CSE 533: The PCP Theorem and Hardness of Approximation **(Autumn 2005)**

Lecture 14: Label Cover Hardness: Application to Set Cover

Lecturer: Venkat Guruswami *Scribe: Prasad Raghavendra*

In the previous lecture, we have seen the complete proof of NP hardness of GapLabelCover Problem. From this point onwards, we shall use the results derived so far to obtain hardness of approximation results for some important problems. Towards this, we start with the problem of Set-Cover in this lecture. However, since the hardness of GapLabelCover will be used in showing so many inapproximability results, we take a closer look at the various parameters of the problem.

1 Label Cover Problem

Definition 1.1. ***Label Cover Instance:***
A label cover instance consists of $(G = (V_1, V_2, E), \Sigma, \Pi)$ *where*

- G *is a bipartite graph between vertex sets* V_1 *and* V_2 *and an edge set* E*.*
- G *is left and right regular. Denote by* D_1 *and* D_2 *the degrees of vertices in* V_1 *and* V_2 *respectively.*
- *Each vertex in* $V_1 \cup V_2$ *is to be assigned a label from the alphabet* Σ*.*
- *For each edge* e*, there is a constraint* Π_e *which is a function from* Σ *on to itself. Thus the set of all constraints in* G *are*
$$\Pi = \{\Pi_e : \Sigma \to \Sigma | e \in E\}$$
The property that each constraint is a function from Σ *on to itself, is also referred to as the Projection Property.*

A labelling of the graph, is a mapping $\sigma : V \to \Sigma$ *which assigns a label for each vertex of* G*. A labelling* σ *is said to satisfy an edge* $e = (u, v)$ *if and only if*

$$\Pi_e(\sigma(u)) = \sigma(v)$$

Let us define the following 'Promise' of the Label-Cover Problem.

Definition 1.2. $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ *Given an instance* G *of Label Cover such that either one of the following is true*

- *There exists a labelling* σ *such that it satisfies all the edges* e *in* G*.*
- *For any labelling* σ *of the vertices, not more than* $\epsilon|E|$ *edges are satisfied by* σ*.*

The computational problem is to determine which of the above two cases hold for G.

We call the above problem, a *Promise* problem because the input is promised to fall in to either one of the two categories. Observe that, we do not care about the output if the input does not fall in to either of the two categories. It is easy to check that, that if $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ is NP-hard then approximating $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ within a factor of ϵ is NP-hard.

In the previous class, we have shown the following theorem.

Theorem 1.3. *For all $\epsilon > 0$ there exists a constant $|\Sigma|$ such that $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ is NP-hard.*

The above theorem essentially implies that: Given a $\mathrm{E3SAT}$ formula, there is a polynomial time reduction that outputs a graph G such that

- If the $\mathrm{E3SAT}$ is satisfiable then there is a labelling that satisfies all edges in G.
- If the $\mathrm{E3SAT}$ is not satisfiable, then no labelling of G satisfies more than ϵ fraction of edges in G.

Observe that our reduction implies something stronger than what is stated in 1.3. For instance, in the graph G produced by the reduction, the degree of a node is independent of n. Therefore let us have a closer look at the various parameters of the graph G produced.

Remember that the in-order to obtain the above reduction, we first reduced $\mathrm{E3SAT}$ to $\mathrm{GapLabelCover}_{1,\mathrm{c}}$ for some constant c. This reduction, was a polynomial time reduction, with the left and the right degrees fixed constants independent of the size of the $\mathrm{E3SAT}$ instance. We then reduced the soundness c to ϵ by using some sort of parralel repetition. The size of the graph produced, and the time taken in this step depends on the value of the ϵ. Therefore, our choice of the Parralel Repetition theorem, dictates the size of the graph produced and the time taken to produce it. In the table below, we summarize the values of the various parameters, for two different parralel Repetition theorems.

