New insights into Probabilistically Checkable Proofs (PCPs)

Eli Ben-Sasson
Computer Science Department
Technion

WoLLIC `06, Stanford, July 2006
Talk outline

- Probabilistically checkable proofs (PCPs)
  - Definition and statement of results
  - Applications

- PCP building blocks
  - Sublinear coding theory
  - PCPs of proximity
  - Soundness preservation/amplification
NP – Efficient proof verification

\[ M_L \]

\[ x \in L \text{ iff } \exists y M_L(x, y) = \text{accept} \]

**Efficiency:** \( M_L \) runs in deterministic polynomial time in \(|x|\)

**Completeness:** \( x \in L \Rightarrow \exists y, M_L(x, y) = \text{accept} \)

**Soundness:** \( x \notin L \Rightarrow \forall y, M_L(x, y) = \text{reject} \)
NP – Efficient proof verification

$M_L \quad x \in L \iff \exists y M_L(x,y) = \text{accept}$

Efficiency: $M_L$ runs in deterministic polynomial time in $|x|$

Completeness: $x \in L \Rightarrow \exists y, M_L(x,y) = \text{accept}$

Soundness: $x \notin L \Rightarrow \forall y, M_L(x,y) = \text{reject}$
NP – Efficient proof verification

\[(x \lor y \lor \bar{z}) \land \vdots \land (\bar{x} \lor y \lor z)\]

\[\mathcal{M}_L(x, y) = \text{accept}\]

Efficiency: \(\mathcal{M}_L\) runs in deterministic polynomial time in \(|x|\)

Completeness: \(x \in L \Rightarrow \exists y, M_L(x, y) = \text{accept}\)

Soundness: \(x \notin L \Rightarrow \forall y, M_L(x, y) = \text{reject}\)
PCP – Super-Efficient Proof Verification

Efficiency: $V$ runs in randomized polynomial time in $|x|$

Completeness: $x \in L \Rightarrow \exists \pi, \Pr[V^\pi(x) = \text{accept}] = 1$

Soundness: $x \notin L \Rightarrow \forall \pi, \Pr[V^\pi(x) = \text{reject}] \geq 1/2$
PCP – Super-Efficient Proof Verification

Pros
• Few queries into proof $\pi$
• Running time polylog($\pi$)

Cons
• Errors possible
• Proofs longer

Efficiency: $V$ runs in randomized polynomial time in $|x|$

Completeness: $x \in L \Rightarrow \exists \pi, \Pr[V^\pi(x) = \text{accept}] = 1$

Soundness: $x \notin L \Rightarrow \forall \pi, \Pr[V^\pi(x) = \text{reject}] \geq 1/2$
Definition: PCP language class

We say $L \in \text{PCP}$ if there exists verifier $V = V_L$ that on input $x$, $|x| = n$, runs in time $t(n)$, makes $q(n)$ queries to a proof of length $l(n)$, such that:

\[
\begin{align*}
\text{Completeness: } & \quad x \in L \implies \exists \pi, \ \Pr[V^\pi(x) = \text{accept}] \geq c(n) \\
\text{Soundness: } & \quad x \notin L \implies \forall \pi, \ \Pr[V^\pi(x) = \text{reject}] \geq s(n)
\end{align*}
\]
PCP Theorems

Thm: \( \text{NP} \subseteq \text{PCP} \)

\[
\begin{array}{c|c|c}
\text{time} & n^{O(1)} & \text{comp.} \\
\text{length} & n^{O(1)} & \geq 1 \\
\text{query} & O(1) & \geq 1/2 \\
\end{array}
\]

Two settings, two applications:

- Hardness of approximation [FGL+91]
## PCP Theorems

Thm: $\text{NP} \subseteq \text{PCP}$

- $t(n)$ time
- $l(n)$ length
- $q(n)$ query
- $\overline{\text{comp.}} \geq c(n)$
- $\overline{\text{sound.}} \geq s(n)$

Two settings, two applications:

- Hardness of approximation [FGL+91]
- Super-efficient proof/computation verification [BFL+91]
PCPs and Hardness of approximation \([FGL+91]\)

Example \([Hås97]\):

Thm: \(\text{NP} \subseteq \text{PCP}\)

<table>
<thead>
<tr>
<th>time (\leq n^{O(1)})</th>
<th>comp. (\geq 1-\epsilon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>length (\leq n^{O(1)})</td>
<td>sound. (\geq 1/2-\epsilon)</td>
</tr>
<tr>
<td>query (\leq 3) bits</td>
<td></td>
</tr>
</tbody>
</table>

