CMPUT 675: Approximation Algorithms	Winter 2005

Lecture 24: April 8

Lecturer: Mohammad R. Salavatipour — *Scribe: Frederick Vizeacoumar*

24.1 Hardness of Clique (contd.)

Last lecture, we started a polynomial hardness for Clique. We started from a $PCP_{1,\frac{1}{2}}(d \log n, q)$ verifier for NP and showed how this leads to the gap of factor 2 for clique. Simulating this PCP verifier algorithm k times, we get a $PCP_{1,\frac{1}{2^k}}(k.d \log n, q.k)$ verifier which lead to the gap of 2^k for clique. To get a gap of n^δ, we need a $PCP_{1,\frac{1}{n}}(O(\log n), O(\log n))$ verifier for NP. The idea is to use an expander graph H on n^d vertices with labels of length $O(\log n)$, do a random walk on H of length $O(\log n)$ and collect "random" stings of length $d \log n$ (i.e. the labels).

Theorem 24.1 $PCP_{1,\frac{1}{2}}(d \log n, q) \subseteq PCP_{1,\frac{1}{n}}(O(\log n), O(\log n))$

Proof: Let $L \in PCP_{1,\frac{1}{2}}(d \log n, q)$ and $\mathcal{F}$ be a verifier for L. We give a $PCP_{1,\frac{1}{n}}(O(\log n), O(\log n))$ verifier $\mathcal{F}'$ for L.

$\mathcal{F}'$ builds graph H (Expander graph), then creates a random walk of length $klogn$ using only $O(logn)$ bits for some constant k (because choosing a neighbor, it takes only constant bits). This random walk yields $k \log n$ "random" string of length $d \log n$ each, which are the labels of the vertices of the walk. Then $\mathcal{F}'$ simulate $\mathcal{F}$ on each of these strings and accepts if and only if all these simulations accept.

Now, if $y \in L$ is a "yes" instance, then there is a proof Π such that, $\mathcal{F}$ accepts with probability 1 given Π (i.e. on all random strings) which means $\mathcal{F}'$ accepts.

Suppose, $y \in L$ is a "no" instance. Thus $\mathcal{F}$ accepts on at most $\frac{n^d}{2}$ of the random strings. Let S be the set of vertices of H with those labels (i.e. these random strings on which $\mathcal{F}$ gives the wrong answer). Note that $|S| \leq \frac{n^d}{2}$. This means that $\mathcal{F}'$ accepts y only if all the random walk is inside S. Now, based on the last Theorem of previous lecture (Theorem 23.4), the probability that a random walk remains entirely in S is at most $\frac{1}{n}$ (if k is sufficiently large constant). Therefore, the probability that $\mathcal{F}'$ accepts y is at most $\frac{1}{n}$. ■

Theorem 24.2 *For some $\delta > 0$, it is NP-hard to approximate clique within a factor of $\Omega(n^\delta)$.*

Proof: Given a SAT formula ϕ, let F be a $PCP_{1,\frac{1}{n}}(d \log n, q \log n)$ verifier for it for some constants d and q. Construct the graph G from F in the same manner as we did in the last lecture. So the size of G is $n^d 2^{q \log n} = n^{d+q}$ and the gap created is equal to soundness probability, i.e. $\frac{1}{n}$. More specifically:

- If ϕ is a yes instance then G has a clique of size n^d.
- If ϕ is a no instance then every clique of G has size at most n^{d-1}.

This creates a gap of n^δ with $\delta = \frac{1}{d+q}$, which is a constant. ■

24.2 Hardness of Set Cover

Our goal from now (until the end of the course) is to present an $\Omega(\log n)$-hardness result for set cover. To achieve this goal we need to define a few other problems and prove hardness results for them. The first one is a variation of Max-3SAT.

24.2.1 Gap-Max-3SAT(5) problem

Given a 3SAT formula with the extra restriction that every variable belongs to 5 clauses.

