Network Design and Game Theory
Spring 2008
Lecture 1

Instructor: Mohammad T. Hajiaghayi
Scribe: Ricardo A. Collado

January 28, 2008

1 Overview

We define and explore three classical problems: Set Cover, Unique Coverage, and Maximum Coverage problems.

2 Definitions

(1) **SET COVER**

Input : A universe set $U = \{e_1, \ldots, e_n\}$, a collection $\mathcal{S} = \{S_1 \ldots S_k\}$ of subsets of U such that $U = \bigcup\{S_1, \ldots, S_k\}$, and a cost function $\mathsf{cost} : \mathcal{S} \to \mathbb{Q}^+$.

Goal: Find a minimum cost $\mathcal{S}' \subseteq \mathcal{S}$ such that S' covers U, i.e. $U = \bigcup_{S \in \mathcal{S}'} S$.

(2) **UNIQUE COVERAGE**

Input : A universe set $U = \{e_1, \ldots, e_n\}$ and a collection $\mathcal{S} = \{S_1 \ldots S_k\}$ of subsets of U such that $U = \bigcup\{S_1, \ldots, S_k\}$.

Goal: Find $\mathcal{S}' \subseteq \mathcal{S}$ that maximizes the number of uniquely covered elements.

(3) **MAXIMUM COVERAGE**

Input : A universe set $U = \{e_1, \ldots, e_n\}$ and a collection $\mathcal{S} = \{S_1 \ldots S_k\}$ of subsets of U such that $U = \bigcup\{S_1, \ldots, S_k\}$. Also, cost and weight functions $\mathsf{cost} : \mathcal{S} \to \mathbb{Q}^+$ and $w : U \to \mathbb{Q}^+$ and a bound $L \in \mathbb{Q}^+$.

Goal: Find $\mathcal{S}' \subseteq \mathcal{S}$ such that the total cost of its sets is at most L while maximizing the total weight of covered elements.

These three problems are NP-complete and can be polynomialy reduced to one another. For example, any instance of Set Cover can be solved with polynomialy many calls to Maximum Coverage by assigning a weight of 1 to each element of U and solving Maximum Coverage for increasing L's until the minimum L that gives a cover of U is found.

Wireless networking has many applications of these problems. For example, suppose that there are different options of where to build transmission towers and each location has its range and cost. Set cover can be used to find a network of towers that cover everything and has minimum cost. If a limiting budget is added then Maximum Coverage could be used to find a network that covers the maximum within the budget's constraints. Also, the Unique Coverage could be used to minimize interference among the transmission towers.

3 Set Cover

Consider an instance of Set Cover with universe U and collection S.

Definition 1 *The* frequency *of* $e_i \in U$ *is the number of sets from* S *that contain* e_i*. Let* F *be the maximum frequency of all elements of* U*.*

Next we will see three approximation algorithms to Set Cover and analyze their approximation ratio.

The next algorithm assumes that the cost of all sets is 1.
Greedy F Approximation Algotithm:

1: Set $A \leftarrow \emptyset$, $U' \leftarrow \emptyset$.

2: While U has an element e not covered by A add all sets containing e to A and add e to U'.

Output: Cover A.

Analysis The number of sets chosen is at most $F|U'|$ and no set can cover two elements of U'. Therefore

$$\mathsf{OPT} \geq |U'| \geq \frac{1}{F}|A|$$

and we conclude that **Greedy** F is an F approximation algorithm.

The next algorithm uses linear programming to find and approximation to Set Cover.
LP-Rouding Approximation Algorithm:

1. Set $A \leftarrow \emptyset$

2. Solve the following linear program in polynomial time

$$\begin{array}{lll} \text{minimum:} & \sum_{j=1}^{|\mathcal{S}|} \mathsf{cost}(S_j) \cdot x_j & \\ \text{subject to} & \sum\{x_j \mid e_i \in S_j\} \geq 1, & i = 1, \ldots, |U| \\ & x_i \geq 0, & i = 1, \ldots, |\mathcal{S}| \end{array} \tag{1}$$

3. In the optimal solution to (1) pick all variables x_i's with value at least $\frac{1}{F}$ and add the corresponding S_i's to A.

Otput: Cover A.

Remark The solution of (1) plus the constraints $x_i \in \{0, 1\}, i = 1, \ldots, |\mathcal{S}|$ is exactly the solution of Set Cover. Therefore (1) is a linear programming relaxation of Set Cover.

Analysis
Let x^* be the optimal solution to the LP relaxation obtained in step 2 of the algorithm.

(i) Suppose that $e \in U$ is not covered by A. Then

$$\sum\{x_j^* \mid e \in S_j\} < \sum\{\frac{1}{F} \mid e \in S_j\} \leq F\frac{1}{F} = 1$$

which contradicts the feasibility of x^*. Therefore A is a cover of U.