\(V\) computes XOR of 3 answer bits

List all possible verifier tests:

\[y_1 \oplus y_2 \oplus y_3 = 1\]
\[y_3 \oplus y_5 \oplus y_{20} = 0\]
\[\vdots\]

Completeness: \(x \in L\): Exists \(y\) satisfying \(1-\epsilon\) fraction of constraints

Soundness: \(x \notin L\): Every \(y\) satisfies \(\leq 1/2-\epsilon\) frac. of constraints

**Corollary:** NP-hard to 2-approximate MAX3LIN.

NP-hard to 8/7-approximate MAX3SAT.
PCPs and Hardness of approximation [FGL+91]

Thm: $\mathsf{NP} \subseteq \mathsf{PCP}$

\[
\begin{array}{cccc}
\text{time} & \leq & n^{O(1)} \\
\text{length} & \leq & n^{O(1)} \\
\text{query} & \leq & O(1) \\
\text{comp.} & \geq & 1 \\
\text{sound.} & \geq & 1/2
\end{array}
\]

- Many hardness of approximation results
  - [Hås96] Clique $n^{1-\varepsilon}$
  - [Hås97] MAX3SAT $8/7 - \varepsilon$
  - [Hås97] MAXCUT $17/16$
  - [Fei98] Set Cover $(1-\varepsilon) \ln n$
  - [DR02] Vertex cover $1.36$
  - ...

...
**PCPs and super-efficient verification [BFL+91]**

Thm [BS05; BGH+05]: \( \text{NTIME}(f(n)) \subseteq \)

- Not enough time to read input \( x \) (!)
- Settle for approximate soundness:
  - If input \( x \) is not in \( L \), then \( V \) rejects.

\[ \begin{align*}
\text{PCP} & \quad \text{time} \quad \leq \quad f^{O(1)}(n) \\
& \quad \text{length} \quad \leq \quad f(n) \cdot \text{polylog} f(n) \\
& \quad \text{query} \quad \leq \quad \text{polylog} f(n) \\
\end{align*} \]

- \( \text{comp.} \geq 1 \)
- \( \text{sound.} \geq 1/2 \)

Proof Carrying Codes
[Necula, Lee]

Software Consumer + Software Producer
PCPs and super-efficient verification [BFL+91]

Thm [BS05; BGH+05]: \( \text{NTIME}(f(n)) \subseteq \)

- Not enough time to read input \( x \) (!)
- Settle for approximate soundness:
  If input \( x \) is not in \( L \), then \( V \) rejects.
  far (in Hamming distance) from

Proof Carrying Codes
[Necula, Lee]

Software Consumer

\[ [\text{Kil92}], [\text{Mic94}] \]

Software Producer
Talk outline

✓ Probabilistically checkable proofs (PCPs)
  ✓ Definition and statement of results
  ✓ Applications

- PCP building blocks
  - Sublinear coding theory
  - PCPs of proximity
  - Soundness preservation/amplification
PCP Blueprint

- Want to verify that $y$ witnesses $x$ is in $L$
- Encode $y$, "spreading" its information. Minimal requirements from code:
  - Locally testable
  - Locally decodable
- Problem: Too many queries/too little soundness
- Solution: Proof composition
Error Correcting Codes

Encoding: $E: \{0,1\}^k \rightarrow \{0,1\}^n$, $C = \{E(m): m \text{ in } \{0,1\}^k\}$

Rate $= k/n$, blowup $= 1/rate$

Distance:
\[
\delta(x, y) = \Pr_{i \in [n]}[x_i \neq y_i]
\]
\[
\delta(C) = \min_{x \neq y \in C} \{\delta(x, y)\}
\]
\[
\delta_C(w) = \min_{x \in C} \{\delta(w, x)\}
\]

Message space $= \{0, 1\}^k$

Code space $= \{0, 1\}^n$
Sub-linear coding algorithms

Running time = $o(n)$, typically $\text{poly}(\log n)$

- Want “good” code (large rate and distance) s.t.
  - Sub-linear time for encoding $i^{\text{th}}$ bit
  - Sub-linear distance estimation
    locally testable code (LTC)
  - Sub-linear decoding of one message-bit
    locally decodable code (LDC)
Locally Testable Code

- \( t(n) = o(n) \), think of polylog \( n \)
- \( q(n) = o(n) \), think of \( O(1) \)
- Comp. : \( w \in C \Rightarrow \Pr[\text{Testerm}=\text{accept}] = 1 \)
- Sound. : \( \delta_C(w) > \delta_0 \Rightarrow \Pr[\text{Testerm}=\text{reject}] > .99 \)

