CSE 533: The PCP Theorem and Hardness of Approximation **(Autumn 2005)**

Lecture 5: PCP Theorem proof: Gap Amplification

Oct. 12, 2005

Lecturer: Ryan O'Donnell *Scribe: Chris Ré and Ryan O'Donnell*

1 Overview

Recall from last lecture that we were in the middle of Dinur's four step process for amplifying the gap of a constraint graph.

1. Degree-Reduce
2. Expanderize
3. Powering
4. Alphabet-Reduce

Recall that the powering stage is the stage where the gap of the problem increases. We will first give a sketch of the proof of this stage in order to give some intuition and then return to details for the remainder of the lecture. The details presented use a slick improvement due to J. Radakrishnan (unpublished).

2 Powering Stage (Sketch)

2.1 Parameter Effects

In this section, we will be sketchy about some details. Entering the powering stage, we have an input constraint graph denoted $(G, \mathcal{C})$. G is an a (n, d, λ)-expander, with $\lambda < d$ universal constants, and the constraints are over some fixed constant alphabet $\Sigma = \Sigma_0$. Our goal is to produce a new constraint graph $(G', \mathcal{C}')$ with a larger gap. We will denote parameters of $(G, \mathcal{C})$ (e.g. size) in the input without a prime, and constraint graph parameters of $(G', \mathcal{C}')$ in the output with a prime (e.g.. size'). Our goal for gap' is as below:

$$\mathrm{gap}' = \begin{cases} 0 & \text{if gap = 0,} \\ \geq \frac{t}{O(1)} \cdot \min(\mathrm{gap}, \frac{1}{t}) & \text{otherwise.} \end{cases}$$

As in previous stages we worry about some ancillary parameters of the graph. The effects of this stage on parameters besides the gap are summarized in Figure 1.

Output Params	Input Params
size'	= $O(\text{size})$
$\lvert\Sigma'\rvert$	$\approx \lvert\Sigma\rvert^{d^t}$
$\deg'$, λ'	we don't care

Figure 1: Effect on parameters

2.2 The beginning of the sketch

Now we begin the sketch in earnest: We need to describe the construction by which we come to the output graph G'. This construction will have an integer parameter $t \geq 1$ in it. For the moment, we will treat t like a variable but it will eventually be filled in with some universal constant, perhaps 10^6.

Vertex Set. The output graph G' will have the same vertex set; i.e. $V' = V$.

The Alphabet. The new alphabet, Σ', will be given by $\Sigma' = \Sigma^{1+d+d^2+\cdots+d^t}$. This alphabet size is chosen because $1 + d + d^2 + \cdots + d^t$ is an upper bound on the number of vertices at distance at most t from a given vertex in G, since G is d-regular. As a result, we can identify for each node, w, in the t-neighborhood of a node v a particular position in the label. We say that the value of this position corresponds to an "opinion" about the value of w in the old graph. Given an assignment $\sigma' : V' \to \Sigma'$, we write $\sigma'(w)_v \in \Sigma$ for the portion of w's label that corresponds to the node v; i.e., the "opinion" of w about v's value.

Constraints. The new edges e' will correspond to paths in G of length about t. (We are being intentionally fuzzy here.) For such a path from a to b in G, we make an edge on (a, b) in G'. The constraint we put on the edge is, roughly speaking, "check everything". For example, in checking the assignment $\sigma' : V' \to \Sigma'$, we can check that for any v on the path that $\sigma'(a)_v$ and $\sigma'(b)_v$ agree. More importantly, we can also check that for each edge (u, v) on the path, the assignment $(\sigma'(a)_u, \sigma'(b)_v)$ is acceptable for the old constraint on (u, v), namely $\mathcal{C}(u, v)$. Actually, when we eventually analyze the gap, we will only worry about such checks.

