On the Standard Written Proof

Johan Håstad

IAS / Royal Inst. of Technology

Overall goal

Inapproximability of basic optimization problems

through

Efficient Probabilistically Checkable Proofs

Can we standardize?

Approximation

Given (NP-hard) optimization problem O.

Find, in polynomial time, a reasonably good solution.

Approximation ratio, maximization problems.

$$\frac{\text{optimal value}}{\text{found value}}$$

Approximation algorithms

An algorithm is a C-*approximation* algorithm if it on *any input* gives an output with ratio at most C.

Key question: For a given NP-hard optimization problem what is the smallest value of C achievable in (probabilistic) polynomial time.

Probabilistically Checkable Proofs (PCP)

A proof that a formula φ is satisfiable.

Checked by a probabilistic Verifier reading a small number of bits.

completeness If φ is satisfiable exists a proof to make V accept with probability c.

soundness If φ is not satisfiable, no proof makes V accept with probability more than s.

$c \approx 1$

s a constant < 1.

More PCPs

The written proof is denoted Π. Verifier V.

One Π can be checked in interesting ways by many different V.

Completeness is usually easy, naturally correct proof easy to analyze.

Soundness usually difficult, has to deal with any weird proof.

Fundamental Result
PCP-theorem [ALMSS]

A polynomial time transformation:

$\varphi \mapsto \psi$

$$\begin{aligned} \varphi \text{ satisfiable} \quad &\Rightarrow \psi \text{ satisfiable} \\ \varphi \text{ not satisfiable} \quad &\Rightarrow \psi \text{ at most}(1-\epsilon)\text{-satisfiable} \end{aligned}$$

No assignment satisfies more than fraction $(1-\epsilon)$ of the clauses.

Each clause of length exactly 3.

Each variable appears exactly 5 times.

ALMSS-transformation is a PCP!

The written proof is an assignment of the variables of ψ.

Checking: Create ψ, pick a random clause and read variables occurring.

Completeness: $c = 1$.

Soundness: $s = 1 - \epsilon$.

Constraint Satisfaction Problems

Satisfy as many as possible of

$$P(x_{i_1}, x_{i_2}, \ldots x_{i_k})$$

- Linear equations mod 2, with k variables in each equation.
- Max-k-Sat.
- Max-cut, set-splitting.

General approach for CSP

To prove inapproximability of CSP given by P.

Design a PCP where accept criteria is given by P.

Π might not need to depend on P, but surely V does!

A Standard Written Proof (SWP(u))

Let φ be an ALMSS-3CNF formula, x^0 a satisfying assignment.

$$\Pi(f) = f(x^0)$$

for all f that depends only on u bits of x^0.

Size $\approx n^u 2^{2^u}$, u constant or $\log\log n$.

Applications

This proof is used for most (all?) strong inapproximability for CSPs. We here study.

- Linear systems of equations mod 2 with 3 variables in each equation.

- Max-4Sat, inapproximability for satisfiable instances.

Alternative description

The *long code* [BGS] of a vector $x \in \{0,1\}^t$ is

$$f(x), \text{all } f : \{0,1\}^t \mapsto \{0,1\}$$

An $SWP(u)$ is the long codes of all restrictions of x^0 to sets of size at most u.

Inapproximability for CSP given by P

Pick k function $f_1, f_2 \ldots f_k$ such that

$$P(f_1(x), f_2(x), \ldots f_k(x))$$

true for each x, possibly only with high probability.

Check

$$P(\Pi(f_1), \Pi(f_2), \ldots \Pi(f_k)).$$

General inapproximability theorem

Theorem: If completeness is c and soundness is s of this PCP then CSP given by P is hard to approximate within c/s.

Proof: The variables are the bits of the proof. Conditions are the checks of PCP.

φ satisfiable	$\Rightarrow$	Can satisfy $c-$ fraction
φ not satisfiable	$\Rightarrow$	Can only satisfy $s-$ fraction

Completeness

For a correct $SWP(u)$ we have $\Pi(f_i) = f_i(x^0)$ for the satisfying assignment x^0.

$$P(\Pi(f_1), \Pi(f_2), \ldots \Pi(f_k)) = \\ P(f_1(x^0), f_2(x^0), \ldots f_k(x^0))$$

which, by construction, is (almost) always true.

Completeness is 1 or almost 1 depending on construction.

Soundness

Needs a clever choice of functions f_i.

Major problem to avoid: Cheating proofs that locally look OK.

I.e. when looking only at functions that depend on the variables in a set U we have the long code of an assignment that satisfies the clauses supported on U.

