PCP Theorem And Its Applications

Presented by Tomáš Masařík, Dušan Knop, Martin Böhm, Vojta Tůma
Spring School 2014

PCP Theorem

D(α-approximation**):** An algorithm A is an α-approximation algorithm for a maximization problem P if the maximization value $v(A) \geq \alpha OPT(P)$, where $OPT(P)$ is the optimal solution.

Motivation: It is easy to recognize a valid α-approximation, but how can we recognize that no α-approximation algorithm exists for a given $\alpha < 1$?

D(PCP-machine**):** A Turing machine T is a PCP-machine if it has access to four tapes: a tape with the *input* and a *work tape* as usual, a random-access tape with a *proof* of possibly exponential size, and a *random tape* containing r random bits.

D(PCP complexity class**):** A language $L \in PCP(p, q)$ if there exists a PCP-machine T such that on input x, T can access p random bits and can also access q bits of the proof. This machine must then satisfy the following:

- If $x \in L$, then there is a proof y that makes T accept with probability 1.
- If $x \notin L$, then for every proof y, T accepts with probability $< 1/2$.

T(PCP Theorem**):** $NP = PCP(\mathrm{O}(\log n), \mathrm{O}(1))$.

T(Hastad**):** $NP = PCP(\mathrm{O}(\log n), 3)$.

T(Weaker PCP**):** For a fixed c, $NP = PCP(n^c, 1)$.

Proof of Weaker PCP

D: For two vectors $x, y \in \{0,1\}^n$, we define $x \circ y = \sum_{i=1}^{n} x_i y_i \bmod 2$. This corresponds to the number of 1-bits x and y have in common.

L(Random substring principle**):** If $u \neq v$ then for a half of the possible choices of $x \in \{0,1\}^n$, $u \circ x \neq v \circ x$.

L(Linearity testing**):** After $\mathrm{O}(1/\delta)$ independently random linearity checks, we can correctly decide with probability at least $1/2$ whether or not f is a function $(1-\delta)$-close to a linear function, that is: $Pr_{x,y}[f(x+y) = f(x) + f(y)] \geq 1 - \delta$.

D(QUADEQ**):** QUADEQ is an NP-complete language of systems of quadratic equations over $\mathbb{Z}_2$ that are satisfiable. In other words, we get a system of equations and ask for a solution.

D(Tensor product**):** For two n-dimensional vectors a, b, we have $a \otimes b = (a_1b_1, a_1b_2, a_1b_3, \ldots, a_nb_n)$. In other words, we do a matrix product $a \cdot b^T$ and read the matrix of size $n \times n$ as a big vector from left to right.

O: QUADEQ is the following problem: given A matrix of size $m \times n^2$ and an m-dimensional vector b, find an n^2-dimensional vector U such that $AU = b$ and U is the tensor product $u \otimes u$ for some n-dimensional vector u.

Relation of PCP and CSP

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

- $x \in L \Rightarrow \mathrm{val}(f(x)) = 1$;
- $x \notin L \Rightarrow \mathrm{val}(f(x)) < \rho$.

D(CSP**):** If $q \in \mathbb{N}$ (*arity*), then a $qCSP$ instance φ is a collection of functions $\varphi_1, \ldots, \varphi_m$ (*constraints*) from $\{0,1\}^n$ to $\{0,1\}$ such that each function φ_i depends on at most q of its input bits.

We say that an *assignment* $u \in \{0,1\}^n$ *satisfies* a constraint φ_i if $\varphi_i(u) = 1$. Let $\mathrm{val}(\varphi)$ denote the relative maximum of satisfied constraints for any assignment u. If $\mathrm{val}(\varphi) = 1$, we say φ is *satisfiable*.

D(Gap-CSP**):** For ever $q \in \mathbb{N}, \rho < 1$, define ρ-GAP $qCSP$ to be the problem of determining the following:

For a given $qCSP$ instance φ whether:

1. $\mathrm{val}(\varphi) = 1$,
2. $\mathrm{val}(\varphi) < \rho$.

