| | |
|---|---|
| **6.893 Approximability of Optimization Problems** | December 1, 1999 |
| Lecture 23 | |
| *Lecturer: Madhu Sudan* | *Scribe: Anna Lysyanskaya* |

# 1 Introduction: custom-made PCPs

In this lecture, we will see how to tweak PCPs to make them yield hardness of approximation results. First, we will extend the notion of a probabilistically checkable proof with a special zero-knowledge property, and see how this special PCP will give us a hardness result for approximating the chromatic number of a graph. Then we will see how manipulating various PCP parameters gives us a tighter exponent for the hardness of approximating CLIQUE. Thus, this lecture is about applying PCP techniques to arriving at hardness of approximation.

# 2 Chromatic number and zero-knowledge PCP

## 2.1 Review: from PCPs to statements about graphs

Recall that $NP \subseteq PCP_{1,1-\epsilon}[r = \log n, 3]$. From here we amplify error without worrying about wasting randomness and queries, and obtain $NP \subseteq PCP_{1,(1-\epsilon)^k}[r + 3k, 3k]$. Then we apply the FGLSS transformation covered a few lectures ago. This transformation is described in Figure 2.1.

Recall how the transformation works. First, we note that there are $2^r$ possible questions, a different question for a different set of random bits of the verifier; and $2^q$ possible answers to each question, corresponding to the $q$ bits of the proof that are read. The graph $G$ has a vertex for every possible question/answer combination that induces the verifier to accept. Two vertices have an edge if they stand for two question/answer pairs that are mutually inconsistent.

As we have seen a few lectures ago, as a result of this transformation, we obtain a graph on $N$ vertices with the feature that $N \approx 2^{6k+r}$ and (1) completeness: if the NP theorem was true, then $G$ has an independent set of size $2^{r+3k} \approx N^{1/2}$; (2) soundness: if the theorem was not true, then the size of the largest independent set in graph $G$ is at most $N^{1/2+\delta}$.

## 2.2 Approaches with regular PCP

What does this tell us about the approximability of the chromatic number? We see that, if the original theorem was false, then the chromatic number of the resulting graph would be large: $\chi(G) \geq \frac{N}{N^{1/2-\delta}} = N^{1/2+\delta}$. In order for this to be of any use for proving hardness of approximating the chromatic number, we need that, in case the theorem was true, we obtain a graph with a small chromatic number, $\chi(G) \leq N^{1/2}$.

Suppose we try to achieve something like that by transforming a verifier for some hard-to-approximate problem to an FGLSS graph. This is just a Karp reduction. Let us see what features of such a transformation might come in handy for proving a hardness of approximation result for chromatic number.

- We want a small number of independent sets to cover all vertices in the graph. (Roughly this means that every vertex belong to a large independent set.)
- Every independent set corresponds to a proof of our theorem. In other words, for every random string, and every possible accepting transcript that can be generated in response to this random string, there exists a proof that gives this transcript.

This is a (very informal) motivation for the zero-knowledge property.

