COM S 783 / OR&IE 634 Approximation Algorithms	**Fall Semester, 2003**

Lecture 21-22: Nov 19 and 24, 2003

?? *Scribe: Bharath Kumar Rangarajan*

1 Hardness of Approximation

It was shown in the first lecture that approximating TSP (without any assumption on the metric) within any factor is NP-hard.

MAX-E3SAT: Given a collection of clauses, either all clauses are satisfied or for every assignment there is some clause that is violated. This problems is very well-known to be NP-Hard.

'NP-hardness' of Approximating **MAX-E3SAT**: Given a collection of clauses, either all clauses are satisfied or for every assignment atleast a certain fraction of the clauses are violated.

Probabilistically Checkable Proof (PCP) (for a language L)
L is the class of all "yes" instances, i.e. instances with a correct proof or witness. PCP is a polynomial time algorithm V with the following properties (V is also called a verifier)

$$\begin{aligned} &\forall \text{ inputs } x \in L\, \exists \text{ a witness } w \text{ such that the } V(x,w)\text{accepts} \\ &\forall \text{ inputs } x \notin L\, \forall \text{ witness } w,\ \Pr\left[V(x,w) \text{ accepts }\right] \le 1/2. \end{aligned} \tag{1}$$

One can think of V as a map from $L \times W$ to $\{0,1\}$, where 0 is reject and 1 is accept) and W to be the space of witnesses. The above definition generalises the definition of NP, where the probability of rejection is 1:

$$\begin{aligned} &\forall \text{ inputs } x \in L\, \exists \text{ a witness } w \text{ such that the } V(x,w)\text{accepts} \\ &\forall \text{ inputs } x \notin L\, \forall \text{ witness } w,\ V(x,w) \text{ rejects }. \end{aligned} \tag{2}$$

Probabilitiscally checkable proofs are parametrised using r and q. The verifier, V, tosses r coins and probes accordingly q bits of the witness w and decides to accept or reject the proof. So, the verifier chooses to access random positions in the proof string and decides to accept or reject. For NP languages, the witness (or proof. these will be used interchangeably) can be restricted to be of polynomial size, while in PCP the witnesses can in principle be of exponential size, however the total random access is restricted to be polynomial in size. PCP(r,q) is the class of languages recognizable by a PCP with parameters $O(r)$, $O(q)$. For every q PCP($\log n, q$) $\subset$ NP is easy to see as there are polynomial in n ($i.e., 2^{O(\log n)}$)

possiblities and all witnesses are of size $O(q)$ and hence in total no more than polynomial number of bits is explored for verification.

PCP Theorem: NP $\subset$ PCP$(\log n, 1)$.

Corollary: It is NP-hard to approximate MAX-E3SAT within a factor better than $1-\epsilon$ for some $\epsilon > 0$.

Proof: For every 3CNF formula ψ $\exists$ poly-time computable proper 3CNF formula ξ such that

$$\begin{aligned} \psi \text{ satisfiable} &\Rightarrow \xi \text{ satisfiable.} \\ \psi \text{ not satisfiable} &\Rightarrow \xi \text{ is atmost } 1-\epsilon \text{ satisfiable.} \end{aligned}$$

This means that there exists a threshold of approximation $\epsilon > 0$ such that for any $\rho > 1-\epsilon$ it is NP hard to distinguish between satisfiable and ρ–satisfiable proper 3CNF formulae.

Let PT be the set of languages L such that there exists a poly-time algorithm that computes a (proper 3CNF) formula $\psi(x)$ on input x, such that

1. if $x \in L$ then $\psi(x)$ is satisfiable, and
2. if $x \notin L$ then $\psi(x)$ is atmost ρ (i.e., $\leq \rho$) satisfiable.

Theorem: PT = PCP $(\log n, 1)$.
Proof: First, PT $\subset$ PCP$(\log n, 1)$. We will construct a PCP for L: The witness w is just an assignment for $\psi(x)$. Let m = number of clauses in $\psi(x)$ and k = minimum integer such that $\rho^k \leq 1/2$.

Verification: Pick k clauses independently and uniformly at random. Note that we have atmost $3k$ variables in these clauses. Accept iff assignment to these variables satisfies the clauses we had chosen.

If $x \in L$ then $\psi(x)$ has a satisfying assignment w and we always accept w. If $x \notin L$, then every assignment w satisfies atmost ρm clauses of $\psi(x)$ because $\Pr[\text{accept}] \leq \rho^k \leq 1/2$ ($\Pr[\text{accept for a clause}] \leq \rho$).

<u>Other direction</u>: PT $\supset$ PCP$(\log n, 1)$: Consider a verifier that uses $r = O(\log n)$ random bits and $q = O(1)$ bit queries. Notation: input I, witness for that input is w_I has length polynomial in the size of I.

