Computational Complexity Theory, Fall 2010 December 10

Lecture 26: PCP: Overview, degree reduction and alphabet reduction

Lecturer: Kristoffer Arnsfelt Hansen Scribe: Martin Sergio Hedevang Faester

The PCP-theorem

In this lecture the main ideas of the proof of the PCP theorem will be discussed. Let us recall the theorem which was seen in the previous lecture

Theorem 1 (**PCP**)**.**

$$\boldsymbol{NP} = \boldsymbol{PCP}(O(\log n), O(1))$$

Notice that this is equivalent to $(1, s) - GAP - qCSP$ *being* ***NP****-hard for* $s < 1$ *and* q *constant!*

We will look at $qCSP_W$, CSP with variables $x_1, x_2, \ldots, x_n \in \{1, 2, \ldots, W\}$ (the size of the alphabet is W) and constraints dependent on q variables.

Let us consider the case with $q = 2$. Given an instance of $2CSP_W$, ϕ defines the *constraint-graph* for ϕ as follows:

- Nodes correspond to the variables $x_1, x_2, \ldots, x_n$.
- The edges correspond to the m constraints of ϕ, and go between the two variables on which a constraint is dependent.

Example 1. **3COLORING**: Given a graph G, is it 3-colorable? This is a $2 - CSP$ problem:

- Assign to each node a variable
- Assign to each edge the constraint that the end-nodes of an edge must have different colors.

The constraint graph for this problem is G itself!

We will from now on work mostly with $2 - CSP$ problems, and will always have the constraint graph in mind of an instance in mind. In particular we will focus on *graph theoretic* properties of the constraint graph as well as the constraint themselves.

So let us return to our main goal: Show $(1, s) - GAP - 2CSP_W$ is **NP**-hard for constants $s < 1$ and W.

Observation 2. **3COLORING** is **NP**-hard $\Rightarrow (1, 1 - \frac{1}{m}) - GAP - 2CSP_3$ is **NP**-hard.

The proof of the **PCP**-theorem below is long so it is useful to first look at the high level proof strategy to get an overview of what to do.

Proof-strategy of the PCP-theorem

Start with the $(1, 1-\frac{1}{m}) - GAP - 2CSP_3$ problem, and run $O(\log m)$ iterations of a procedure that does the following

- The number of variables and constraints are increased by at most a constant factor.
- The size of the alphabet W is left unchanged.
- The constraint-gap is increased by at least a constant factor (unless it is already larger than a fixed constant)

Thus after k iterations the number of variables is at most $C_1^k \cdot n$, the number of constraints at most $C_2^k \cdot m$ and the constraint gap at least $\min(C_4, C_3^k \frac{1}{m})$, for some constants C_1, C_2, C_3, C_4.

We can choose $k = O(\log m)$ such that $C_3^k \frac{1}{m}$ is larger than a constant. With this the number of variables is $n^{O(1)}$ and the number of constraints is $m^{O(1)}$.

Each iteration consists of 3 parts

Degree reduction + expanderization: This involves

- The graph is transformed into an constant degree expander graph.
- Side-effect: The constraint-gap is decreased by some constant k_1.

Gap amplication: This involves

- The constraint-gap can be increased by some chosen constant k_2.
- Side-effect: The degree and size of the alphabet are increased by a constant dependent on k_2.

Alphabet reduction: This involves

- Fix size of the alphabet as a constant k_3.
- Side-effect: The constraint gap is decreased by a constant k_4.
- Side-effect: Transform the graph such that it is no longer an expander graph.

The crucial thing that makes this work is that the increase in the constraint-gap in step 2 can be chosen large enough to counteract the side-effects of the other steps, decreasing the constraint-gap, by a fixed constant.

We will need the following lemma about expander graphs, essentially saying that in an expander graph there are no small cuts.

Lemma 3. *Let $G = (V, E)$ be a (n, d, λ)-expander graph and $S \subseteq V$. If $|S| \leq \frac{n}{2}$ then $|E(S, \bar{S}| \geq \frac{(1-\lambda)d}{2}|S|$.*

Proof. The Expander mixing lemma (EML) says

$$\left| |E(S, \bar{S})| - \frac{d}{n}|S||\bar{S}| \right| \leq \lambda d \sqrt{|S| \left(1 - \frac{|S|}{n}\right)\left(1 - \frac{\bar{S}}{n}\right)} = \frac{\lambda d |S||\bar{S}|}{n}$$

Hence,

$$|E(S, \bar{S}| \geq \frac{d}{n}|S||\bar{S}| - \lambda d \frac{|S||\bar{S}|}{n} = (1-\lambda)d\frac{|S||\bar{S}|}{n} \geq \frac{(1-\lambda)d}{2}|S|$$

□

Degree reduction

We a given a constraint graph by a $2-CSP$ instance ϕ with m constraints. Let $(G_k)_{k=1}^{\infty}$ be a

