PCP Theorem

[PCP Theorem is] the most important result in complexity theory since Cook's Theorem.

Ingo Wegener, 2005

S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems. J. ACM, 1998. FOCS 1992.

Irit Dinur. The PCP Theorem by Gap Amplification. J. ACM, 2007. STOC 2006.

The 1^{st} proof is algebraic, the 2^{nd} one is combinatorial and non-elementary.

Two ways to view the PCP Theorem:

- It is a result about locally testable proof systems.
- It is a result about hardness of approximation.

PCP = Probabilistically Checkable Proof

Synopsis

1. Approximation Algorithm
2. Two Views of PCP Theorem
3. Equivalence of the Two Views
4. Inapproximability
5. Fourier Transform Technique
6. Efficient Conversion of NP Certificate to PCP Proof
7. Proof of PCP Theorem
8. Håstad's 3-Bit PCP Theorem
9. Historical Remark

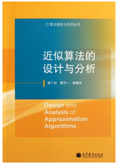
Approximation Algorithm

Since the discovery of NP-completeness in 1972, researchers had been looking for approximate solutions to NP-hard optimization problems, with little success.

The discovery of PCP Theorem in 1992 explains the difficulty.

---

1. V. Vazirani. Approximation Algorithms. Springer, 2001.
2. D. Williamson, D. Shmoys. The Design of Approximation Algorithms. CUP, 2010.



Suppose $\rho : \mathbf{N} \rightarrow (0,1)$. A ρ-approximation algorithm $\mathbb{A}$ for a maximum, respectively minimum optimization problem satisfies

$$\frac{\mathbb{A}(x)}{Max(x)} \geq \rho(|x|),$$

respectively

$$\frac{Min(x)}{\mathbb{A}(x)} \geq \rho(|x|)$$

for all x.

SubSet-Sum

Given m items of sizes $s_1, s_2, \ldots, s_m$, and a positive integer C, find a subset of the items that maximizes the total sum of their sizes without exceeding the capacity C.

- There is a well-known dynamic programming algorithm.
- Using the algorithm and a parameter ϵ a $(1-\epsilon)$-approximation algorithm can be designed that runs in $O\left((\frac{1}{\epsilon}-1)\cdot n^2\right)$ time.
- We say that SubsetSum has an FPTAS.

$\mathrm{poly}(n, \frac{1}{\epsilon})$.

KnapSack

Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of items to be packed in a knapsack of size C. For $1 \leq j \leq m$, let s_j and v_j be the size and value of the j-th item, respectively.

The objective is to fill the knapsack with items in U whose total size is at most C and such that their total value is maximum.

- There is a similar dynamic programming algorithm.
- Using the algorithm and a parameter ϵ one can design a $(1-\epsilon)$-approximation algorithm of $O\left((\frac{1}{\epsilon} - 1) \cdot n^{\frac{1}{\epsilon}}\right)$ time.
- We say that `KnapSack` has a PTAS.

Max-3SAT

For each 3CNF φ, the value of φ, denoted by $\mathtt{val}(\varphi)$, is the maximum fraction of clauses that can be satisfied by an assignment to the variables of φ.

- φ is satisfiable if and only if $\mathtt{val}(\varphi) = 1$.

Max-3SAT is the problem of finding the maximum $\mathtt{val}(\varphi)$.

- A simple greedy algorithm for Max-3SAT is $\frac{1}{2}$-approximate.
- We say that Max-3SAT is in APX.

By definition, FPTAS $\subseteq$ PTAS $\subseteq$ APX $\subseteq$ OPT.

We will see that the inclusions are strict assuming **P** $\neq$ **NP**.

$\frac{7}{8}$-Approximation Algorithm for Max-3SAT

Let $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_m$. Let $L = \{\varphi_1, \ldots, \varphi_m\}$.

For $j \in [m]$, define the weight $w(\varphi_j) = \frac{1}{2^{|\varphi_j|}}$, where $|\varphi_j|$ is the number of variables in φ_j. $\frac{m}{8}$

Suppose $x_1, \ldots, x_{i-1}$ have been assigned values.

Let L_i be those in L that contain x_i, and $\overline{L_i}$ those that contain $\overline{x_i}$.

1. If $\sum_{C \in L_i} w(C) \geq \sum_{C \in \overline{L_i}} w(C)$, let $x_i := 1$. Remove L_i from L, and remove $\overline{x_i}$ from L.
2. If $\sum_{C \in L_i} w(C) < \sum_{C \in \overline{L_i}} w(C)$, let $x_i := 0$. Remove $\overline{L_i}$ from L, and remove x_i from L.

- Initially the overall weight is $m/8$. This is an invariant property of the algorithm.
- Upon termination, each clause in L has weight 1. There are no more than $\frac{1}{8}m$ clauses left.
- At least $\frac{7}{8}m$ clauses have been removed.

[1] D. Johnson. Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci 9, 256–278, 1974.

Max-IS

Min-VC + Max-IS $= m$.

A $\frac{1}{2}$-approximation algorithm for Min-VC. It turns out to be the best one could have.

1. Pick up a remaining edge and collect the two end nodes.
2. Remove all edges adjacent to the two nodes.
3. Goto Step 1 if there is at least one remaining edge.

- Is Min-VC in PTAS?
- Is Max-IS in APX?

---

1. S. Khot, O. Regev. Vertex Cover Might be Hard to Approximate to within $2-\epsilon$. 18th IEEE Annual Conference on Computational Complexity, 2003.

A breakthrough in the study of approximation algorithm was achieved in early 1990's.

[1991]. There is no $2^{\log^{1-\epsilon}(n)}$-approximation algorithm for Max-IS unless $\mathtt{SAT} \in \mathbf{SUBEXP}$.
[1992]. Max-IS is not in APX if $\mathbf{P} \neq \mathbf{NP}$.

1. U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive Proofs and the Hardness of Approximating Cliques. FOCS 1991. JACM, 1996.
2. S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. FOCS 1992. JACM, 1998.

Two Views of PCP Theorem

Surprisingly, **IP** = **PSPACE**. Even more surprisingly, **MIP** = **NEXP**.

The latter can be interpreted as saying that nondeterminism can be traded off for

randomness + interaction.

Interactive Proof Viewpoint

Suppose L is an NP problem and x is an input string.

1. Prover provides a proof π of polynomial length.
2. Verifier uses at most logarithmic many random bits, and makes a constant number of queries on π.

- A query i is a location of logarithmic length. The answer to query i is $\pi(i)$.
- We assume that verifier is nonadaptive in that its selection of queries is based only on input and random string.

Probabilistically Checkable Proofs

Suppose L is a language and $q, r : \mathbf{N} \rightarrow \mathbf{N}$.

L has an $(r(n), q(n))$-PCP verifier if a P-time verifier $\mathbb{V}$ exists satisfying the following.

- Efficiency. On input x and given access to any location of a proof π of length $\leq q(n)2^{r(n)}$, the verifier $\mathbb{V}$ uses at most $r(n)$ random bits and makes at most $q(n)$ nonadaptive queries to the proof π before it outputs '1' or '0'.
- Completeness. If $x \in L$, then $\exists\pi.\mathrm{Pr}[\mathbb{V}^{\pi}(x) = 1] = 1$.
- Soundness. If $x \notin L$, then $\forall\pi.\mathrm{Pr}[\mathbb{V}^{\pi}(x) = 1] \leq 1/2$.

$\mathbb{V}^{\pi}(x)$ denotes the random variable with x and π fixed.

Probabilistically Checkable Proofs

1. Proof length $\leq q(n)2^{r(n)}$. At most $q(n)2^{r(n)}$ locations can be queried by verifier.
2. $L \in \mathbf{NTIME}(q(n)2^{O(r(n))})$.
    - An algorithm guesses a proof of length $q(n)2^{r(n)}$.
    - It executes deterministically $2^{r(n)}$ times the verifier's algorithm.
    - The total running time is bounded by $q(n)2^{r(n)} \cdot 2^{r(n)} \cdot T(n) \log T(n) = q(n)2^{O(r(n))}$.

Both random bits and query time are resources. An $(r(n), q(n))$-PCP verifier has

- randomness complexity $r(n)$ and
- query complexity $q(n)$.

Sometimes one is concerned with proof complexity $q(n)2^{r(n)}$.

PCP Hierarchy

A language is in $\mathbf{PCP}(r(n), q(n))$ if it has a $(cr(n), dq(n))$-PCP verifier for some c, d.

$$\mathbf{PCP}(r(n), q(n)) \subseteq \mathbf{NTIME}(q(n)2^{O(r(n))}).$$

- $\mathbf{PCP}(0, \log) = \mathbf{P}$.
- $\mathbf{PCP}(0, \mathrm{poly}) = \mathbf{NP}$.
- $\mathbf{PCP}(\log, \mathrm{poly}) = \mathbf{NP}$.

