NP-COMPLETE PROBLEMS

Presumably infeasible to decide in worst case

Hence also infeasible to compute optimal solutions in worst case
(Search problems can be reduced to decision problems since we can "map certificates back" – Levin reductions)

WHAT ABOUT APPROXIMATE SOLUTIONS?

Considered after discovery of NP-completeness in 1971.
Also seemed hard for most problems.
Show that NP-complete problems are hard not only to solve exactly, but also to approximate? This, too, seems hard...

For almost 20 years, not too much happened.

2nd half of 1980s: Invention/discovery of interactive proofs [first paper took long to get published and recognized]
Used for cryptographic applications.

[Goldwasser, Micali, Rackoff '85]

[Babai, Fortnow, Lund '90]

$MIP = NEXP$

Then everything happened very swiftly...

"Sealed claims" to NP

- Proofs for NP that can be probabilistically verified
- Connection to hardness of approximation

[Arora, Safra '92]

$NP = PCP[\log n, \sqrt{\log n}]$ ← Check only sublogarithmically many bits of proof!

[Arora, Lund, Motwani, Sudan, Szegedy '92]

$NP = PCP[\log n, 1]$ ← Check only constant number of bits

"PCP Theorem"

Quite a bombshell...

Has led to whole new area of research in computational complexity: Hardness of approximation

Many results showing that for many problems even computing approximate solutions is NP-hard

Many tight results

- Can approximate up to some limit L
- Doing better than $L + \varepsilon$ is NP-hard

Many such results obtained by members of TCS group at KTH

Plan for rest of course:

2 lectures giving overview of PCP Theorem and connections to hardness of approximation
[+start with examples of approximation algorithms]

2-3 lectures going into details of the proof of the PCP Theorem

---

Instance of NP-complete problem can often be viewed as tuple

$x = \langle O, v \rangle$ "Does O have a solution of value $\geq v$?"

Ex VERTEX COVER

$\langle G, k \rangle$ Does G have a vertex cover of size $\leq k$?

INDEPENDENT SET

$\langle G, k \rangle$ Does G have an independent set of size $\geq k$?

3SAT

$\langle \varphi, \#\text{clauses in } \varphi \overset{m=}{} \rangle$ Does the 3-CNF formula φ have an assignment satisfying $\geq m = (\#\text{clauses in } \varphi)$ clauses?

Can classify such problems as maximization or minimization problems.

Can define the optimal value of $\mathcal{O}$, $val(\mathcal{O})$ or $opt(\mathcal{O})$, as the best value v for which $\langle \mathcal{O}, v \rangle$ is a "yes" instance

MINVERTEXCOVER
What is the minimal size of a vertex cover of G?

MAXINDEPENDENTSET
What is the maximal size of an independent set of G?

MAX-3SAT
What is the maximal # clauses of φ that can be satisfied?

DEFINITION (variant of 11.1) [and slightly informal]
Let $\mathcal{L} = \{ \langle \mathcal{O}, v \rangle \mid \mathcal{O}$ has solution of value $v \}$ be a language in NP

For a maximation problem, let $val(\mathcal{O}) = \max \{ v \mid \langle \mathcal{O}, v \rangle \in \mathcal{L} \}$.
For $\rho \leq 1$ we say that A is a ρ-approximation algorithm if for every $\mathcal{O}$, $A(\mathcal{O})$ outputs a solution of value $\geq \rho \cdot val(\mathcal{O})$

For a minimization problem, let
$val(\mathcal{O}) = \min \{ v : \langle \mathcal{O}, v \rangle \in \mathcal{L} \}$

For $\rho \leq 1$ we say that A is a ρ-approximation algorithm if for every $\mathcal{O}$, $A(\mathcal{O})$ outputs a solution of value $\leq \frac{1}{\rho} val(\mathcal{O})$

Intuitively, A gets within $(\rho \cdot 100)\%$ of the optimal solution.

