Kungl
Tekniska
Högskolan

Inapproximability
some history and open problems

Johan Håstad
Royal Institute of Technology
Stockholm, SWEDEN

Disclaimer

I make no attempt to be complete.

I try to be accurate in mathematical facts but do not follow the historical developments.

A biased, personal view of a subset of the area.

If I make you upset, protest.

The NP-complete problems

In theory equally hard but in practice not.

- Some easy on average, some not.
- For some we can find reasonably good solutions for some not.

Quote from someone in operations research: Now we have reduced the problem to Set Cover and this is easy to solve.

Approximation ratio

Given (NP-hard) optimization problem O.

Approximation ratio, maximization problems.

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

Approximation ratio, minimization problems.

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

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 polynomial time.

Sometimes we allow C to be achieved on the average, but for each input.

Max-3Sat

Given a 3Sat formula with exactly 3 literals in each clause and m clauses, satisfy as many clauses as possible.

A random assignment satisfies $7m/8$ clauses and by calculating conditional expectations an assignment that satisfies $7m/8$ clauses can be found deterministically.

This gives and 8/7-approximation algorithm [J].

Gut response

This is very simple algorithm, must be possible to improve.

Not so easy, try Max-2Sat.

Relaxing Max-2Sat

Given Max-2Sat instance

$$\varphi = (x_1 \vee \bar{x}_7) \wedge (\bar{x}_3 \vee x_4) \ldots$$

replace each clause by a degree two real polynomial in two real variables. Let Boolean values be represented by $\{-1, 1\}$ with -1 true.

Thus replace $(x_i \vee x_j)$ by $1-(1+x_i)(1+x_j)/4$ etc.

Summing we get a quadratic function giving the number of satisfied clauses. Make homogeneous by removing the constant and replacing x_i by $x_i x_0$.

Semi-definite programming [GW]

Replace $x_i x_j$ by variable y_{ij}.

Optimize over y_{ij} being a positive semidefinite matrix.

Obtains vector solution for x_i.

Round in suitable way to Boolean solution x_i.

Due to Feige and Goemans approximation ratio around 1.074.

Open problem I

Design a combinatorial algorithm for Max-2Sat with a performance ratio better than that of picking a random assignment, i.e. 4/3.

Hard to define combinatorial but I know what I mean.

Open problem II

Do all constraint satisfaction problems in two variables admit non-trivial approximation?

True for Boolean predicates, for some special predicates with variables in Z_d but unknown in general.

How all-mighty is semidefinite programming?

General problems to discuss

- Max-3Sat.
- Maximum satisfiability of linear equations mod p.
- Max-Clique.
- Coloring of graphs.
- Vertex cover.
- Set cover.
- Bisection.

Lower and upper bounds on approximability.

Success stories

- Max-3Sat. Upper 8/7, lower $8/7 - \epsilon$ [H].
- Max-Lin-p. Upper p, lower $p - \epsilon$ [H].
- Max-Clique & Coloring. Upper $n/O((\log n)^c)$, lower $n^{1-o(1)}$ [H,EH,K,FK].
- Set-cover. Upper $\ln n$, lower $\ln n(1-o(1))$ [F].

Work to be done

- Vertex cover, upper 2, lower $10\sqrt{5}-21-\epsilon \approx 1.36[DS]$.

- Max-2Sat, upper 1.0741 [FG], lower $\frac{22}{21}-\epsilon$ [H].

- Bisection upper $O((\log n)^2)$ [FK], lower 1.

- Color of 3-colorable graph, upper $\tilde{O}(n^{3/14})$ [BK], lower 5/3 [KLS, GK].

Why do we know so little?

Why can we not get best results for all problems?

Right question

How have we achieved so much?

What stops us from extending the tight results?

Proving Max-3Sat hard

A polynomial time transformation:

$\varphi \mapsto \psi$

$$\begin{array}{rl} \varphi \text{ satisfiable} & \Rightarrow \psi \text{ satisfiable} \\ \varphi \text{ not satisfiable} & \Rightarrow \psi \text{ at most}(\dfrac{7}{8} + \epsilon)\text{-satisfiable} \end{array}$$

No assignment satisfies more than fraction $(\frac{7}{8} + \epsilon)$ of the clauses.

Inapproximability consequence

Suppose we have a $(\frac{7}{8}+\delta)^{-1}$ approximation algorithm, A, for Max-3Sat for $\delta > \epsilon$.

Run A on ψ.

If φ is satisfiable then so is ψ and A returns an assignment that satisfies at least a $(\frac{7}{8}+\delta)$ fraction of the clauses.

If φ is not satisfiable then ψ is at most $(\frac{7}{8}+\epsilon)$-satisfiable and A returns and assignment that satisfies at most a $(\frac{7}{8}+\epsilon)$ fraction of the clauses.

Solving an NP-hard problem.