PCP Hierarchy

1. $\mathbf{PCP}(\mathrm{poly}, \mathrm{poly}) \subseteq \mathbf{NEXP}$.
2. $\mathbf{PCP}(\log, \log) \subseteq \mathbf{NP}$.
3. $\mathbf{PCP}(\log, 1) \subseteq \mathbf{NP}$.

In three influential papers in the history of PCP, it is proved that the above '$\subseteq$' can be strengthened to '$=$'.

The PCP Theorem

PCP Theorem. $\mathbf{NP} = \mathbf{PCP}(\log, 1)$.

Every NP-problem has specifically chosen certificates whose correctness can be verified probabilistically by checking only 3 bits.

Example

$\texttt{GNI} \in \textbf{PCP}(\mathrm{poly}, 1)$.

- Suppose both G_0 and G_1 have n vertices.
- Proofs of size 2^{n^2} are indexed by adjacent matrix representations.
  - If the location, a string of size n^2, represents a graph isomorphic to G_i, it has value i.
- The verifier picks up $b \in \{0, 1\}$ at random, produces a random permutation of G_b, and queries the bit of the proof at the corresponding location.

Can we scale down PCP Theorem further?

Fact. If $\mathbf{NP} \subseteq \mathbf{PCP}(o(\log), o(\log))$, then $\mathbf{P} = \mathbf{NP}$.

Scale-Up PCP Theorem

Theorem. $\mathbf{PCP}(\mathrm{poly}, 1) = \mathbf{NEXP}$.

Hardness of Approximation Viewpoint

For many NP-hard optimization problems, computing approximate solutions is no easier than computing the exact solutions.

The PCP Theorem, Hardness of Approximation

PCP Theorem. There exists $\rho < 1$ such that for every $L \in \mathbf{NP}$ there is a P-time computable function $f\colon L \to 3\mathtt{SAT}$ such that

$$\begin{array}{lcl} x \in L & \Rightarrow & \mathtt{val}(f(x)) = 1, \\ x \notin L & \Rightarrow & \mathtt{val}(f(x)) < \rho. \end{array}$$

- Figure out the significance of the theorem by letting $L = 3\mathtt{SAT}$.

The PCP Theorem, Hardness of Approximation

PCP Theorem cannot be proved using Cook-Levin reduction.

- $\mathtt{val}(f(x))$ tends to 1 even if $x \notin L$.

"The intuitive reason is that computation is an inherently unstable, non-robust mathematical object, in the sense that it can be turned from non-accepting to accepting by changes that would be insignificant in any reasonable metric."

Papadimitriou and Yannakakis, 1988

Corollary. There exists some $\rho < 1$ such that if there is a P-time ρ-approximation algorithm for `Max-3SAT` then $\mathbf{P} = \mathbf{NP}$.

- The ρ-approximation algorithm for `Max-3SAT` is NP-hard.

From PCP Verifier to Nonapproximability

Suppose $\mathbb{V}^?$ is a PCP-verifier for L that, upon receiving an input x of length n, generates a $c \log n$ long random string and asks a constant number ℓ of questions.

pcp configuration: $\langle r, q_1, a_1, \ldots, q_\ell, a_\ell \rangle$, or simply $\langle r, a_1, \ldots, a_\ell \rangle$. $|\text{pcp cfg.}| = c\log(n) + \ell$.

Graph G_x:

- the vertices are pcp configurations that force $\mathbb{V}$ to accept.
- $\langle r, q_1, a_1, \ldots, q_\ell, a_\ell \rangle$ and $\langle r', q'_1, a'_1, \ldots, q'_\ell, a'_\ell \rangle$ are connected iff they are consistent in the sense that for all $g, h \in [\ell]$ the equality $q_g = q'_h$ implies the equality $a_g = a'_h$.

Let $\omega(G_x)$ be the largest clique.

Lemma. $\omega(G_x) = \max_\Pi \mathtt{Pr}[\mathbb{V}^\Pi(x) = 1] \cdot 2^{c \log n}$.

Corollary. `MaxClique` does not have $\frac{1}{2}$-approximation algorithm.

Equivalence of the Two Views

CSP, Constraint Satisfaction Problem

If q is a natural number, then a $q\mathtt{CSP}$ instance φ with n variables is a collection of constraints $\varphi_1, \ldots, \varphi_m : \{0,1\}^n \to \{0,1\}$ such that for each $i \in [m]$ the function φ_i depends on q of its input locations.

We call q the arity of φ, and m the size of φ.

Every constraint is of size $O(\log n)$.

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

$$\mathtt{val}(\varphi) = \max_{\mathbf{u}\in\{0,1\}^n} \left\{ \frac{\sum_{i=1}^{n} \varphi_i(\mathbf{u})}{m} \right\}.$$

We say that φ is satisfiable if $\mathtt{val}(\varphi) = 1$.

$q\mathtt{CSP}$ is a generalization of $\mathtt{3SAT}$.

1. We assume that $n \leq qm$.
2. Since every φ_i can be described by a formula of size $q2^q$, and every variable can be coded up by $\log n$ bits, a qCSP instance can be described by $O(mq2^q \log n)$ bits.
3. The greedy algorithm for MAX-3SAT can be applied to MAXqCSP to produce an assignment satisfying $\geq \frac{1}{2^q}\mathtt{val}(\varphi)m$ constraints.

If we think of φ_i as a circuit, it is of constant size.

Gap CSP

Suppose $q \in \mathbf{N}$ and $\rho \leq 1$.

Let ρ-$\texttt{GAP}q\texttt{CSP}$ be the promise problem of determining if a qCSP instance φ satisfies either (1) $\texttt{val}(\varphi) = 1$ or (2) $\texttt{val}(\varphi) < \rho$.

We say that ρ-$\texttt{GAP}q\texttt{CSP}$ is NP-hard if for every NP-problem L some P-time computable function $f : L \to \rho$-$\texttt{GAP}q\texttt{CSP}$ exists such that

$$\begin{aligned} x \in L &\quad\Rightarrow\quad \mathrm{val}(f(x)) = 1, \\ x \notin L &\quad\Rightarrow\quad \mathrm{val}(f(x)) < \rho. \end{aligned}$$

PCP Theorem. There exists some $\rho \in (0, 1)$ such that ρ-$\texttt{GAP3SAT}$ is NP-hard.

PCP Theorem. There exist $q \in \mathbf{N}$ and $\rho \in (0,1)$ such that ρ-$\mathtt{GAP}q\mathtt{CSP}$ is NP-hard.

Equivalence Proof

PCP Theorem ⇒ **PCP Theorem**.

This is essentially the Cook-Levin reduction.

1. Suppose $\mathbf{NP} \subseteq \mathbf{PCP}(\log, 1)$. Then $\mathtt{3SAT}$ has a PCP verifier $\mathbb{V}$ that makes q queries using $c\log n$ random bits.
2. Given input x with $|x| = n$ and random string $r \in \{0,1\}^{c\log n}$, $\mathbb{V}(x, r)$ is a Boolean function of type $\{0,1\}^q \to \{0,1\}$.
3. $\varphi = \{\mathbb{V}(x, r)\}_{r\in\{0,1\}^{c\log n}}$ is a P-size $q\mathtt{CSP}$ instance.
   - By completeness, $x \in \mathtt{3SAT} \Rightarrow \mathtt{val}(\varphi) = 1$.
   - By soundness, $x \notin \mathtt{3SAT} \Rightarrow \mathtt{val}(\varphi) \leq \rho \stackrel{\text{def}}{=} 1/2$.
4. The map from $\mathtt{3SAT}$ to $\frac{1}{2}$-$\mathtt{GAP}q\mathtt{CSP}$ is P-time computable.
   - $\mathbb{V}$ runs in P-time.

Equivalence Proof

PCP Theorem $\Leftarrow$ **PCP Theorem**.

Suppose $L \in \mathbf{NP}$ and ρ-$\mathtt{GAP}q\mathtt{CSP}$ is NP-hard for some $q \in \mathbf{N}$, $\rho < 1$. By assumption there is some P-time reduction $f : L \to \rho$-$\mathtt{GAP}q\mathtt{CSP}$.

1. The verifier for L works as follows:
   - On input x, compute the $q\mathtt{CSP}$ instance $f(x) = \{\varphi_i\}_{i \in [m]}$.
   - A PCP proof π is an assignment to the variables. The verifier randomly chooses $i \in [m]$ and checks if φ_i is satisfied by reading the relevant q bits of the proof.
2. If $x \in L$, the verifier always accepts; otherwise it accepts with probability $< \rho$.

Equivalence Proof

PCP Theorem $\Rightarrow$ **PCP Theorem**.

This is very much like the equivalence between SAT and 3SAT.

1. Let $\epsilon > 0$ and $q \in \mathbf{N}$ be such that $(1-\epsilon)$-GAPqCSP is NP-hard.
2. Let $\varphi = \{\varphi_i\}_{i=1}^{m}$ be a qCSP instance with n variables.
3. Each φ_i is the conjunction of at most 2^q clauses, each being the disjunction of at most q literals.
4. If all assignments fail at least an ϵ fragment of the constraints of φ, then all assignments fail at least a $\frac{\epsilon}{2^q}$ fragment of the clauses of the SAT instance.
5. Consequently all assignments fail at least a $\frac{\epsilon}{q2^q}$ fragment of the clauses of the 3SAT instance.

