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

Lecture 11: Label Cover and 2-Prover 1-Round Game

Nov. 7, 2005

Lecturer: Ryan O'Donnell *Scribe: Ning Chen*

1 LabelCover Problem

The "Label-Cover problem" (also called "Raz's Verifier") plays an important role in proving hardness of approximation for other problems: It's the mother of all (or most) known "sharp" hardness-of-approximation reductions. In this lecture we will introduce and discuss the Label-Cover problem. We will then spend two lectures proving a hardness for it. The remainder of the course will involve showing strong hardness of approximation for various problems by reduction from Label-Cover and its variants.

Definition 1.1. (Label-Cover problem over Σ) *The Label-Cover problem with alphabet Σ is the same as the MAX-CG(Σ) problem with the following three restrictions:*

- *The constraint graph is bipartite, with vertex set (V_1, V_2).*
- *The graph is regular on the left; i.e., each vertex in V_1 has the same degree.*
- *The constraints have the "projection property", where a constraint $C_{(u,v)}$, $u \in V_1, v \in V_2$, is said to have the "projection property" if for every label $a \in \Sigma$ for u, $C_{(u,v)}$ accepts exactly one label for v.*

When the constraints have the projection property, we can think of them as *functions* $\pi_{(u,v)} : \Sigma \to \Sigma$, indicating what label for v is required, given a label for u.

As usual, we denote the gap decision version of the Label-Cover problem by Gap-Label-Cover. The main hardness theorem we wish to prove is:

Theorem 1.2. *For any $\epsilon > 0$, there exists Σ, such that Gap-Label-Cover$(\Sigma)_{1,\epsilon}$ is NP-hard.*

In general, the major difficulty of proving Theorem 1.2 is showing that GAP-CG$(\Sigma)_{1,\epsilon}$ is NP-hard. In other words, it's not really the bipartiteness, regularity, and projection property that make proving the theorem difficult — it's simply getting the soundness down to ϵ that is hard. Note that the homework demonstrates that Dinur's gap amplification methods don't seem to be able to push the soundness value below $1/2$.

Theorem 1.2 was first proved by Raz in'95 [4] as a consequence of his Parallel Repetition Theorem. His proof also has the very nice property that $|\Sigma| = \mathrm{poly}(1/\epsilon)$. Feige and Kilian in '94 [2]

had earlier proved most of Theorem 1.2 — they just didn't obtain the projection property. (Part of the reason for this is that at the time, it was not known that the projection property would be important.) Later, in the journal version of their work, they showed how to get projection, thus giving another (easier) proof of Theorem 1.2. A downside of their proof, though, is that it requires $|\Sigma| = 2^{\text{poly}(1/\epsilon)}$. Since Raz's proof is very difficult, we will only prove the Feige-Kilian version in class. In fact, we will also not obtain the projection property, since this makes the proof a tiny bit less complicated.

Before looking at either of these proofs, though, we will begin with a basic starting point:

Lemma 1.3. *There exists a universal $\epsilon > 0$ and Σ_0 of constant size, such that Gap-Label-Cover$(\Sigma_0)_{1,1-\epsilon}$ is NP-hard.*

Proof. In fact, no proof is needed! If one inspects the proof we gave that MAX-CG$(\Sigma_0)_{1,1-10^{-6}}$ that we gave (i.e., the PCP Theorem), one can see that the final constraint graph built already satisfies the three extra properties of Label-Cover! This comes from looking at the $O(1)$-query Assignment Tester to 2-query Assignment Tester reduction.

For clarity though, we will give an explicit proof. By the PCP Theorem, we know that GAP-E3SAT$_{1,1-\epsilon}$ is NP-hard for some explicit $\epsilon > 0$. Given an instance of GAP-E3SAT$_{1,1-\epsilon}$, where we have clauses $C_1, \ldots, C_m$ over variables $x_1, \ldots, x_n$, in the "YES" case we have that all clauses can be simultaneously satisfied and in the "NO" case at most a $1 - \epsilon$ fraction of clauses can be simultaneously satisfied. We construct a Label-Cover instance over Σ with $|\Sigma| = 7$ as follows: Let $C = \{C_1, \ldots, C_m\}$ and $X = \{x_1, \ldots, x_n\}$ be the two sets of vertices of the bipartite graph. For any $C_j \in C$ and $x_i \in X$, there is an edge between C_j and x_i if x_i appears in the clause C_j. Note that this construction guarantees that all vertices in C have degree three.