Steps

- Using tools from 2 Prover protocols we relate local properties to global properties.

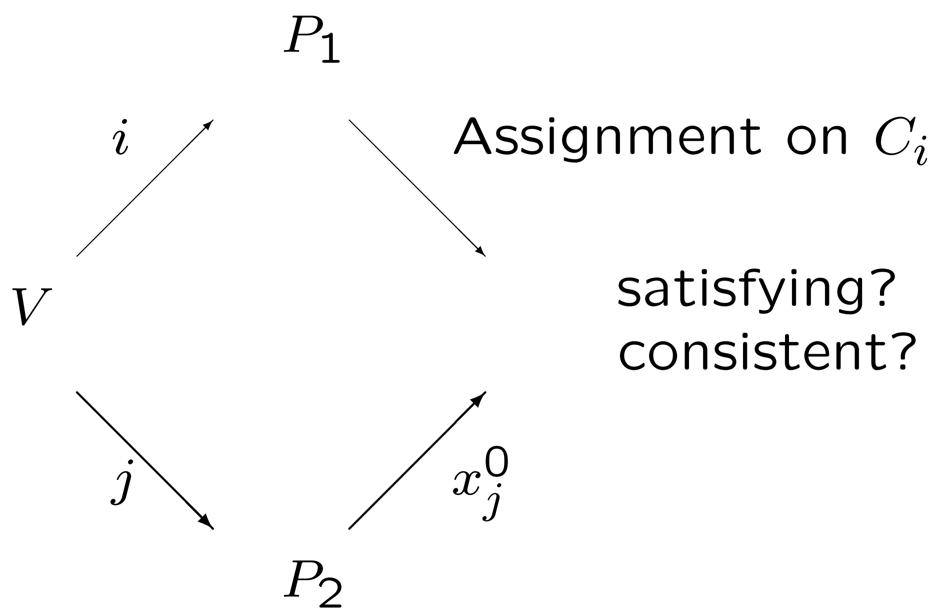
- We extract local information from the local long codes.

2-Prover protocols

Provers cooperate, make plans but may not communicate during protocol.

$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$.

$i \in_R [m]$, $j : x_j \in_R C_i$.



Completeness 1.

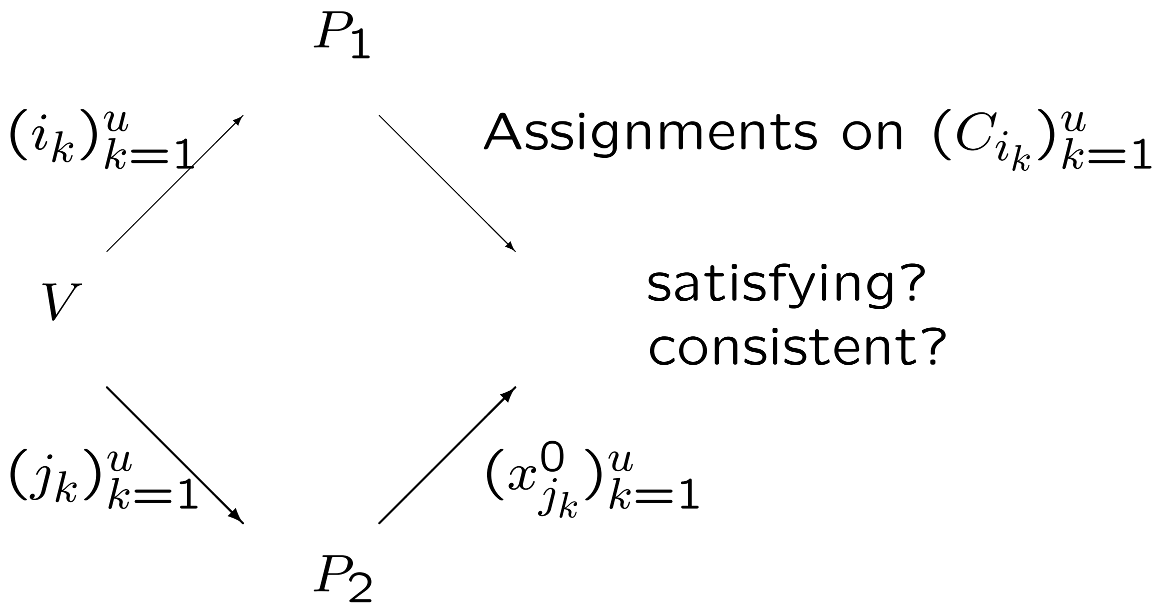
Soundness $1 - \epsilon/3$.