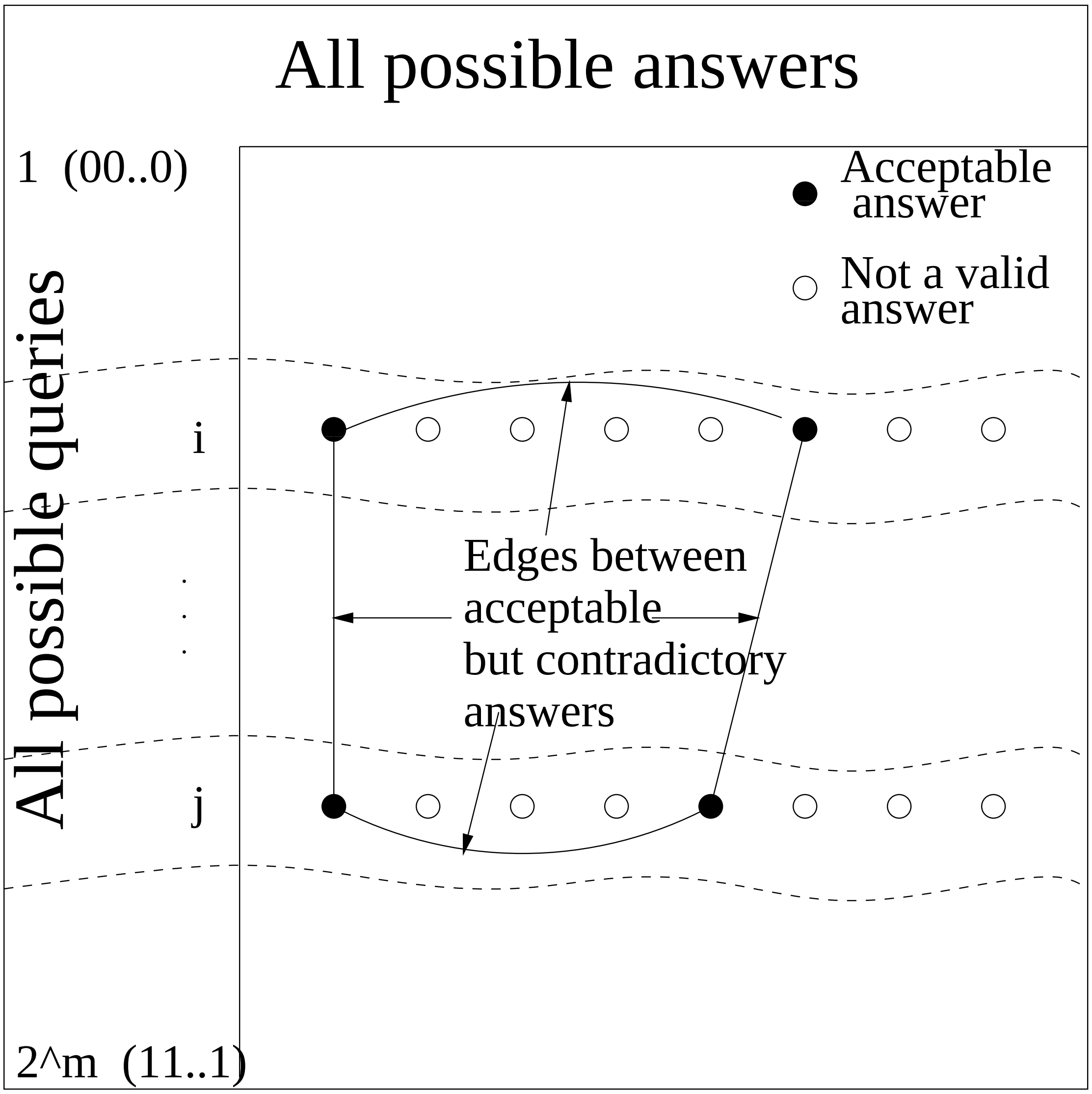


**Figure 1**: The FGLSS graph

## 2.3 Zero-knowledge PCP

Let us now formulate what we mean by zero-knowledge probabilistically checkable proofs.

**Definition 1** *$L \in ZKPCP_{1,s}[r,q]$ if there exists a verifier such that the usual completeness and soundness conditions are satisfied, and, beyond that, the zero-knowledge property:*

**Zero-knowledge property:** *if $x \in L$, then there exists a distribution $D$ on valid PCPs of this theorem, such that if $V$'s randomness is $R$, and $P$ picks proof $\pi$ from distribution $D$, then the transcript of their interaction is random and accepting.*

The reason that this notion is called "zero-knowledge" is that the intuition is similar to the cryptographic zero-knowledge: it captures that the honest verifier does not learn anything from the process, since he can generate a random accepting transcript himself. Of course, here we don't care whether the verifier learns anything or not, all we want is for the FGLSS graph $G$ to have chromatic number $\chi(G)$ equal to the number of valid probabilistically checkable proofs of a given theorem; and if the theorem does not have a valid proof, then the chromatic number of the FGLSS graph cannot be small.

## 2.4 Relevance to the chromatic number problem

And now, more formally, why would it help us to know that $NP \subseteq ZKPCP_{1,s}[r,q]$? It turns out that this will indeed imply hardness of approximation for chromatic number.