Proof View	Inapproximability View
PCP verifier $\mathbb{V}$	$\mathtt{CSP}$ instance φ
PCP proof π	assignment to variables $\mathbf{u}$
proof length $\lvert\pi\rvert$	number of variables n
number of queries q	arity of constraints q
number of random bits r	logarithm of number of constraints $\log m$
soundness parameter ϵ	maximum fraction ρ of the violated constraints of no instances
$\mathbf{NP} \subseteq \mathbf{PCP}(\log, 1)$	ρ-$\mathtt{GAP}q\mathtt{CSP}$ is NP-hard

The equivalence of the proof view and the inapproximability view is essentially due to the Cook-Levin Theorem for PTM.

Inapproximability

Min-VC and Max-IS are inherently different from the perspective of approximation.

- $\texttt{Min-VC} + \texttt{Max-IS} = n$.
- ρ-approximation algorithm of Max-IS $\Rightarrow$ $\frac{n-IS}{n-\rho IS}$-approximation algorithm of Min-VC.

Lemma. There is a P-time computable function f that maps an m clause 3CNF φ to a $7m$-vertex graph $f(\varphi)$ whose independent set is of size $\mathtt{val}(\varphi)m$.

The standard Karp reduction from `3SAT` to `Max-IS` is as follows:

- Each clause is translated to a clique of 7 nodes, each node represents a (partial) assignment that validates the clause.
- Two nodes from two different cliques are connected if and only if they conflict.

A formula φ of m clauses is translated to a graph with $7m$ nodes, and an assignment satisfying l clauses of φ if and only if the graph has an independent set of size l.

Theorem. The following statements are valid.

1. $\exists\rho' \in (0,1)$. ρ'-approximation to `Min-VC` is NP-hard, and
2. $\forall\rho \in (0,1)$. ρ-approximation to `Max-IS` is NP-hard.

[$\exists\rho$. ρ-approximation to `Max-IS` is NP-hard.]* By PCP Theorem, ρ-approximation to `Max-3SAT` is NP-hard for some ρ. So by Lemma ρ-approximation to `Max-IS` is NP-hard.

1. Referring to the map of Lemma, the minimum vertex cover has size $7m - \texttt{val}(\varphi)m$. Let $\rho' = \frac{6}{7-\rho}$. Suppose `Min-VC` had a ρ'-approximation algorithm.

- If $\texttt{val}(\varphi) = 1$, it would produce a vertex cover of size $\leq \frac{1}{\rho'}(6m) = (7-\rho)m$.
- If $\texttt{val}(\varphi) < \rho$, the minimum vertex cover has size $> (7-\rho)m$.
  The ρ'-approximation algorithm must return a vertex cover of size $> (7-\rho)m$.

The first proposition is established. The second will be proved by making use of [_]*.

2. We prove that if `Max-IS` were $\frac{\rho}{2}$-approximate, it would be ρ-approximate as well.

   1. Let G be the input graph. Let K, k be such that $\frac{\rho}{2}\binom{K}{k} > \binom{\rho K}{k}$. Consider G^k:
      - The vertices are k-size subsets of V_G;
      - Two vertices S_1, S_2 are disconnected if $S_1 \cup S_2$ is an independent set of G.
   2. Apply the $\frac{\rho}{2}$-approximation algorithm $\mathbb{I}$ to G^k, and derive an independent set of G from the output of $\mathbb{I}$.

- Suppose the size of the largest independent set of G is at least K. Then the size of the largest independent set of G^k is $\geq \binom{K}{k}$.
- The output of $\mathbb{I}$ is an independent set of size $\geq \frac{\rho}{2}\binom{K}{k} > \binom{\rho K}{k}$.
- An independent set of size $> \rho K$ can be derived from the output. A contradiction.

Given an input graph, apply brutal force to see if there is an independent set of size at most K. If the answer is yes, we are done. Otherwise use the above self-reduction.

Fourier Transform Technique

A Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is a linear function if

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y}),$$

where "$+$" is the addition operation in $\mathbf{F}_2$. Linear Functions can be seen as vectors.

$f(\mathbf{x}) = x_1 f(\mathbf{e}_1) + \ldots + x_n f(\mathbf{e}_n)$.

Fourier Transform over $\mathbf{F}_2^n$

Boolean functions have been studied using Fourier transform over $\mathbf{F}_2^n$.

We shall use $\{+1,-1\}$ instead of $\{0,1\}$ whenever it is technically convenient.

- $0 \leftrightarrow (-1)^0 = 1$ and $1 \leftrightarrow (-1)^1 = -1$.
- $\{0,1\}^n$ is turned into $\{\pm 1\}^n$.
- "addition in $\mathbf{F}_2$" is turned into "integer multiplication".

The notation $\mathbf{y}\cdot\mathbf{z}$ stands for $\langle y_1 z_1, \ldots, y_n z_n \rangle$, where $\mathbf{y}, \mathbf{z} \in \mathbf{R}^{\{\pm 1\}^n}$.

Fourier Transform over $\mathbf{F}_2^n$

The 2^n-dimensional Hilbert space $\mathbf{R}^{\{\pm1\}^n}$ is defined as follows: For $f, g \in \mathbf{R}^{\{\pm1\}^n}$,

1. $(f+g)(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$,
2. $(cf)(\mathbf{x}) = cf(\mathbf{x})$, and
3. expectation inner product: $\langle f, g\rangle = \mathrm{E}_{\mathbf{x}\in\{\pm1\}^n}[f(\mathbf{x})\cdot g(\mathbf{x})]$.

Standard orthogonal basis: $\{\mathbf{e}_\mathbf{x}\}_{\mathbf{x}\in\{\pm1\}^n}$.

Fourier Transform over $\mathbf{F}_2^n$

Fourier Basis: $\{\chi_\alpha\}_{\alpha\subseteq[n]}$, where $\chi_\alpha(\mathbf{x}) = \prod_{i\in\alpha} x_i$.

1. $\chi_\emptyset = \mathbf{1}$.
2. Fourier basis functions are linear functions. The converse is also true.

Fourier basis is orthonormal.

- $\langle\chi_\alpha,\chi_\alpha\rangle = \mathrm{E}_{\mathbf{x}\in\{\pm1\}^n}[\chi_\alpha(\mathbf{x})\chi_\alpha(\mathbf{x})] = \mathrm{E}_{\mathbf{x}\in\{\pm1\}^n}[\chi_\alpha(\mathbf{x}\cdot\mathbf{x})] = \mathrm{E}_{\mathbf{x}\in\{\pm1\}^n}[1] = 1$.
- $\langle\chi_\alpha,\chi_\beta\rangle = \mathrm{E}_{\mathbf{x}\in\{\pm1\}^n}[\chi_\alpha(\mathbf{x})\chi_\beta(\mathbf{x})] = 0$ if $\alpha \neq \beta$.

Random Subsum Principle.

- If $\mathbf{u} \neq \mathbf{v}$ then for exactly half the choices of $\mathbf{x}$, $\mathbf{u}\odot\mathbf{x} \neq \mathbf{v}\odot\mathbf{x}$.

Fourier Transform over $\mathbf{F}_2^n$

$f = \sum_{\alpha\subseteq[n]} \widehat{f}_\alpha \chi_\alpha$ for every $f \in \mathbf{R}^{\{\pm1\}^n}$, where $\widehat{f}_\alpha$ is the αth Fourier coefficient of f.

Lemma. (i) $\langle f, g\rangle = \sum_{\alpha\subseteq[n]} \widehat{f}_\alpha \widehat{g}_\alpha$. (ii) (Parseval's Identity) $\langle f, f\rangle = \sum_{\alpha\subseteq[n]} \widehat{f}_\alpha^2$.

Proof.

$\langle f, g\rangle = \langle \sum_{\alpha\subseteq[n]} \widehat{f}_\alpha \chi_\alpha, \sum_{\beta\subseteq[n]} \widehat{g}_\beta \chi_\beta\rangle = \sum_{\alpha,\beta\subseteq[n]} \widehat{f}_\alpha \widehat{g}_\beta \langle \chi_\alpha, \chi_\beta\rangle = \sum_{\alpha\subseteq[n]} \widehat{f}_\alpha \widehat{g}_\alpha$. □

Fourier Transform over $\mathbf{F}_2^n$

Example.

1. Majority function of 3 variables $= \frac{1}{2}u_1 + \frac{1}{2}u_2 + \frac{1}{2}u_3 - \frac{1}{2}u_1u_2u_3$.
2. Projection function $\lambda x_1 \ldots x_n.x_i$. Here $\widehat{f}_\alpha$ is 1 if $\alpha = \{i\}$ and is 0 if $\alpha \neq \{i\}$.

