Georgia Tech Fall'04
CS 8002 : PCPs and Hardness of Approximation

Lecture 2: The PCP Theorem

Lecturer: *Subhash Khot* Scribe: *Nayantara Bhatnagar*

In this lecture we will state the **PCP** Theorem and show that it is equivalent to the existence of a reduction from any **NP**-complete language L to Gap-MAX-3SAT with a constant gap.

1 Gap Preserving Reductions

Definition 1 *Let P and P' be maximization problems. A* gap preserving reduction *from P to P' is a polynomial time algorithm which given an instance I of P with $|I| = n$, produces an instance I' of P' with $|I'| = n'$ such that if*

- $OPT(I) \geq h(n)$*, then* $OPT(I') \geq h'(n')$
- $OPT(I) \leq g(n)h(n)$*, then* $OPT(I') \leq g'(n')h'(n')$

for some functions $h(n), g(n), h'(n'), g'(n')$ with $g(n), g'(n') \leq 1$.

Observe that if *Gap-*$P_{g(n)}$ is **NP**-hard, then the problem *Gap-*$P'_{g'(n')}$ is also **NP**-hard.

Example 1 Let $G = (V, E)$ be an undirected graph. An independent set of G is a set $S \subseteq V$ such that for every pair of vertices $u, v \in S$ the edge $(u, v) \notin E$. We will see that the usual reduction from MAX 3SAT to Independent Set(IS) is gap preserving. Let ϕ be an instance of MAX3SAT with n variables and m clauses. We construct the graph G_ϕ from ϕ as follows. The graph G_ϕ has a vertex $v_{i,j}$ for every occurrence of the variable x_i in clause C_j. All the vertices corresponding to literals from the same clause are joined by an edge (thus forming triangles). Also, if a variable x_i occurs in clause C_j and its negation $\overline{x}_i$ occurs in clause $C_{j'}$, we join the vertices $v_{i,j}$ and $v_{i,j'}$ by an edge. Verify that there is an independent set of size $\geq k$ in G_ϕ if and only if there is an assignment which satisfies $\geq k$ of the clauses of ϕ. Hence, we have

- $OPT(\phi) = 1 \Rightarrow OPT(G_\phi) \geq m$
- $OPT(\phi) \leq c \Rightarrow OPT(G_\phi) \leq cm$

where $c < 1$ is an absolute constant. By the **PCP** Theorem, we know that there exists a polynomial time reduction from SAT to Gap-3SAT, hence, we get

Theorem 1
IS is hard to approximate within $\frac{1}{c}$.

In fact, we can amplify this constant by a reduction from an instance of IS with a gap of $\frac{\alpha}{\beta}$ to an instance of IS with gap $\left(\frac{\alpha}{\beta}\right)^2$. By repeating any constant k times, we can show

Theorem 2
IS is hard to approximate within a factor of $\left(\frac{1}{c}\right)^k$ *for every constant integer* $k \geq 1$.

Proof: Given an instance $G = (V, E)$, construct the graph $G' = (V', E')$ where $V' = V \times V$ and $E' = \{((u, v), (u', v')) \mid (u, u') \in V \textit{ or } (v, v') \in E\}$. Let $I \subseteq V$ be an independent set of G. The $I \times I$ is an independent set of G' by construction. Hence $OPT(G') \geq OPT(G)^2$. On the other hand, let I' be an optimal independent set of G' with vertices $(u_1, v_1), \cdots, (u_k, v_k)$. By construction, the vertices $u_1, \cdots, u_k$ and $v_1, \cdots, v_k$ are independent sets in G. Hence each contains at most $OPT(G)$ distinct vertices, and therefore $OPT(G') \leq OPT(G)^2$. Thus we have $OPT(G') = OPT(G)^2$. Hence

- $OPT(G) \geq \alpha n \Rightarrow OPT(G') \geq \alpha^2 n^2$
- $OPT(G) \leq \beta n \Rightarrow OPT(G') \leq \beta^2 n^2$ □

Since a maximum clique of G is a maximum independent set of the complement $\overline{G}$, the same amplification property (and hardness result) is true for Maximum Clique.

In the next section, we define probabilistic verifiers for **NP** and state the PCP Theorem in terms of existence of efficient probabilistic verifiers.