**Theorem 2** *If $NP \subseteq ZKPCP_{1,s}[r,q]$, then approximating chromatic number to within a factor of $\frac{1}{s \log N}$ is hard in $N$-vertex graphs, where $N \approx 2^{r+q}$.*

**Proof** We will prove the theorem by considering the FGLSS transformation of the ZKPCP of an NP theorem.

**Lemma 3** *Suppose the theorem is true. Then $\chi(G) \leq 2^q \log N \approx 2^q(r+q)$.*

We prove the lemma by probabilistic method. Recall that, by the zero-knowledge property, there is a proof for every accepting transcript. Therefore, for every vertex of $G$, there exists a proof that covers it. Then the probability that a random proof covers a given vertex $v$ is at least $2^{-q}$. Then, after $2^q l$ proofs are chosen at random, the probability that $v$ is still not covered is at most $2^{-l}$. By the union bound, the probability that there exists a vertex $v$ that is not covered after $2^q l$ proofs have been selected at random, is at most $N2^{-l}$. Therefore, there exists a way of covering the entire graph with $2^q l$ proofs provided that $2^l < N$. Since each proof is an independent set, we may associate a color with each proof, and cover a vertex according to the color of one of the proofs it belongs to. Therefore, we obtain that $A = \chi(G) \leq$ Number of covering proofs $\leq 2^q \log N$.

**Lemma 4** *Suppose the theorem is false. Then $\chi(G) \geq \frac{2^q}{s}$.*

Recall that if the theorem is false, then no independent set of $G$ can have more than $s2^r$ vertices. Then we immediately get the claimed bound:

$$B = \chi(G) \geq \frac{2^{r+q}}{s2^r} \geq \frac{2^q}{s}$$

Let us put Lemma 3 and Lemma 4 together. This gives us the result that approximating chromatic number in graphs on $N = 2^{r+q}$ vertices is hard to approximate to within $\frac{B}{A} = \frac{2^q}{s}\frac{1}{2^q \log N} = 1/s \log N$. ■

Suppose that $NP \subseteq ZKPCP_{1,1-\epsilon}[r = \log n, q = 3]$. Our goal is to prove that approximating chromatic number to within a factor of $N^\delta$ is NP-hard, for some constant $\delta$. $N^\delta = 1/s \log N$, so we need

$s \approx N^{-\delta}$, which we would achieve if we could push the soundness $s$ of the ZKPCP to a value like this, i.e. if our result was $NP \subseteq ZKPCP_{1,(1-\epsilon)^k}[3k \log n, 3k]$.

How do we amplify it to achieve $NP \subseteq ZKPCP_{1,(1-\epsilon)^k}[3k \log n, 3k]$, improving soundness? Let $k = c \log n$. Let $N = 2^{\log n + 6k}$, then new $r \approx \frac{1}{2} \log N$, and new $q \approx \frac{1}{2} \log N$. We can apmplify the usual way, the only point to take care of here is that we need to show there is a distribution on the PCPs for the new theorem that satisfies the zero-knowledge property.

## 2.5 NP is in ZKPCP

What remains to be shown is that $NP \subseteq ZKPCP_{1,q-\epsilon}[\log n, 3]$. We need to pick an appropriate NP-complete problem for which there is a PCP that satisfies the zero-knowledge property. Consider the following problem:

**Not-All-Equal MAX SAT** :

**Instance:** $n$ variables $x_1, \ldots, x_n$, and $n$ clauses, $c_1, \ldots, c_n$. Each clause $c_i$ is a list of three variables, $x_{i_1}$, $x_{i_2}$, $x_{i_3}$.

**Task:** An assignment to $\{x_j\}$ satisfies a clause $c_i$ if $x_{i_1}$, $x_{i_2}$, and $x_{i_3}$ are not all assigned the same value. Find an assignment that satisfies the maximum number of clauses.

For this problem, it is known that it is hard to distinguish between an instance that is satisfiable, and an instance where at most a $1 - \epsilon$ fraction of the clauses are satisfiable, where $\epsilon$ is some positive constant. This can be obtained via a gadget reduction to MAX 3SAT (using the same gadget constructed by Schaefer to reduce the decision version of this problem to 3SAT).