D: We say that ρ-GAP $qCSP$ is NP-hard for every language $L \in NP$ if there is a polynomial-time function f mapping strings to $qCSP$ instances satisfying:

1. $x \in L \Rightarrow \mathrm{val}(f(x)) = 1$,
2. $x \notin L \Rightarrow \mathrm{val}(f(x)) < \rho$.

T(GAP Hardness**):** There exist constants $q \in \mathbb{N}, \rho \in (0,1)$ such that ρ-GAP $qCSP$ is NP-hard.

L: PCP Theorem implies GAP Hardness.

L: Hardness of Approximation View is equivalent to GAP Hardness.

Exercise Session 1

Exercise 1. Prove that the theorem GAP Hardness implies the PCP Theorem.

Exercise 2. Prove that any language L that has a PCP-verifier using r random bits and q *adaptive* queries also has a non-adaptive verifier using r random bits and 2^q queries.

Exercise 3. Prove that:

- $PCP(0,0) = PCP(0, \mathrm{O}(\log n)) = P$.
- $PCP(0, \mathrm{O}(\mathrm{poly}(n))) = NP$.
- $PCP(\mathrm{O}(\mathrm{poly}(n)), 0)) = co - RP$.
- $PCP(\mathrm{O}(\log n), \mathrm{O}(1)) = PCP(\mathrm{O}(\log n), \mathrm{O}(\mathrm{poly}(n)))$.

Exercise 4. Prove that $PCP(\mathrm{O}(\mathrm{poly}(n)), \mathrm{O}(1)) \subseteq NP$.

Reductions using PCP

D: Let P be a maximization problem. A *gap-introducing reduction* from some NP-hard problem H to P is a reduction that comes with two parameters, f and α. Given an instance i of the problem H, we want to output an instance $p \in P$ such that:

- if $i \in H$: $OPT(p) \geq f(p)$;
- if $i \notin H$: $OPT(p) < \alpha(|p|)f(p)$.

D: Let R, P be maximization problems. In a *gap-preserving reduction* from R (with associated f_1, α) to P (with associated f_2, β), we want for every instance $r \in R$ output $p \in P$ such that:

- if $OPT(r) \geq f_1(r)$, then $OPT(p) \geq f_2(p)$.
- if $OPT(r) < \alpha(|r|)f_1(r)$ then $OPT(p) < \beta(|p|)f_2(p)$.

T(Stronger Hastad**):** $NP \subseteq PCP_{1-\varepsilon, 1/2+\varepsilon}(\mathrm{O}(\log n), 3)$, and the verifier can only use functions odd and even on the three bits.

T(CSP view of Stronger Hastad**):** There exists no $\alpha > 1/2$ approximation algorithm for the odd/even CSP unless P=NP.

D: MAX 3-LIN is a maximization problem where the goal is to satisfy as many linear equations as possible. GAP 3-LIN is the gap version of MAX 3-LIN.

O: GAP-3-LIN with parameters $1-\varepsilon, 1/2+\varepsilon$ is hard to approximate due to Stronger Hastad, even for equations modulo 2.

T(E3SAT 13/14**):** There exists no $\alpha > 13/14$-approximation algorithm for E3SAT unless P=NP.

T(E3SAT 7/8**):** There exists no $\alpha > 13/14$-approximation algorithm for E3SAT unless P=NP.

T(VERTEX COVER 7/6**):** For all constants $\varepsilon > 0$, VERTEX COVER is NP-hard to approximate within a factor of $7/6 - \varepsilon$.

Exercise Session 2

Exercise 1. Show an 1/2-approximation algorithm for the odd/even CSP problem.

Exercise 2. Let MAX 3-MAJ be the optimization problem where the input is a set of constraints over 3 boolean literals each, where each constraint is of type "the majority of these three variables have value 1". Show that there is no $2/3 + \varepsilon$-approximation for MAX 3-MAJ unless P=NP.

Exercise 3. Let MAX 3-SAT(k) be a MAX 3-SAT problem where each variable occurs at most k times. Give a gap-preserving reduction from MAX 3-SAT(29) to MAX 3-SAT(5), with appropriate parameters, to show hardness for MAX-3SAT(5).