Result	Alphabet Size $\lvert\Sigma\rvert$	Left Degree D_1	Right Degree D_2	Number of vertices	Reduction Time
Feige/Kilian[1]	$(2^{\frac{1}{\epsilon}})^{O(1)}$	$(2^{\frac{1}{\epsilon}})^{O(1)}$	$(2^{\frac{1}{\epsilon}})^{O(1)}$	$(n^{\frac{1}{\epsilon}})^{O(1)}$	$(n^{\frac{1}{\epsilon}})^{O(1)}$
Raz[3]	$(\frac{1}{\epsilon})^{O(1)}$	$(\frac{1}{\epsilon})^{O(1)}$	$(\frac{1}{\epsilon})^{O(1)}$	$n^{O(\log \frac{1}{\epsilon})}$	$n^{O(\log \frac{1}{\epsilon})}$

It is apparent from the table that, for a given ϵ, the size of the graphs produced by using Raz's parralel repetition theorem are much smaller. This means that, the gaps that can be produced by Raz's theorem in a fixed polynomial time are much higher, and larger gaps usually translate to better inapproximability results. For the purposes of this lecture, we will be using the $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ instance obtained using Raz's parralel repetition theorem.

However there is one technical point that needs clarification. Observe that the reduction from $\mathrm{E3SAT}$ to $\mathrm{LabelCover}$ used in the proof of theorem1.3, does not produce a graph that is right regular. In fact, the degree of a vertex corresponding to a variable, is equal to the number of different clauses in which the variable appears. Fortunately, there is a very easy way to obtain,

a graph that is right-regular. Instead of creating, a $\mathrm{GapLabelCover}_{1,c}$ instance starting from a $\mathrm{GapE3SAT}$ instance, we start with $\mathrm{GapE3SAT}(5)$ instance. A $\mathrm{GapE3SAT}(5)$ is nothing but the $\mathrm{GapE3SAT}$ instance with the additional constraint that each variable occurs in exactly 5 clauses. Therefore, the right degree of the graph G if there are k-parralel repetitions, is given by 5^k.

Note that the number of times we perform parralel repetition, can be function of n, and still the above results continue to hold. So choosing $\epsilon = \frac{1}{\log^3 n}$, we get a reduction from $\mathrm{E3SAT}$ instances of size n to $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ instances of size $O(n^{O(\log\log n}))$. Hence a polynomial time algorithm for $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ will imply a $O(n^{O(\log\log n)})$ algorithm for $\mathrm{E3SAT}$. Therefore we conclude the following theorem

Theorem 1.4. *The problem* $\mathrm{GapLabelCover}_{1,\frac{1}{\log^3 \mathrm{n}}}$ *is not in* P *unless* $\mathrm{NP} \in \mathrm{DTIME}(n^{O(\log\log n)})$

The $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ problem is in some sense the mother of all inapproximability results for NP-hard problems. In other words, it occupies the same distinction that the problem $\mathrm{E3SAT}$ does for proving NP hardness of problems. By now, a large number of inapproximability results use $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ as the starting point for the reduction. This is probably because of the following reasons

- The constraints $\pi_e(a) = b$, in $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ are very simple, and are natural to model using gadgets in several contexts
- It also helps to have arbitrarily large gap $\frac{1}{\epsilon}$, in the input to a reduction.
- Prior success using $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$ for reduction, prompts one to attempt a reduction from $\mathrm{GapLabelCover}_{1,\epsilon}(\Sigma)$. Thus with more attempts to use, $GAPLC$, there could have been more reductions.

2 Set Cover

The set cover problem, is one of the classic NP-hard problems, for which an approximation algorithm was obtained. In this lecture, we will be dealing with the unweighted version of Set Cover.

Definition 2.1. *Unweighted Set Cover*
Given a universe U with $|U| = n$, and sets $S_1 \ldots S_m \subset U$, such that

$$\bigcup_i S_i = U$$

The computational task is to find a set of indices $I \in \{1 \ldots M\}$ with the minimum cardinality such that

$$\bigcup_{i \in I} S_i = U$$

2.1 Algorithm

A simple greedy algorithm produces the optimal approximation for Set-Cover. Let U' - denote the set of elements yet to be covered. Let I_C - denote the set of indices i for which the set S_i is already chosen to the cover.