A canonical proof that instance $\phi$ is in $NAE - SAT$ is just the assignment to the variables that satisfies every clause. Note that if $NAE - SAT$ has a satisfying assignment $\vec{x}$, then the assignment $\vec{y}$ obtained by negating every variable in $\vec{x}$ is also satisfying. To see this, note that if clause $c_i$ is satisfied by assignment $x_{i_1} = b_1$, $x_{i_2} = b_2$, and $x_{i_3} = b_3$, then not all $b_j$'s are equal, and so if we negate them *all* then they are still not all equal. Note that from here it follows that if $\phi$ is satisfiable, then there is a proof that assigns 0's to exactly two out of the three variables in $c_i$, and there is also a proof that assigns 0's to exactly one of the the three variables in $c_i$. Based on these observations, consider the following PCP system:

**Prover's output** : The Prover outputs a proof $\pi$ that consists of the following:

1. The canonical proof: a satisfying assignment $a_1, \ldots, a_n$.
2. For each clause $c_i$, a permutation $\sigma_i \ : \ \{1,2,3\} \mapsto \{1,2,3\}$.
3. For each clause $c_i$, an assignment to the variables in the clause, in the order permuted by $\sigma$: $b_i^1 = a_{i_{\sigma(1)}}$, $b_i^2 = a_{i_{\sigma(2)}}$, $b_i^3 = a_{i_{\sigma(3)}}$.

**Verifier's actions** : The verifier picks $1 \leq i \leq n$ uniformly at random and performs the following checks:

1. Read $b_i^1$, $b_i^2$, $b_i^3$, and check that they are not all equal.
2. Read $\sigma_i$ (check that it is a permutation), and check that $b_i^1 = a_{i_{\sigma(1)}}$.
3. Read $\sigma_i$ (check that it is a permutation), and check that $b_i^2 = a_{i_{\sigma(2)}}$.
4. Read $\sigma_i$ (check that it is a permutation), and check that $b_i^3 = a_{i_{\sigma(3)}}$.

So far, this is not a zero-knowledge verifier, since he cannot generate a valid accepting transcript by himself. For example, we can imagine that no satisfying assignment can exist in which $x_1$ and $x_2$ are the same, while a random accepting transcript might allow them to be the same.

However, to make it zero-knowledge, we just make the verifier pick $1 \le k \le 4$ at random, and have him execute Check $k$ only. Now, the zero-knowledge property is satisfied, because if the theorem is true, then any accepting transcript can be generated by some proof:

1. Suppose $k = 1$. Then an accepting transcript is just a transcript of three bits that are not all equal, without indices that would tell which bit corresponds to which variable. Any such three bits correspond to a valid proof, because: (1) if there is a satisfying assignment for which $c_i$ has one 0, then there is a satisfying assignment for which $c_i$ has two 0's (see the observations above); and (2) for any satisfying assignment for which $c_i$ has $m$ 0's, there exists a permutation $\sigma_i$ that places these 0's in the right places.

2. Suppose $2 \le k \le 4$. Then an accepting transcript consists of a permutation $\sigma_i$ and two bits $b_i^{k-1}$ and $a_{i_{\sigma(k-1)}}$ that must be equal to each other. The permutation in a real proof is just a random permutation, thus any permutation seen in a valid transcript can be seen in a valid proof. But can $b_i^{k-1} = a_{i_{\sigma(k-1)}}$ take any value? Yes, since for any satisfying assignment $\vec{a}$, the negation of $\vec{a}$ is also a satisfying assignment (see the observations above).

### 2.6 Conclusions

We just proved the following theorem:

**Theorem 5** $NP \subseteq ZKPCP_{1,1-\frac{1}{4n}}[\log n + 2, c]$.

From which, as we have discussed above, it follows that:

**Theorem 6** $NP \subseteq ZKPCP_{1,(1-\epsilon)^k}[3k \log n, 3k]$.

In turn, as we have shown in Theorem 2, this implies a hardness of approximation result for chromatic number:

**Theorem 7** *Chromatic number of a graph is hard to approximate to within a factor of $N^\delta$ for some constant $\delta$.*

## 3 Getting at the right exponent

Recall that we have a related result for CLIQUE:

**Theorem 8** *CLIQUE is hard to approximate to within a factor of $N^\delta$ for some constant $\delta$.*

(We saw this result a few lectures ago; recall that CLIQUE is the flipside of the independent set problem).

The question is, what is the best possible exponent? We may obtain insight into this matter by manipulating the various parameters of PCPs. From $NP \subseteq PCP_{1,1/2}[r,q]$, we amplify and get $NP \subseteq PCP_{1,2^{-k}}[r+ck, qk]$, which, by the FGLSS reduction, implies that approximating clique size to within $2^k$ in an $N = 2^{k(c+q)+r}$ vertex graph is NP-hard. Let us ignore $r$, and we get the exponent $\delta = N^{\frac{1}{c+q}}$ such that approximating maximum clique size up to a factor of $N^\delta$ in an $N$ vertex graph is NP hard.