Def: Implicit in [BFL+91], explicit in [Aro94; Spi95; FS95]
Locally Decodable Code

Let $F$ be family of Boolean functions on $k$ bits

$F$ is loc. dec. from $E$ if $t(n), q(n)=o(n)$ and

for all $f$ in $F$,

Comp.: $\delta(w, E(m)) < \delta_0 \Rightarrow \Pr[\text{Dec. } w(f) = f(m)] \geq .99$

Remark: No soundness requirement

Def: Implicit in [BFL+91; Sud92], explicit in [KT00]
LTCs and LDCs – brief comparison

- Applications (other than PCPs and coding theory)
  - LTCs: Property testing
  - LDCs: Derandomization, Cryptography, Private Information Retrieval

- Rate comparison for $q = O(1)$
  - LTCs: $n = k \cdot \text{polylog } k$ [BS05; Din06]
  - LDCs: $n = \exp(k^\epsilon)$ [BIK+02]
LTCs – results

- Positive (constructions)
  - Hadamard codes [BLR90; BCH+96]
  - Reed-Muller codes [BFL+91; ALM+92; AS97; RS97 …]
  - Derandomized Hadamard/Reed-Muller testers [GS02; BSV+03; BGH+04; SW04; BS05; RM06]
  - Tensor codes [BS04; DSW06]

- Negative (lower bounds)
  - $q=2$ [BGS03]
  - LDPC expander codes [BHR03]
  - Cyclic codes [BSS05]
  - Two-wise tensor [Val05; CR05]
  - Very little known…
LDCs - results

- Positive (lower bounds)
  - Hadamard codes [BLR90]
  - Reed-Muller codes [BF90]
  - Improvements [Amb97; IK99; BI01; BIKR02]

- Negative (lower bounds)
  - [Man98; KT00; GKS+02; Oba02]
  - Exponential lower bounds for $q=2$ [KdW03]
  - Very little known ...
LTCs, LDCs and PCP Blueprint

Given $x$ as input, request $E(y)$, where
- $E$ is Locally testable
- “Interesting” $F$ is locally decodable from $E$

Use $F$ to locally test that $y$ witnesses $x$ is in $L$
**Example: Hadamard-Walsh based PCP**

Given $x$ as input, request $E(y)$, where
- $E$ is Locally testable
- “Interesting” $F$ is locally decodable from $E$

Use $F$ to locally test that $y$ witnesses $x$ is in $L$

$E$ is a LTC, with 3 queries [BLR90]

Every linear function is Loc. Dec. from $E$, with 2 queries

Verifying $x$ is in $L$ can be reduced to decoding a constant number of linear functions [ALM+91]

Problem: rate... $E : \{0, 1\}^k \rightarrow \{0, 1\}^{2^k}$
Talk outline

✓ Probabilistically checkable proofs (PCPs)
  ✓ Definition and statement of results
  ✓ Applications

- PCP building blocks
  ✓ Sublinear coding theory
- PCPs of proximity
- Soundness preservation/amplification
Proof Composition [AS91]

**Problems**

- If $q(n) = O(1)$, $s(n) = 1/2$, then $l(n) = \exp(n^2)$
- If $l(n) = \text{poly}(n)$, $q(n) = O(1)$, then $s(n) = 1/n$
- If $l(n) = \text{poly}(n)$, $s(n) = 1/2$, then $q(n) = \text{polylog}(n)$

**Solution**

Proof composition
PCPs of Proximity/Assignment testers
[BGH+05; DR05]

Let $L_2 = \{(x,y) : M_L(x,y) = \text{accept}\}$

Let $L_x = \{y : M_L(x,y) = \text{accept}\}$

A PCPP-verifier $V$ verifies that $y$ is close to $L_x$
PCPs of Proximity/Assignment testers
[BGH+05; DR05]

Definition:
We say $L_2 \in \text{PCPP}$

\[
\begin{bmatrix}
\text{time} & \leq t(n) \\
\text{length} & \leq l(n) \\
\text{query} & \leq q(n)
\end{bmatrix}
\begin{array}{c}
\text{comp.} = 1 \\
\text{sound.} \geq 0.99
\end{array}
\]

If there exists a nonadaptive PCPP verifier $V$ running in time $t(n)$, making $q(n)$ queries to a proof of length $l(n)$, such that:

Completeness: $y \in L_x \Rightarrow \exists \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_x)] = 0$