- Is A deterministic or randomized? Both are of interest.
- Does A need to find the solution (as above) or just report the value? Both are of interest — second task can be (seemingly) easier.

EXAMPLE 1.1.2 ½-approximation for MAX-3SAT
3-CNF formula φ with m clauses
Each clause has ≤ 3 literals
(but could be less)

Fix variables in some order $x_1, x_2, \ldots, x_n$
Set $\varphi' := \varphi$
for $i = 1$ up to n
  Let $\varphi'_{x_i} = \{C \in \varphi' \mid x_i \in C\}$; $\varphi'_{\bar{x}_i} = \{C \mid \bar{x}_i \in C\}$
  If $|\varphi'_{x_i}| \geq |\varphi'_{\bar{x}_i}|$
    $x_i := T$; $\varphi' := \varphi' \restriction_{x_i = T}$
  else
    $x_i := \bot$; $\varphi' := \varphi' \restriction_{x_i = \bot}$

clearly satisfies at least half of all clauses. Clearly $val(\varphi) \leq m$.

For MAX-E3SAT, a random assignment satisfies $\geq 7/8$ of the clauses [in expectation] Can be derandomized

Same result can be obtained for MAX-3SAT via so-called semidefinite programming

EXAMPLE 11.3 1/2-approximation for MINVERTEXCOVER

For undirected graph $G=(V,E)$, $S \subseteq V$ is a VERTEX COVER if for all $(u,v) \in E$ it holds that $\{u,v\} \cap S \neq \emptyset$.

1/2-approximation = find vertex cover of at most twice the optimal size.

$|E(G)| = m$

Fix edges $e_1, e_2, \ldots, e_m \in E(G)$ in some order

$S := \emptyset$

For all $e_i \in E$

Let $e_i = (u_i, v_i)$

if $S \cap \{u_i, v_i\} = \emptyset$

$S := S \cup \{u_i, v_i\}$

important to add both vertices

Claims:

- At end of algorithm, S is a vertex cover
- Furthermore, S is a matching.

[rather; the edges added to S form a matching of size $|S|/2$]

Let M be any matching in G.
Then any vertex cover S is of size $\geq |M|$ since every edge in M should be covered by a vertex in S.

Hence, S is at most twice the size of a smallest vertex cover.

Now let's shift focus to the PCP Theorem
Can be viewed in 2 ways

(A) Statement about extremely robust locally testable proofs

(B) Statement about hardness of approximating combinatorial optimization problems

Let us first give a formal definition of PCPs

Verifier: Probabilistic poly-time algorithm
Oracle access to certificate / proof π
that $x \in L$

Formally TM with oracle tape for reading bits of π, but we will be happy with more informal specification of V.

Resources to keep track of
coin flips r
queries q into proof

Following Arora-Barak, we define the verifier to be nonadaptive — i.e., given x and random coin flips, decide in one go all q positions to query (don't read them one-by-one and adapt along the way

DEFINITION (variant of Def 11.4)

Let L be a language

Let $q, r : \mathbb{N} \to \mathbb{N}$ be functions

Let $c, s \in [0,1]$ be constants, $c > s$

We say that L has an $(r(n), q(n))$-PCP verifier if there is a probabilistic algorithm V running in time $\text{poly}(|x|)$ satisfying the following properties

[with completeness c and soundness error s]

"EFFICIENCY"

Given $x \in \{0,1\}^n$ and oracle access to $\pi \in \{0,1\}^*$, V uses at most $r(n)$ random coin flips and makes at most $q(n)$ nonadaptive queries to π, after which it outputs $V^{\pi}(x) \in \{0,1\}$

random variable depending on coin flips

COMPLETENESS

If $x \in L$, then $\exists \pi$ of length $\leq q(n) 2^{r(n)}$ such that

$$\Pr\left[V^{\pi}(x) = 1 \right] \geq c$$

SOUNDNESS

If $x \notin L$, then $\forall \pi'$ it holds that

$$Pr[V^{\pi'}(x) = 1] \leq s$$

We call c the COMPLETENESS parameter and s the SOUNDNESS ERROR.

