PCPs and Inapproxiability CIS 6930 October 5, 2009

Lecture Hardness of Set Cover

Lecturer: Dr. My T. Thai Scribe: Ying Xuan

1 Preliminaries

1.1 Two-Prover-One-Round Proof System

A new PCP model **2P1R** – Think of the proof system as a game between two provers and one verifier.

- Prover Model:
  - each tries to cheat – convince the verifier that a "No" instance is "Yes";
  - each cannot communicate with the other on their answers;
- Verifier Model:
  - tries not to be cheated – ensures the probability to be cheated is upper bounded by some small constant;
  - can cross-check the two provers' answers;
  - is only allowed to query **one** position in each of the two proofs.

Definition 1 *Given three parameters: completeness c, soundness s and the number of random bits provided to the verifier $r(n)$, assume the two proofs are written in two alphabets Σ_1 and Σ_2 respectively, then a language L is in **2P1R**$_{c,s}(r(n))$ if there is a polynomial time bounded verifier V that receives $O(r(n))$ truly random bits and satisfies:*

- *for every input $x \in L$, **there is** a pair of proofs $y_1 \in \Sigma_1^*$ and $y_2 \in \Sigma_2^*$ that makes V accept with probability $\geq c$;*
- *for every input $x \notin L$ and **every pair** of proofs $y_1 \in \Sigma_1^*$ and $y_2 \in \Sigma_2^*$ makes V accept with probability $< s$;*

2P1R is actually a mechanism to solve Label-Cover: $\mathcal{L}(G = (U, V; E), \Sigma, \Pi) \Leftrightarrow$ $u \in U$ and $v \in V$, we have

- v: proof P_1; u: proof P_2;
- $uv \in E$: u,v have answers for the same bit;
- a labeling L is said to satisfy an edge uv iff $\Pi_{uv}(L(u)) = L(v)$: the two answers are consistent, otherwise they are cheating.

You can see this from the next proof and the proof of NP-hard for Gap-Max-Label-Cover$\Sigma_{1,\eta}$ in previous lecture.

Theorem 2 (29.21) *There is a constant $\epsilon_p > 0$ such that* ***NP*** $=$ ***2P1R***$_{1,1-\epsilon_p}(\log(n))$.

Proof:

- **2P1R**$_{1,1-\epsilon_p}(\log(n)) \subseteq$ **NP** (exercise);
- **NP** $\subseteq$ **2P1R**$_{1,1-\epsilon_p}(\log(n))$ (shown here).

Given a SAT formula ϕ, we employ a **gap-producing** reduction to obtain a MAX-E3SAT(E5) instance ψ, that is, given m is the number of clauses in ψ:

- if ϕ is satisfiable, $OPT(\psi) = m$;
- if ϕ is not satisfiable, $OPT(\psi) < 1 - \epsilon_p$ for some constant ϵ_p

Goal: Prove $SAT \in$ **2P1R**$_{1,1-\epsilon_p}(\log(n))$
Protocol:

- V selects an index of a clause, sends it to the first prover, selects a random variable in the clause, and sends it to the second prover.
- First prover returns 3 bits (the assignment of the selected clause), second returns 1 bit.
- V accepts if the following conditions hold:
  - Clause check: the assignment sent by the **first** prover satisfies the clause.
  - Consistency check: the assignment sent by the second prover is identical to the assignment for the same variable sent by the first prover.

Reduction

- if ϕ is satisfiable, then there exists two proofs such that both give a satisfiable assignment, so they will surely be accepted.
- otherwise,
  - The strategy of the second prover defines an assignment τ to the variables;
  - If V selects a clause that is not satisfied by τ (with probability ϵ_p), the first prover must set one of the variables differently from τ, otherwise, its assignment does not satisfy this clause.
  - The consistency check fails with probability $= 1/3$, since we have only $1/3$ probability to catch the disagreed variable from the three in the clause.

■

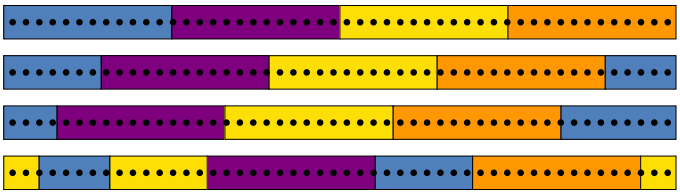

Figure 1:

1.2 Parallel Repetition Theorem and Gap Amplification

Motivation: To improve the soundness for SAT, i.e., to decrease the acceptance probability for false proofs.