Proof checking consequence

A proof of satisfiability is given by a satisfying assignment. Checking: Going over clauses and checking that they are verified.

For φ we need to check all clauses to have any guarantee.

For ψ if we randomly pick k clauses then the probability of being cheated is $(\frac{7}{8}+\epsilon)^k$.

Ideal for lazy proof checkers!

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.

Equivalence

A PCP reading three bits with $O(\log n)$ randomness is exactly the same as a gap making reduction from 3Sat.

Bits of proof is the variables of the formula.

All checks by the verifier give the clauses.

PCP-theorem [ALMSS],I

There is a polynomial time verifier V that uses $O(\log n)$ random coins and on input φ reads three bits of a proof.

If φ is satisfiable there is a proof that makes V accept with probability 1.

If φ is not satisfiable no proof makes V accept with probability above $1-\epsilon$.

PCP-theorem [ALMSS],II

A polynomial time transformation:

$\varphi \mapsto \psi$

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

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

Proof components

- Coding of objects as low degree polynomials.

- Recursion also called proof composition [AS].

The final proof is in fact a proof that there is a proof that there is a proof that φ is satisfiable.

Open problem III

Find a more accessible proof of the PCP theorem.

Using the PCP theorem

The PCP-theorem gives some inapproximability constant for Max-3Sat but a very bad one.

Getting the best constant we need to add additional layers of proof composition.

We start with a ψ that is either satisfiable or not $(1-\epsilon)$ satisfiable and it is NP-hard to tell which.

Standard steps

For many tight results we add the following steps.

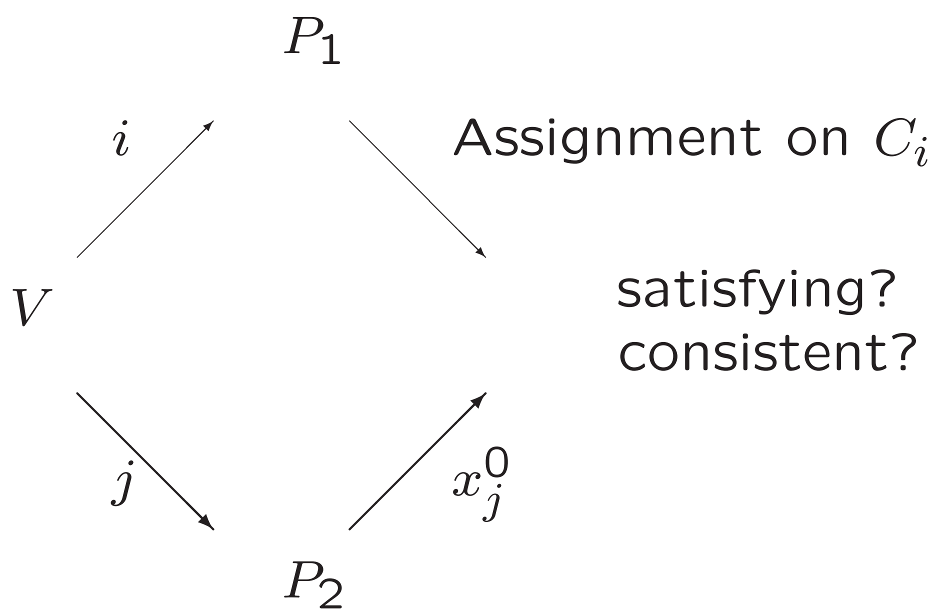
- Make into a two prover protocol.
- Improve two prover protocol.
- Code answers by provers in a suitable way to get better PCP targeted towards problem to prove lower bounds for.

Two-Prover protocol

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

$\psi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, satisfiable or at most $(1-\epsilon)$-satisfiable.

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



Completeness 1.