Note that in Ch 11, Arora-Barak fix $c = 1$ (perfect completeness) and $s = 1/2$

Note that w.l.o.g. any proof π has size $\leq q(n) 2^{r(n)}$, since this is max possible # distinct locations V can read.

DEFINITION $PCP_{c,s}[r(n), q(n)]$

We say that a language L is in $PCP_{c,s}[r(n), q(n)]$ if there are constants $k_1, k_2 > 0$ such that L has a $(k_1 \cdot r(n), k_2 \cdot q(n))$-PCP verifier with soundness error s and completeness c.

There are many different theorems about which complexity classes are characterized by PCPs with which parameters. However, "The" PCP Theorem is the following statement

X

THEOREM (A) (11.5) PCP Theorem

[Arora, Safra '92], [Arora, Lund, Motwani, Sudan, Szegedy '92]

$$NP = PCP_{2,1/2}(\log n, 1)$$

Some remarks:

- Proving completeness is usually easy and proving soundness is usually hard for PCPs.
- $PCP_{c,s}(r(n), q(n)) \subseteq NTIME(2^{O(r(n))} q(n))$

  Just guess a proof π; check for all possible $2^{O(r(n))}$ random coin flip strings that probability of acceptance $> c$

  Hence $NP \supseteq PCP_{1,1/2}(\log n, 1)$ is easy. It is the other direction that is hard.
- The constant $O(1)$ can be different for different languages, but can be upper-bounded by a universal constant (since every $L \in NP$ can be reduced to SAT)
- As usual, the exact value of the soundness error $s = 1/2$ is inconsequential as long as it is < 1.

Another view of the PCP theorem:
For many NP-hard optimization problems, computing approximate solutions is no easier than computing exact solutions.

For concreteness, consider MAX-3SAT.
Is this problem hard to approximate?
Or does it have efficient approximation algorithms for all $\rho < 1$?
Before the PCP Theorem this was not known. But PCP Theorem says that there exists a $\rho^* < 1$ such that there are no efficient ρ^*-approximation algorithms for MAX-3SAT

THEOREM (B) (11.9)

There exists a $\rho^* < 1$ such that for every $L \in NP$ there is a polynomial-time function f mapping strings to 3-CNF formulas such that

$$x \in L \implies val_N(f(x)) = 1$$
$$x \notin L \implies val_N(f(x)) < \rho^*$$

Define $val_N(\varphi) = \dfrac{\text{max \# satisfied clauses}}{\text{total \# clauses}} \in [0,1]$

Normalize just to get value between 0 and 1

COROLLARY 11.10

There exists a constant $\rho^* < 1$ such that if there is a poly-time ρ^*-approximation algorithm for MAX-3SAT, then P=NP

In fact [Håstad '97] showed that we can pick $\rho^* = 7/8 + \varepsilon$ for any arbitrarily small $\varepsilon > 0$.

So for E3-SAT a stupid random assignment that doesn't even look at the formula satisfies 7/8 of the clauses in expectation.
And this is best possible unless P=NP!

Proof of Cor 11.10:

Take your favourite NP-complete language L. Given instance x, compute $f(x)$ and run ρ^*-approximation algorithm A

For an instance $x \notin L$, clearly

$$\frac{A(f(x))}{\#\text{clauses in } f(x)} \leq \text{val}_N(f(x)) < \rho^*$$

For $x \in L$, by the approximation guarantee we have

$$\frac{A(f(x))}{\#\text{clauses}} \geq \rho^* \cdot \text{val}_N(f(x)) = \rho^*$$

Hence $x \in L$ iff $A(f(x)) \geq \rho^* \cdot (\#\text{clauses in } f(x))$, and A and f both run in poly time.