Parallel repetition

We repeat the basic protocol u times in parallel.

$i_1, i_2, \ldots i_u \in_R [m]$, independently.

$j_1, j_2, \ldots j_u : x_{j_k} \in_R C_{i_k}$, independently.



Completeness 1.

Soundness c_ϵ^u [R].

Moral

Finding locally satisfying, locally consistent assignments is sufficient to guarantee a global satisfying assignment.

Locally satisfying Assignments on $3u$ variables from u clauses. Call such a set W.

Locally consistent Assignments on u variables one from each of u clauses. Call such a set U.

Extracting local assignments

Take a incorrect long code on U, i.e. $\Pi(f)$ is not constructed as $f(x)$ for any x. How to find an associated assignment?

Naive Take x such that $\Pi(f) = f(x)$ for many x.

Better Look at the Fourier transform of Π.

Fourier transforms

Move to ± 1 notation, $0 \mapsto 1$, $1 \mapsto -1$, xor becomes multiplication.

$$\hat{\Pi}_\alpha = 2^{-2^u} \sum_f \Pi(f) \prod_{x \in \alpha} f(x),$$

with inversion

$$\Pi(f) = \sum_\alpha \hat{\Pi}_\alpha \prod_{x \in \alpha} f(x)$$

$\alpha \subseteq \{-1, 1\}^U$.

Plancherel: $\sum_\alpha \hat{\Pi}_\alpha^2 = 1$.

Generic strategy

Let A be the long code on U. Take α such that

- $\widehat{A}_\alpha^2$ is large.

- $|\alpha|$ small.

A local assignment used by a prover given the set U is given by a random $x \in \alpha$.

Naive is special case

If α is a singleton set $\{x\}$ then $\hat{\Pi}_\alpha$ is the fraction more than half for which $\Pi(f) = f(x)$.

Assignments that qualify for the naive condition for output qualify also for the better one.

Conditioning

To make sure that local assignments satisfy local constraints we proceed as follows.

f chosen function.

h constrains on the active variables.

$\Pi'(f) = \Pi(f \wedge h)$.

This mechanism ensures

$\hat{\Pi}'_\alpha \neq 0$, $x \in \alpha \Rightarrow h(x)$ true.

Local constraints simply disappear.

Folding

Another useful restriction on cheating provers.

$\Pi(f) = -\Pi(-f)$

by having one bit for each pair $(f, -f)$.

Needs negation to be present in optimization problem, does not work for Max-Cut or Set-Splitting.

Fourier transform side. Makes all nonzero coefficients be of odd size (they are all nonempty).

Overall plan

To prove inapproximability for constraint satisfaction given by P.

- Choose $f_1,\ f_2, \ldots f_k$ with

  $$P(f_1(x), f_2(x), \ldots f_k(x))$$

  true for (almost) all x.

- Analyze using Fourier expansion giving strategies for u-parallel 2-prover protocol.

Easiest case

P parity of 3 inputs.

U as in 2-prover protocol, $(i_k)_{k=1}^u$.

W as in 2-prover protocol, variables of $(C_{j_k})_{k=1}^u$.

f_1 random function on U

f_2 random function on W

$f_3(x) = f_2(x) f_1(\pi(x))$ except for fraction ϵ.

Check if $\Pi(f_1)\Pi(f_2)\Pi(f_3) = 1$.

The analysis

Lemma: The test accepts a correct SWP with probability $1-\epsilon$.

Lemma: For any δ, sufficiently large u the soundness is $(1+\delta)/2$.

Theorem: For any $\epsilon > 0$, it is NP-hard to approximate maximum linear equations mod 2 within $2-\epsilon$.

Outline of soundness proof

A long code on U.

B long code of W.

$A(f_1)B(f_2)B(f_3) = 1$ if accept and -1 otherwise.

Acceptance with probability $\frac{1+\delta}{2}$ implies

$E[A(f_1)B(f_2)B(f_3)] = \delta$

After simple calculations

$$E[A(f_1)B(f_2)B(f_3)] = \sum_\beta \widehat{B}_\beta^2 \widehat{A}_{\pi_2(\beta)}(1-2\epsilon)^{|\beta|},$$

where

$$\pi_2(\beta) = \{x \mid \text{odd \# of } y \in \beta, \pi(y) = x\}.$$

Strategies in two prover game

P1 Upon receiving W, pick a β with probability $\widehat{B}_\beta^2$ and returns a random $y \in \beta$.