2 The PCP Theorem

NP can be defined as the class of languages that are accepted by polynomial time non-deterministic Turing machines. Equivalently it can be defined in terms of existence of a polytime deterministic verifier that can check membership proofs :

Definition 2 *A language L is in* **NP** *if and only if there exists a deterministic polynomial time verifier V such that given an input x, and a proof π such that $|\pi| = |x|^{O(1)}$ it satisfies*

- *Completeness: $x \in L \Rightarrow \exists\ \pi$ such that $V(x, \pi) = 1$*
- *Soundness: $x \notin L \Rightarrow \forall\ \pi,\ V(x, \pi) = 0$*

Now let us define probabilistic verifiers that are restricted to look at only a few bits of the proof instead of reading the whole proof.

Definition 3 *A $(r(n), q(n))$-restricted verifier is one that is restricted to using at most $r(n)$ random bits, runs in probabilistic polynomial time, and queries $q(n)$ bits from the proof.*

Definition 4 *A language L is in* **PCP**$(r(n), q(n))$ *if and only if there exists a $(r(n), q(n))$-restricted verifier such that given an input x, $|x| = n$ and a proof π, the verifier satisfies*

- *Completeness: $x \in L \Rightarrow \exists\ \pi$ such that* $\Pr[V(x, \pi) = 1] = 1$
- *Soundness: $x \notin L \Rightarrow \forall\ \pi$,* $\Pr[V(x, \pi) = 1] \leq \frac{1}{2}$

where V accepts or rejects on the basis of the bits read from the proof and the probabilities are computed over the choice of random bits. Since there are $2^{r(n)}$ possible "runs" of the verifier and every run reads $q(n)$ bits, one can place an a priori bound $|\pi| \leq q(n) \cdot 2^{r(n)}$.

PCP for Graph Non-Isomorphism (GNI)

Two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ on n vertices are said to be isomorphic if there exists a permutation in S_n, $\pi : V_1 \to V_2$ such that $(\pi(u), \pi(v)) \in E_2$ iff $(u, v) \in E_1$. Two graphs are non-isomorphic if there exists no such permutation. Denote isomorphism by $G_1 \approx G_2$.

THEOREM 3
$GNI \in \mathbf{PCP}(O(n \log n), 1)$

PROOF: The input $x = (G_1, G_2)$, is a pair of graphs on n vertices. Each bit of the proof π corresponds to a labeled graph on n vertices H, and the bit is supposed to be 1 or 2 respectively if H is isomorphic to G_1 or G_2. If H is isomorphic to neither, the bit may be arbitrary. The verifier uses $O(n \log(n))$ random bits to choose $i \stackrel{R}{\longleftarrow} \{1, 2\}$ and to choose a random permutation. She applies the permutation to the vertices of G_i to obtain its random isomorphic copy, say H. She queries π at the position corresponding to H and accepts if and only if the bit queried is i.

- Completeness: Suppose $G_1 \not\approx G_2$. Since every graph H is isomorphic to either G_1 or G_2 but not both, we can construct a proof π such that $\mathbf{Pr}[V((G_1, G_2), \pi) = 1] = 1$.
- Soundness: Suppose $G_1 \approx G_2$. Since H could "arise" from G_1 or G_2 with probability 1/2 each, no matter which bit the location H contains, the verifier accepts with probability exactly 1/2. Thus for any proof π, $\mathbf{Pr}[V((G_1, G_2), \pi) = 1] = \frac{1}{2}$ □

Now we are ready to state the PCP Theorem :

THEOREM 4 (**PCP** THEOREM)
$\mathbf{NP} = \mathbf{PCP}(O(\log(n)), O(1))$

The inclusion $\mathbf{PCP}(O(\log(n), O(1)) \subseteq \mathbf{NP}$ is easy to see. Let V be a verifier for $L \in PCP(O(\log(n), O(1))$. The proof π has at most $2^{O(\log(n))}$ bits. Hence we can construct a polynomial time deterministic verifier V' by simulating all possible coin flips and computing the probability that V accepts. The verifier V' accepts if and only if this probability is 1.

Now we will show that $\mathbf{NP} \subseteq \mathbf{PCP}(O(\log n), O(1))$ if and only if there is a reduction from any **NP**-complete language to Gap-MAX-3SAT. Thus the PCP Theorem is same as an inapproximability result for MAX-3SAT.

