Advanced Algorithms (XIII)

Yijia Chen
Fudan University
The PCP Theorem

**Theorem**

\[ \text{NP} = \text{PCP}(\log n, 1). \]
Motivation
Approximate solutions

**Definition (Approximation of MAX–3SAT)**
For every 3CNF formula $\varphi$, the value of $\varphi$

$$\text{val}(\varphi) := \max_{\text{assignment } z} \frac{\# \text{ clauses in } \varphi \text{ satisfied by } z}{\# \text{ clauses in } \varphi}$$

Thus

$$\varphi \text{ is satisfiable } \iff \text{val}(\varphi) = 1.$$

For every $\rho \leq 1$, an algorithm $A$ is a $\rho$-approximation algorithm for MAX–3SAT if for every 3CNF formula $\varphi$ with $m$ clauses, $A(\varphi)$ outputs an assignment satisfying at least

$$\rho \cdot \text{val}(\varphi) \cdot m$$

of $\varphi$’s clauses.
The algorithm assigns values to the variables one by one in a greedy fashion, whereby the $i$th variable is assigned the value that results in satisfying at least $1/2$ the clauses in which it appears.
The minimum vertex cover problem

**MIN-VERTEX-COVER**

*Input:* A graph $G = (V, E)$.

*Solution:* $S \subseteq V$ with $S \cap e \neq \emptyset$ for every $e \in E$.

*Cost:* $|S|$.

*Goal:* max.

For $\rho \leq 1$, a $\rho$-approximation algorithm for MIN-VERTEX-COVER is an algorithm that on input a graph $G$ outputs a vertex cover whose size is at most $1/\rho$ times the size of the minimum vertex cover.
1/2-Approximation for MIN-VERTEX-COVER

1. Start with $S \leftarrow \emptyset$.
2. Pick any edge $e = \{u, v\}$.
3. $S \leftarrow S \cup \{u, v\}$.
4. Delete $u$, $v$, and all edges adjacent to them.
5. If the graph is not empty, then goto 2, else output $S$. 
Two Views of the PCP Theorem
- New, extremely robust proof systems.
- Approximating combinatorial optimization problems.
Let $\mathcal{A}$ be any one of the usual axiomatic systems of mathematics for which proofs can be verified by a deterministic TM in time that is polynomial in the length of the proof.

Then

$$L = \{ \langle \varphi, 1^n \rangle \mid \varphi \text{ has a proof in } \mathcal{A} \text{ of length } \leq n \}.$$ 

The PCP Theorem asserts that $L$ has probabilistically checkable certificates: proofs are probabilistically checkable by examining only a constant number of bits in them.
A standard definition of NP

$L \in \text{NP}$ if there is a polynomial time Turing machine $V$ that, given input $x$, checks certificates (or membership proofs) to the effect that $x \in L$:

$$x \in L \implies \exists \pi V^\pi(x) = 1$$
$$x \notin L \implies \forall \pi V^\pi(x) = 0,$$

where $V^\pi$ denotes "a verifier with access to certificate $\pi$."
**Definition**

Let $L$ be a language and $q, r : \mathbb{N} \rightarrow \mathbb{N}$. We say that $L$ has an $(r(n), q(n))$-PCP verifier if there is a polynomial time probabilistic algorithm $V$ satisfying:

1. **Efficiency:** On input $x \in \{0, 1\}^n$ and given random access to a string $\pi \in \{0, 1\}^*$ of length at most $q(n)2^{r(n)}$, $V$ uses at most $r(n)$ random coins and makes at most $q(n)$ nonadaptive queries to locations of $\pi$.

2. **Completeness:** If $x \in L$, then there exists a proof $\pi \in \{0, 1\}^*$ with
   \[
   \Pr[V^\pi(x) = 1] = 1.
   \]

3. **Soundness:** If $x \notin L$ then for every proof $\pi \in \{0, 1\}^*$,
   \[
   \Pr[V^\pi(x) = 1] \leq 1/2.
   \]
Definition
$L \in \text{PCP}(r(n), q(n))$ if there are some constants $c, d > 0$ such that $L$ has a $(c \cdot r(n), d \cdot q(n))$-PCP verifier.

Theorem

$\text{NP} = \text{PCP}(\log n, 1)$. 
Remarks

1. By the soundness, if $x \notin L$ then the verifier has to reject every proof with probability at least $1/2$, the most difficult part of the proof.

2. Proofs checkable by an $(r, q)$-verifier are of length at most $q2^r$, since such a verifier can look on at most this number of locations.

3. $\text{PCP}(r(n), q(n)) \subseteq \text{NTIME}\left(2^{O(r(n))} q(n)\right)$.

4. For different NP languages the $(O(\log n), O(1))$-verifiers might use a different constant number of query bits. However, since every NP language is reducible to SAT, all these numbers can be upper bounded by the number of query bits required by a verifier for SAT.

