Proofs and Randomness

Johan Håstad

KTH Numerical Analysis
and Computer Science

August 5, 2005

Basic concepts

- Efficient computation is given by polynomial time computation.
- We allow algorithms to use randomness by flipping unbiased random bits.
- Measure resources in terms of n, the length of the input.

Complexity classes

P	Polynomial time.
BPP	Probabilistic polynomial time, allowing errors.
NP	Non-deterministic polynomial time.
PSPACE	Polynomial space.

Does randomness help for basic questions?

For polynomial space, provably not.

For polynomial time, probably not and most people think that $BPP = P$.

How about for verifying proofs?

Proofs

A way to convince a efficient, sceptical, rational, verifier V of the truth of a statement.

Completeness Can give proofs of a given type for every correct statement.

Soundness Cannot give a proof that is accepted for an incorrect statement. May happen with low probability.

Statements to think about

- This graph is 3-colorable.
- The following formula is true:

$$\forall x_1 \exists x_2 \ldots Q x_n (x_1 \vee \bar{x}_2) \wedge (x_7 \vee x_2) \ldots$$

- The program M halts on any input of length n it at most 2^n steps.

Types of proof

Written proofs which can be accessed at random places.

Interaction with one or more provers.

Types of proof

Written proofs which can be accessed at random places.

Can be exponentially large!

Interaction with one or more provers.

Types of proof

Written proofs which can be accessed at random places.

Can be exponentially large!

Interaction with one or more provers.
Cross-examination.

Deterministic verifier

Nothing interesting happens.

A written proof of polynomial size is the only interesting case, and gives exactly NP.

In a large proof with random access only write on pages that the verifier would look at.

For an interactive proof write down the path of inquiry followed by the verifier.

Inclusions, probabilistic V

Increasing order of power.

- Written proof of polynomial size.
- 1-prover interactive proofs.
- 2-prover interactive proofs.
- m-prover interactive proofs.
- Written proof of exponential size.

Written proofs of polynomial size

The complexity class MA.
Possibly barely more than NP, but not much.

The power of one prover

Take co-NP-complete statement.
3SAT-formula

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_4 \vee x_7) \ldots$$

is unsatisfiable, i.e. every assignment falsifies some clause.
An efficient proof of this?

Arithmetization

If φ has m clauses we can write polynomial $P = P_\varphi$, which is easy to evaluate, of degree $3m$ such that

$$\begin{aligned} \varphi(x) \text{ true} &\Rightarrow & P(x) = 1 \\ \varphi(x) \text{ false} &\Rightarrow & P(x) = 0 \end{aligned}$$

Need to verify

$$\sum_{x \in \{0,1\}^n} P(x) = 0.$$

The idea

$$\sum_{x_1=0}^{1}\sum_{x_2=0}^{1}\cdots\sum_{x_n=0}^{1} P(x) = 0$$

to be verified.
Define

$$Q(x_1) = \sum_{x_2=0}^{1}\cdots\sum_{x_n=0}^{1} P(x_1, x_2 \ldots x_n)$$

Suggesting protocol

P Sends $Q(x_1)$ as a polynomial mod p for large p.

Q Checks $Q(0) + Q(1) = 0$, picks random $\alpha_1 \in Z_p$.

recursively verify value of $Q(\alpha_1)$.

In general

We use

$$Q_i(x_i) = \sum_{x_{i+1}=0}^{1} \cdots \sum_{x_n=0}^{1} P(\alpha_1, \ldots \alpha_{i-1}, x_i, x_{i+1} \ldots x_n) \mod p$$

On step i, V knows value of $Q_{i-1}(\alpha_{i-1})$

- P Sends $Q_i(x_i)$ as a polynomial mod p.
- Q Checks $Q_i(0) + Q_i(1) = Q_{i-1}(\alpha_{i-1})$, picks random $\alpha_i \in Z_p$, sends α_i to P.

Finally V makes sure that $Q_n(\alpha_n) = P(\alpha_1, \alpha_2, \ldots \alpha_n)$.

The analysis

Completeness is straightforward. P sends correct polynomial at each point.