Thus, the lower the values of $c+q$ that we can tolerate, the tighter the exponent $\delta$. The best known value of $c$ is approximately 2. Therefore, by this kind of reduction, we get that clique size is hard to approximate to within a factor of $N^{1/5}$.

We can improve this by using a randomized reduction to get from $NP \subseteq PCP_{1,1/2}[r,q]$ to $NP \le_R PCP_{1,2^{-k}}[r+ck, qk]$ (where $\le_R$ denotes a randomized reduction). It turns out that using such randomized reductions, we can get $c$ arbitrarily close to 1.

We can also manipulate the soundness in the reduction and save a constant factor of $\delta$: if we start with better soundness but higher query complexity, e.g. from $NP \subseteq PCP_{1,1/4}[r,5]$, then it turns out that $\delta$ we obtain will be higher. This motivates the study of amortized query complexity.

### 3.1 Amortized query complexity of PCP

Suppose that a PCP has perfect completeness, makes $q$ queries, has soundness $2^{-k}$ and makes $q$ queries, then the amortized query complexity is defined as $\bar{q} = \frac{q}{k}$. Then we get hardness of approximating the clique problem to within a factor of $N^{\frac{1}{c+\bar{q}}}$. Best known (and actually the best possible) results give $\bar{q} = 1$ and thus give hardness results for clique approxmation to within $N^{1/3-\epsilon}$, unless NP=P, and to within $N^{1/2-\epsilon}$, unless NP=RP. This is still not getting us to the right answer and so we study the proof systems one more time.

### 3.2 Free bits

Observe that it may be that the third bit read from a valid proof is determined by the first two. In that case, the third bit is not "free." How is this relevant to hardness of approximation? Suppose we are given a $PCP[r, q]$. Observe that in the FGLSS graph (recall the graph from Figure 2.1), we always upper bounded the number of vertices by $2^{r+q}$. But this is a really weak upper bound. For example, in Håstad's verifier, the third bit read depended on the first two. Our usual bound on the number of vertices in this graph would be $82^r$, where $r$ is the length of the random string used by this verifier. But the correct count is actually less — the number of vertices is only $42^r$. Thus our hardness result actually held for a smaller sized graph than we were claiming. This factor of two does not appear to be a significant savings; but it turns out to be very significant, when we take into account the fact that we are going to take graph products and amplify the clique size of this graph. Keeping this in mind, we define a formal parameter, called free bit, that studies the number of degrees of freedom of a given verifier.

**Definition 9** *If a PCP verifier has $2^f$ accepting configurations, then the free bit complexity of this PCP system is $f$.*

(Note that a bit is "free" in the sense that its value is not determined by other bits. But it might be an unfortunate name, because it is exactly the free bits that we end up "paying" for: the more free bits, the worse the hardness result we get.)

Let us use FPCP$[r, f]$ to describe languages whose verifiers are restricted to using $r(\cdot)$ bits of randomness and $f$ free bits. The FGLSS-reduction can now be refined to state that if NP $\subseteq$ FPCP$_{c,s}[r, f]$, then approximating the clique size to within a factor of $c/s$ in a graph with $2^{r+f}$ vertices is NP-hard. Similarly, we can define amortized free bit complexity, and get further refinement on the hardness exponent, which now states that if NP $\subseteq$ FPCP$_{1,2^{-k}}[\log n, k\bar{f}]$ then approximating clique size to within $N^{\frac{1}{c+\bar{f}}}$ is hard, unless NP=RP. Håstad shows how to get $\bar{f} = \epsilon$ for any positive $\epsilon$ and thus shows that clique is hard to approximate to within $N^{1-\epsilon}$ for any positive $\epsilon$. Feige & Kilian show this verifier can also be endowed with the Zero-Knowledge property and thus get a matching result for approximating the chromatice number.

# 4 References


Feige, Goldwasser, Lovasz, Safra and Szegedy: approximating clique. JACM, MArch 1996.
Lund and Yannakakis: approximating $\chi(G)$. JACM, September 1994.
Feige and Kilian: ZKPCP. JCSS, October 1998.
Håstad: $N^{1-\delta}$ hardness for clique. FOCS 1996.
Samorodnitsky and Trevisan: Alternative proof of above and $\bar{q} = 1$. To be accepted in STOC 2000.