Definition 3 *Parallel Repetition is: the verifier picks k clauses randomly and independently from the first prover P_1, then choose a random variable from each of these clauses and send the index of these variables to the second prover P_2. That is, it queries P_1 on the k variables and P_2 on the k clauses, and accepts iff all answers are accepting.*

Theorem 4 *If the error probability of a **2P1R** proof system is $\delta < 1$, then the error probability on k parallel repetitions is at most δ^{dk}, where d is a constant that depends only on the length of the answers of the original proof system.*

1.3 Set System

Definition 5 *Given a universal set U and all its subsets $C_1, \cdots, C_m$ and their complement set $\bar{C}_1, \cdots, \bar{C}_m$. A* good cover *is a collection of such subsets, which contains at least one subset C_i and its complement set $\bar{C}_i$. Otherwise, it is a* bad cover.

Theorem 6 *There is a randomized algorithm which generates a set system $(U, C_1, \cdots, C_m, \bar{C}_1, \cdots \bar{C}_m)$ with elements $|U| = 2^l$ for every m and l. With probability $> 1/2$ that every bad cover of such set system is of $size > l$.*

2 Main Reduction

Theorem 7 (Main) *From SAT to the cardinality set cover problem, there exists a constant $c > 0$ for which we can find a randomized **gap-producing** reduction ζ of polynomial time $n^{O(\log\log n)}$, which transforms a SAT instance ϕ to a set cover instance (U, S) where U is a universal set of size $n^{O(\log\log n)}$ such that*

- *if ϕ is satisfiable, $OPT(S) = 2n^k$;*
- *if ϕ is not satisfiable, **Pr**$[OPT(S) > kcn^k \log n] > 1/2$;*

*where n, polynomial in the size of ϕ, is the length of **each proof** for SAT in **2P1R**, k is $O(\log\log n)$.*

Sketch of Proof:
Step 1 – Equally long proof: For MAX-E3SAT(E5) with N variables, P_1 is an assignment for N variables, so $|P_1| = N$ and $P_1 \in \Sigma_1^*$ with $\Sigma = \{0, 1\}$; P_2 is the

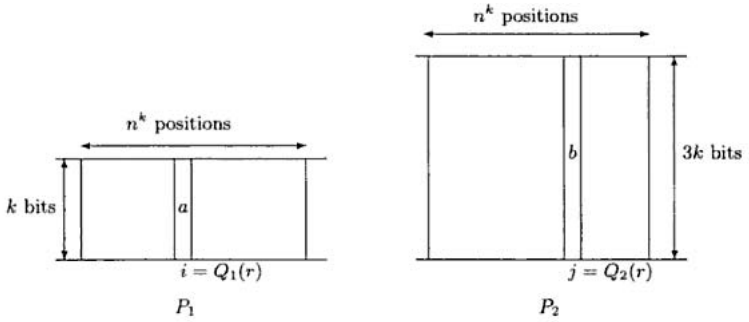

Figure 2:

assignment for $5N/3$ clauses, so $P_2 = 5N/3$. Repeat P_1 5 times and P_2 3 times. $\rightarrow$ $|P_1| = |P_2| = 5N$, so $n = 5N$. The verifier will query a **random** copy of each proof.

Step 2– Gap Amplification: Execute k parallel repetitions to both proofs, so $|P_1| = |P_2| = n^k$. (Every time query k positions independently from n positions). The result for each position in P_1 and P_2 is of k-bit and $3k$-bit respectively.

Step 3– Position Pick: choose one position (k clauses) from $P_2 \rightarrow 1$ out of n^k, then choose one position (k variables) each of which occurs in one chosen clause from $P_1 \rightarrow k$ out of 3^k. We need $n^k 3^k$ random strings.

Step 4– Acceptance Mapping: for any random string r, $Q_1(r)$ – picked position in P_1, $Q_2(r)$ – picked position in P_2. Let b be the answer for $Q_2(r)$, and $\pi(r, b)$ be the answer in P_1 agrees with b. (The same assignment for k-bit corresponding to the k-clauses).

Step 5– Set System Construction: construct $\mathcal{T} = (U, C_1, \cdots, C_{2^k}, \bar{C}_1, \cdots, \bar{C_{2^k}})$. $2^k \leftrightarrow$ all possible assignments to k bits $\leftrightarrow$ each C_a corresponds to one k-bit answer a in P_1. **Remark**: C_a corresponds to an answer in P_1, $\bar{C_{\pi(b)}}$ corresponds to another answer in P_1, if $\pi(b) = a$ (agreement in 2P1R), then C_a and $\bar{C_{\pi(b)}} = \bar{C}_a$ is a cover of U.

Step 6– Set System Copy: Copy the system $(3n)^k$ times to be $\mathcal{T}^1, \cdots, \mathcal{T}^r, \cdots, \mathcal{T}^{3n^k}$ so each copy corresponds to the results of a different random string r. Note that $\mathcal{T}^r = (U^r, C_1^r, \cdots, C_{2^k}^r, \bar{C}_1^r, \cdots, \bar{C_{2^k}^r})$ and **universal set** U^r **are pairwise disjoint**. The union of U^r for all r is the universal set $\mathcal{U}$ of the set system, where $\mathcal{U} = (3n)^k |U| = n^{O(\log \log n)}$.