The analysis

Completeness is straightforward. P sends correct polynomial at each point.

Soundness: If $p \geq 2^n$, P needs to lie about Q_1, giving $Q_1' \neq Q_1$.

$$Pr[Q_1(\alpha_1) = Q_1'(\alpha_1)] \leq \frac{3m}{p}$$

The analysis

Completeness is straightforward. P sends correct polynomial at each point.

Soundness: If $p \geq 2^n$, P needs to lie about Q_1, giving $Q_1' \neq Q_1$.

$$Pr[Q_1(\alpha_1) = Q_1'(\alpha_1)] \leq \frac{3m}{p}$$

P needs to lie about Q_2, etc

The analysis

Completeness is straightforward. P sends correct polynomial at each point.

Soundness: If $p \geq 2^n$, P needs to lie about Q_1, giving $Q_1' \neq Q_1$.

$$Pr[Q_1(\alpha_1) = Q_1'(\alpha_1)] \leq \frac{3m}{p}$$

P needs to lie about Q_2, etc

Probability that V accepts is at most $\frac{3nm}{p}$.

General result

The technique extends to any question solvable in PSPACE [LFKN,S] and was in fact discovered in this generality.

This is the true power of one-prover interactive proofs.

Shamir: IP=PSPACE (1990)

Looking at other proof systems

Usually easy: An honest prover can convince a verifier of a correct statement using a correct proof.

The hard part: Proving that the verifier cannot be cheated to believe incorrect statements.

In the given example: We get a sequence of incorrect polynomials.

In general: Analyzing (loosing) probabilistic games.

Improving soundness

If a verifier can be cheated with probability q then doing two independent checks decrease cheating probability to q^2.

In a standard game-theoretic setting this is true even if we have two interactive games running in parallel.

An interesting game with two provers

V picks (q_1, q_2) from $\{(0,0),(0,1),(1,0)\}$ and sends q_i to P_i.

P_i returns $a_i \in \{0,1\}$ $1 \geq a_i \geq q_i$.

V accepts iff $a_1 \neq a_2$.

An interesting game with two provers

V picks (q_1, q_2) from $\{(0,0),(0,1),(1,0)\}$ and sends q_i to P_i.

P_i returns $a_i \in \{0,1\}$ $1 \geq a_i \geq q_i$.

V accepts iff $a_1 \neq a_2$.

Maximal accept probability is $2/3$

Two games played in parallel

V picks (q_1^1, q_2^1) and (q_1^2, q_2^2) independently and sends (q_i^1, q_i^2) to prover P_i.

If P_i construct a_i^1 and a_i^2 independently they can win only with probability $(2/3)^2 = 4/9$.

Two games played in parallel

V picks (q_1^1, q_2^1) and (q_1^2, q_2^2) independently and sends (q_i^1, q_i^2) to prover P_i.

If P_i construct a_i^1 and a_i^2 independently they can win only with probability $(2/3)^2 = 4/9$.

Strategy: If $q_i^1 = q_i^2 = 0$ set $a_i^1 = a_i^2 = 0$ and otherwise set $a_i^1 = a_i^2 = 1$.

Two games played in parallel

V picks (q_1^1, q_2^1) and (q_1^2, q_2^2) independently and sends (q_i^1, q_i^2) to prover P_i.

If P_i construct a_i^1 and a_i^2 independently they can win only with probability $(2/3)^2 = 4/9$.

Strategy: If $q_i^1 = q_i^2 = 0$ set $a_i^1 = a_i^2 = 0$ and otherwise set $a_i^1 = a_i^2 = 1$.

Succeeds in **both** games with probability $2/3$.

Why does parallel repetition fail?

Do not really know.

Maybe the fact that the provers can assume that they have won previous games creates a channel of information not available in a single game.

Raz Parallel repetition

After the question being open for 5 years Raz proved.

Theorem: Assume we have a 2-prover game with answer size bounded by d and soundness $c < 1$. Then there exists a constant $c_{c,d} < 1$ such that the soundness of the k-parallel 2-prover game is bounded by $c_{c,d}^k$.

Exponential decrease but at lower rate!

Challenge

Find an easy to follow proof of Raz parallel repetition!