The labels on vertices in C represent *how* the clauses are satisfied. For each $C_j \in C$, there are seven different assignments (corresponding to seven symbols in Σ) that satisfy C_j. The labels for the variable vertices simply indicate a $\{0,1\}$-labeling (5 of the 7 labels go unused here). The constraint for any edge (C_j, x_i) is satisfied if the labels on C_j and x_i are consistent. That is, the label on x_i is exactly same as the assignment of x_i implied by the label on C_j. Thus, for any label on C_j, there is exactly one label on x_i such that edge (C_j, x_i) is satisfied. This implies that "projection property" is satisfied.

If the GAP-E3SAT$_{1,1-\epsilon}$ instance is "YES", then we can just label vertices according to the satisfying assignment, and thus all edges are satisfied. On the other hand, if the GAP-E3SAT$_{1,1-\epsilon}$ instance is "NO", any assignment of variables violates at least an ϵ fraction of clauses, and for each of those, at least one of the the corresponding edges is violated. Overall, this means that every assignment to the new constraint graph violates at least an $\epsilon/3$ fraction of the edge-constraints. □

2 2-Prover 1-Round Games

An equivalent way to look at bipartite CG problems is through the language of "2-prover 1-round games" (2P1R).

Definition 2.1. (2-Prover 1-Round Game) *A 2P1R game G is played by two players (or provers) P_1 and P_2, and has the following parameters:*

- *a set of questions X for P_1*
- *a set of questions Y for P_2*
- *an answer set A*
- *a probability distribution λ on $X \times Y$*
- *an acceptance predicate V on $X \times Y \times A \times A$.*
- *strategy $f_1 : X \to A$ and $f_2 : Y \to A$*

The game is played as follows:

- *a verifier picks $(x, y) \in X \times Y$ according to λ, and asks x to P_1 and y to P_2*
- *P_1 answers $a = f_1(x)$ and P_2 answers $b = f_2(y)$*
- *the verifier tests the predicate $V(x, y, a, b)$; the players P_1 and P_2 win if the predicate is satisfied, and lose otherwise.*

We stress that 2P1R games are precisely equivalent to *weighted*, *bipartite* constraint-graph sastisfaction problems; we have $X = V_1$, $Y = V_2$, $A = \Sigma$ and λ is the weighted distribution on edges.

Definition 2.2. *Given a 2P1R game G, we say the* value *of G, denoted by $\omega(G)$, is the probability (over λ) that P_1 and P_2 win, when they use optimal strategies.*

Value corresponds to the maximum fraction of simultaneously satisfiable constraints in a bipartite constraint graph.

3 Transformation on 2P1R games; parallel repetition

To prove Theorem 1.2 from Lemma 1.3, we would like to come up with a polynomial time reduction that greatly reduces the value of Label-Cover instances. In other language, given a 2P1R game G with $\omega(G) < 1$, how can we get a new 2P1R game G' with $\omega(G') < \epsilon$ for any $\epsilon > 0$? (The transformation should also have the property that $\omega(G') = 1$ when $\omega(G) = 1$.) Fortnow, Rompel, and Sipser in '88 [3] proposed the following answer to this question: repeat the game G k times in parallel. That is, construct game G^k as follows:

- P_1 has question set X^k, and P_2 has question set Y^k
- the answer set is A^k

- the verifier picks $(x_1, y_1), \ldots, (x_k, y_k)$ independently according to λ
- the verifier sends all $x_1, \ldots, x_k$ simultaneously to P_1, and $y_1, \ldots, y_k$ simultaneously to P_2
- the verifier gets back answers $a_1, \ldots, a_k$ from P_1, and $b_1, \ldots, b_k$ from P_2.
- G^k's acceptance predicate is $V(x_1, y_1, a_1, b_1) \wedge \cdots \wedge V(x_k, y_k, a_k, b_k)$. That is, the players win if they win in every coordinate.

