C is the collection of sets given by the **Algorithm 1** (with randomized strategy), which may not be a **valid set cover** (i.e., cover all the elements in U).

To derive the probability that C is a valid set cover, we start from deriving the probability that an element $u \in U$ is covered by C. Assume u is included in k subsets, saying $\{S_1,S_2,\cdots,S_k\}$. Recall S_i is selected with the probability x_i and we have the constraint $x_1+x_2+\cdots+x_k \ge 1$ w.r.t. u. Then, the probability that u is not covered by C (i.e., no subset in $\{S_1,S_2,\cdots,S_k\}$ includes u) is

$$\begin{aligned} P(u \text{ is not covered by } C) &= (1-x_1)(1-x_2)\cdots(1-x_k) \\ &\le (1-1/k)^k \\ &\le 1/e \end{aligned},$$

where we have $(1-1/k)^k \le \lim_{k\to\infty}(1-1/k)^k = 1/e$. Thus, the probability that u is covered by C (i.e., at least one subset in $\{S_1,S_2,\cdots,S_k\}$ includes u) is

$$\begin{aligned} P(u \text{ is covered by } C) &= 1-P(u \text{ is not covered by } C) \\ &= 1-(1-x_1)(1-x_2)\cdots(1-x_k) \\ &\ge 1-(1-1/k)^k \\ &\ge 1-1/e \end{aligned}.$$

The result indicates that each element $u \in U$ is covered by C with a **constant probability**. As there are $|U|$ elements that need to be cover, we would expect **a constant fraction of the elements to be covered** and **a constant fraction of elements not to be covered**. In fact, one can show that this is the situation with '**high probability**'. Especially, '**high probability**' refers to the probability $p \ge 1-\frac{1}{\text{poly(input size)}}$, e.g., $1-\frac{1}{n^2}$ and $1-\frac{1}{2^n}$, etc.

(**Algorithm 3**) (**Revised Randomized Rounding**) For the **ILP** of the original **Weighted Set Cover** problem, obtain the **optimal solution** $\mathbf{x}=[x_1,x_2,\cdots,x_n]$ to the corresponding **LP-Relaxation**.

Then, pick each **subset** S_i with the corresponding **probability** x_i, which derive the collection C.

Let C' be the **union** of such collections C for **multiple runs** of the subset picking procedure. Check whether the following two conditions are satisfied:

(1) *C*' is a **validate set cover** (i.e., all elements are covered by *C*');

(2) the cost of *C*' is at most $cost(C') \le 2^{c-1} c \log |U| \cdot Opt_{LP}(I)$, where *c* is a pre-set parameter and $Opt_{LP}(I)$ denotes the **optimal value** of the **LP-Relaxation**.

If the two conditions are not satisfied, we independently repeat the aforementioned procedure (except solving the LP-Relaxation).

Suppose we **independently** pick *c*log|*U*| such collections *C* and then take their **union** to form a final collection *C*', where *c* is a suitable constant.

In the aforementioned revised algorithm, the probability that an element $u \in U$ is not covered by *C*' (i.e., *u* is not covered in all the *c*log|*U*| iterations) is

$$P(u \text{ is not covered by } C') \le (\frac{1}{e})^{c\log|U|} = (\frac{1}{e^{\log|U|}})^c = \frac{1}{|U|^c}.$$

Then, the probability that *C*' is not a valid set cover is

$$\begin{aligned} P(C' \text{ is not a valid set cover}) &= P(u_1 \text{ is not covered in } C' \text{ OR } u_2 \text{ is not covered in } C' \text{ OR} \cdots) \\ &\le P(u_1 \text{ is not covered in } C') + P(u_2 \text{ is not covered in } C') + \cdots \\ &\le |U| \cdot \frac{1}{|U|^c} = \frac{1}{|U|^{c-1}} \end{aligned},$$

where *c* should be a positive constant larger than 1. The probability that *C*' is a valid set cover is

$$\begin{aligned} P(C' \text{ is a valid set cover}) &= 1 - P(C' \text{ is not a valid set cover}) \\ &\ge 1 - \frac{1}{|U|^{c-1}} \end{aligned},$$

which is a '**high probability**' (with the form of $1 - \frac{1}{\text{poly(input size)}}$).