Robust Soundness: $\forall \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_x)] \geq 0.99 \cdot \delta(y, L_x)$
Theorem [BS05; Din06]: If $L \in \text{NTIME}(f(n))$, then

$L_2 \in \text{PCPP}$

$$
\begin{align*}
\text{time} & \leq f^{O(1)}(n) \\
\text{length} & \leq f(n) \cdot \text{polylog}(f(n)) \\
\text{query} & \leq O(1)
\end{align*}
$$

<table>
<thead>
<tr>
<th>comp.</th>
<th>sound.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$= 1$</td>
<td>$\geq 0.99$</td>
</tr>
</tbody>
</table>

Completeness: $y \in L_x \Rightarrow \exists \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_{x'})] = 0$

Robust Soundness: $\forall \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_{x'})] \geq 0.99 \cdot \delta(y, L_x)$
PCPs of Proximity/Assignment testers
[BGH+05; DR05]

PCPPs - History

- Holographic proofs - PCPPs where assignment $y$ is encoded. [BFL+91]
- PCPP - implicit in low-degree tests [RS92; ALM+91]
- PCPPs - special case of “PCP Spot Checkers” [EKR99]
- PCPP – extension of Property Testing [RS92; GGR96]
Applications of PCPPs

- PCPPs yield PCPs
- Simpler proof composition, essential in
  - Shorter PCPs [BGH+05; BS05; BGH+06]
  - PCPs via gap amplification [Din06]
- Coding
  - Locally Testable Codes [GS02; BSV+03; BGH+05...]
  - Relaxed Locally Decodable Codes [BGH05+]
- Property testing
  - Every property is locally testable (with a little help)
  - Lower bounds for tolerant testing [FF05]
Completeness:  \( y \in L_x \Rightarrow \exists \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_x)] = 0 \)

Soundness:  \( \forall \pi \mathbb{E} [\delta (\langle y, \pi \rangle | Q, L_x)] \geq 0.99 \cdot \delta(y, L_x) \)
Talk outline

✓ Probabilistically checkable proofs (PCPs)
  ✓ Definition and statement of results
  ✓ Applications

- PCP building blocks
  ✓ Sublinear coding theory
  ✓ PCPs of proximity
- Soundness preservation/amplification
Putting it all together

- Algebraic approach
  - Encode using LTCs/LDCs based on polynomials, specifically, Reed-Solomon and Reed-Muller codes
  - Large $q$, large $s$
  - PCPP Composition to reduce $q$, while preserving $s$

- Expander-based approach [Din06]
  - Constant $q$, small $s$
  - Randomness-efficient repetition to boost $s$ (but $q$ also increases)
  - Encode using simple, rate-inefficient LTCs/LDCs
  - PCPP Composition to reduce $q$, while preserving $s$
**PCP via gap amplification** [Din06]

**Gap amplification:** There exists $c>0$ s.t. for $s(n)<c$,

<table>
<thead>
<tr>
<th>PCP</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$\leq t(n)$</td>
<td>comp.</td>
<td>$\geq 1$</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>$\leq l(n)$</td>
<td>sound.</td>
<td>$\geq s(n)$</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>$\leq 2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCP</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$\leq O(t(n))$</td>
<td>comp.</td>
<td>$\geq 1$</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>$\leq O(l(n))$</td>
<td>sound.</td>
<td>$\geq 2\cdot s(n)$</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>$\leq 2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Proof of PCP Theorem:**

<table>
<thead>
<tr>
<th>NP $\subseteq$ PCP</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$\leq n^{O(1)}$</td>
<td>comp.</td>
<td>$\geq 1$</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>$\leq n^{O(1)}$</td>
<td>sound.</td>
<td>$\geq 1/n$</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>$\leq 2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Apply gap amplification log $\log n$ times...

<table>
<thead>
<tr>
<th>$\subseteq$ PCP</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$\leq n^{O(1)}$</td>
<td>comp.</td>
<td>$\geq 1$</td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>$\leq n^{O(1)}$</td>
<td>sound.</td>
<td>$\geq c$</td>
<td></td>
</tr>
<tr>
<td>query</td>
<td>$\leq 2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QED
PCP via gap amplification [Din06]

Gap amplification: There exists $c > 0$ s.t. for $s(n) < c$,

\[
\begin{array}{ccc}
\text{time} & \leq & t(n) \\
\text{length} & \leq & l(n) \\
\text{query} & \leq & 2 \\
\end{array}
\quad \begin{array}{ccc}
\text{comp.} & \geq & 1 \\
\text{sound.} & \geq & s(n) \\
\end{array}
\subseteq
\begin{array}{ccc}
\text{time} & \leq & O(t(n)) \\
\text{length} & \leq & O(l(n)) \\
\text{query} & \leq & 2 \\
\end{array}
\quad \begin{array}{ccc}
\text{comp.} & \geq & 1 \\
\text{sound.} & \geq & 2 \cdot s(n) \\
\end{array}
\]