Fourier Transform over $\mathbf{F}_2^n$

Theorem. Suppose $f\colon \{\pm 1\}^n \to \{\pm 1\}$ satisfies $\Pr_{\mathbf{x},\mathbf{y}}[f(\mathbf{x}\cdot\mathbf{y}) = f(\mathbf{x})f(\mathbf{y})] = \frac{1}{2}+\epsilon$. Then there is some $\alpha \subseteq [n]$ such that $\widehat{f}_\alpha \geq 2\epsilon$.

The assumption is equivalent to $\mathrm{E}_{\mathbf{x},\mathbf{y}}[f(\mathbf{x}\cdot\mathbf{y})f(\mathbf{x})f(\mathbf{y})] = \frac{1}{2}+\epsilon-(\frac{1}{2}-\epsilon) = 2\epsilon$. Now

$$\begin{aligned}
2\epsilon = \mathrm{E}_{\mathbf{x},\mathbf{y}}[f(\mathbf{x}\cdot\mathbf{y})f(\mathbf{x})f(\mathbf{y})] &= \mathrm{E}_{\mathbf{x},\mathbf{y}}[(\sum_{\alpha}\widehat{f}_\alpha\chi_\alpha(\mathbf{x}\cdot\mathbf{y}))(\sum_{\beta}\widehat{f}_\beta\chi_\beta(\mathbf{x}))(\sum_{\gamma}\widehat{f}_\gamma\chi_\gamma(\mathbf{y}))] \\
&= \mathrm{E}_{\mathbf{x},\mathbf{y}}[\sum_{\alpha,\beta\ \gamma}\widehat{f}_\alpha\widehat{f}_\beta\widehat{f}_\gamma\chi_\alpha(\mathbf{x})\chi_\alpha(\mathbf{y})\chi_\beta(\mathbf{x})\chi_\gamma(\mathbf{y})] \\
&= \sum_{\alpha,\beta,\gamma}\widehat{f}_\alpha\widehat{f}_\beta\widehat{f}_\gamma\mathrm{E}_{\mathbf{x},\mathbf{y}}[\chi_\alpha(\mathbf{x})\chi_\alpha(\mathbf{y})\chi_\beta(\mathbf{x})\chi_\gamma(\mathbf{y})] \\
&= \sum_{\alpha,\beta,\gamma}\widehat{f}_\alpha\widehat{f}_\beta\widehat{f}_\gamma\mathrm{E}_{\mathbf{x}}[\chi_\alpha(\mathbf{x})\chi_\beta(\mathbf{x})]\mathrm{E}_{\mathbf{y}}[\chi_\alpha(\mathbf{y})\chi_\gamma(\mathbf{y})] \\
&= \sum_{\alpha}\widehat{f}^3_\alpha \leq (\max_\alpha \widehat{f}_\alpha)\sum_{\alpha}\widehat{f}^2_\alpha = \max_\alpha \widehat{f}_\alpha\langle f,f\rangle = \max_\alpha \widehat{f}_\alpha.
\end{aligned}$$

The last equality is due to the fact that f is Boolean.

Fourier Transform over $\mathbf{F}_2^n$

Suppose $f\colon \{\pm 1\}^n \to \{\pm 1\}$ satisfies

$$\mathrm{Pr}_{\mathbf{x},\mathbf{y}\in\{\pm 1\}^n}\left[f(\mathbf{x}\cdot\mathbf{y}) = f(\mathbf{x})f(\mathbf{y})\right] \ = \ \frac{1}{2} + \epsilon.$$

By the previous theorem $\widehat{f}_\alpha \geq 2\epsilon$ for some Fourier coefficient $\widehat{f}_\alpha$ of f. Thus

$$\langle f, \chi_\alpha \rangle \geq 2\epsilon.$$

In other words f coincides with the basis function χ_α on $\geq \frac{1}{2} + \epsilon$ fraction of inputs.

Random Test of Linearity

The basis functions are precisely the linear functions.

$\chi_\alpha(\mathbf{x}\cdot\mathbf{y}) = \chi_\alpha(\mathbf{x})\chi_\alpha(\mathbf{y})$.

Suppose $f\colon \{0,1\}^n \to \{0,1\}$ satisfies

$$\Pr_{\mathbf{x},\mathbf{y}\in\{0,1\}^n}[f(\mathbf{x}+\mathbf{y}) \text{ is equal to } f(\mathbf{x})+f(\mathbf{y})] = \frac{1}{2}+\epsilon.$$

Then f coincides with a linear function on at least $\frac{1}{2}+\epsilon$ fraction of inputs.

Random Test of Linearity

Let $\rho \in [0,1]$. The functions $f, g : \{0,1\}^n \rightarrow \{0,1\}$ are ρ-close if

$$\Pr_{\mathbf{x} \in_{\mathrm{R}} \{0,1\}^n}[f(\mathbf{x}) = g(\mathbf{x})] \geq \rho.$$

Theorem. Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be such that for some $\rho > \frac{1}{2}$,

$$\Pr_{\mathbf{x},\mathbf{y} \in_{\mathrm{R}} \{0,1\}^n}[f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})] \geq \rho.$$

Then f is ρ-close to a linear function.

Local Testing of Linear Functions

A local test of f checks if f is a linear function by making a constant number of queries.

- It accepts every linear function, and
- it rejects every function that is far from being linear with high probability.

For $\delta \in (0, 1/2)$ a $(1-\delta)$-linearity test rejects with probability $> \frac{1}{2}$ any function not $(1-\delta)$-close to a linear function by testing

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$

randomly for $\frac{1}{\delta}$ times.

The acceptance probability is $\leq (1-\delta)^{\frac{1}{\delta}} \approx \frac{1}{e} < \frac{1}{2}$.

If the test accepts with probability greater than $1/2$, then the function is $(1-\delta)$-close to a linear function.

Local Decoding

Suppose $\delta < \frac{1}{4}$ and f is $(1-\delta)$-close to some linear function $\widetilde{f}$.

Given $\mathbf{x}$ one can learn $\widetilde{f}(\mathbf{x})$ by making only two queries to f.

1. Choose $\mathbf{x}' \in_{\mathrm{R}} \{0,1\}^n$;
2. Set $\mathbf{x}'' = \mathbf{x} + \mathbf{x}'$;
3. Output $f(\mathbf{x}') + f(\mathbf{x}'')$.

By union bound $\widetilde{f}(\mathbf{x}) = \widetilde{f}(\mathbf{x}') + \widetilde{f}(\mathbf{x}'') = f(\mathbf{x}') + f(\mathbf{x}'')$ holds with probability $\geq 1 - 2\delta$.

Efficient Conversion of NP Certificate to PCP Proof

Proofs of PCP Theorems involve some interesting ways of encoding NP-certificates and the associated methods of checking if a string is a valid encoding.

One idea is to amplify any error that appears in an NP-certificate.

We shall showcase how it works by looking at a problem to which the amplification power of Walsh-Hadamard Code can be exploited.

Theorem. $\mathbf{NP} \subseteq \mathbf{PCP}(\mathrm{poly}(n), 1)$.

Walsh-Hadamard Code

The Walsh-Hadamard function $\mathtt{WH} : \{0,1\}^n \to \{0,1\}^{2^n}$ encodes a string of length n by a function in n variables over $\mathbf{F}_2$:

$$\mathtt{WH}(\mathbf{u}) : \ \mathbf{x} \ \mapsto \ \mathbf{u} \odot \mathbf{x},$$

where $\mathbf{u} \odot \mathbf{x} = \sum_{i=1}^{n} u_i x_i \pmod 2$.

Walsh-Hadamard code is an error correcting code with distance $1/2$.

Walsh-Hadamard Codeword

We say that f is a Walsh-Hadamard codeword if $f = \mathtt{WH}(\mathbf{u})$ for some $\mathbf{u} \in \{0,1\}^n$.

Walsh-Hadamard codewords are precisely the linear functions.

Proof.

A linear function f is the same as $\mathtt{WH}(\mathbf{f})$, where

$$\mathbf{f} = \begin{pmatrix} f(\mathbf{e}_1) \\ f(\mathbf{e}_2) \\ \vdots \\ f(\mathbf{e}_n) \end{pmatrix}.$$

□

Quadratic Equation in $\mathbf{F}_2$

Suppose $\mathbf{A}$ is an $m \times n^2$ matrix and $\mathbf{b}$ is an m-dimensional vector with values in $\mathbf{F}_2$. Let $(\mathbf{A}, \mathbf{b}) \in \texttt{QUADEQ}$ if there is an n-dimensional vector $\mathbf{u}$ such that

$$\mathbf{A}(\mathbf{u} \otimes \mathbf{u}) = \mathbf{b},$$

where $\mathbf{u} \otimes \mathbf{u}$ is the tensor product of $\mathbf{u}$.

$$\mathbf{u} \otimes \mathbf{u} = (u_1 u_1, \ldots, u_1 u_n, \ldots, u_n u_1, \ldots, u_n u_n)^{\dagger}.$$

Quadratic Equation in $\mathbf{F}_2$

An instance of QUADEQ over u_1, u_2, u_3, u_4, u_5:

$$\begin{array}{rcl} u_1 u_2 + u_3 u_4 + u_1 u_5 & = & 1 \\ u_1 u_1 + u_2 u_3 + u_1 u_4 & = & 0 \end{array}$$

A satisfying assignment is $(0, 0, 1, 1, 0)$.