FRS claimed that $w(G^k) = w(G)^k$. However, this is not correct; Fortnow found a counterexample. We give here another very simple counterexample, due to Feige. It is based on the "Noninteractive Agreement" (NA) game.

Definition 3.1. (Noninteractive Agreement 2P1R Game) *Call the provers P_0 and P_1. The NA 2P1R game is as follows:*

- *The verifier flips two independent coins b_0 and b_1, and sends b_0 to P_0 and b_1 to P_1. (I.e., $X = Y = \{0, 1\}$ and λ is uniform.)*
- *Each prover answers an element from the set $\{P_0, P_1\} \times \{0, 1\}$, representing a "guess" as to the coin flip for one of the players. The players win the game if both of them give the same answer* and *this answer agrees with reality (that is, if they both answer is $(P_0, 0)$, then b_0 should equal 0).*

Note that if a prover decides to answer herself, then of course it should be consistent with the reality, so the only problem is how should the other prover replies.

Lemma 3.2. $\omega(NA) \leq 1/2$ *(actually, it's $= 1/2$).*

Proof sketch. One player always has to guess a coin flip that he knows nothing about. □

Lemma 3.3. $\omega(NA^2) \geq 1/2$ *(actually, it's $= 1/2$).*

Proof idea. Consider the following strategy for P_0 and P_1: Both provers always answer with a guess for P_0's round one coin and with a guess for P_1's round two coin. One part of these guesses they can correctly fill in for themselves; for the other part, they operate under the assumption that P_0's first round coin equals P_1's second round coin. Whenever this actually happens (with probability $1/2$), both players will be completely right on both rounds. □

For a fuller discussion of this counterexample, see the side note on the course web page.

It's of interest to analyze what happens when this game is repeated k times. For simplicity, let k be even. By using the above strategy $k/2$ times on pairs of rounds, it's easy to see that the players can with with probability at least $2^{-k/2}$. Feige showed this is sharp:

Theorem 3.4. (Feige'91 [1]) *When k is even, $\omega(NA^k) = 2^{-k/2} = (1/\sqrt{2})^k$.*

This theorem shows that for this 2P1R game NA, $\omega(G^k)$ goes indeed go down exponentially in k — just with a different base in the exponent than $\omega(G)$.

In fact, this *always* happens; this is the content of Raz's Parallel Repetition Theorem:

Theorem 3.5. (Parallel Repetition Theorem (Raz'95 [4])) *Let $s < 1$ and $|A|$ be two constants. There exists $s' < 1$ (only depending on s and $|A|$) such that for any 2P1R game G with answer set A and $\omega(G) = s$, $\omega(G^k) < (s')^k$, for any $k \geq 1$.*

We will not prove the theorem in class. Note that if $k = O(\log 1/\epsilon)$, we have $\omega(G^k) < \epsilon$ and an answer size of $\mathrm{poly}(1/\epsilon)$. As a corollary of the Parallel Repetition Theorem, we get Theorem 1.2:

Proof. By reduction from Lemma 1.3: View a given Label-Cover instance as a 2P1R game, and repeat the game $O(\log 1/\epsilon)$ times in parallel. Then view this game as a bipartite CG problem again. It is easy to check that it is in fact a Label-Cover instance: parallel repetition preserves regularity and the projection property. □

References

[1] U. Feige, On the success probability of the two provers in one-round proof systems, Structures (CCC) 1991.

[2] U. Feige, J. Kilian, Two prover protocols: low error at affordable rates, STOC 1994, 172-183.

[3] L. Fortnow, J. Rompel, M. Sipser, On the power of multi-prover interactive protocols, Structures (CCC) 1988.

[4] R. Raz, A parallel repetition theorem, STOC 1995, 447-456.