Furthermore, the expected objective value of *C*' is

$$E[cost(C')] \le c\log|U| \cdot E[cost(C)] = c\log|U| \cdot Opt_{LP}(I) \le c\log|U| \cdot Opt_{WSC}(I).$$

Note that in multiple runs of random selecting procedure, one may obtain duplicated subsets, so $E[C'] \le c\log|U| \cdot E[C]$, but not $E[C'] = c\log|U| \cdot E[C]$. The aforementioned result indicates that the **expected approximation ratio** of **Algorithm 3** is *O*(log|*U*|), i.e.,

$$\frac{E[cost(C')]}{Opt_{WSC}(I)} \le \frac{c\log|U| \cdot Opt_{WSC}(I)}{Opt_{WSC}(I)} = c\log|U| = O(\log|U|).$$

Usually, we have $|U| \ge 2$. For the probability that *C*' is not a valid set cover, we have

$$P(C' \text{ is not a valid set cover}) \le \frac{1}{|U|^{c-1}} \le \frac{1}{2^{c-1}}.$$

By the **Markov's Inequality**, the probability that $cost(C') \ge 2^{c-1} c\log|U| \cdot Opt_{WSC}(I)$ is

$$P[cost(C') \ge 2^{c-1} c\log|U| \cdot Opt_{WSC}(I)] \le \frac{E[cost(C')]}{2^{c-1} c\log|U| \cdot Opt_{WSC}(I)} = \frac{1}{2^{c-1}}.$$

Further, the probability that (i) *C*' is not a valid set or (ii) $cost(C') \ge 2^{c-1} c\log|U| \cdot Opt_{WSC}(I)$ is

$$\begin{aligned}&P(C' \text{ is not a valid set cover OR } cost(C') \geq 2^{c-1} c\log|U|\cdot Opt_{WSC}(I))\\&\leq P(C' \text{ is not a valid set cover}) + P(cost(C') \geq 2^{c-1} c\log|U|\cdot Opt_{WSC}(I))\\&\leq \frac{1}{2^{c-1}}+\frac{1}{2^{c-1}}\\&=\frac{1}{2^{c-2}}\end{aligned}.$$

Thus, the probability that (i) *C*' is a valid set or (ii) $cost(C') \leq 2^{c-1} c\log|U|\cdot Opt_{WSC}(I)$ is

$$\begin{aligned}&P(C' \text{ is a valid set cover AND } cost(C') \leq 2^{c-1} c\log|U|\cdot Opt_{WSC}(I))\\&\geq 1-\frac{1}{2^{c-2}}\end{aligned}.$$

The best lower bound of such a probability is obtained when we set c=3, where we have

$$P(C' \text{ is a valid set cover AND } cost(C') \leq 2^{c-1} c\log|U|\cdot Opt_{WSC}(I)) \geq \frac{1}{2}.$$

In **polynomial time**, we can **verify** the following **two conditions**:

(1) *C*' is a valid cover;

(2) $cost(C') \leq 2^{c-1} c\log|U|\cdot Opt_{LP}(I)$, where we use the **optimal value** of the **LP-Relaxation** $Opt_{LP}(I)$ to be a good lower bound of $Opt_{WSC}(I)$, since we can obtain the optimal solution to the LP-Relaxation in polynomial time.

If in one iteration of **Algorihtm3**, the two conditions are not satisfied, then we repeat the procedure of randomly selecting subsets (based on **x**). The **expected number of repetitions** is 2. In **expected polynomial time**, we will get **a valid set cover** whose **cost** is at most $O(\log|U|)Opt_{WSC}(I)$.

(**Theorem 1**) (**Markov's Inequality**) If x is a non-negative random variable and t>0, then

$$P(x \geq t) \leq \frac{E[x]}{t}.$$

Proof of **Theorem 1**.

$$\begin{aligned}E[x] &= \int_{-\infty}^{\infty} xP(x)dx\\&= \int_{0}^{\infty} xP(x)dx\\&\geq \int_{t}^{\infty} xP(x)dx\\&\geq \int_{t}^{\infty} tP(x)dx\\&= t\int_{t}^{\infty} P(x)dx\\&= tP(x\geq t)\end{aligned}.$$

Thus, we have

$$P(x \geq t) \leq \frac{E[x]}{t}.$$

(**Theorem 2**) If for some constants c, there is a **polynomial** c**-approximation algorithm** for the **Weighted Set Cover** problem, then $P=NP$.

If for some constant $\varepsilon > 0$, there is a polynomial-time $(1-\varepsilon)\ln|U|$-approximation algorithm, then $P=NP$.