For each of the 2^r possible outcomes of tosses or equivalently of the 2^r random bit values, we run the verifier and find the q bit locations $i_1^s, \ldots, i_q^s$ it checks. Assign a variable x_i to each position i in w_I. Based on positions $i_1^s, \ldots, i_q^s$ verifier decides to accept or reject. This can be done using a truth table for accepting $x_{i_1^s}, \ldots, x_{i_q^s}$. Now represent the truth table as a q-CNF formula of size atmost 2^q (a constant). Denote this formula by $\psi_q(s)$ and similarly for other outcomes we construct their q-CNF formulae.

Convert $\psi_q(s)$ into a 3CNF formula ϕ^s using less than $q2^q$ auxillary variables.

$$x_1 \vee \ldots \vee x_q = (\bar{y}_1 \vee x_1 \vee y_2) \wedge (\bar{y}_2 \vee x_2 \vee y_3) \wedge \ldots$$

ϕ_I is the conjunction of all formulae ϕ^s. If $I \in L$ we want ϕ_I to be satisfiable. Because, $\exists$ w such that every possible s is accepted. Hence, for the truth table for s, the assignment for those values gives a true value of 1. Therefore, ϕ^s is satisfiable. Hence, ϕ_I is satisfiable.

If $I \notin L$, then given any witness w_I) atmost $\frac{1}{2}2^r = 2^{r-1}$ values of s are accepted. This means that the assignment to corresponding truth table gives false. That is ψ_q^s is not satisfied by w_I. Then ϕ^s is not satisfied by w_I plus the corresponding assignment to the auxillary variable. That is, at least one clause is not satisfied in ϕ^s.

How many are these? The fraction of unsatisfied clauses is at least $\frac{1}{2}\frac{1}{q2^q}$. Let us denote this constant by ϵ. Then an ϵ–fraction is always unsatisfied. Therefore, we have generated a gap. $\square$

The PCP theorem is proven in 3 steps (Ultimately, we would like to give sharp thresholds on approximation guarantees.)

Structure of the PCP Theorem:

Step 1: (Outer Verifier) First step is to prove that: NP $\subset$ PCP$(\log n, \text{poly}(\log n))$. The proof of this is nor presented here. This if adapted to earlier proof of equivalence would give ϕ_s of quasi-polynomial size.

Step 2: Now take the language L in PCP$(\log n, \text{poly}(\log n))$. This will now be verified using an inner verfier applied recursively. i.e., to check poly$(\log n)$ bits and deciding to accept or not, we re-apply this PCP and use poly$(\log\log n)$ bits of the proof.

Problem: Note if we take a arbitrary outer verfier then we'll have trouble keeping the time spent verifying to polynomial in n. And total number of positions we should have probed will not be $O(1)$ if we do this recurrence many times. We actually will stop recurrence very early (may be just one time).

Step 3: At the bottom, i.e., when we reach a sufficiently small size in the number of bits to verify, PCP$(\hat{n}^2, 1)$ can be used allowing exponential sized (in $\hat{n}$) witness. If $\hat{n}$ is of the order of poly$(\log\log n)$ then the whole process will still be polynomial in n.

Definition of **3LIN**: A conjunction of clauses each being an exclusive OR of 3-literals. $(x_1 \oplus x_2 \oplus \bar{x}_3)$.

Satisfiability for 3LIN is known to be in P, i.e., decision version is in , but the optimization version is not.

Theorem: $\forall \epsilon > 0$, is is NP-hard to distinguish between $\geq (1-\epsilon)$ satisfiable and $(1/2+\epsilon)$-satisfiable 3LIN formulae.

We can give a $\frac{1}{2}$-approximation by randomly assigning each variable to 0 or 1. This theorem says we can't do any better.

Corollary: For all $\epsilon > 0$, it is NP-hard to distinguish between $\geq (1-\epsilon)$–satisfiable and $\leq (\frac{7}{8}+\epsilon)$–satisfiable for proper 3CNF formulae and note that $\frac{7}{8}$ is tight. Recall that a proper 3CNF formula has exactly 3 literals in each clause.

Proof Replace each 3LIN clause by four 3CNF clauses such that if 3LIN is satisfied then the 3CNF clauses are satisfied. And the following is the mechanism for conversion:

$$x_1 \oplus x_2 \oplus x_3 \text{ is replaced by } (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3).$$

$x_1 \oplus x_2 \oplus x_3$ is satisfied if one or 3 variables is satisfied. $\square$

References

[1] S. Sahni and T. Gonzalez, *P–complete approximation problems*, J. Assoc. Comput. Mach., 23 (1976), pp. 555–565.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy *Proof verification and intractability of approximation problems*, In Proc. of 33rd IEEE Symp. on Foundations of Computer Science (1992), pp. 13–22.

[3] S. Arora, R. Motwani, M. Safra, M. Sudan, M. Szegedy *PCP and approximation problems*, Manuscipt (1992).

[4] C. Håstad *Some optimal inapproximability results*, Proc. 29th Annual ACM Symp. on Theory of Computing (1997), pp. 1–10.