Backing up

Is parallel repetition obvious for one-prover interactive games?

Backing up

Is parallel repetition obvious for one-prover interactive games?

YES, but write a careful proof, and prove it by induction on the rounds.

The true power of 2-prover games

[BFL]: Two-prover interactive games is exactly NEXPTIME.

The true power of 2-prover games

[BFL]: Two-prover interactive games is exactly NEXPTIME.

[FL]: Even for one-round variants. One question to each prover.

The true power of 2-prover games

[BFL]: Two-prover interactive games is exactly NEXPTIME.

[FL]: Even for one-round variants. One question to each prover.

Randomness does help verification!

Written proofs

Still NEXPTIME in the setting where the only restriction is an efficient verifier.

Let us scale down and get very efficient proofs for NP.

Resources to consider

For a really efficient written proof.

- **Size** Could be polynomial, possibly allowing an efficient prover given a witness.
- **Bits read** What could we hope for?
- **Randomness** Do we care how many random coins V uses?
- **Completeness** Assumed perfect. (Do we care if it drops to .999?)
- **Soundness** At most probability $1/2$ of accepting a false claim.

Reading bits

Reading all bits of the proof puts us at a verifier that is almost an NP-verifier, even allowing randomness.

How few bits could we use?

Reading one-bit

V computes (probabilistically) an address a and bit b and checks that the bit at this address, π_a, has value b.

Reading one-bit

V computes (probabilistically) an address a and bit b and checks that the bit at this address, π_a, has value b.

Sampling we can either see that V sometimes wants the same bit to have opposite values

or

It is easy to construct a proof that V (almost) always accepts.

Reading one-bit

V computes (probabilistically) an address a and bit b and checks that the bit at this address, π_a, has value b.

Sampling we can either see that V sometimes wants the same bit to have opposite values

or

It is easy to construct a proof that V (almost) always accepts.

We do not need prover to answer questions and we have only proofs for languages in BPP.

Reading two bits

Assume V on some random coins reads bit 17 and bit 297 and rejects if $\pi_{17} = 1$ and $\pi_{297} = 0$.

Reading two bits

Assume V on some random coins reads bit 17 and bit 297 and rejects if $\pi_{17} = 1$ and $\pi_{297} = 0$.

Corresponds to $(\bar{x}_{17} \vee x_{297})$.

Reading two bits

Assume V on some random coins reads bit 17 and bit 297 and rejects if $\pi_{17} = 1$ and $\pi_{297} = 0$.

Corresponds to $(\bar{x}_{17} \vee x_{297})$.

Sample V's coins to write down a 2Sat formula.

Two bits continued

Get a 2Sat-formula which is satisfiable if the statement is true and very unsatisfiable if the statement is false.

Can check efficiently determine which is the case.

Again only proofs for BPP.

Reading three bits

As we move to 3Sat formulas it seems hard to rule out this possibility by similar methods....

The famous PCP-theorem

Arora, Lund, Motwani, Sudan and Szegedy [ALMSS] building on work by Arora and Safra [AS] proved in 1992.

Theorem: Any statement in NP has a polynomial size proof that can be verified by a probabilistic polynomial time verifier V that reads three bits such that

- V always accepts a correct proof of a correct statement.
- V rejects any proof of an incorrect statement with probability $c > 0$.
- V uses only a logarithmic number of random bits.

The famous PCP-theorem

Arora, Lund, Motwani, Sudan and Szegedy [ALMSS] building on work by Arora and Safra [AS] proved in 1992.

Theorem: Any statement in NP has a polynomial size proof that can be verified by a probabilistic polynomial time verifier V that reads three bits such that

- V always accepts a correct proof of a correct statement.
- V rejects any proof of an incorrect statement with probability $c > 0$.
- V uses only a logarithmic number of random bits.

PCP=Probabilistically Checkable Proofs.

The original proof

Uses many ideas.

- Representing objects by interpolation of multivariate polynomials.

$$P\left(\hat{i}\right) = x_i,$$

  look at P on larger domain.
- Low degree testing, using non-coding points.
- Proof composition of different types of proofs.