(ii) Let $\mathsf{cost}(\mathsf{OPT}), \mathsf{cost}(\mathsf{LP})$ be the cost of the optimal solution and the cost of the solution to the LP relaxation obtained in step 2, respectively. Then

$$\begin{aligned} \mathsf{cost}(\mathsf{OPT}) &\geq \mathsf{cost}(\mathsf{LP}) \\ &= \sum_{j=1}^{|\mathcal{S}|} \mathsf{cost}(S_j) \cdot x_j^* \\ &= \sum_{S \in A} \mathsf{cost}(S) \cdot x_S^* + \sum_{Z \notin A} \mathsf{cost}(Z) \cdot x_Z^* \\ &\geq \sum_{S \in A} \mathsf{cost}(S) \cdot x_S^* \\ &\geq \frac{1}{F} \sum_{S \in A} \mathsf{cost}(S). \end{aligned}$$

Therefore $\mathsf{cost}(\mathsf{OPT}) \geq \frac{1}{F}\mathsf{cost}(A)$, where A is the cover given by the algorithm, so **LP-Rounding** is an F approximation algorithm.

Greedy $O(\lg n)$ Approximation Algorithm:

1. Set $A \leftarrow \emptyset, C \leftarrow \emptyset$.

2. While $C \neq U$ do:
   (a) Find a set S which is most cost effective: i.e. C minimizes
   $$\alpha = \frac{\mathsf{cost}(S)}{S - C},$$
   (b) add S to A and for each $e \in S - C$ set $\mathsf{price}(e) = \alpha$, and
   (c) set $C \leftarrow C \cup \{S\}$.

Output: Cover A.

Analysis
(i) Let $\{S_1, \ldots, S_k\}$ be an optimal solution to Set Cover at iteration t. Then

$$\begin{aligned}
\mathsf{cost}(\mathsf{opt}_t) &= \sum_{i=1}^{k} \mathsf{cost}(S_i) \\
&= \sum_{i=1}^{k} \frac{\mathsf{cost}(S_i)}{|S_i|}|S_i| \\
&\geq \sum_{i=1}^{k} \alpha_t |S_i| \\
&\geq \alpha_t |U - C|.
\end{aligned}$$

Therefore at iteration t
$$\alpha_t \leq \frac{\mathsf{cost}(\mathsf{OPT}_t)}{|U - C|}.$$

(ii)

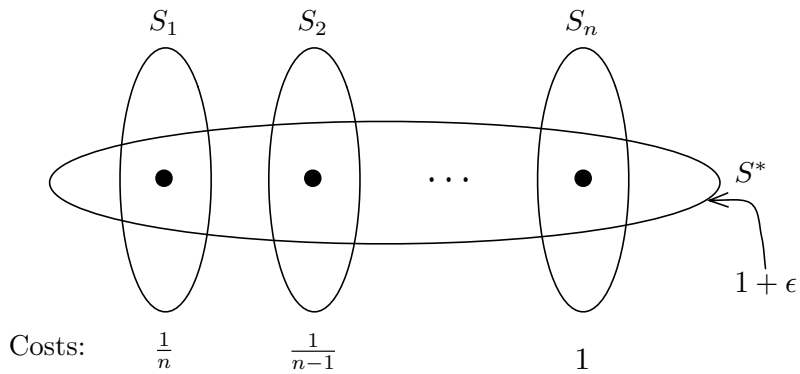
$$\begin{aligned}
\mathsf{cost}(\mathsf{ALG}) &= \sum_{e \in U} \mathsf{price}(e) \\
&= \sum_{i=1}^{n} \mathsf{price}(e_i) \\
&\leq \sum_{i=1}^{n} \frac{\mathsf{cost}(\mathsf{OPT}_i)}{|U - C|} \\
&\leq_{(*)} \sum_{i=1}^{n} \frac{\mathsf{cost}(\mathsf{OPT}_i)}{n - i + 1} \\
&\leq \mathsf{cost}(\mathsf{OPT}) \sum_{i=0}^{n} \frac{1}{i} \\
&= \mathsf{cost}(\mathsf{OPT}) \lg(n),
\end{aligned}$$

where $\leq_{(*)}$ is obtained because by definition of e_i at the iteration in which e_i is added $|U - C| \geq n - i + 1$. Therefore

$$\mathsf{cost}(\mathsf{OPT}) \geq \frac{1}{\lg(n)}\mathsf{cost}(\mathsf{ALG})$$

and **Greedy** $O(\lg n)$ is an $O(\lg n)$ approximation algorithm.

By considering the following example we see that the previous analysis is tight:



Here OPT chooses S^* while **Greedy** $O(\lg n)$ picks $S_1, S_2, \ldots, S_n$. The cost of OPT is $1 + \epsilon$ while the cost of **Greedy** $O(\lg n)$ is $\lg n$.

Important theorem to remember:

Theorem 1 (Ferge) *There is no* $(1 - \epsilon)\lg n$ *approximation algorithm to* SET COVER *unless* $\mathsf{NP} \subseteq \mathsf{DTIME}\left(n^{\lg \lg n}\right)$.