Goal: To find the maximum number of clauses satisfied by an assignment.

Theorem 24.3 *There is a gap preserving reduction form Max-3SAT to Max-3SAT(5).*

Given a Max-3SAT(5) instance ϕ let $Opt(\phi)$ denote the number of clauses that can be satisfied by a truth assignment. Then it is NP-hard to decide if:

- $opt(\phi) = m$
- $opt(\phi) \leq (1-\epsilon)m$ for some constant $\epsilon > 0$.

24.2.2 Label Cover Problem

The label cover problem is a graph theoric modeling of a 2-pover 1-round proof system for NP. An instance of label cover consists of the followings:

- $G(V \cup W, E)$ is a bipartite graph.
- $[N] = \{1...N\}$, $[M] = \{1...M\}$ are 2 sets of labels, $[N]$ for the vertices in V and $[M]$ for the vertices in W.
- $\{\Pi_{v,w}\}_{(v,w)\in E}$ denotes a (partial) function on every edge (v,w) such that $\Pi_{v,w} : [M] \to [N]$

A labeling $l : V \to [N], W \to [M]$ is said to cover edge (v,w) if $\Pi_{v,w}(l(w)) = l(v)$.

Goal: Given an instance of label cover, find a labeling that covers maximum fraction of the edge.

Theorem 24.4 *Given an instance $\mathcal{L}(G, M = 7, N = 2, \{\Pi_{v,w}\})$ it is NP-hard to decide if*

- $opt(\mathcal{L}) = 1$*, or*
- $opt(\mathcal{L}) \leq 1 - \epsilon$

Proof: We use Theorem 24.3. Given a Max-3SAT(5) formula ϕ, we construct an instance $\mathcal{L}$ as follows.

Let the variables of ϕ be $\{x_1, \ldots, x_n\}$ and clauses be $\{C_1, \ldots, C_m\}$.

Define $V = \{x_1...x_n\}$, and $W = \{C_1, \ldots, C_m\}$, i.e. create a vertex in V for every variable of ϕ and a vertex in W for every clause of ϕ. Two vertices x_i and C_j are adjacent iff $x_i \in C_j$. Note that the degree of every vertex in V is 5, because each variable is in 5 clauses and the degree of every vertex in W is 3.

For $\Pi_{v,w}$: V gets labels from $\{0,1\}$ and for every clause $C_j \in V$ let $[7]$ be the set of seven satisfying assignments that a clause C_j can have. Then, $\Pi_{x_i,C_j} : [7] \rightarrow [2]$ is basically the bijection of the assignment of variable x_i in the given satisfying assignment of C_j.

Example: if $C_j = x_1 \vee x_2 \vee x_3$, then $\Pi_{x_1,C_j}(101) = 1$ and $\Pi_{x_2,C_j}(101) = 0$

If ϕ is satisfiable (i.e. a Yes instance), then there is a truth assignment satisfying all clauses. Consider the labeling of x_i's defined by this truth assignment and also every clause (of course satisfied) has one of those seven labels and these labels are consistent on every edge. This means, all the edges of $\mathcal{L}$ are covered. So $opt(\mathcal{L}) = 1$.

Now assume that ϕ is a "no" instance (i.e. $opt(\phi) \leq (1-\epsilon)m$. Consider any solution to $\mathcal{L}$. Labels on V give a truth assignment to the variables. This means, the fraction of satisfied clauses of ϕ by this truth assignment is at most $(1-\epsilon)m$. Consider any unsatisfied clauses $C_j = x_1 \vee x_2 \vee x_3$ by this truth assignment. Any label (satisfying truth assignment) to C_j is inconsistent with the truth assignment to at least one of its variables (or else clause C_j is satisfied). Therefore, at least one of the the 3 edges incident with C_j must not be covered. This implies that at last $\frac{\epsilon}{3}$ fraction of edges are not covered. Thus $Opt(\mathcal{L}) \leq (1-\frac{\epsilon}{3})$. ■