CS880: Approximations Algorithms

Scribe: Matt Elder **Lecturer:** Shuchi Chawla
Topic: Greedy Approximations: Set Cover and Min Makespan **Date:** 1/30/06

3.1 Set Cover

The *Set Cover* problem is: Given a set of elements $E = \{e_1, e_2, \ldots, e_n\}$ and a set of m subsets of $E, \mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, find a "least cost" collection C of sets from $\mathcal{S}$ such that C covers all elements in E. That is, $\cup_{S_i \in C} S_i = E$.

Set Cover comes in two flavors, unweighted and weighted. In unweighted Set Cover, the cost of a collection C is number of sets contained in it. In weighted Set Cover, there is a nonnegative weight function $w : \mathcal{S} \to \mathbb{R}$, and the cost of C is defined to be its total weight, i.e., $\sum_{S_i \in C} w(S_i)$.

First, we will deal with the unweighted Set Cover problem. The following algorithm is an extension of the greedy vertex cover algorithm that we discussed in Lecture 1.

Algorithm 3.1.1 Set Cover(E, $\mathcal{S}$):

1. $C \leftarrow \emptyset$.
2. While E contains elements not covered by C:
   (a) Pick an element $e \in E$ not covered by C.
   (b) Add all sets S_i containing e to C.

To analyze Algorithm 3.1.1, we will need the following definition:

Definition 3.1.2 *A set E' of elements in E is* independent *if, for all $e_1, e_2 \in E'$, there is no $S_i \in C$ such that $e_1, e_2 \in S_i$.*

Now, we shall determine how strong an approximation Algorithm 3.1.1 is. Say that the *frequency* of an element is the number of sets that contain that element. Let F denote the maximum frequency across all elements. Thus, F is the largest number of sets from $\mathcal{S}$ that we might add to our cover C at any step in the algorithm. It is clear that the elements selected by the algorithm form an independent set, so the algorithm selects no more than $F|E'|$ elements, where E' is the set of elements picked in Step 2a. That is, ALG $\leq F|E'|$. Because every element is covered by some subset in an optimal set cover, we know that $|E'| \leq$ OPT for any independent set E'. Thus, ALG $\leq F$ OPT, and Algorithm 3.1.1 is therefore an F–approximation.

Theorem 3.1.3 *Algorithm 3.1.1 is an F–approximation to Set Cover.*

Algorithm 3.1.1 is a good approximation if F is guaranteed to be small. In general, however, there could be some element contained in every set of $\mathcal{S}$, and Algorithm 3.1.1 would be a very poor approximation. So, we consider a different unweighted Set Cover approximation algorithm which uses the greedy strategy to yield a $\ln n$–approximation.

Algorithm 3.1.4 Set Cover(E, $\mathcal{S}$):

1. $C \leftarrow \emptyset$.
2. While E contains elements not covered by C:
   (a) Find the set S_i containing the greatest number of uncovered elements.
   (b) Add S_i to C.

Theorem 3.1.5 *Algorithm 3.1.4 is a* $\ln \frac{n}{OPT}$*–approximation.*

Proof: Let $k = \text{OPT}$, and let E_t be the set of elements not yet covered after step i, with $E_0 = E$. OPT covers every E_t with no more than k sets. ALG always picks the largest set over E_t in step $t+1$. The size of this largest set must cover at least $|E_t|/k$ in E_t; if it covered fewer elements, no way of picking sets would be able to cover E_t in k sets, which contradicts the existence of OPT. So, $|E_{t+1}| \leq |E_t| - |E_t|/k$, and, inductively, $|E_t| \leq n\,(1 - 1/k)^t$.

When $|E_t| < 1$, we know we are done, so we solve for this t:

$$\begin{aligned} &\left(1 - \frac{1}{k}\right)^t < \frac{1}{n} \\ \Rightarrow\ & n < \left(\frac{k}{k-1}\right)^t \\ \Rightarrow\ & \ln n \leq t \ln\left(1 + \frac{1}{k-1}\right) \approx \frac{t}{k} \\ \Rightarrow\ & t \leq k \ln n = \text{OPT} \ln n. \end{aligned}$$

Algorithm 3.1.4 finishes within $\text{OPT} \ln n$ steps, so it uses no more than that many sets. We can get a better analysis for this approximation by considering when $|E_t| < k$, as follows:

$$\begin{aligned} &n\left(1 - \frac{1}{k}\right)^t = k \\ \Rightarrow\ & n\frac{1}{e^{t/k}} = k \text{ (because } (1-x)^{1/x} \leq \frac{1}{e} \text{ for all } x). \\ \Rightarrow\ & e^{t/k} = \frac{n}{k} \\ \Rightarrow\ & t = k \ln \frac{n}{k}. \end{aligned}$$

Thus, after $k \ln \frac{n}{k}$ steps there remain only k elements. Each subsequent step removes at least one element, so $\text{ALG} \leq \text{OPT}\left(\ln \frac{n}{OPT} + 1\right)$. ■

Theorem 3.1.6 *If all sets are of size* $\leq B$*, then there exists a* $(\ln B + 1)$*–approximation to unweighted Set Cover.*

Proof: If all sets have size no greater than B, then $k \geq n/B$. So, $B \geq n/k$, and Algorithm 3.1.4 gives a $(\ln B + 1)$–approximation. ■