Step 7– Reduction Construction: The subsets of $\mathcal{U} = \bigcup_r U^r$ are of two kinds:

- $S_{i,a} = \bigcup_{r:Q_1(r)=i} C_a^r$: for **ALL** random strings, if position i is picked in P_1, union the sets corresponding to all answers for this position from P_1.
- $S_{j,b} = \bigcup_{r:Q_2(r)=j} \bar{C}_{\pi(r,b)}^r$: for **ALL** random strings, if position j is picked in P_2, union the complement sets corresponding to the answer $\pi(r, b)$, *iff* b is satisfiable to all k picked clauses.

Remark: For any cover C of a sub-universal set $\mathcal{U}^r$, if C contains a pair of such two set kinds: $S_{i,a}$ and $S_{j,b}$ with $\pi(r,b) = a$ (called pair-set), i.e., it contains a *good cover*. Otherwise, it contains a *bad cover*, so every bad cover is of size $> l$, i.e. there are more than l subsets, but no pair-sets.

Step 8– Yes Instance: Since for a satisfiable instance ϕ, the results a and b for respectively position i in P_1 and j in P_2 should agree, so for each such pair (a,b) choose pair-set $S_{i,a}$ and $S_{j,b}$ and put into S. Therefore, the size of S is $2n^k$.(the position i in P_1 and j in P_2 are one-to-one mapped.) Certainly S contains a good cover for each U^r and hence $\mathcal{U}$.

Step 9– No Instance: Assume S is an optimal cover of $\mathcal{U}$, try to prove S has a lower bound on its size with probability $> 1/2$;

- For each position i and j in P_1 and P_2 respectively, let $A(i) = \{a | S_{i,a} \in S\}$ and $A(j) = \{b | S_{j,b} \in S\}$;

- Arbitrarily choose answers from $A(i)$ and $A(j)$ to the verifier queries for position i in P_1 and j in P_2, to construct proofs (y_1, y_2);

- Differentiate random strings. Let $B_1 = \{r || A(Q_1(r))| > l/2\}$ and $B_2 = \{r || A(Q_2(r))| > l/2\}$, and $G = B_1 \bar{\cup} B_2 = \bar{B_1} \cap \bar{B_2}$;

- So if a random string r is from G, then at most $l/2$ sets of kind $S_{i,a}$ and $l/2$ sets of kind $S_{j,b}$ will be included in S, so the size of S is less than l, so it should be a good cover, with probability $> 1/2$;

- Assume the pair-set within S above is $S_{i,a} \cup S_{j,b}$, but by randomly choosing, we can only pick up these pair (i,j) with probability $\geq 1/l \cdot 1/l$, since $|A(Q_1(r))| \leq l/2$ and $|A(Q_2(r))| \leq l/2$ for r in G (choose 1 from $l/2$ independently). So with proof (y_1, y_2), the verifier accepts on random string r with probability $\geq (2/l)^2$.

- According to the gap amplification, the error probability decreases from δ to δ^{dk} for k-repetition. Take $\delta = 1/2$, $k = O(\log\log n)$. Denote the fraction of random strings in G is f_G, then for any random strings, the error probability is (to have an error verification, the random string r should be within G with prob. f_G, and the verifier should accept the proof with prob. $\geq (2/l)^2$) $f_G(2/l)^2 \leq \delta^{dk}$, which will lead to $f_G < 1/2$ (need confirmation).

- Therefore, $B_1 \cup B_2$ contains at least half the random strings, so at least one of B_1 and B_2 contains a quarter of the random strings, say it is B_i. Now we only consider the set cover for all U^r where $r \in B_i$.

- For r in B_i, there are least $l/2$ subsets of kind $S_{i,a}$ or $S_{j,b}$. Since each random string r is mapped to one position pair (i,j) with $Q_1(r) = i$ and $Q_2(r) = j$, so there are at least a fraction $1/4$ of such position pairs (i,j). Moreover, for each position pair, at least $l/2$ subsets (of either kind) are contained in S, so there are

in total at least $l/2 * n^k * 1/4 = ln^k/8 = \Sigma(kn^k \log n)$.

■

Corollary 8 *There is a constant b such that if there exists a $b \log n$-approx algorithm for the cardinality set cover problem, then $NP \subseteq$ **ZTIME**$(n^{O(\log \log n)})$, i.e.,a class of problems for which there is a randomized algorithm running in expected time $O(n^{O(\log \log n)})$.*

References

[1] V.V. Vazirani, Approximation Algorithms, Springer, 2001.