5. The constant $1/2$ in the soundness is arbitrary, in the sense that changing it to any other positive constant smaller than 1 will not change the class of languages defined. A PCP verifier with soundness $1/2$ that uses $r$ coins and makes $q$ queries can be converted into a PCP verifier using $cr$ coins and $cq$ queries with soundness $2^{-c}$ by just repeating its execution $c$ times.
The graph non-isomorphism problem is defined as

$$\text{GNI} = \{(G_0, G_1) \mid G_0 \text{ and } G_0 \text{ are two non-isomorphic graphs}\}.$$

- Assume $G_0$ and $G_1$ are both of $n$ vertices.
- The proof $\pi$ contains, for each graph $H$ with $n$ nodes, $\pi(H) \in \{0, 1\}$ corresponding to whether $H \equiv G_0$ or $H \equiv G_1$, otherwise, arbitrary.
- The verifier randomly picks $b \in \{0, 1\}$ and a permutation. She applies the permutation to $G_b$ to obtain an isomorphic graph $H$. Then she accepts iff $\pi(H) = b$.
- If $G_0 \not\equiv G_1$, then the correct $\pi$ makes the verifier accept with probability 1.
- If $G_1 \equiv G_2$, then the probability that any $\pi$ makes the verifier accept is at most $1/2$. 
Theorem

\[ \text{NEXP} = \text{PCP}(\text{poly}(n), 1). \]
Theorem

PCP Theorem: Hardness of approximation view There exists $\rho < 1$ such that for every $L \in \text{NP}$ there is a polynomial time function $f$ mapping strings to (representations of) 3CNF formulas such that

\[
x \in L \implies \text{val}(f(x)) = 1
\]
\[
x \notin L \implies \text{val}(f(x)) < \rho.
\]

Corollary

There exists some constant $\rho < 1$ such that if there is a polynomial time $\rho$-approximation algorithm for $\text{MAX-3SAT}$ then $P = \text{NP}$.
Equivalence of the Two Views
Constraint satisfaction problems (CSP)

**Definition**
Let $q \in \mathbb{N}$. Then a $q$CSP instance $\varphi$ is a collection of functions $\varphi_1, \ldots, \varphi_m$ (i.e., constraints) from $\{0, 1\}^n$ to $\{0, 1\}$ such that each function $\varphi_i$ depends on at most $q$ of its input locations: for every $i \in [m]$ there exist $j_1, \ldots, j_q \in [n]$ and $f : \{0, 1\}^q \to \{0, 1\}$ such that

$$\varphi_i(u) = f(u_{j_1}, \ldots, u_{j_q})$$

for every $u \in \{0, 1\}^n$.

An assignment $u \in \{0, 1\}^n$ satisfies $\varphi_i$ if $\varphi_i(u) = 1$. Let

$$\text{val}(\varphi) = \max_u \frac{\sum_{i \in [m]} \varphi_i(u)}{m}.$$ 

$\varphi$ is satisfiable if $\text{val}(\varphi) = 1$.

We call $q$ the *arity* of $\varphi$. 
Remarks

1. The size of a $q$CSP-instance is the number of constraints it has.

2. We can always assume $n \leq qm$.

3. A $q$CSP instance over $n$ variables with $m$ constraints can be described by a binary string of length $O(mq2^q \log n)$.

4. The simple greedy approximation algorithm for 3SAT can be generalized for the MAX$q$CSP problem of maximizing the number of satisfied constraints in a given $q$CSP instance. For any $q$CSP instance $\varphi$ with $m$ constraints, this algorithm will output an assignment satisfying $\text{val}(\varphi)m/2^q$ constraints.
**Definition**

Let \( q \in \mathbb{N} \) and \( \rho \leq 1 \). Then the \( \rho\text{-GAP}q\text{CSP} \) is the problem of determining for a given \( q\text{CSP} \)-instance \( \varphi \) whether:

1. \( \text{val}(\varphi) = 1 \) (\( \varphi \) is a YES instance of \( \rho\text{-GAP}q\text{CSP} \)), or
2. \( \text{val}(\varphi) < \rho \) (\( \varphi \) is a NO instance).

\( \rho\text{-GAP}q\text{CSP} \) is NP-hard if there is a polynomial time function \( f \) mapping strings to (representations of) \( q\text{CSP} \) instances satisfying:

1. Completeness: \( x \in L \implies \text{val}(f(x)) = 1 \).
2. Soundness: \( x \notin L \implies \text{val}(f(x)) \leq \rho \).

**Theorem**

*There exist constants \( q \in \mathbb{N} \) and \( \rho \in \mathbb{R} \) with \( 0 < \rho < 1 \) such that \( \rho\text{-GAP}q\text{CSP} \) is NP-hard.*
Theorem

The following are equivalent.