Relies on many properties of polynomials.

New proof of PCP-theorem

By Dinur in 2005.
Uses combinatorics.

- Expander graphs, walks on expanders.
- Efficient PCPs of constant size.

An iterative construction inspired by Reingold's result that *st*-connectivity is in L, logarithmic space.

Many parameters to improve

What is the size of the proof?

How few bits can we read?

What is the soundness?

Minimizing several parameters at the same time.

My favorite

Read few bits.
Accept from simple condition.
Get good soundness.

Non-adaptive if the location of each read bit does not depend on values of previously read bits.

Reading three bits

Exists non-adaptive proof system that reads three bits, always accepts a correct proof and has soundness $3/4 + \epsilon$.[H]

If we (very) rarely reject a correct proof we can improve soundness to $1/2 + \epsilon$.[H]

Reading three bits

Exists non-adaptive proof system that reads three bits, always accepts a correct proof and has soundness $3/4 + \epsilon$.[H]

Cannot for sure not be improved further than $5/8 + \epsilon$. [Z]

If we (very) rarely reject a correct proof we can improve soundness to $1/2 + \epsilon$.[H]

Best possible [Z].

Proof systems that sometimes reject correct proofs have some benefits?!

Reading q bits

It is possible to push soundness to $2^{O(\sqrt{q})-q}$ [ST].

Even with perfect completeness [HK].

Note that a random proof is accepted with probability 2^{-q}.

One view of the (optimized) PCP-theorem

We take a Boolean formula φ and produce a 3Sat formula ψ such that

$$\begin{aligned} \varphi \text{ satisfiable} &\Rightarrow \psi \text{ satisfiable} \\ \varphi \text{ not satisfiable} &\Rightarrow \psi\ (7/8+\epsilon)-\text{satisfiable} \end{aligned}$$

Can only simultaneously satisfy a fraction $(7/8+\epsilon)$ of the clauses.

Cook's theorem

We take a Boolean formula φ and produce a 3Sat formula ψ such that

$$\begin{aligned} \varphi \text{ satisfiable} &\Rightarrow \psi \text{ satisfiable} \\ \varphi \text{ not satisfiable} &\Rightarrow \psi \text{ not satisfiable} \end{aligned}$$

Consequences

Of Cook's theorem

Theorem: It is NP-complete to determine whether a 3Sat formula is satisfiable.

Of optimized PCP theorem

Theorem: It is NP-hard to approximate Max-3Sat within a factor $7/8 + \epsilon$.
We cannot find approximate solutions to hard optimization problems!

Approximating NP-hard optimization problems

The connection between PCPs and approximability has given us fantastic hardness results.

Many new efficient approximation algorithms, many based on semi-definite programming [GW].

Just an overview would fill a complete talk, in particular the first talk I had planned.

Some optimal inapproximability results

It is hard to

- approximate Max-Linear equations mod 2 within a factor $1/2 + \epsilon$ [H].
- approximate Set Cover within $\ln n(1 - o(1))$ if universe size is n [F] .
- approximate Max-Clique within $n^{1-\epsilon}$ on n-node graphs [H].
- approximate Graph Coloring within $n^{1-\epsilon}$ on n-node graphs [FK].

There are many more

My recent favorite by Khot:

Theorem: For any constant C, it is NP-hard to find the shortest vector in an integer lattice within C. This is true for any L_p-norm $p > 1$.

Positive result by GW

The first and still striking

Theorem: It is possible to approximate Max-Cut in probabilistic polynomial time within a factor α_{GW} where

$$\alpha_{GW} = \min_{\Theta} \frac{2\Theta}{\pi(1 - \cos\Theta)} \approx .878567$$

Positive result by GW

The first and still striking

Theorem: It is possible to approximate Max-Cut in probabilistic polynomial time within a factor α_{GW} where

$$\alpha_{GW} = \min_{\Theta} \frac{2\Theta}{\pi(1-\cos\Theta)} \approx .878567$$

Recent results [KKMO] suggest this might be the correct constant.

Final words

Proofs and randomness mix very well and make a fantastic cocktail.
Gives a lot of information about approximating NP-hard optimization problems.