

Introduction to the PCP Theorem

Pradipta Mitra
Department of CS
Yale University
`pradipta.mitra@yale.edu`

January 31, 2005

Overview

1. Motivation
2. Proving Hardness of Approximation
3. Probabilistically Checkable Proofs
4. How does 3 help with 2
5. Bit of History

Why?

The theory NP hardness has provided a fairly complete characterization of natural computational problems, at least as far as determining whether a problem is solvable in polytime or not (provided $P \neq NP$). Notable exception: Graph Isomorphism.

The field in which a lot more has to be done is hardness of approximation

Hardness results usually fall into the following 3 classes (for minimization problems):

1. $constant > 1$
2. $\Omega(\log n)$
3. n^ϵ

The hope is: PCP together with other techniques will give better understanding of hardness of approximation.

Proving Hardness of Approximation

- Always had a way: Show if we have a α (possibly a function of the input size) approximation to problem A , we could solve the NP-hard problem B *exactly*.
- Example: It can be shown that a $2 - \epsilon$ (for any $\epsilon > 0$) approximation algorithm for the metric k -center problem can be used to solve the dominating set problem.
- But reductions from exact problems is not always easy. We want to use already proved hardness of *approximation* results to prove new results.
- Need Gap Introducing and Gap Preserving reductions.

Gap Introducing Reduction:

This is the bootstrapping part. We prove statements like:

Let Π be a minimization problem. A gap-introducing reduction from SAT to Π is one, that given an instance ϕ of SAT, it outputs in polynomial time, an instance x of Π such that

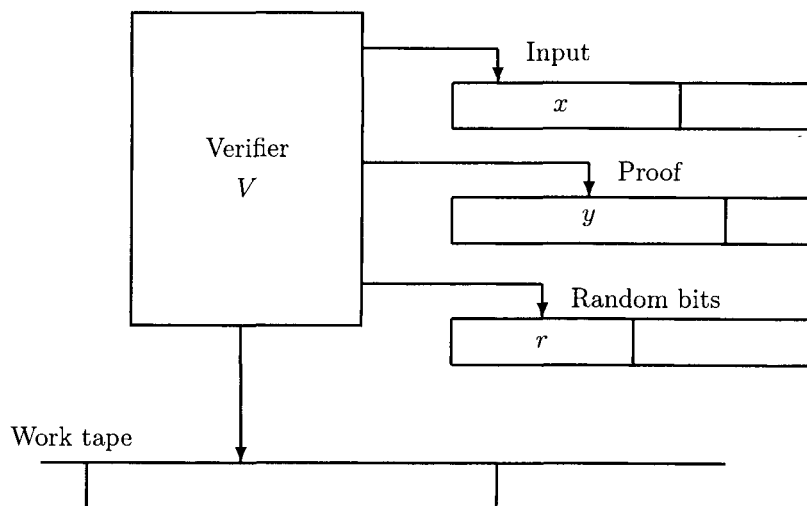
- if ϕ is not satisfiable $OPT(x) \leq f(x)$
- if ϕ is satisfiable $OPT(x) > \alpha(|x|)f(x)$

Gap Preserving Reduction:

A gap preserving reduction comes with four parameters, f_1, α, f_2 and β . Given an instance x of Π_1 (minimization problem), it (poly) computes an instance y of Π_2 such that,

- if $OPT(x) \leq f_1(x)$ then $OPT(y) \geq f_2(x)$
- if $OPT(x) > \alpha(|x|)f_1(x)$ then $OPT(y) < \beta(|x|)f_2(x)$

The PCP System



- Probabilistic characterization uses the familiar concept of a verifier and a proof.
- A Probabilistically checkable proof system comes with two parameters, the number of random bits required by the verifier, and the number of bits that the verifier is allowed to examine.
- The most useful setting of these parameters is $O(\log n)$ and $O(1)$, respectively. This defines the class $PCP(\log n, 1)$.

More formally, a language $L \in PCP(\log n, 1)$ if there is a verifier V , with constants c and q such that on input x , V obtains a random string, r , of length $c \log |x|$ and queries q bits of the proof. Moreover,

- if $x \in L$, then there is a proof y that makes V accept with probability 1,
- if $x \notin L$, then for every proof y , V accepts with probability $< \frac{1}{2}$.

An easy relation between NP and PCP can be, $NP = PCP(0, poly(n))$.

The PCP Theorem

In 2 landmark papers, a number of people (Arora, Lund, Motwani, Safra, Sudan and Szegedy) proved the following theorem:

Theorem 1. $NP = PCP(\log n, 1)$

Proof.

1. $PCP(\log n, 1) \subseteq NP$: Simulate all possible random bits and all possible proof bits. Small detail?
2. $NP \subseteq PCP(\log n, 1)$: Left to the reader.