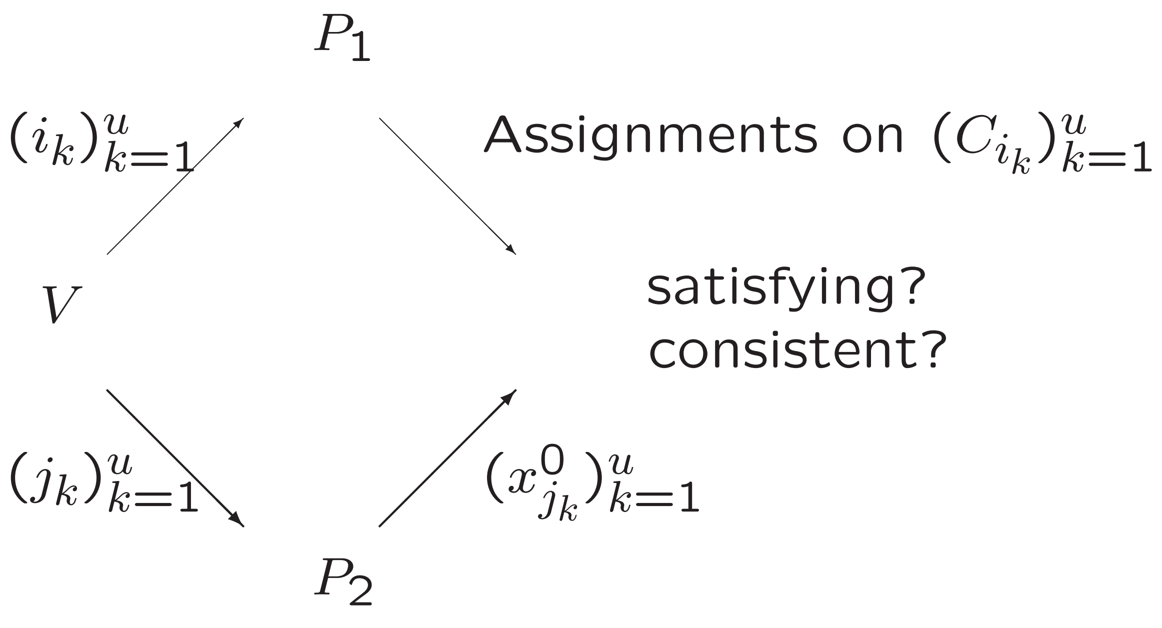
Soundness $1 - \epsilon/3$.

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].

The approach

Code each answer by the two provers in a suitable way.

The key is to make sure that the answers are consistent.

Thus we have two answers x and y and need to check that the bits of x agree with some subset of bits of y.

x answers of P_2, y answers of P_1, $\pi(y) = x$.

The long code

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

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

Use this to code answers both of P_1 and P_2 in two-prover game.

Long code of 101

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

As a function of f it is a coordinate function.

The PCP for Max-3Sat

Choose f, g_1 and g_2 such that

$g_1(y) \vee g_2(y) \vee f(\pi(y))$ is true for all y.

These are coordinates in the long code and corresponds to positions in the proof Π.

Functions of answers given by P_1 and P_2.

Check

$$\Pi(g_1) \vee \Pi(g_2) \vee \Pi(f).$$

Basis of the analysis

Key technical tool, Discrete Fourier Transform to analyze cheating strategies of prover.

Small set giving large coefficients of supposed long codes give a good strategy for P_1 and P_2.

Any proof beating a random one gives a non-trivial strategy for P_1 and P_2.

This general approach apply to many constraint satisfaction problems.

Doing clique

Clique comes very close to proofs. Make a graph of local accepting views.

Connect if consistent.

A clique is a set of consistent views and corresponds exactly to a standard proof. Size is the probability of accepting.

Need work to get best parameters.

Key technical result

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

Technical limits

Studying graphs with really big cliques or independent sets.

To get vertex cover: Want it to be hard to decide if a graph has an independent set of size $(\frac{1}{2} - \epsilon)n$ or of size ϵn.

For 3-colorable graphs: Is the graph 3-colorable or is the size of the largest independent set ϵn.

My guess

Both these problems are hard.

Need some new idea to complete the proof, a more rigid object than the parallelized two-prover protocol.

More rigid?

Our consistency test is rigid one way but not the other. Given x there are many possible y.

The best result [DS] for Vertex cover uses a consistency check that is at most 2-to-1 both ways.

The ultimate rigidity

A two-prover protocol where for each answer there is a unique answer from the other prover to make the verifier accept.

[KR] Suppose we for each $\epsilon > 0$ have such a two-prover protocol with constant answer size and ϵ soundness, then it follows that Vertex Cover is hard to approximate within $2-\delta$.

Open problem IV

Present any evidence in favor or against this unique two prover protocol.

I.e. given φ we want to know if it is satisfiable. We have the two prisoners to be in different cells, ask one question to each, the answer of each determines the answer of the other, almost perfect completeness, almost perfect soundness.

Easy to see that we have to hang some innocent prisoners to be this efficient.

Open problem V

Is the existence of the unique two-prover game equivalent to the lower bound for Vertex Cover?

3-coloring

Not clear how to proceed even given the unique two-prover conjecture. Thus might be hard.

On the other hand not known to imply the PCP theorem. Could be very easy with direct proof via some gadget-type proof.

Best lower bounds of 5 was first done with PCP theorem and then redone without!

Bisection

Would like any lower bound for bisection.

Inapproximability 4/3 from assumption that it is hard to separate satisfiable 3Sat formulas from random 3Sat formulas [F].

Weak evidence it might be hard, but not compelling.

Problem seems to come from combination of two complications.

- Constraints work on pairs, little freedom for lower bound construction.

- One global constraint which is essential.

Open problems VI

- Min-2Sat deletion. Can we do constant approximation?

- Max-2Sat. Get correct constant.

Final words

We have come a long way but to get next set of problems it seems like a new idea is needed. Could be just around the corner.

Why are there not more problems that are neither NP-hard nor in P?