$\texttt{QUADEQ}$ is NP-Complete

$\texttt{CKT-SAT} \leq_K \texttt{QUADEQ}$.

- All wires are turned into variables.
- Boolean equality $x \vee y = z$ relating the inputs to the output is turned into algebraic equality $(1-x)(1-y) = 1-z$ in $\mathbf{F}_2$, which is equivalent to $xx + yy + xy + zz = 0$.
- $\neg x = z$ is turned into $xx + zz = 1$.
- $x \wedge y = z$ is turned into $xy + zz = 0$.

From NP Certificate to PCP Proof

A certificate for $(\mathbf{A}, \mathbf{b})$ is an n-dimensional vector $\mathbf{u}$ witnessing $(\mathbf{A}, \mathbf{b}) \in \mathtt{QUADEQ}$.

- To check if $\mathbf{u}$ is a solution, one reads the n bits of $\mathbf{u}$ and checks the m equations.

We convert an NP-certificate $\mathbf{u}$ to the PCP-proof $\mathtt{WH}(\mathbf{u})\mathtt{WH}(\mathbf{u} \otimes \mathbf{u})$.

- The proof is a string of length $2^n + 2^{n^2}$.
- Using the proof it is straightforward to verify probabilistically if $(\mathbf{A}, \mathbf{b}) \in \mathtt{QUADEQ}$.

Verifier for QUADEQ

Step 1. Verify that f, g are linear functions.

1. Perform a 0.999-linearity test on f, g.

If successful we may assume that $f(\mathbf{r}) = \mathbf{u} \odot \mathbf{r}$ and $g(\mathbf{z}) = \mathbf{w} \odot \mathbf{z}$.

The test makes a constant number of queries to f, g.

Verifier for QUADEQ

Step 2. Verify that g encodes $(\mathbf{u} \otimes \mathbf{u}) \odot \_$.

1. Get independent random $\mathbf{r}, \mathbf{r}'$.
2. Reject if $f(\mathbf{r})f(\mathbf{r}') \neq g(\mathbf{r} \otimes \mathbf{r}')$.
3. Repeat the test 10 times.

- In a correct proof $f(\mathbf{r})f(\mathbf{r}') = (\sum_i u_i r_i)(\sum_j u_j r'_j) = \sum_{i,j} u_i u_j r_i r'_j = g(\mathbf{r} \otimes \mathbf{r}')$.
- Assume $\mathbf{w} \neq \mathbf{u} \otimes \mathbf{u}$. Let matrices W and U be $\mathbf{w}$ and respectively $\mathbf{u} \otimes \mathbf{u}$. One has
  - $g(\mathbf{r} \otimes \mathbf{r}') = \mathbf{w} \odot (\mathbf{r} \otimes \mathbf{r}') = \sum_{i,j} w_{ij} r_i r'_j = \mathbf{r}W\mathbf{r}'$, and
  - $f(\mathbf{r})f(\mathbf{r}') = (\mathbf{u} \otimes \mathbf{r})(\mathbf{u} \otimes \mathbf{r}') = (\sum_i u_i r_i)(\sum_j u_j r'_j) = \mathbf{r}U\mathbf{r}'$.

  $\mathbf{r}W$, $\mathbf{r}U$ differ for at least $\frac{1}{2}$ of $\mathbf{r}$'s; and $\mathbf{r}W\mathbf{r}'$, $\mathbf{r}U\mathbf{r}'$ differ for at least $\frac{1}{4}$ of $(\mathbf{r}, \mathbf{r}')$'s.
- The overall probability of rejection is at least $1 - (\frac{3}{4})^{10} > 0.9$.

The verification makes a constant number of queries to f, g.

Verifier for QUADEQ

Step 3. Verify that g encodes a solution.

1. Take a random subset S of $[m]$.
2. Reject if $g(\sum_{k\in S} A_{k,\_}) \neq \sum_{k\in S} b_k$.

- There is enough time to check $\mathbf{A}(\mathbf{u}\otimes\mathbf{u}) = \mathbf{b}$.
- However since m is part of the input, the number of queries, which must be a constant, should not depend on m.
- If $\{k\in[m] \mid g(A_{k,\_}) \neq b_k\} \neq \emptyset$, then $\Pr_{S\subseteq_R[m]}[|S\cap\{k\in[m] \mid g(A_{k,\_})\neq b_k\}| \text{ is odd}] = \frac{1}{2}$.

  Note that $g(\sum_{k\in S} A_{k,\_}) = \sum_{k\in S} g(A_{k,\_})$ by linearity.

A constant number of queries are made.

Suppose the PCP verifier for QUADEQ makes a total of q_0 queries.

It follows from the completeness of QUADEQ that all NP problems have PCP verifiers that toss coins for a polynomial number of time and make precisely q_0 queries.

Proof of PCP Theorem

CSP with Nonbinary Alphabet

$q\mathtt{CSP}_W$ is analogous to $q\mathtt{CSP}$ except that the alphabet is $[W]$ instead of $\{0,1\}$.

The constraints are functions of type $[W]^q \to \{0,1\}$.

For $\rho \in (0,1)$, we define the promise problem ρ-$\mathtt{GAP}q\mathtt{CSP}_W$ analogous to ρ-$\mathtt{GAP}q\mathtt{CSP}$.

$\texttt{3COL}$ is a case of $\texttt{2CSP}_3$.

PCP Theorem states that ρ-$\texttt{GAP}q\texttt{CSP}$ is NP-hard for some q, ρ.

The proof we shall describe is based on the following observation:

1. If φ of m constraints is unsatisfied, then $\texttt{val}(\varphi) \leq 1 - \frac{1}{m}$.
2. There is a construction that increases the gap.

The idea is to start with an NP-problem, then apply Step 2 for $\log(m)$ times.

Let f be a function mapping CSP instances to CSP instances.

It is a complete linear-blowup reduction (CL-reduction) if it is P-time computable and the following are valid for every CSP instance φ.

1. Completeness. If φ is satisfiable then $f(\varphi)$ is satisfiable.
2. Linear Blowup. If φ has n variables, $f(\varphi)$ has no more than Cn variables. If φ has m constraints, $f(\varphi)$ has no more than Cm constraints.
    - C, W depend only on the problem parameter q, hence linearity.
    - C, W do not depend on anything of the problem instance $\varphi(n, m)$.

We will define two CL-reductions that will be repeated $\log(m)$ times.

Main Lemma. There exist constants $q_0 \geq 3$, $\epsilon_0 > 0$ and CL-reduction f such that for every q_0CSP instance φ and every $\epsilon < \epsilon_0$, $f(\varphi)$ is a q_0CSP instance satisfying

$$\texttt{val}(\varphi) \leq 1 - \epsilon \Rightarrow \texttt{val}(f(\varphi)) \leq 1 - 2\epsilon.$$

q_0CSP Instance	Arity	Alphabet	Constraint	Gap
φ	q_0	binary	m	$1-\epsilon$
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
$f(\varphi)$	q_0	binary	Cm	$1-2\epsilon$

The q_0 is the number of queries of the PCP verifier for QUADEQ.

In the following proof, ϵ_0, ℓ, W, W', t are functions of q_0, and d is an absolute value.

Proof of PCP Theorem

Let $q_0 \geq 3$ and $\epsilon_0 > 0$ be given by the Main Lemma. A CL-reduction from q_0CSP to $(1{-}2\epsilon_0)$-GAPq_0CSP is obtained as follows:

1. q_0CSP is NP-hard.
2. For a q_0CSP instance φ with m constraints, apply Main Lemma for $\log m$ times to amplify the gap. We get an instance ψ.
3. If φ is satisfiable, then ψ is satisfiable. Otherwise according to Main Lemma

$$\mathtt{val}(\psi) \leq 1 - 2^{\max\{\log(m),\log(2\epsilon_0)\}} \cdot \frac{1}{m} \leq 1 - 2\epsilon_0.$$

4. $|\psi| \leq C^{\log(m)} m = \mathtt{poly}(|\varphi|)$. Conclude that $(1{-}2\epsilon_0)$-$\mathrm{GAP}q_0\mathtt{CSP}$ is NP-hard.

C depends on two constants, q_0 and 2 (the size of alphabet).

Main Lemma is proved in three steps.