Greedy Algorithm

while U' not empty

- Find the set S_i such that $|S_i \bigcap U'|$ is maximized.
- Add S_i to the set cover, and $U' = U' - S_i$

Output the cover I_C

For the above algorithm, a clever analysis yields the following approximation result.

Theorem 2.2. *The Greedy Algorithm is a* $\ln n$ *approximation algorithm, where* n *is the size of the universe* U.

3 Inapproximability of Set Cover

In this section, we will show the following inapproximability result for SET-COVER.

Theorem 3.1. *There exists* $c > 0$*, such that no polynomial time* $c \log N$ *approximation algorithm exists for SET COVER unless* $\mathrm{NP} \subset \mathrm{DTIME}(n^{O(\log \log n)})$.

The above theorem implies that the greedy algorithm achieves the optimal inapproximability results except up to constant factors. Infact, it can been shown that the greedy algorithm achieves the exact optimal approximation ratio, in other words the following theorem has been shown in [2]

Theorem 3.2. *For every* $\epsilon > 0$*, there is no* $(1 - \epsilon) \ln N$ *approximation algorithm exists for SET COVER unless* $\mathrm{NP} \subset \mathrm{DTIME}(n^{O(\log \log n)})$.

In order to show theorem 3.1 we will be making a gadget reduction from the Label-Cover to Set cover problem. Essentially for each edge we need a gadget to check a constraint like $\pi_e(a) = b$. The result of the reduction is an instance of the Set-Cover problem. Therefore, in some sense we want a collection of sets such that if the constraint for the edge is satisfied, then there is a small set cover and vice versa. The gadgets that fit this requirement are called (m, l) set systems which we describe below.

Definition 3.3. *An* (m, l)*-set system consists of a universe* B *and collection of subsets* $\{C_1, \ldots, C_m\}$ *such that: If the Union of any collection of* l*-sets in* $\{C_1, \ldots, C_m, \overline{C_1}, \ldots, \overline{C_m}\}$ *is* B*, then the collection must contain both* C_i *and* $\overline{C_i}$ *for some* i.

From the definition of an (m, l)-set system, it is not apparent that they can be constructed efficiently. Fortunately there are constructions of (m, l)-set systems, whose size and time to construct are reasonably bounded. For the purposes of this lecture, we will assume that the following theorem is true. We will see a proof-sketch in the coming lecture.

Theorem 3.4. *An (m, l)-set system with a universe size $|B| = O(2^{2l}m^2)$ exists, and can be constructed in $2^{O(l)}m^{O(1)}$ time.*

Using the gadgets just described, we show a reduction from Label-Cover to Set-Cover below.

Label Cover Instance:

$$(G = (V_1, V_2, E), \Sigma, \Pi)$$

The alphabet consists of $\Sigma = \{1, \ldots, m\}$, and thus $|\Sigma| = m$

Set Cover Instance:
Let B be a (m, l) set system. The universe for the set cover instance consists of $E \times B$. Define the following subsets of $E \times B$

- For all vertices $v \in V_2$, and $x \in \Sigma$, define the subset $S_{v,x} \subset E \times B$ as follows

$$S_{v,x} = \bigcup_{e \ni v} \{e\} \times C_x$$

- For all vertices $u \in V_1$, and $y \in \Sigma$, define the subset $S_{v,y} \subset E \times B$ as follows

$$S_{u,y} = \bigcup_{e \ni u} \{e\} \times \overline{C_{\pi_e(y)}}$$

The notation $e \ni u$ denotes that the edge e is incident at vertex u.
The Set Cover Instance produced by the reduction consists of

$$\big(E \times B, \{S_{w,x} | w \in V_1 \cup V_2, x \in \Sigma\}\big)$$

Lemma 3.5. *(Completeness) If the LabelCover instance G, has a labelling that satisfies all the edges $e \in E$, then the Set-Cover instance produced has a set cover of size $|V_1| + |V_2|$*