The Effect on the Gap. Notice that if $\text{gap} = 0$, then in $(G', \mathcal{C}')$ one can just use the labelling that honestly writes down the actual opinions giving the satisfying assignment for $(G, \mathcal{C})$. This labelling will pass all the constraints in $(G', \mathcal{C}')$ (indeed, the constraints were chosen with this in mind); hence gap' will still be 0, as desired. When the gap does not equal 0, we hope to make $gap' \geq \frac{t}{O(1)} \cdot \text{gap}$. (Note that if gap is already at least $\frac{1}{t}$ then this might not make sense; hence the reason for actually writing $\frac{t}{O(1)} \cdot \min(\text{gap}, \frac{1}{t})$.) So our approach is to take a best assignment for $(G', \mathcal{C}')$, call it σ', and extract from it some assignment for the original constraint graph, $\sigma : V \to \Sigma$. This assignment by definition must violate at least a gap fraction of the constraints in $(G, \mathcal{C})$. We use this fact to get a lower bound on how many constraints σ' violates in $(G', \mathcal{C}')$.

2.3 Rough analysis

Recall that given σ', we write $\sigma'(w)_v$ for the "opinion" of w about v. To define the extracted assignment $\sigma : V \to \Sigma$, given a vertex $v \in V$, we consider the vertices w at distance about t from it (again, we are intentionally fuzzy here). Each of these has an opinion $\sigma'(w)_v$ about what v's label in Σ should be, and for $\sigma(v)$ we take the plurality of these opinions.

Having defined σ, we know that it violates the constraints on some edges $F \subseteq E$ with $|F|/|E| \geq \text{gap}$. Throw away some edges from F if necessary so that $|F|/|E| = \min(\text{gap}, \frac{1}{t}) =: \gamma$.

Now we want to show that σ' has about t times more unsatisfied edge constraints. Consider an edge e' in E'; i.e., a path $a \to b$ in G. If this path passes through an edge in F, say (u, v), then there is a "good chance" that $\sigma'(a)_u = \sigma(u)$ and $\sigma'(b)_v = \sigma(v)$. This is because $\sigma(u)$ and $\sigma(v)$ are the plurality opinions about u and v among neighbors at distance around t — and a and b are such neighbors (fuzzily speaking). Further, if these things happen, then σ' violates the constraint on edge $e' = (a, b)$!

This discussion leads us to conclude that, roughly speaking,

$$\text{gap}' = \mathop{\mathbf{P}}_{e'}[\sigma' \text{ violates } e'] \geq \frac{1}{O(1)} \cdot \mathop{\mathbf{P}}_{e'}[e' \text{ passes through } F].$$

Finally, we can easily analyze the probability that a random path of length about t hits F, using the fact that G is an expander:

$$\begin{aligned}
\text{gap}' &\geq \frac{1}{O(1)} \cdot \mathop{\mathbf{P}}_{e'}[e' \text{ passes through } F] \\
&= \frac{1}{O(1)} \cdot (1 - \mathop{\mathbf{P}}_{e'}[e' \text{ completely misses } F]) \\
&= \frac{1}{O(1)} \cdot \left(1 - \left(1 - \frac{|F|}{|E|}\right)^t\right) && \text{(since } G \text{ is an expander)} \\
&\approx \frac{1}{O(1)} \cdot (1 - (1 - t\gamma)) && \text{(since } (1-\alpha)^t \approx 1 - \alpha t \text{ for } \alpha \leq 1/t) \\
&= \frac{t}{O(1)} \cdot \gamma,
\end{aligned}$$

as desired.

3 Details for the gap amplification argument

We now explicitly describe the powering step. Recall we start with $G = (V, E)$ an (n, d, λ)-expander and constraints $\mathcal{C}$ over Σ on the edges E. We are building a new constraint graph $(G', \mathcal{C}')$. As mentioned, the new vertex V' is the same as the old vertex set, and the new alphabet is $\Sigma' = \Sigma^{1+d+d^2+\cdots+d^t}$, where a label in this set is thought of as giving an "opinion" on what label

from Σ should be given to all vertices w within *shortest-path graph-distance t in G*. We will write $\mathrm{dist}_G(v, w)$ for this distance between v and w.