1. $\text{NP} = \text{PCP}(\log n, 1)$.

2. There exist constants $q \in \mathbb{N}$ and $\rho \in \mathbb{R}$ with $0 < \rho < 1$ such that $\rho$-$\text{GAPqCSP}$ is $\text{NP}$-hard.
From $\text{NP} \subseteq \text{PCP}(\log n, 1)$ we show $1/2$-GAP$q$CSP is NP-hard for some $q \in \mathbb{N}$.

- 3SAT has a verifier $V$ which makes $q$ queries and uses $c \log n$ random coins.
- Let $x \in \{0, 1\}^n$ and $r \in \{0, 1\}^{c \log n}$. Then $V_{x,r}$ is the function that on input $\pi$ outputs 1 if $V$ accepts $\pi$ on input $x$ and coins $r$.
- $V_{x,r}$ depends on at most $q$ locations. Thus for every $x \in \{0, 1\}^n$

$$\varphi = \{ V_{x,r} \}_{r \in \{0,1\}^{c \log n}}$$

is a polynomial-sized $q$CSP instance.
- Since $V$ runs in polynomial time, $x \mapsto \varphi$ is computable in polynomial time.
- By the completeness, if $x \in \text{3SAT}$, then $\text{val}(\varphi) = 1$.
- By the soundness, if $x \notin \text{3SAT}$, then $\text{val}(\varphi) \leq 1/2$. 
Assume $\rho$-$\text{GAP}_q\text{CSP}$ is NP-hard for some $\rho < 1$ and $q \in \mathbb{N}$. Then this easily translates into a PCP with $q$ queries, $\rho$ soundness, and logarithmic randomness for any language $L \in \text{NP}$:

- Given an input $x$, the verifier $V$ will run the reduction $f(x)$ to obtain a $q\text{CSP}$ instance $\varphi = \{\varphi_i\}_{i \in [m]}$.
- $V$ expects $\pi$ to be an assignment to the variables of $\varphi$, which it will verify by choosing a random $i \in [m]$ and checking that $\varphi_i$ is satisfied (by making $q$ queries).
- If $x \in L$, then the verifier will accept with probability 1.
- If $x \notin L$, it will accept with probability at most $\rho$. 
NP ⊆ PCP(poly(n), 1)
The **Walsh-Hadamard code** encode bit strings of length $n$ by linear functions in $n$ variables over $\text{GF}(2)$. Let $u \in \{0, 1\}^n$. Then

$$\text{WH}(u) = x \mapsto u \odot x = \sum_{i \in [n]} u_i x_i \pmod{2}.$$
Random Subsum Principle

**Theorem**

Let \( u \neq v \). Then

\[
\Pr_x \left[ u \odot x \neq v \odot x \right] = 1/2.
\]
Definition
Let $\rho \in \mathbb{R}$ with $0 \leq \rho \leq 1$. Two functions $f, g : \{0, 1\}^n \to \{0, 1\}$ are $\rho$-close if
\[
\Pr_x [f(x) = g(x)] \geq \rho.
\]

$f$ is $\rho$-close to a linear function if there exists a linear function $g$ such that $f$ and $g$ are $\rho$-close.

Theorem
Let $f : \{0, 1\}^n \to \{0, 1\}$ be such that
\[
\Pr_{x, y} [f(x + y) = f(x) + f(y)] \geq \rho
\]
for some $\rho > 1/2$. Then $f$ is $\rho$-close to a linear function.
Local decoding of Walsh-Hadamard code

Let $\delta < 1/4$ and $f : \{0, 1\}^n \to \{0, 1\}$ be $(1 - \delta)$-close to some linear function $\tilde{f}$. Then $\tilde{f}$ is uniquely determined.

Suppose we are given $x \in \{0, 1\}^n$ and random access to $f$. Can we obtain the value $\tilde{f}(x)$ using only a constant number of queries?

1. Choose $x' \in_R \{0, 1\}^n$.
2. $x'' \leftarrow x + x'$.
3. $y' \leftarrow f(x')$ and $y'' \leftarrow f(x'')$.
4. Output $y' + y''$.

Lemma
\[ \Pr[\tilde{f}(x) = y' + y''] \geq 1 - 2\delta. \]
Example
The all one assignment satisfies

\[ u_1 u_2 + u_3 u_4 + u_1 u_5 = 1 \]
\[ u_2 u_3 + u_1 u_4 = 0 \]
\[ u_1 u_4 + u_3 u_5 + u_3 u_4 = 1. \]

QUADEQ is the problem, given \( m \times n^2 \) matrix \( A \) and an \( m \)-dimensional vector \( b \), of finding an \( n^2 \)-dimensional vector \( U \) satisfying

1. \( AU = b \),
2. \( U \) is the tensor product \( u \otimes u \) of some \( n \)-dimensional vector \( u \)

Lemma

QUADEQ is NP-complete.