Proof: Let $\sigma : V_1 \cup V_2 \to \Sigma$ denote a labelling for G that satisfies all the edges E. Pick the following set of sets $\mathcal{S} = \{S_{w,\sigma(w)} | w \in V_1 \cup V_2\}$. The number of sets in $\mathcal{S}$ is $|V_1 + V_2|$. We claim that $\mathcal{S}$ is a valid set cover for $E \times B$. Towards this, we show the following for every edge $e = (u, v)$

$$\{e\} \times B \subset S_{u,\sigma(u)} \cup S_{v,\sigma(v)} \tag{1}$$

By definition of $S_{u,\sigma(u)}, S_{v,\sigma(v)}$ we have

$$S_{v,\sigma(v)} \supseteq \{e\} \times C_{\sigma(v)} \tag{2}$$

$$S_{u,\sigma(u)} \supseteq \{e\} \times \overline{C_{\pi_e(\sigma(u))}}$$

But since σ satisfies all the edges, and in particular the edge e, we have $\pi_e(\sigma(u)) = \sigma(v)$. Therefore, we can write,

$$S_{u,\sigma(u)} \supseteq \{e\} \times \overline{C_{\pi_e(\sigma(u))}} = \{e\} \times \overline{C_{(\sigma(v)}} \tag{3}$$

From equations 2 and 3, we can conclude equation 1. Further taking union of equation 1, over all edges e, we get

$$E \times B \subseteq \bigcup_{u \in V_1 \cup V_2} S_{u,\sigma(u)}$$

This concludes the proof □

Lemma 3.6. *(Soundness) If the LabelCover instance G, has no labelling that satisfies more than $\frac{2}{l^2}$ fraction of the edges, then the Set-Cover instance has no set cover of size $\leq \frac{l}{8}(|V_1| + |V_2|)$*

Proof: We will prove the contrapositive of the above lemma. Suppose, there is a set cover $\mathcal{S}$ with $|\mathcal{S}| < \frac{l}{8}(|V_1| + |V_2|)$, then for each vertex w define the set of labels

$$L_w = \{c \in \Sigma | S_{w,c} \in \mathcal{S}\}$$

L_w is in some sense the set of all labels, that the set cover solution 'assigns' to vertex w. Therefore the total number of sets chosen in the cover $\mathcal{S}$ is given by

$$\sum_{w \in V_1 \cup V_2} |L_w| = |\mathcal{S}|$$

Therefore the average cardinality of L_w satisfies

$$\frac{\sum_{w \in V_1 \cup V_2} |L_w|}{|V_1| + |V_2|} = \frac{\mathcal{S}}{8} \leq \frac{l}{8}$$

For atleast $\frac{3}{4}$ of the vertices $w \in V_1 \cup V_2$ we have $|L_w| \leq \frac{l}{2}$. Otherwise, there will be more than $\frac{1}{4}$ of vertices with $|L_w| > \frac{l}{2}$, thus making the sum $\sum_w |L_w| > \frac{1}{4} \times \frac{l}{2}$, a contradiction.

Observe that by regularity, atleast $\frac{1}{2}$ of the edges have both endpoints (u, v) with $|L_u| < \frac{l}{2}$ and $|L_v < \frac{l}{2}$.

Let us call an edge $e = (u, v)$ to be *Frugally Covered* if it satisfies $|L_u| < \frac{l}{2}$ and $|L_v| < \frac{l}{2}$. Now we will obtain a labelling σ for G as follows :
"For each vertex $w \in V_1 \cup V_2$, define $\sigma(w) = x$ where x is uniformly randomly chosen from L_w."

We will show the following fact about σ, which will complete the proof □

Fact 3.7. *For the labelling σ obtained by the above random experiment, the expected fraction of the edges satisfied is atleast $\frac{2}{l^2}$.*

Proof: We show that each Frugally-Covered edge is satisfied by σ with probability $\geq \frac{4}{l^2}$. Since there are more than $\frac{|E|}{2}$ Frugally-Covered edges, the expected fraction of edges satisfied is atleast $\frac{1}{2} \times \frac{4}{l^2} = \frac{2}{l^2}$

Let $e = (u, v)$ be a Frugally-Covered edge. Let $L_u = \{a_1, \ldots, a_p\}$ and $L_v = \{b_1, \ldots, b_q\}$ As e is Frugally-Covered, p and q are less than $\frac{l}{2}$.