To define the new edge set and constraints, we will go through the notion of a *verifier*; the verifier will be a randomized process for generating an edge (a, b) for E' along with a constraint to go with it, a subset of $\Sigma' \times \Sigma'$. Note that this does not exactly give a new proper constraint graph. There are two reasons for this: First, our verifier will actually give a probability distribution over *all possible edges*; i.e., all of $V \times V$. Second, a probability distribution over edges/constraints can really only be viewed as a *weighted* constraint graph, where the weight associated to an edge is equal (or proportional) to the probability the verifier chooses that edge. In fact, we actually have to view the verifier as generating a weighted constraint graph with *multiple parallel edges*; the reason is that the verifier may produce the same pair of endpoints (a, b) in different ways with different constraints when it does its random test. So once we describe our verifier, what we've really described a *weighted* constraint graph on the complete graph, with some multiple parallel edges.

This is not so good, since we want an unweighted constraint graph (we don't mind multiple parallel edges) and since we get at least n^2 edges ($n = |V|$), rather than $O(n)$; i.e., the size has gone up way too much. Nevertheless, we will disregard these two problems for the duration of this lecture and just try to do the gap analysis for the verifier we present. In the next lecture we will show how to slightly change the verifier so that we get a weighted constraint graph with *constant* degree (the constant depends on t, d, and $|\Sigma|$). Having fixed this problem, we can also easily get an equivalent unweighted constraint graph by replacing weighted edges by (constantly many) multiple parallel unweighted edges.

3.1 The verifier

We now describe our randomized verifier by describing how it picks a random edge $e' = (a, b) \in E'$ and what test it performs on the resulting labelling $(\sigma'(a), \sigma'(b)) \in \Sigma' \times \Sigma'$. First, the verifier does an After-Stopping Random Walk (A.S.R.W. — see Lecture 4) starting from a random vertex a and ending at some random vertex b. *This walk may be of any finite length.* (Note: it ends after finitely many steps with probability 1. Also note: it can start and end at a and b in multiple different ways.) Once the verifier completes its walk, it does the following test:

For every step $u \to v$ it took, if $\mathrm{dist}_G(u, a) \le t$ and $\mathrm{dist}_G(v, b) \le t$ and if $(\sigma'(a)_u, \sigma'(b)_v)$ is a violation of the edge-constraint $\mathcal{C}(u, v)$ from the old constraint graph, then V rejects. If the verifier doesn't reject for any of these $u \to v$ steps, then it accepts.

Note: the verifier may take a step $u \to v$ more than once on its path; if so, it will be equally satisfied or unsatisfied about each of these steps. The reason is that it doesn't matter *when* the verifier takes the $u \to v$ step; u is either within distance t of a or it isn't, and the same goes for v and b.

Having defined our verifier, we now want to analyze its satisfiability gap. To that end, let σ' be the best assignment $\sigma' : V' \to \Sigma'$ for this verifier. We want to "extract" from it an assignment $\sigma : V \to \Sigma$. Here is how we do this:

Definition 3.1. *To define $\sigma(v)$: Consider the probability distribution on vertices $w \in V$ gotten as follows:*

- *Do a Before-Stopping Random Walk (B.S.R.W. — see Lecture 4) starting from v, ending on w.*
- *Condition on this walk stopping within t steps.*

Since any walk from v of at most t steps ends at a vertex that is within distance t of v in G, the above gives a probability distribution on vertices w that have opinions on v. *Thus we get a well-defined associated probability distribution $\mathcal{L}$ on letters of Σ — namely, the distribution $\sigma'(w)_v$. Finally, given this distribution, $\sigma(v)$ is defined to be the letter in Σ with highest probability under distribution $\mathcal{L}$. Ties can be broken by, say, lexicographical order on Σ.*

Note: this is not a random assignment; given σ', the assignment σ is fixed. Note also that this definition is a completely abstract notion we make for analysis purposes. Our overarching poly-time deterministic reduction does not need to do or understand this definition.