1. Prove that every $q\mathtt{CSP}$ instance can be turned into a "nice" $q\mathtt{CSP}_W$ instance.
2. Gap Amplification. Construct a CL-reduction f that increases both the gap and the alphabet size of a "nice" $q\mathtt{CSP}$ instance. [Dinur's proof]
3. Alphabet Reduction. Construct a CL-reduction g that decreases the alphabet size to 2 with a modest reduction in the gap. [Proof of Arora et al.]

---

1. 1st Proof: Algebraic (first part) + Algebraic (second part)
2. 2nd Proof: Combinatorial (first part) + Algebraic (second part)

Gap Amplification. For all numbers ℓ, q, there are number W, $\epsilon_0 \in (0,1)$ and a CL-reduction $g_{\ell,q}$ such that for every $q\mathrm{CSP}$ instance φ, $\psi = g_{\ell,q}(\varphi)$ is a $2\mathrm{CSP}_W$ instance that satisfies the following for all $\epsilon < \epsilon_0$.

$$\mathrm{val}(\varphi) \leq 1 - \epsilon \Rightarrow \mathrm{val}(\psi) \leq 1 - \ell\epsilon.$$

Alphabet Reduction. There exist a constant q_0 and a CL-reduction h such that for every $2\mathrm{CSP}_W$ instance φ, $\psi = h(\varphi)$ is a $q_0\mathrm{CSP}$ instance satisfying

$$\mathrm{val}(\varphi) \leq 1 - \epsilon \Rightarrow \mathrm{val}(\psi) \leq 1 - \epsilon/3.$$

CSP Instance	Arity	Alphabet	Constraint	Gap
φ	q_0	binary	m	$1-\epsilon$
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
$f(\varphi)$	2	nonbinary	$C'm$	$1-6\epsilon$
$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
$g(f(\varphi))$	q_0	binary	$C''C'm$	$1-2\epsilon$

Dinur makes use of expander graphs to construct new constraints.

Let φ be a $2\mathtt{CSP}_W$ instance with n variables. The constraint graph G_φ of φ is defined as follows:

1. the vertex set is $[n]$, and
2. (i, j) is an edge if there is a constraint on the variables u_i, u_j. Parallel edges and self-loops are admitted.

A $2\mathtt{CSP}_W$ instance φ is nice if the followings are valid:

1. There is a constant d such that G_φ is a $(d, 0.9)$-expander.
2. At every vertex half of the adjacent edges are self loops.

A nice CSP instance looks like an expander. In a nice CSP a $t+1$ step random walk is very much like a t step random walk.

Lemma. Let G be a d-regular n-vertex graph, S be a vertex subset and $T = \overline{S}$. Then

$$|E(S,T)| \geq (1-\lambda_G)\frac{d|S||T|}{|S|+|T|}. \tag{1}$$

The vector $\mathbf{x}$ defined below satisfies $\|\mathbf{x}\|_2^2 = |S||T|(|S|+|T|)$ and $\mathbf{x}\perp\mathbf{1}$.

$$\mathbf{x}_i = \begin{cases} +|T|, & i \in S, \\ -|S|, & i \in T. \end{cases}$$

Let $Z = \sum_{i,j} A_{i,j}(\mathbf{x}_i - \mathbf{x}_j)^2$. By definition $Z = \frac{2}{d}|E(S,T)|(|S|+|T|)^2$. On the other hand

$$Z = \sum_{i,j} A_{i,j}\mathbf{x}_i^2 - 2\sum_{i,j} A_{i,j}\mathbf{x}_i\mathbf{x}_j + \sum_{i,j} A_{i,j}\mathbf{x}_j^2 = 2\|\mathbf{x}\|_2^2 - 2\langle \mathbf{x}, A\mathbf{x}\rangle.$$

Since $\mathbf{x}\perp\mathbf{1}$, $\langle \mathbf{x}, A\mathbf{x}\rangle \leq \lambda_G\|\mathbf{x}\|_2^2$ (cf. Rayleigh quotient).

Let $G=(V,E)$ be an expander and $S\subseteq V$ with $|S|\le|V|/2$. The following holds.

$$\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in S]\le\frac{|S|}{|V|}\left(\frac{1}{2}+\frac{\lambda_G}{2}\right). \tag{2}$$

Observe that $|S|/|V|=\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in S]+\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in\overline{S}]$. And by (1), one has

$$\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in\overline{S}]=E(S,\overline{S})/d|V|\ge\frac{|S|}{|V|}\cdot\frac{1}{2}\cdot(1-\lambda_G).$$

We are done by substituting $|S|/|V|-\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in S]$ for $\mathrm{Pr}_{(u,v)\in E}[u\in S,v\in\overline{S}]$.

$$\mathrm{Pr}_{(u,v)\in E^\ell}[u\in S,v\in S]\le\frac{|S|}{|V|}\left(\frac{1}{2}+\frac{\lambda_G^\ell}{2}\right). \tag{3}$$

Step 1: Reduction to Nice Instance

The reduction consists of three steps.

$$\begin{aligned} q_0\texttt{CSP}\ \text{instance} &\overset{\text{Step1.1}}{\Longrightarrow} 2\texttt{CSP}_{2^{q_0}}\ \text{instance} \\ &\overset{\text{Step1.2}}{\Longrightarrow} 2\texttt{CSP}_{2^{q_0}}\ \text{instance with regular constraint graph} \\ &\overset{\text{Step1.3}}{\Longrightarrow} \text{nice } 2\texttt{CSP}_{2^{q_0}}\ \text{instance.} \end{aligned}$$

In all the three steps the fraction of violated constraints decreases by a constant factor.

Step 1: Reduction to Nice Instance

Step 1.1. There exists a CL-reduction that maps a q_0CSP instance φ to a $2\mathrm{CSP}_{2^{q_0}}$ instance ψ such that

$$\mathtt{val}(\varphi) \le 1 - \epsilon \Rightarrow \mathtt{val}(\psi) \le 1 - \frac{\epsilon}{q_0}.$$

Suppose φ has variables $x_1, \ldots, x_n$ and m constraints.

- The new instance ψ has variables $x_1, \ldots, x_n, y_1, \ldots, y_m$, where y_i takes value in $\{0,1\}^{q_0}$. A value in $\{0,1\}^{q_0}$ codes up an assignment to the variables in φ_i.
- For each variable x_j in φ_i, construct the constraint $\psi_{i,j}$ stating that y_i satisfies φ_i and y_i is consistent with x_j.

- An assignment that satisfies ψ is the same thing as an assignment that satisfies φ.
- A constraint splits into q_0 constraints.

Step 1: Reduction to Nice Instance

Step 1.2. There exist an absolute constant d' and a CL-reduction that maps a $2\mathtt{CSP}_W$ instance φ to a $2\mathtt{CSP}_W$ instance ψ such that

$$\mathtt{val}(\varphi) \leq 1 - \epsilon \Rightarrow \mathtt{val}(\psi) \leq 1 - \frac{\epsilon}{100Wd'}$$

and that G_ψ is d'-regular.

$W = 2^{q_0}$ by the construction of Step 1.1.

Let $\{G_k\}_k$ be an explicit $(d'-1, 0.9)$-expander. We get ψ by replacing each k-degree node of G_φ by G_k and adding the identity constraint (of the form $y_i^j = y_i^{j'}$, where $i \in [n]$) to each edge (j, j') of G_k. If φ has m constraints, ψ has $d'm$ constraints.

Suppose $\mathtt{val}(\varphi) \leq 1 - \epsilon$ and $\mathbf{v}$ is an unsatisfying assignment to ψ.

It suffices to prove that $\mathbf{v}$ violates at least $\frac{\epsilon m}{100W}$ constraints of ψ.

Continue on the next slide.

Fact. For every $c \in (0, 1)$ there is a constant d and an algorithm that, given input n, runs in $\mathtt{poly}(n)$ time and outputs an (n, d, c)-expander.

Step 1: Reduction to Nice Instance

Let $\mathbf{u}$ be the assignment to φ that is defined by the plurality of the assignment $\mathbf{v}$ to ψ.

Let t_i be the number of v_i^j's, where $j \in [k]$, that disagree with u_i. Clearly $t_i \leq k(1-\frac{1}{W})$.

If $\sum_{i=1}^n t_i$ is large, then on average G_k already contains enough violated constraints.

1. $\sum_{i=1}^n t_i \geq \frac{1}{4}\epsilon m$. Let $S_i = \{y_i^j \mid v_i^j = u_i\}$ and let $\overline{S_i} = \{y_i^1, \ldots, y_i^k\} \setminus S_i$. The number of constraints of G_k violated by $\mathbf{v}$ is at least

   $$E(S_i, \overline{S_i}) \overset{(1)}{\geq} (1-\lambda_{G_k})\frac{(d'-1)|S_i||\overline{S_i}|}{|S_i|+|\overline{S_i}|} = \frac{1}{10}\frac{d'-1}{k}|S_i||\overline{S_i}| \geq \frac{d'-1}{10k}\frac{k}{W}|\overline{S_i}| \geq \frac{d'-1}{10k}\frac{k}{W}t_i \geq \frac{1}{10W}t_i.$$

   Now $\sum_{i\in[n]} E(S_i, \overline{S_i}) \geq \frac{\epsilon m}{40W} = \frac{\epsilon}{40Wd'} \cdot d' m$.