(**Example 3**) For the **LP-Relaxation** w.r.t. the **Weighted Set Cover** problem:

$$\begin{aligned}&\min \sum\nolimits_{i=1}^{n} w_i x_i \\ &\text{s.t. } \sum\nolimits_{u \in S_i} x_i \geq 1 \text{ for each } u \in U, \\ &\quad x_i \geq 0 \text{ for } i \in \{1, 2, \cdots, n\}\end{aligned}$$

we have the following **Dual LP**:

$$\begin{aligned}&\max \sum\nolimits_{u \in U} y_u \\ &\text{s.t. } \sum\nolimits_{u \in S_i} y_u \leq w_i \ \text{ for } i \in \{1, 2, \cdots, n\}. \\ &\quad y_u \geq 0\end{aligned}$$

In the aforementioned **Dual LP**, we assign a 'charge' to each element $u \in U$, subject to the condition that for each set S_i, the sum of the charge on the elements in S_i is at most the weight of S_i, i.e., w_i. The goal of **Dual LP** is to <u>maximize the total charge of all the elements</u>.

Especially, the **cost** of any **feasible solution** to the **Dual LP** is a **lower bound** on the weight of the **optimal Set Cover**.

(**Algorithm 4**) (**Greedy Algorithm**)

1: $I \leftarrow \varnothing$
2: **while** there is an element of U that hasn't been covered
3: let D be the set of uncovered elements
4: **for** every set S_i, let $e_i = |D \cap S_i| / w_i$ be the '**cost-effectiveness**' of S_i
5: let S_{i^*} be a set with the highest cost-effectiveness
6: $I \leftarrow I \cup \{i^*\}$
7: return I

(**Theorem 3**) Let n be the number of sets and $m = |U|$. The approximation ratio of **Algorithm 4** is $O(\log m)$.

Proof of **Theorem 3**. Let $u_1, u_2, \cdots, u_m$ be the enumeration of the elements of U in the order where they are covered by **Algorithm 4**. Let c_j be the <u>**cost-effectiveness** of the set S_k that was picked at the step where **Algorithm 4** covers u_j for the **first time**</u>.

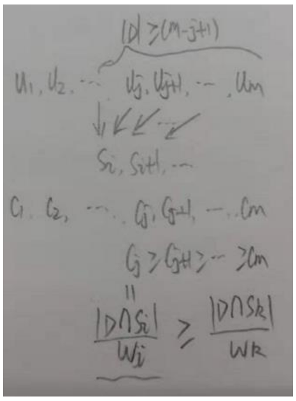
For example, assume that S_k is the set first picked by **Algorithm 4** and there're 4 elements in S_k. Then, we have

$$c_1 = c_2 = c_3 = c_4 = \frac{4}{w_k} \neq c_5.$$

Consider the set D of elements that were uncovered <u>**just before the step** in which we **cover the element** u_j</u>. At this moment, u_j hasn't been covered. Suppose the first set we pick that covers u_j is S_i, there should be multiple elements in S_i (which may include elements $u_{(j-l)}$). Hence, we have

$$|D| \geq (m - j + 1),$$

i.e., D has at least $(m$-j+1) elements.



According to the **greedy strategy** in **Algorithm 4**, we also have

$$c_j = \frac{|D \cap S_i|}{w_i} \geq \frac{|D \cap S_{i+l}|}{w_{i+l}},$$

where $S_{(i+l)}$ is a set picked after S_i. For a set $S_{i\text{-}l}$ picked before S_i, we also have $|D \cap S_{i-l}| = 0$, so

$$c_j = \frac{|D \cap S_i|}{w_i} \geq \frac{|D \cap S_{i-l}|}{w_{i-l}} = 0 \cdot$$

Thus, we can obtain

$$c_j \geq \frac{|D \cap S_k|}{w_k},$$

where S_k represents an **arbitrary set**. We further have

$$|D \cap S_k| \leq c_j w_k,$$

i.e., every set S_k (that hasn't been picked) of weight w_k contains at most $c_j w_k$ elements of D.

