IITM-CS6840: Advanced Complexity Theory	Feb 21, 2012
Lecture No. 27 : Inapproximability	
Lecturer: Jayalal Sarma M N	*Scribe: Balagopal*

THEME: Inapproximability
LECTURE PLAN:Inapproximability of Independent set Problem. GAPCSP to GAPIS, PCP for LIN, Attempts, Proof in the long-code form. Need of linearity testing.

Our aim is to show the inapproximability of MAXINDSET. For this purpose, we introduce the problem $\mathsf{GAPINDSET(s,c)}$. An instance of $\mathsf{GAPINDSET(s,c)}$ is a graph G which is guaranteed to either have an independent set of size at least cn or to have no independent set of size sn (i.e., all independent sets are of size less than sn). Note that $0 \leq s < c \leq 1$ for otherwise the problem is same as the INDSET problem.

Let us define the notion of an approximation algorithm for MAXINDSET. An ϵ-approximation A for MAXINDSET is an algorithm that takes a graph G as input and yields an independent set of size at least ϵk where k is the size of the maximum independent set in G.

Now we connect the existence of good approximation algorithms for MAXINDSET to algorithms solving GAPINDSET. Note that if A is an ϵ-approximation for MAXINDSET, then A can be used to solve $\mathsf{GAPINDSET(s,c)}$ where $s < \epsilon c$. For example, let us take $\epsilon = 1/2$. Suppose A is a 1/2-approximation for MAXINDSET. Then we can use A to solve $\mathsf{GAPINDSET(c/2,c)}$. We run A on the input graph G and output "yes" iff A outputs an independent set of size greater than $(c/2)n$. If G had an independent set of size at least cn, then A is guaranteed to output an independent set of size at least $(c/2)n$. Otherwise, by the promise, the largest independent set in G has size less than $(c/2)n$ and A outputs an independent set of size less than $(c/2)n$. This shows that if a 1/2-approximation to MAXINDSET exists, then $\mathsf{GAPINDSET(c/2,c)}$ can be solved in polynomial time. In other-words, by showing that $\mathsf{GAPINDSET(s,c)}$, where $s/c < \epsilon$, is NP-complete, we may conclude that an ϵ-approximation to MAXINDSET does not exist unless $\mathsf{P} = \mathsf{NP}$.

1 qGAPCSP $\leq_m^p$ $\mathsf{GAPINDSET(m,m/2)}$

We now present a reduction from qGAPCSP to $\mathsf{GAPINDSET(m,m/2)}$. Here q stands for the number of variables in each constraint of CSP. The parameter m is the number of constraints. The promise in qGAPCSP problem is that either all constraints can be satisfied or less than 1/2 the fraction of the constraints can be satisfied (This is where the $m/2$ comes from in $\mathsf{GAPINDSET(m,m/2)}$). The following algorithm constructs a graph from an instance of qGAPCSP.

```
1.  Create m clusters of vertices, one for each constraint.
2.  The vertices in cluster i are in one-to-one correspondence with
..  satisfying assignments for ψ_i.  That is, for each (global) assignment
..  that satisfies ψ_i, we add a vertex to cluster i that corresponds to the
..  restriction of the global assignment satisfying ψ_i.
3.  All vertices within a cluster are connected.  Two vertices
..  u and v in different clusters are connected iff they are not
..  contradictory.  That is, there does not exist any x_i such
..  that x_i = 1 in u and x_i = 0 in v or viceversa.
```

The running time of the algorithm is polynomial since each cluster contains at most 2^q vertices and there are only a linear number of clusters.

We now prove the correctness of the reduction. Suppose ψ is a yes instance. Then we claim there is an independent set of size m. Let x be the (global) assignment that satisfies ψ. Then, x satisfies each ψ_i. Choose the vertex corresponding to x from the i^{th} cluster for each i. Since the assignment from each cluster is the same, there is no edge between any of the vertices. Now suppose ψ was a no instance. We claim that no independent set of size $m/2$ exists in G. Suppose we were able to select $m/2$ vertices. Then, by construction each vertex would be from a different cluster. We are also guaranteed that they are not contradictory. So there exists a way to extend the partial assignment to yield a global assignment satisfying $m/2$ constraints which violates our assumption that ψ is a no instance of the promise problem.

The above result combined with hardness of qGAPCSP shows the inapproximability of MAXINDSET.

2 Towards the PCP theorem

As a first step towards proving the PCP theorem $\mathsf{NP} \subseteq \mathsf{PCP}(O(\log n), O(1))$ we prove the result $\mathsf{LIN} \in \mathsf{PCP}(O(\log n), O(1))$ where LIN is the language of all linear system of equations solvable over $\mathbb{F}_2$. Assume that the proof Π is the satisfying assigment. Then it seems impossible to verify with high probability the correctness by looking at only a constant number of bits. We get around this problem by demanding a different sort of proof from the prover [1]. Note that the system could be written as $Ax = b$ where $A \in \mathbb{F}_2^{m \times n}$, $x, b \in \mathbb{F}_2^{n \times 1}$. Suppose the system is solvable, then for any $r \in \mathbb{F}_2^{m \times 1}$, we have $r^T Ax = r^T b$. If we let $r^T A = a$, we may rewrite this as $a.x = r^T b$. Note that the right hand side could be computed without looking at the proof. We now describe the structure of the proof Π. The proof $\Pi \in \mathbb{F}_2^{2^m}$ where the i^{th} bit of Π is the value of $r^T Ax$ for the i^{th} r. If the system is satisfiable, an honest prover could compute $a.x$ for each choice of a with the satisfying assignment x. In the next lecture, we will see that if the system is unsatisfiable, then verifier has a strategy to reject with high probability.

[1] The proof in long-code form is simply the Hadamard encoding of the satisfying assignment