P2 Upon receiving U, pick a α with probability $\widehat{A}_\alpha^2$ and returns a random $x \in \alpha$.

Accept-probability at least

$$\sum_\beta \widehat{B}_\beta^2 \widehat{A}_{\pi_2(\beta)}^2 |\beta|^{-1}$$

which is at least

$$\epsilon \left(\sum_\beta \widehat{B}_\beta^2 \widehat{A}_{\pi_2(\beta)} (1-2\epsilon)^{|\beta|} \right)^2 .$$

Another example

Theorem $\forall \epsilon > 0$, it is NP-hard to distinguish satisfiable 4-CNF formulas from those where only a fraction $\frac{15}{16} + \epsilon$ can be satisfied.

Need to find f_1, f_2, f_3, f_4 such that

$f_1(x) \vee f_2(x) \vee f_3(x) \vee f_4(x)$ is true for any x.

Details 4-Sat PCP

U, W as in 2-prover protocol.

f_1 random function on U

f_2, f_3 random function on W

$$f_4(x) = \begin{cases} \text{random} & f_2(x) \text{ true} \\ -f_1(\pi(x))f_3(x) & f_2(x) \text{ false} \end{cases}$$

Check that $\Pi(f_1) \vee \Pi(f_2) \vee \Pi(f_3) \vee \Pi(f_4)$ is true.

Lemmas to prove

Lemma: The verifier always accepts a correct SWP.

Lemma: If the verifier accepts with probability $\frac{15}{16} + \epsilon$ then we can find a strategy for the provers in the two-prover game that makes the verifier accept with probability $\epsilon^2/4$.

Establishing soundness

It turns out that

$$\Pi(f_1) \vee \Pi(f_2) \vee \Pi(f_3) \vee \Pi(f_4) =$$
$$-1 + \frac{1}{8}\prod_{i=1}^{4}(1 + \Pi(f_i)) =$$
$$-\frac{7}{8} + \frac{1}{8}(\Pi(f_1) + \Pi(f_2) + \Pi(f_3) + \Pi(f_4) +$$
$$\Pi(f_1)\Pi(f_2) + \Pi(f_1)\Pi(f_3) + \ldots$$
$$\Pi(f_1)\Pi(f_2)\Pi(f_3) + \Pi(f_1)\Pi(f_2)\Pi(f_4) + \ldots$$
$$\Pi(f_1)\Pi(f_2)\Pi(f_3)\Pi(f_4))$$

Take expected value of each term, due to folding only

$E[\Pi(f_1)\Pi(f_3)\Pi(f_4)]$ and $E[\Pi(f_1)\Pi(f_2)\Pi(f_3)\Pi(f_4)]$

are nonzero. These are analyzed by Fourier expansion yielding strategies for P_1 and P_2.

Some more results

Theorem $\forall\epsilon > 0$, it is NP-hard to distinguish satisfiable 3-CNF formulas from those where only a fraction $(7+\epsilon)/8$ of the clauses can be satisfied.

Proof Define functions f_1, f_2 and f_3 with suitable distribution. Test $\Pi(f_1) \vee \Pi(f_2) \vee \Pi(f_3)$.

Theorem $\forall\epsilon > 0$, it is NP-hard to distinguish set-systems of size 4 sets where all sets can be split from those where only a fraction $(7+\epsilon)/8$ of the clauses can be satisfied.

Proof Find functions f_1, f_2, f_3, and f_4 with suitable distribution. Test that not all $\Pi(f_1)$, $\Pi(f_2)$, $\Pi(f_3)$ and $\Pi(f_4)$ are equal.

Other results:

Theorem[ST] For any $\epsilon, \delta > 0$ and integer k, there is a PCP with makes $2k+k^2$ question, has completeness $1-\epsilon$ and soundness $2^{-k^2}+\delta$.

Again SWP is the written proof.

Theorem[GHS] For any k it is NP-hard to distinguish 2-colorable hypergraphs from k colorable hypergraphs. Edge-size 4.

Uses many copies of SWP.

An efficient PCP not using (exactly) SWP

The recent result of Khot giving the best lower bounds for approximability of clique $n^{1-O((\log n)^{-\gamma})}$ uses a different proof.

The long code is too long.

Area of remaining problems

"Two position predicates"

- Coloring problems for graphs.
- Vertex-cover.
- Min-2Sat deletion.
- Max-2Sat.

Is the SWP sufficiently connected when we only ask questions that are very related pair-wise?

Final question

How Standard is SWP?