With σ defined, as in class we let $F \subset E$ be the subset of edges from the old constraint graph it violates. (And, we throw away some edges from F if necessary so that $|F|/|E| = \min(\text{gap}, 1/t)$.) Given this F, we introduce the notion of a *faulty step* within the verifier's walk:

Definition 3.2. *Within the verifier's A.S.R.W., we say a particular step $u \to v$ is* faulty *if:*

- $(u, v) \in F$*;*
- $\text{dist}_G(u, a) \leq t$ and $\sigma'(a)_u = \sigma(u)$*;*
- $\text{dist}_G(v, b) \leq t$ and $\sigma'(b)_v = \sigma(v)$*;*

Define also N to be the random variable counting the number of faulty steps the verifier makes in its A.S.R.W.

The key observation about this definition is that whenever $N > 0$, the verifier rejects. This is by definition of the verifier's actions. (Note that the verifier may reject in other cases too. For instance, it may reject because it made a step $u \to v$ where u is within distance t of a, v is within distance t of b, and $(\sigma'(a)_u, \sigma'(b)_v)$ is a bad assignment for $\mathcal{C}(u, v)$; however, this can happen without $\sigma'(a)_u$ equalling $\sigma(u)$ or $\sigma'(b)_v$ equalling $\sigma(v)$, which is required for faultiness.) Note also that if a verifier makes the step $u \to v$ many times, then *either all of them are faulty or none is faulty*.

We would like to show that $\mathbf{P}[N > 0]$ — and hence gap' — is large. We can do this by using the *second moment method*, a basic tool from probability:

Lemma 3.3. *(Second Moment Method) If X is a nonnegative random variable then*

$$\mathbf{P}[X > 0] \geq \frac{\mathbf{E}[X]^2}{\mathbf{E}[X^2]}.$$

Proof.

$$\mathbf{E}[X] = \mathbf{E}[X \cdot \mathbf{1}[X > 0]] \leq \sqrt{\mathbf{E}[X^2]}\sqrt{\mathbf{E}[(\mathbf{1}[X > 0])^2]} = \sqrt{\mathbf{E}[X^2]}\sqrt{\mathbf{P}[X > 0]},$$

where we used Cauchy-Schwarz in the the inequality. Rearrangement completes the proof. □

Thus the first thing we should show is that $\mathbf{E}[N]$ is large. Having done this, we will see the upper bound on $\mathbf{E}[N^2]$ and also the fix that truncates the verifier's walks in the next lecture.

Lemma 3.4. $\mathbf{E}[N] \geq \frac{1}{4|\Sigma|^2} \cdot t\frac{|F|}{|E|}$.

Proof. By linearity of expectation, it is enough to argue that for any particular edge $(u, v) \in F$, the expected number of faulty $u \to v$ steps is at least $\frac{1}{8|\Sigma|^2} \cdot \frac{t}{|E|}$. So for $(u, v) \in F$,

$$\begin{aligned}&\mathbf{E}[\text{\# faulty } u \to v \text{ steps}] \\ &\quad = \sum_{k \geq 1} \mathbf{E}[\text{\# faulty } u \to v \text{ steps} \mid \text{exactly } k\ u \to v \text{ steps}] \cdot \mathbf{P}[\text{exactly } k\ u \to v \text{ steps}]. \quad (1)\end{aligned}$$

As we said before, for a given random path, either all $u \to v$ steps are faulty or none is. So we have the relation

$$\mathbf{E}[\text{\# faulty } u \to v \text{ steps} \mid \text{exactly } k\ u \to v \text{ steps}] = k \cdot \mathbf{P}[u \to v \text{ steps are faulty} \mid \text{exactly } k\ u \to v \text{ steps}].$$