2. $\sum_{i=1}^n t_i < \frac{1}{4}\epsilon m$. Since $\mathtt{val}(\varphi) \leq 1-\epsilon$, there is at least ϵm constraints violated in φ by $\mathbf{u}$. These ϵm constraints are also in ψ with variables being $\mathbf{v}$.

   Since every constraint has two variables, less than $\frac{1}{4}\epsilon m + \frac{1}{4}\epsilon m$ constraints have values in ψ different from those in φ. So at least $\frac{1}{2}\epsilon m$ constraints are violated.

Step 1: Reduction to Nice Instance

Step 1.3. There is an absolute constant d and a CL-reduction that maps a $2\mathtt{CSP}_W$ instance φ with G_φ being d'-regular for some $d' \leq d$ to a $2\mathtt{CSP}_W$ instance ψ such that

$$\mathtt{val}(\varphi) \leq 1 - \epsilon \Rightarrow \mathtt{val}(\psi) < 1 - \frac{\epsilon}{10d}$$

and that G_ψ is nice, $4d$-regular, and half of the edges adjacent to each vertex are self-loops.

There is a constant d and an explicit $(d, 0.1)$-expander $\{G_k\}_{k\in\mathbf{N}}$. We assume that φ contains n variables and that φ is d-regular (adding self-loops if $d' < d$).

Let $G_n^{2d\circlearrowleft}$ be G_n extended with $2d$ self-loops on each node, all constraints being tautological.

We get ψ from φ by overlapping the nodes of G_φ and the nodes of $G_n^{2d\circlearrowleft}$. Then

$$\lambda_{G_\psi} \leq \frac{1}{4}\cdot\lambda_{G_\varphi} + \frac{3}{4}\cdot\lambda_{G_n^{2d\circlearrowleft}} < 0.9.$$

Notice that "adding" decreases ϵ by a factor $\leq d$ and "overlapping" by a further factor ≤ 4.

A path of length t defines the conjunction of t constraints. If t is large enough, the chance for the conjunction being violated is significant.

For this simple idea to work, the following issues must be addressed:

1. The arity $2t$ is dependent of m and the number of constraints is exponential.
2. Even if $t = O(\log m)$, adding constraints to guarantee consistency of assignment would blow up the number of constraints non-linearly.

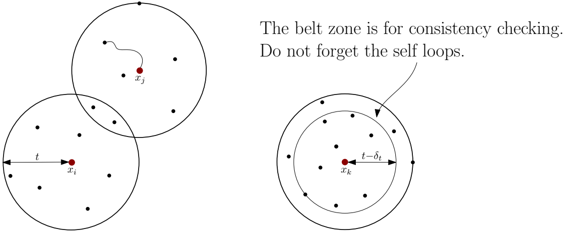
We look for a constant t.

Step 2: Gap Amplification, from ψ to ψ^t

Construction of the $2\mathrm{CSP}_{W'}$ instance ψ^t: Variables.

1. Let $x_1, \ldots, x_n$ denote the variables of ψ.
2. ψ^t contains n variables $\mathbf{y}_1, \ldots, \mathbf{y}_n \in W' < W^{d^{5t}}$, where $\mathbf{y}_i$ is an assignment to those of $x_1, \ldots, x_n$ reachable within t steps from x_i.

For $i, j \in [n]$ we say that an assignment to $\mathbf{y}_i$ claims a value for x_j.



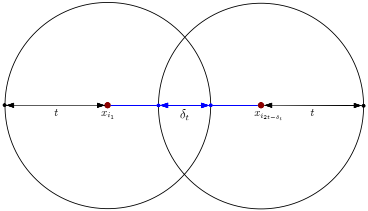
The belt zone is for consistency checking.
Do not forget the self loops.

Step 2: Gap Amplification, from ψ to ψ^t

Construction of the $2\mathtt{CSP}_{W'}$ instance ψ^t: Constraints.

Introduce in ψ^t a constraint $C_p = \bigwedge_{j\in[t-\delta_t,t]} C_p^j$ for each $2t-\delta_t$ step path $p = (x_{i_0}, \ldots, x_{i_{2t-\delta_t}})$ such that the following are valid.

1. $\delta_t = o(\sqrt{t})$.
2. For each $j \in [t-\delta_t, t]$, C_p^j is obtained from the constraint of the j-th edge by replacing $x_{i_j}, x_{i_{j+1}}$ respectively by $\mathbf{y}_{i_1}$'s claim for x_{i_j} and $\mathbf{y}_{i_{2t+1}}$'s claim for $x_{i_{j+1}}$.



Do not forget the self-loops.

Step 2: Gap Amplification

Lemma. An algorithm exists that given $\delta_t = o(\sqrt{t})$ and a nice $2\mathtt{CSP}_W$ instance ψ with n variables, $m = \frac{dn}{2}$ edges, d-degree G_ψ, produces a $2\mathtt{CSP}_{W'}$ instance ψ^t satisfying 1-4.

1. $W' < W^{d^{5t}}$ and ψ^t has $\frac{d^{2t-1}}{t} \cdot m$ constraints. take $\delta_t = \log(t)$.
2. If ψ is satisfiable then ψ^t is satisfiable.
3. The formula ψ^t is produced in $\mathrm{poly}(m, W^{d^{5t}})$ time. $W^{d^{5t}}$ is a constant.
4. If $\mathtt{val}(\psi) = 1 - \epsilon$ for $\epsilon < \frac{1}{d\delta_t}$, then $\mathtt{val}(\psi^t) \leq 1 - \ell\epsilon$ for $\ell = \frac{\delta_t}{44dW^4}$. $\frac{1}{\delta_t}$ is subtle.

Gap Amplification follows immediately from the lemma.

- If $\ell = 6$ and $W = 2^{q_0}$, we get a constant t and a constant $\epsilon_0 = \frac{1}{d\delta_t}$.
- In this case ψ^t is produced in $\mathtt{poly}$ time.

The conditions 1, 2 and 3 of the lemma are satisfied by the constraints of ψ^t.

Step 2: Gap Amplification, random generation of constraints

A random $2t-\delta_t$ step path $p=(x_{i_0},\ldots,x_{i_{2t-\delta_t}})$ is picked up with probability $\frac{2}{n\cdot d^{2t-\delta_t}}$.

Equivalently such a random path can be chosen in any of the following manners.

1. Pick up a random node, and take a $2t-\delta_t$ step random walk from the node.
2. Pick up a random node, and then take a j-step random walk from the node and a $(2t-\delta_t-j)$-step random walk from the node.
3. Pick up a random edge, and then take a j-step random walk from one node of the edge and a $(2t-\delta_t-j-1)$-step random walk from the other node of the edge.

---

a random constraint of ψ^t = a random path of length $2t-\delta_t$.

Step 2: Gap Amplification, plurality assignment

Fix an arbitrary assignment $\mathbf{v}_1, \ldots, \mathbf{v}_n$ to $\mathbf{y}_1, \ldots, \mathbf{y}_n$.

We would like to define an assignment to $x_1, \ldots, x_n$ from $\mathbf{v}_1, \ldots, \mathbf{v}_n$.

1. Let $Z_i \in [W]$ be a random variable defined by the following.
   - Starting from the vertex i, take a t step random walk in G_ψ to reach some vertex k; output $\mathbf{v}_k$'s claim for x_i.

   Let w_i denote the most likely value of Z_i.
2. We call $w_1, \ldots, w_n$ the plurality assignment to $x_1, \ldots, x_n$. Clearly $\Pr[Z_i{=}w_i] \geq \frac{1}{W}$.

There is a set F of $\epsilon m = \epsilon \frac{dn}{2}$ constraints in ψ violated by the plurality assignment.

Step 2: Gap Amplification

For $j \in [t-\delta_t, t]$ the edge $(x_{i_j}, x_{i_{j+1}})$ in $p = (x_{i_0}, \ldots, x_{i_{2t-\delta_t}})$ is truthful if $\mathbf{v}_{i_0}$ claims the plurality value for x_{i_j} and $\mathbf{v}_{i_{2t-\delta_t}}$ claims the plurality value for $x_{i_{j+1}}$.

If p has an edge that is both truthful and in F, the constraint C_p is violated.

Step 2: Gap Amplification

Suppose $p = (x_{i_0}, \ldots, x_{i_{2t-\delta_t}})$ is a random path, and $j \in [t-\delta_t, t]$.

Claim. $\Pr_p[(x_{i_j}, x_{i_{j+1}}) \text{ is truthful}] \geq \frac{1}{2W^2}$.

Claim. $\Pr_p[(x_{i_j}, x_{i_{j+1}}) \text{ is truthful and is in } F] \geq \frac{1}{2W^2}\epsilon$.

Proof.