The sets in $\mathcal{S}$, completely cover $E \times B$, and in particular they cover $e \times B$. Note, that for any vertex w other than u,v we have $|S_{w,x} \cap \{e \times B\}| = 0$ for all $x \in \Sigma$. In other words, no element of the set $e \times B$ can be covered by any of the sets $S_{w,x}$ for any vertex w other than u, v. Therefore the set $e \times B$ is covered by the sets chosen for vertices u and v.

$$\{e\} \times B \subseteq (\bigcup_{i=1}^{p} S_{u,a_i}) \cup (\bigcup_{j=1}^{q} S_{v,b_j})$$

But note that by definition of $S_{v,x}$ its interesection with $\{e\} \times B$ is $\{e\} \times C_x$. Similarly intersection of $S_{u,y}$ with $\{e\} \times B$ is $\{e\} \times \overline{C}_{\pi_e(y)}$. Restricting the sets S_{u,a_i} and S_{v,b_j} to $\{e\} \times B$ in the above containment we get

$$\{e\} \times B \subseteq (\{e\} \times \bigcup_{i=1}^{p} \overline{C}_{\pi(a_i)}) \cup (\{e\} \times \bigcup_{j=1}^{q} C_{b_j})$$

Therefore we get

$$B \subseteq (\bigcup_{i=1}^{p} \overline{C}_{\pi(a_i)}) \cup (\bigcup_{j=1}^{q} C_{b_j})$$

In other words, the set B is covered by $p + q < \frac{l}{2} + \frac{l}{2} = l$ sets, all of which are C_i or $\overline{C}_i$. Since (B, C_i) form an (m, l) set system, for some i, both C_i and $\overline{C}_i$ are present among the $p + q$ sets. This implies that for some a_i, b_j we have $\pi_e(a_i = b_j$.

We are choosing the two labels $\sigma(u)$ and $\sigma(v)$ uniformly at random from L_u and Lw. Hence with probability $\frac{1}{p} \cdot \frac{1}{q}$ we choose $\sigma(u) = a_i$ and $\sigma(v) = b_j$. Thus the probability that e is satisfied by σ is atleast $\frac{1}{p} \cdot \frac{1}{q} \geq \frac{4}{l^2}$. This completes the proof. □

Proof of theorem 3.1: From theorem 1.4, we know that there is no polynomial time algorithm to distinguish between Label Cover instances that are completely satisfiable, and those for which atmost $\frac{1}{\log^3 n}$ are satisfiable. Given a $\mathrm{GapLabelCover}_{1, \frac{1}{\log^3 n}}$ instance G, apply the reduction to Set Cover with $l = \log n$, $m = \Sigma$. From theorem 3.4, we know that the size of the (m, l) set-system is polynomially bounded and can be constructed in time polynomial in n. The size of the Set-Cover instance is thus a polynomial multiple of the size of the graph G. Therefore the reduction time is bounded by the size of the graph G, which is $O(n^{O(\log \log n)})$.

From lemma 3.5, we know that if the instance G was satisfiable, then there is a Set Cover of size $|V_1| + |V_2|$. The lemma 3.6 implies that if G is atmost $\frac{1}{\log^3 n} < \frac{2}{\log^2 n}$ satisfiable, then the set cover size is atleast $\frac{\log n}{8}(|V_1| + |V_2|)$. If there is a polynomial time $\frac{\log n}{8}$ approximation algorithm, then we can distinguish between the two cases. Therefore if there is a polynomial time $\frac{\log n}{8}$ approximation algorithm for Set cover, then we can solve a $\mathrm{E3SAT}$ or in general any problem in NP in time bounded by $O(n^{O(\log \log n)})$. □

References

[1] Uri Feige and Joe Kilian. Two prover protocols: low error at affordable rates. In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 172–183, New York, NY, USA, 1994. ACM Press.

[2] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[3] Ran Raz. A parallel repetition theorem. In *STOC '95: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 447–456, New York, NY, USA, 1995. ACM Press.