Advanced Algorithms - Lecture November 10, 2005

Lecture 2

Lecturer: Yair Bartal *Scribe: Yuri Kosharovski*

1 Weighted Set Cover

1.1 Definitions

Definition 1 *Weighted Set Cover:*
Universe of n members U=1...n.
$S = \{s_1 ... s_t\}, \quad s_i \in U, \quad \bigcup_i = U$
$c : S \rightarrow \Re^+$
We are to find a cover $C \subseteq S$ *i.e* $\bigcup_{s \in C} \{s\} = U$
Target: minimize C's weight i.e minimize $\sum_{s \in C} [c(s)]$

Remark We can examine the problem of Weighted Vertex Cover as a private case of WSC in the following way:
$\{member \leftrightarrow edge\}$ $\{set \leftrightarrow vertex\}$
and each member should be in exactly 2 sets

Definition 2 *frequency function:*
for each $u \in U$, $f(u) := \{$ *Number of sets A in S* $| u \in A\}$
F:=$\max\{f(u)\}_{u \in U}$

Theorem 3 *The approximation relation of WSC is:* $\min\{F, \ln(n) + 1\}$.

The following two section will prove the theorem.

1.2 Greedy Algorithm for WSC

In each step there is a group A⊆U of covered members.

Definition 4 *For each set s, $S \ni s \subsetneq A$, lets define:*

$$\gamma(s) = \frac{c(s)}{|s - A|} = price(e)$$

$e \in S - A$ *i.e* e *is a member that will be added (covered) to A, if we add* s *to A.*

Algorithm:
1) A $\leftarrow \emptyset$
2) choose an s with minimal $\gamma(s)$
3) $A \leftarrow A \bigcup s$
4) if A is *not* a cover, go back to (2)
5) return C = A

Lemma 5 *For each* $1 \leqslant k \leqslant n$

$$price(e_k) \leqslant \frac{OPT}{n - k + 1}$$

$e_1 ... e_k$ *is the order in which the members were added to A.*

Proof: Lets examine the optimal cover C* = $\{T_1 ... T_L\}$ $T_i \in S$.

Definition 6

$$J := \{1 \leq i \leq L \quad | \quad T_i \bigcap \overline{A} \neq \emptyset\}$$

i.e J is a set of indexes i, so that in a given time, every T_i can be added to the current A.

$$OPT = \sum_{1 \leq i \leq L} \left[c(T_i) \right] \quad \geq \quad \sum_{i \in J} \left[|T_i - A| \underbrace{\frac{c(T_i}{|T_i - A|}} \right]$$

.
.
.

$\gamma(T_i) \geq \gamma(S) = price(e_k)$ S is the set that *contains* e_k and it was chosen because its γ is *minimal*

$$\geq \sum_{i \in J} |T_i - A| price(e_k) \quad \geq \quad |\overline{A} * |price(e_k| \quad \geq \quad (n - k + 1) * price(e_k)$$

.
.

$\bigcup_{i \in J} \left(T_i - A \right)$

we already have done k-1 steps and added 1 member in each step ■

Theorem 7 .
The approximation relation of the greedy algorithm is H_n *(*$=1+\frac{1}{2}+\frac{1}{3}+...+\frac{1}{n} \leqslant \ln n+1$*)*

Proof: Lets mark with A_s the set A when the set s was added to it.
$\Rightarrow \; GREEDY(I) = \sum_{s\in C} c(s) = \sum_{s\in C} 1 * c(s) = \sum_{s\in C} |S - A_s| \frac{c(s)}{|S-A_s|} = \sum_{s\in C} \sum_{e\in S-A_s} price(e) = \sum_{e\in U} price(e)$

and according to the lemma:

$\leqslant \frac{OPT}{n} + \frac{OPT}{n-1} + ... + \frac{OPT}{1} = OPT * H_n$ ■

1.3 Linear programming

ILP - Integer Linear Programming
Each $s \in S$ will have a variable $X_s \in \{0,1\}$ which defines if s is in the cover.
$\Rightarrow$ The cost of the problem is: $\sum_{s\in S} c(s) * X_s$ and this is what we strive to minimize. Thus, we can write the problem in the following way:

$$\forall e \in U \quad \sum_{s|e\in s} X_s \geq 1 \quad X_s \in \{0,1\}$$

$$\min \sum_{s\in S} c(s) * X_s$$

ILP relaxation: instead of $X_s \in \{0,1\}$ we requre that $0 \leq X_s \leq 1$. This weakens the demand and the optimal solution OPT_F for this problem can be a lower bound for the original problem.