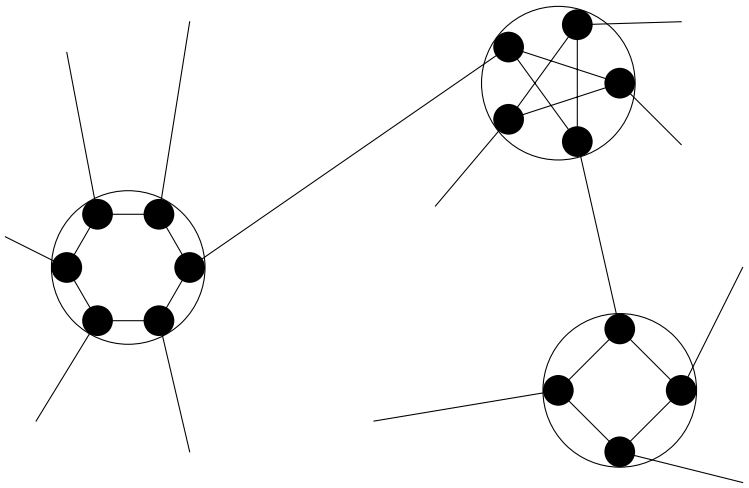

Figure 1: Placing small expander graphs inside the nodes of ϕ to obtain ψ.

$(k, d, \frac{1}{10})$-expander family. For a node in ϕ of degree k, we replace that node by a copy of G_k and every of the k old edges to that node are connected to a (unique) node in the new copy of G_k.

We impose on the new edges in G_k equality-constraints.

Denote by ψ the new graph obtained from ϕ. It is now a $(d+1)$-regular graph and has $(d+1)m$ constraints. We will now consider the new constraint gap. Clearly if $\text{val}(\phi) = 1$ then $\text{val}(\psi) = 1$.

Now suppose $\text{val}(\phi) \leq 1 - \epsilon$ (ϵ is the gap). Choose an optimal assignment y on ψ.

From this we define an assignment u on ϕ as follows: For a node in ϕ, we choose the most popular value in the inserted graph G_k.

By assumption u violates $\epsilon \cdot m$ constraints. If a constraint is violated in ϕ it holds that it *least* one of the following must hold

- the same constraint is violated by y in the constraint graph ψ, or
- One of the endpoints does not have the popular value in ψ.

For a node v define $S_v = \{\text{nodes in the } G_k \text{ placed in } v \text{ that do not have the popular value}\}$. We can write

$$\epsilon \cdot m \leq \#(\text{constraints violated by } u \text{ in } \phi) \leq \#(\text{constraints violated by } y \text{ in } \phi) + \sum |S_v|$$

Note that there are 2 cases

(a) $\#(\text{constraints violated by } y \text{ in } \phi) \geq \frac{\epsilon \cdot m}{2}$. In this case, since $\frac{\epsilon \cdot m}{2} = \frac{\epsilon \cdot m'}{2(d+1)}$, the gap is decreased by a factor at most $\frac{1}{2(d+1)}$.

(b) $\sum |S_v| \geq \frac{\epsilon \cdot m}{2}$. We analyse this case below.

For a node v and value a, let S_v^a be the set of nodes in S_v with value a. Then

$$|S_v^a| \leq \frac{\deg(v)}{2},$$

since otherwise a would be the popular value.

Note that all constraints in $E(S_v^a, \bar{S}_v^a$ are violated. By the lemma for expander graphs we have

$$|E(S_v^a, \bar{S}_v^a)| \geq \frac{9/10 \cdot d}{2} |S_v^a|$$

Hence the number of constraints violated is at least

$$\sum_v \sum_a \frac{9/10d}{2} |S_v^a| = \frac{9/10 \cdot d}{2} \sum_v |S_v| \overset{(b)}{\geq} \frac{9/10 \cdot d}{2} \frac{\epsilon \cdot m}{2} = \frac{9d}{40(d+1)} \epsilon m'$$

We now have a $(d+1)$-regular constraint graph ψ with n' nodes. The next step is to ensure it is an expander graph as well. Choose $d' \geq d+1$ such that there exists an $(n', d', \frac{1}{10})$-expander.

Put this on top of ψ (i.e. take the union of the two graphs) with dummy constraints that is always satisfied on the new edges. We will also ensure that at least half of the edges from each node are self-loops. To this end add also to each node $d' + d + 1$ self-loops, again with a dummy constraints. Let $\hat{\psi}$ be the result. Then

$$\lambda(\hat{\psi}) \leq \frac{3}{4} + \frac{1}{4} \cdot \frac{1}{10} = \frac{31}{40}$$

since at least a $\frac{1}{4}$ fraction of all edges from each node comes from the $(n', d', \frac{1}{10})$-expander graph (the inequality uses the problem from the homework exercises).

Alphabet reduction

In the lecture we also started discussin the alphabet reduction phase, indicating some of the ideas.