Claim: Let $Opt_{WSC}(I)$ be the **optimal value** of the **Weighted Set Cover problem**. We have

$$Opt_{WSC}(I) \geq \frac{(m-j+1)}{c_j} \cdot$$

Proof 1: Since **every set** S_k with weight w_k contains <u>at most $c_j w_k$ elements of D</u> (i.e., $|D \cap S_k| \leq c_j w_k$), $Opt_{WSC}(I)$ must <u>incur at least a cost of</u> $w_k/(c_j w_k) = 1/c_j$ to <u>cover each element of D</u>. Since there are <u>at least $(m\text{-}j+1)$ elements in D</u> (i.e., $|D| \geq (m-j+1)$), we have

$$Opt_{WSC}(I) \geq (m-j+1) \times \frac{1}{c_j} = \frac{(m-j+1)}{c_j} \cdot$$

Proof 2: Without loss of generality, assume that the **optimal set cover** consists of K sets, i.e., $\{S_1, \cdots, S_K\}$. Consider the number of elements in $D \cap S_k$ (for $k \in \{1, \cdots, K\}$), we have

$$(m-j+1) \leq |D \cap S_1| + |D \cap S_2| + \cdots |D \cap S_K| \leq w_1 c_j + w_2 c_j + \cdots + w_K c_j,$$

which indicates that

$$w_1 + w_2 + \cdots + w_K \geq \frac{(m-j+1)}{c_j} \cdot$$

Thus, we have

$$Opt_{WSC}(I) = \sum_{k=1}^{K} w_k \geq \frac{(m-j+1)}{c_j}.$$

Proof 3: (**Using Weak Duality**) For any set S_k with weight w_k, it contains at most $c_j w_k$ elements in D, i.e., $|D \cap S_k| \leq c_j w_k$. Assign $y_{u_1} = y_{u_2} = \cdots = y_{u_{j-1}} = 0$ and $y_{u_j} = y_{u_{j+1}} = \cdots = y_{u_m} = 1/c_j$. The sum of charges of all elements in S_k satisfies:

$$\sum_{u \in S_k} y_u \leq c_j w_k \cdot \frac{1}{c_j} = w_k,$$

which indicates that the aforementioned assignment of **y** is a **feasible solution** to the **Dual LP** (i.e., satisfies all the constraints of the **Dual LP**).

Thus, by weak duality, the cost

$$\sum_{u \in U} y_u = \frac{(m-j+1)}{c_j}$$

provides a lower bound of $Opt_{WSC}(I)$. Namely, we have

$$Opt_{WSC}(I) \geq \frac{(m-j+1)}{c_j}.$$

Claim: The greedy algorithm (**Algorithm 4**) produces a **set cover** with the **cost** $\sum_{j=1}^{m} 1/c_j$.

Proof: At each step, we pick a set S_{i^*} of weight w_{i^*} that covers t elements, so the cost-effectiveness of this set is t/w_{i^*}. For each element u_j covered in this step, we set it **cost-effectiveness** as $c_j = t/w_{i^*}$.

For $c_j = t/w_{i^*}$, we further have $w_{i^*} = t/c_j$. Thus, summing the value $1/c_j$ cover t elements in S_{i^*} gives the weight w_{i^*}. Further, summing the value $1/c_j$ over all the elements in U gives a feasible set cover with cost $\sum_{j=1}^{m} 1/c_j$, which finishes the proof of the claim.

We further finish the proof of **Theorem 3**. Note that we already have

$$Opt_{WSC}(I) \geq \frac{(m-j+1)}{c_j},$$

so we also have

$$\frac{1}{c_j} \leq \frac{Opt_{WSC}(I)}{(m-j+1)}.$$

For all the elements, we have

$$\begin{aligned}\sum_{j=1}^{m} \frac{1}{c_j} &\leq \sum_{j=1}^{m} \frac{Opt_{WSC}(I)}{(m-j+1)} \\ &= Opt_{WSC}(I) \sum_{j=1}^{m} \frac{1}{(m-j+1)} \\ &= Opt_{WSC}(I) \cdot [\frac{1}{m} + \frac{1}{m-1} + \cdots + 1] \\ &= Opt_{WSC}(I)[\ln m + O(1)]\end{aligned}.$$

Note that

$$H_m = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{m} = \ln m + O(1)$$

is called as the **Harmonic series**, which can be proved to have the value closed to ln$m$.

In summary, the **approximation ratio** of **Algorithm 4** is

$$\frac{\sum_{j=1}^{m} 1/c_j}{Opt_{WSC}(I)} \leq \frac{Opt_{WSC}(I) \cdot H_m}{Opt_{WSC}(I)} = H_m = O(\log m) \cdot$$