Lecture 18
PCP theorem and hardness of approximation

Ishan Banerjee
Undergraduate Department
Indian Institute of Science

Instructor
Chandan Saha
Computer Science and Automation
Indian Institute of Science

November 7, 2015

Definition 0.1 (MAX − 3SAT)**.** Given m clauses $a_1 \dots a_m$ with each clause the disjunction of at most three literals, let $\varphi = \bigwedge a_i$. We define $val(\varphi)$ to be the maximum number of clauses that can be satisfied by an assignment.

The MAX − 3SAT problem is to find an assignment to the literals such that the fraction of clauses satisfied is $val(\varphi)$.

It is easy to see that:

- MAX − 3SAT is NP hard.
- 3SAT $\leq_p$ MAX − 3SAT.
- $val(\varphi) = 1 \iff \varphi \in$ SAT.

Definition 0.2 (ρ approximation for a maximization problem)**.** For any $\rho < 1$ a ρ approximate algorithm for a maximization problem outputs an answer, that is greater than or equal to ρ times the optimum.

We can analogously define ρ approximations for minimization problems. In that case ρ will be greater than 1.

1 $\frac{1}{2}$ approximation for MAX − 3SAT

To give a $\frac{1}{2}$ approximate algorithm for MAX − 3SAT we have to give an algorithm which on input φ always outputs an answer that is at least $\frac{1}{2}val(\varphi)$ fraction of the clauses. It suffices to find an assignment that satisfies at least $\frac{1}{2}$ the clauses. We will do so in the following way:

- Pick any variable x
- Count the number of clauses where x appears and where $\neg x$ appears.
- If the number of clauses where x appears is higher, simply assign x the value 1 otherwise assign x the value 0.
- Remove all clauses that have their value fixed by this assignment and repeat with other variables.

This algorithm works as each time, more than half the clauses in which x appears are fixed to 1.
Remark: There exists a $\frac{7}{8}$ approximation algorithm for MAX − 3SAT.

2 2 approximation algorithm for MIN − VERTEXCOVER

Vertex Cover of a graph $G(V, E)$ is a subset of vertices S such that every edge in G is incident to at least one vertex in S. MIN − VERTEXCOVER of graph $G(V, E)$ is a vertex cover $\mathcal{V}$ of G such that the size of every other vertex cover S of G is at least $|\mathcal{V}|$. An easy 2 approximation algorithm for MIN − VERTEXCOVER is as follows:

- Find a maximal matching of the graph G. Take all the vertices of the matching. This forms a vertex cover as any edge must have at least one endpoint in the matching(otherwise we could extend it).
- Any vertex cover must contain atleast one of the two vertices corresponding to an edge in the matching. Thus every other vertex cover must have size at least half of this one.

3 PCP-verifier

Let L be a language in NP. We want to recast the notion of verification into the notion of having a PCP verifier.

Definition 3.1 (PCP$(r(n), q(n))$)**.** Let $r, q : \mathbb{N} \to \mathbb{N}$. A language $\mathsf{L} \subseteq \{0,1\}^*$ has a $(r(n), q(n))$-PCP verifier, if there is a probabilistic TM M^π (with a special random access tape for the proof π) such that the following are true:

- M^π uses $r(n)$ many random bits on an i/p of length n, to generate $q(n)$ addresses on the proof tape and queries the proof at not more than $q(n)$ bits kept at those indices. M^π's output is either 0 or 1.
- $x \in \mathsf{L} \iff \exists\pi$ such that $Pr_r(M^\pi(x) = 1) = 1$
- $x \notin \mathsf{L} \iff \forall\pi Pr_r(M^\pi(x) = 1) \leq \frac{1}{2}$

Definition 3.2 (PCP$(r(n), q(n))$)**.** A language $\mathsf{L} \in$ PCP $(r(n), q(n))$ if it has a PCP $(cr(n), dq(n))$ verifier for some $c, d \in \mathbb{R}$.

Theorem 3.1 (PCP theorem - proof version)**.** NP = PCP $(\log n, 1)$

Theorem 3.2 (PCP theorem - hardness of approximation version)**.** *There is a constant $\rho < 1$ such that for every language $\mathsf{L} \in \mathsf{NP}$, there is a deterministic polynomial time reduction from L to 3SAT such that if $x \in \mathsf{L}$ $val(\phi_x) = 1$ and if $x \notin \mathsf{L}$, $val(\phi_x) < \rho$.*

Corollary 3.2.1. *There is a constant $\rho < 1$ such that ρ approximation of* MAX − 3SAT *is* NP *hard.*

Proof: Let $\mathsf{L} \in \mathsf{NP}$. Reduce L to 3SAT as above. Suppose $x \mapsto \phi_x$ under the reduction. Given a ρ approximation for 3SAT, I simply use it to approximate $val(\phi_x)$. If I obtain an answer greater than ρ the machine says that $x \in \mathsf{L}$. If not, $x \notin \mathsf{L}$. □