Constraint graph

- Vertices: Proof symbols
- Edges: constraints over pair of queries
- $x \in L \Rightarrow$ All constraints can be satisfied
- $x \notin L \Rightarrow$ At least $s(n)$ frac. of constraints reject
PCP via gap amplification [Din06]

Gap amplification: There exists $c > 0$ s.t. for $s(n) < c$,

$$
\begin{align*}
\text{PCP} & : \begin{cases}
\text{time} & \leq t(n) \\
\text{length} & \leq l(n) \\
\text{query} & \leq 2
\end{cases} \quad \text{comp.} \geq 1 \\
\text{sound.} & \geq s(n)
\end{align*}
\subseteq
\begin{align*}
\text{PCP} & : \begin{cases}
\text{time} & \leq O(t(n)) \\
\text{length} & \leq O(l(n)) \\
\text{query} & \leq 2
\end{cases} \quad \text{comp.} \geq 1 \\
\text{sound.} & \geq 2 \cdot s(n)
\end{align*}
$$

Boosting soundness – 1st attempt

- Query 100 edges (sequential repetition)
- $x \in L \Rightarrow$ all constraints can be satisfied
- $x \notin L \Rightarrow$ at least $10s(n)$ frac. of constraints reject
- Problem: $q$ is large
PCP via gap amplification [Din06]

Gap amplification: There exists $c > 0$ s.t. for $s(n) < c$,

$$
\begin{array}{c|c|c}
\text{PCP} & \text{time} & \leq t(n) \\
\text{PCP} & \text{length} & \leq l(n) \\
\text{PCP} & \text{query} & \leq 2
\end{array}
\begin{array}{c|c|c}
\text{comp.} & \geq 1 \\
\text{sound.} & \geq s(n)
\end{array}
\subseteq
\begin{array}{c|c|c}
\text{time} & \leq O(t(n)) \\
\text{length} & \leq O(l(n)) \\
\text{query} & \leq 2
\end{array}
\begin{array}{c|c|c}
\text{comp.} & \geq 1 \\
\text{sound.} & \geq 2 \cdot s(n)
\end{array}
$$

Boosting soundness – 2\textsuperscript{nd} attempt

- Encode ass. to every 100-tuple of vertices using LDC/LTC
- Pick 100 edges, make 2 queries to get ass. to endpoints
- Use PCPPs to prove codewords satisfy all constraints
- $q=2, c=1, \text{sound.} > 9s(n)$
- Problems: (1) $l=n^{100}$, (2) consistency
PCP via gap amplification [Din06]

Gap amplification: There exists $c > 0$ s.t. for $s(n) < c$,

\[
\begin{array}{c|c|c}
\text{time} & t(n) & \geq 1 \\
\text{length} & l(n) & \geq s(n) \\
\text{query} & 2 & \geq 2 \cdot s(n) \\
\end{array}
\]

Boosting soundness – 3rd (final) attempt

- W.l.o.g. $G$ is constant degree regular expander graph
- Encode assignment to ball of radius 100 around every $v$ using LDC/LTC
- Pick $u,v$ at distance 150, query balls around $u,v$
- Use PCPPs to prove balls agree and satisfy intersection
  - $q=2, c=1$, sound. $> 4s(n), l=O(n)$ (deg$(G)=O(1)$)
  - Problem: consistency. Solution: $G$ is an expander... QED
Summing up

- PCPs are fundamental computational objects used in:
  - Hardness of approximation
  - Super-efficient verification of proofs
- Main building blocks:
  - Locally testable and decodable codes
  - PCPP composition
  - Soundness amplification/preservation
- Open question:

\[
NP \subseteq \text{PCP} \left[ \begin{array}{c}
\text{time} \leq n^{O(1)} \\
\text{length} \leq n \log^{O(1)} n \\
\text{query} \leq 3 \text{ bits}
\end{array} \right] \begin{array}{c}
\text{comp.} \geq 1 - \epsilon \\
\text{sound.} \geq 1/2 - \epsilon
\end{array}
\]
New insights into Probabilistically Checkable Proofs (PCPs)

Eli Ben-Sasson
Computer Science Department
Technion

Thank you

WoLLIC `06, Stanford, July 2006