□

A feel for the proof

Relatively easy to construct a verifier for 3SAT whose error probability (i.e., probability of accepting unsatisfiable formulae) is $\leq 1 - 1/m$, where m is the number of clauses in the input 3SAT formula, say ϕ . The verifier accepts a satisfying truth assignment to ϕ a proof. Reads a random clause, and tests the assignment on it. Clearly, error probability $\leq 1 - 1/m$.

Great intellectual achievement, but also has applications.

Hardness of approximation

Let's try (as far as we can) to prove the following theorems:

Theorem 2. *There is a gap introducing reduction from SAT to MAX-3SAT such that:*

- *if ϕ is satisfiable, $OPT(\psi) = m$, and*
- *if ϕ is not satisfiable, $OPT(\psi) < (1 - \epsilon_M)m$*

ϕ is the instance of SAT and ψ is the instance of MAX-3SAT.

Theorem 3. *There is a gap preserving reduction from MAX-3SAT to MAX-3SAT(29) such that:*

- *if $OPT(\phi) = m$, then $OPT(\psi) = m'$, and*
- *if $OPT(\phi) < (1 - \epsilon_M)m$, then $OPT(\psi) < (1 - \epsilon_b)m'$*

ϕ is the instance of MAX-3SAT and ψ is the instance of MAX-3SAT(29), m and m' are the number of clauses respectively and $\epsilon_b = \epsilon_M/43$.

Theorem 4. *There is a gap preserving reduction from MAX-3SAT(29) to VC(30) [Vertex Cover with degree ≤ 30] such that:*

- if $OPT(\phi) = m$, then $OPT(G) \leq \frac{2}{3}V$, and
- if $OPT(\phi) < (1 - \epsilon_b)m$, then $OPT(G) > (1 + \epsilon_v)\frac{2}{3}V$

ϕ is the instance of MAX-3SAT(29) and $G = (V, E)$ is the instance of VC(30), and $\epsilon_v = \epsilon_b/2$.

Shall we?

Theorem 2

Proof. MAX k-FUNCTION SAT Given n boolean variables $x_1, x_2 \dots x_n$ and m functions $f_1, f_2 \dots f_m$, each of which is a function of k of the boolean variables (k is a constant), find a truth assignment that maximizes the number of functions satisfied.

Lemma There is a constant k for which there is a gap introducing reduction from SAT to MAX k-FUNCTION SAT such that

- if ϕ is satisfiable $OPT(I) = m$
- if ϕ is not satisfiable $OPT(I) < \frac{1}{2}m$

Proof. V is a $PCP(\log n, 1)$ verifier for SAT, with parameters c and q . Total qn^c different bits might be read. Have a boolean variable for each of these bits. Set $k = q$. Corresponding to each possible random

string r , define boolean function f_r as the restriction of acceptance/rejection of V to the corresponding q bits.

We can compute f_r in polynomial time. The ψ is satisfiable, all f_r 's are as well and $OPT(I) = m = n^c$. If ϕ isn't, the acceptance probability $< 1/2$ and clearly less than half the f_r 's can be satisfied.

□

Now to prove the theorem we set out to prove in the iron ages, convert SAT to MAX k-FUNCTION SAT. What remains is to a reduction from that to MAX3-SAT.

Write each f_r as a SAT formula ψ_r . Create a MAX-SAT instance by taking a huge conjunct of all these formulae, i.e. $\psi = \bigwedge_r \psi_r$.

Exists standard trick to convert this MAX-SAT instance to a MAX 3-SAT problem. One clause with k literals is replaced by $k - 2$ clauses. So we end up with $n^c 2^q (q - 2)$ clauses. If ϕ is not satisfiable any truth assignment will leave $> \frac{1}{2} n^c$ clauses unsatisfied. Setting $\epsilon_M = 1/2^{q+1} (q - 2)$ gives the theorem.

□

Theorem 3

Proof. Critically uses expander graphs. □

Theorem 4

Proof. Assume wlog that each clause has exactly 3 literals. For each clause, G will have 3 vertices, totalling $3m$ vertices. For each clause, there is a complete graph between the three vertices. And there is an edge between two vertices in V if they are negations of each other. G has degree at most 30.

Claim: The size of the maximum independent set in G is precisely $OPT(\phi)$.

Now, the complement of a maximum independent set in G is a minimum vertex cover. Hence if $OPT(\phi) = m$ then $OPT(G) = 2m$. If $OPT(\phi) < (1 - \epsilon_b)m$, then $OPT(G) > (2 + \epsilon_b)m$. Theorem follows.

□

A bit of tabular History

NP-completeness	Cook, Levin, Karp
RP, BPP and ZPP	Gill, Rabin
MAX-SNP-completeness	Papadimitrou, Yannakakis
Interactive Proof Systems	Goldwasser, Micali, Rackoff, Babai
PCP based hardness results	Feige, Goldwasser, Lovasz, Safra, Szegedy
PCP Systems	Arora, Safra
Proof of the PCP Theorem	Arora, Safra, Lund, Motwani, Sudan, Szegedy
More Hardness results based on PCP	Hastad ...

Thank you!