We will argue that

$$\mathbf{P}[u \to v \text{ steps are faulty} \mid \text{exactly } k\ u \to v \text{ steps}] \geq \frac{1}{4|\Sigma|^2}. \quad (2)$$

Hence we can lower-bound (1) by

$$\sum_{k \geq 1} k \cdot \frac{1}{4|\Sigma|^2} \cdot \mathbf{P}[\text{exactly } k\ u \to v \text{ steps}] = \frac{1}{4|\Sigma|^2} \cdot \mathbf{E}[\text{number of } u \to v \text{ steps}] = \frac{1}{4|\Sigma|^2} \cdot \frac{t}{2|E|},$$

where the last step follows from: a) the expected total number of steps is easily seen to be t; b) each step is equally likely to be any of the $2|E|$ possibilities (this uses the fact that G is regular); c) linearity of expectation.

It now remains to prove (2). So suppose we condition the verifier's walk on making exactly k $u \to v$ steps. Since (u, v) is in F we have that $u \to v$ steps are faulty for this walk iff:

(i) a is within distance t of u in the graph G and $\sigma'(a)_u = \sigma(u)$;

(ii) b is within distance t of v in the graph G and $\sigma'(b)_v = \sigma(v)$.

We now invoke the lemma from Lecture 4 which says that conditioned on the walk taking exactly k $u \to v$ steps, a is distributed as a B.S.R.W. from u, b is distributed as a B.S.R.W. from v, and a and b are independent. Our task is to show that $\mathbf{P}[\text{(i) and (ii)}] \geq \frac{1}{4|\Sigma|^2}$; since a and b are independent, this probability is equal to $\mathbf{P}[\text{(i)}] \cdot \mathbf{P}[\text{(ii)}]$. We will show that $\mathbf{P}[\text{(ii)}] \geq \frac{1}{2|\Sigma|}$ and the proof for (i) is the same; thus this will complete the proof.

So finally, we must prove $\mathbf{P}[\text{(ii)}] = \mathbf{P}[\text{dist}(b, v) \leq t \text{ and } \sigma'(b)_v = \sigma(v)] \geq \frac{1}{2|\Sigma|}$. By the lemma, if we let w denote a random vertex generated by taking a B.S.R.W. starting at v, then

$$\mathbf{P}[\text{dist}(b, v) \leq t \text{ and } \sigma'(b)_v = \sigma(v)] = \mathbf{P}[\text{dist}(w, v) \leq t \text{ and } \sigma'(w)_v = \sigma(v)]. \tag{3}$$

Now consider the B.S.R.W. that generated w and condition on it *having taken at most t steps*. So

$$\begin{aligned}
(3) &= \mathbf{P}[\text{dist}(w, v) \leq t \text{ and } \sigma'(w)_v = \sigma(v) \mid \text{at most } t \text{ steps taken to generate } w] \times \\
&\qquad \times \mathbf{P}[\text{at most } t \text{ steps taken to generate } w] \\
&= \mathbf{P}[\sigma'(w)_v = \sigma(v) \mid \text{at most } t \text{ steps taken to generate } w] \times \mathbf{P}[\text{at most } t \text{ steps taken to generate } w] \\
&\geq (1/2) \cdot \mathbf{P}[\sigma'(w)_v = \sigma(v) \mid \text{at most } t \text{ steps taken to generate } w], \qquad (4)
\end{aligned}$$

where the last step is because a B.R.S.W. stop within t steps with probability at least $1-(1-1/t)^t \geq 1/2$. But now, finally, if we look at the probability in (4), we see that the distribution on w is *precisely the distribution used in defining $\sigma(v)$* — specifically, that generated by taking a B.S.R.W. from v and conditioning on stopping within t steps. Hence the probability in (4) is actually that the most common (plurality) event occurs for a Σ-valued random variable; this is clearly at least $\frac{1}{|\Sigma|}$, so we are done. □