THEOREM 5
$\mathbf{NP} = \mathbf{PCP}(O(\log(n)), O(1)) \Leftrightarrow \exists$ *a polytime reduction from any* **NP***-complete language* L *to MAX-3SAT, mapping instances* x *for* L *to instances* ϕ *for MAX-3SAT such that,*

- $x \in L \Rightarrow OPT(\phi) = 1$
- $x \notin L \Rightarrow OPT(\phi) \leq c, \quad c < 1$

PROOF: ($\Leftarrow$:) Assume that for $L \in \mathbf{NP}$ there is a reduction f as in the statement of the theorem to MAX-3SAT where $f(x) = \phi$, and ϕ has variables $Y_1, \cdots Y_n$ and clauses $C_1, \cdots, C_m$ where $m = n^{O(1)}$.

We show that L has a PCP with $O(\log(n))$ random bits and $O(1)$ queries. The verifier V reads the input x and produces the formula ϕ. Using $\log(m) = O(\log(n))$ random bits, she chooses a uniformly random clause $C_j = Y_i^* \vee Y_k^* \vee Y_\ell^*$ of ϕ, where * denotes the variable or its complement. The proof π corresponds to an assignment to the variables. The verifier reads the bits corresponding to the variables appearing in C_j and accepts if and only if the values satisfy C_j. Then we have,

- Completeness: If $x \in L$, there exists a satisfying assignment for ϕ. Setting π to the satisfying assignment ensures that $\mathbf{Pr}[V(x, \pi) = 1] = 1$.
- Soundness: If $x \notin L$, any assignment to the variables $Y_1, \cdots, Y_n$ satisfies at most a fraction c of the clauses. Hence, for all π, since V chooses a clause uniformly at random $\mathbf{Pr}[V(x, \pi) = 1] \leq c$.

The soundness probability can be amplified by a constant number of repetitions. Now we prove the other direction.

($\Rightarrow$:) Assume that $L \in \mathbf{NP}$ and has a PCP using $O(\log(n))$ random bits and $O(1)$ queries. We first reduce L to an intermediate constraint satisfaction problem whose variables $Y_1, \cdots, Y_{|\pi|}$ are the bits in the proof π. Consider a fixed random string τ used by the verifier. Let $Y_{i_1}^\tau, \cdots, Y_{i_q}^\tau$ be the query bits fixed by τ. Let $C_\tau = C(Y_{i_1}^\tau, \cdots, Y_{i_q}^\tau)$ denote the constraint that V tests for acceptance. The number of constraints is $2^{O(\log(n))} = n^{O(1)}$. The maximization problem is to find an assignment to the variables Y_i's which maximizes the fraction of satisfied constraints. This is the same as the problem of constructing a proof π maximizing the probability that V accepts. The completeness and soundness conditions guarantee

- If $x \in L$, $\exists \pi$ such that $\mathbf{Pr}[V(\pi) = 1] = 1$. Hence, there is an assignment to the variables $Y_1, \cdots, Y_{|\pi|}$ such that all the constraints C_τ are satisfied.
- If $x \notin L$, for all π, $\mathbf{Pr}[V(\pi) = 1] \leq \frac{1}{2}$. Hence, any assignment to the variables $Y_1, \cdots, Y_{|\pi|}$ satisfies at most $\frac{1}{2}$ fraction of the constraints.

We will take an instance of this constraint satisfaction problem I and map it to an instance of MAX-3SAT ϕ such that

- $OPT(I) = 1 \Rightarrow OPT(\phi) = 1$
- $OPT(I) \leq \frac{1}{2} \Rightarrow OPT(\phi) \leq 1 - \frac{1}{q2^{q+1}}$

We can write any constaint $C(Y_{i_1}^\tau, \cdots, Y_{i_q}^\tau)$ as a CNF with at most 2^q clauses and q literals in each clause. We can write the CNF as a 3SAT formula by using at most q extra variables, and with at most $q2^q$ clauses in total.

If there is a staisfying assignment for the Y_i's, then this gives a satisfying assignment for the 3SAT. If for any assignment, at least $\frac{1}{2}$ of the constraints are unsatisfied, then any assignment will satisfy at most $1 - \frac{1}{q2^{q+1}}$ fraction of clauses of the 3SAT formula ϕ. $\square$