We know that $\Pr[v_{i_0} \text{ claims the plurality value for } x_{i_t}] \geq \frac{1}{W}$. We can prove that

$$\Pr[v_{i_0} \text{ claims the plurality value for } x_{i_j}] \geq \frac{3}{4} \cdot \frac{1}{W}.$$

It follows that $\Pr_p[(x_{i_j}, x_{i_{j+1}}) \text{ is truthful}] \geq (\frac{3}{4} \cdot \frac{1}{W})(\frac{3}{4} \cdot \frac{1}{W}) \geq \frac{1}{2W^2}$. □

Let S_j be the random variable for the number of heads in a j-step random walk. Now

$$\begin{aligned}
\mathtt{Pr}[S_j = k] - \mathtt{Pr}[S_j = k+1] &= \frac{1}{2^j}\binom{j}{k} - \frac{1}{2^{j+1}}\binom{j+1}{k} \\
&= \frac{1}{2^j}\left[\binom{j}{k} - \frac{1}{2}\left[\binom{j}{k} + \binom{j}{k-1}\right]\right] \\
&= \frac{1}{2^{j+1}}\left[\binom{j}{k} - \binom{j}{k-1}\right].
\end{aligned}$$

It follows that

$$\begin{aligned}
\sum_{k=0}^{j+1}\left|\mathtt{Pr}[S_j = k] - \mathtt{Pr}[S_j = k+1]\right| &= \frac{1}{2^{j+1}}\sum_{k=0}^{j+1}\left|\binom{j}{k} - \binom{j}{k-1}\right| \\
&= \frac{1}{2^j}\sum_{k=0}^{(j+1)/2}\left(\binom{j}{k} - \binom{j}{k-1}\right) \\
&\leq \frac{1}{2^j}\binom{j}{j/2} \\
&\leq \frac{1}{2^j}\frac{j!}{((j/2)!)^2}.
\end{aligned}$$

By the Stirling formula,

$$\frac{1}{2^j}\frac{j!}{((j/2)!)^2} \;\leq\; \frac{\sqrt{2\pi j}\left(\frac{j}{e}\right)^j e^{\frac{1}{12j}}}{\left(\sqrt{2\pi(j/2)}\left(\frac{j/2}{e}\right)^{j/2} e^{\frac{1}{6j+1}}\right)^2} \;\leq\; \frac{4}{5}\cdot\frac{2^j}{\sqrt{j}}.$$

It follows that

$$\Delta(S_j, S_{j+1}) \;=\; \frac{1}{2}\sum_{k=0}^{j+1}|\mathtt{Pr}[S_j = k] - \mathtt{Pr}[S_j = k+1]| \;\leq\; \frac{2}{5}\cdot\frac{1}{\sqrt{j}}.$$

By the triangle inequality

$$\Delta(S_{t-\delta_t}, S_t) \;\leq\; \frac{2}{5}\left(\frac{1}{\sqrt{t-\delta_t}} + \ldots + \frac{1}{\sqrt{t}}\right) \;<\; \frac{\delta_t}{\sqrt{t}} \;\leq\; \frac{1}{4W}$$

for appropriate t and δ_t.

Step 2: Gap Amplification

Let V be the random variable for the number of edges among the middle δ_t edges that are truthful and in F.

$\Pr_p[V > 0] = \Pr_p[C_p \text{ is violated}]$. If $\Pr_p[V > 0] \geq \epsilon'$, then at least ϵ' fraction of ψ^t's constraints are violated.

Claim. $\mathtt{E}[V] \geq \delta_t \cdot \frac{\epsilon}{2W^2}$.

Proof.

By the previous claim, the probability of an edge in the middle interval of size $\delta\sqrt{t}$ that is truthful and in F is at least $\frac{\epsilon}{2W^2}$. Then $\mathtt{E}[V] \geq \delta_t \cdot \frac{\epsilon}{2W^2}$ by linearity. □

Step 2: Gap Amplification

Claim. $\mathtt{E}[V^2] \leq 11\delta_t d\epsilon$.

Proof.

- Let V' be the number of edges in the middle interval that are in F. Now $V \leq V'$. It suffices to show that $\mathtt{E}[V'^2] \leq 11\delta_t d\epsilon$.
- For $j \in [t-\delta_t, \ldots, t]$, let I_j be an indicator random variable that is 1 if the jth edge is in F and 0 otherwise. Then $V' = \sum_{j\in[t-\delta_t,\ldots,t]} I_j$.
- Let S be the set of the end points of the edges in F. Then $\frac{|S|}{dn} = \epsilon$. continue on next slide.

□

Step 2: Gap Amplification

$$\begin{aligned}
\mathrm{E}[V^2] &= \mathrm{E}\left[\sum_j I_j^2\right] + \mathrm{E}\left[\sum_{j\neq j'} I_j I_{j'}\right] \\
&= \epsilon\delta_t + 2\sum_{j<j'} \Pr[j\text{th edge is in } F \wedge j'\text{th edge is in } F] \\
&\leq \epsilon\delta_t + 2\sum_{j<j'} \Pr[j\text{th vertex of walk lies in } S \wedge j'\text{th vertex of walk lies in } S] \\
&\leq \epsilon\delta_t + 2\sum_{j} \Pr[j\text{th vertex of walk lies in } S] \cdot \sum_{j'>j} \Pr_{(a,b)\in E^{j'-j}}[a\in S, b\in S] \\
&\overset{(3)}{\leq} \epsilon\delta_t + 2\sum_{j} d\epsilon \cdot \sum_{j'>j} d\epsilon \left(\frac{1}{2} + \frac{(\lambda_G)^{j'-j}}{2}\right) && \frac{|S|}{n} = d\epsilon \\
&\leq \epsilon\delta_t + (\delta_t)^2(d\epsilon)^2\left(1 + \sum_{k\geq 1}(\lambda_G)^k\right) && \lambda_G < 0.9 \text{ and } \delta_t d\epsilon < 1 \\
&\leq 11\delta_t d\epsilon.
\end{aligned}$$

Finally we conclude that

$$\Pr[V>0] \;\geq\; \frac{\mathrm{E}[V]^2}{\mathrm{E}[V^2]} \;\geq\; \left(\delta_t\cdot\frac{\epsilon}{2W^2}\right)^2\cdot\frac{1}{11\delta_t d\epsilon} \;>\; \frac{\delta_t}{44dW^4}\epsilon = \ell\epsilon.$$

Lemma. For every non-negative random variable V, $\Pr[V>0] \geq \frac{\mathrm{E}[V]^2}{\mathrm{E}[V^2]}$.

Proof.

$\mathrm{E}[V|V{>}0]^2 \leq \mathrm{E}[V^2|V{>}0]$ by convex property. The lemma follows from the following.

- $\mathrm{E}[V^2|V{>}0] = \sum_i i^2\cdot\Pr[V{=}i|V{>}0] = \sum_i i^2\cdot\frac{\Pr[V=i]}{\Pr[V>0]} = \frac{\mathrm{E}[V^2]}{\Pr[V>0]}$.
- $\mathrm{E}[V|V{>}0]^2 = (\sum_i i\cdot\Pr[V{=}i|V{>}0])^2 = \left(\sum_i i\cdot\frac{\Pr[V=i]}{\Pr[V>0]}\right)^2 = \left(\frac{\mathrm{E}[V]}{\Pr[V>0]}\right)^2$.

□

Step 3: Alphabet Reduction

We look for an algorithm that transforms a $2\mathrm{CSP}_W$ instance to a $q_0\mathrm{CSP}$ instance.

A simple idea is to turn a variable over $[W]$ to $\log W$ boolean variables.

- A constraint can be turned into a circuit C of size bounded by $2^{2\log W} < W^4$.
- This would produce a CSP instance of arity $2\log W$.

The problem with this idea is that $2\log W$ is greater than q_0 (in fact $W \geq 2^{q_0}$).

- If we apply **Gap Amplification** and **Alphabet Reduction** for $\log m$ times, we would get a CSP instance whose arity depends on the input size.

Step 3: Alphabet Reduction

A more sophisticated idea is to design a PCP checker for constraint checking!

1. We turn the $2\mathrm{CSP}_W$ problem to evaluation checking problem for `CKT-SAT`.
2. We further turn it to solution checking problem for `QUADEQ`.
3. We then apply the construction of the PCP verifier (with q_0 queries!) for `QUADEQ`.
4. Finally we turn the PCP verifier to a q_0CSP instance.

PCP of Proximity, Verifier Composition, Proof Composition.

- In some occasions a verifier is allowed to make only a small or constant number of queries. In other words it cannot read any complete assignment to variables.
- A solution is to see an assignment as part of a proof. Consequently a verifier can only get to see a fragment of the proof.

This is the PCP verifier for `QUADEQ`!

Step 3: Alphabet Reduction

Suppose a constraint has been converted to a $\texttt{QUADEQ}$ instance.

- Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be assignments to $\log W$ variables.
- Let $\mathbf{c}$ be bit string of size $\ell = \texttt{poly}(W)$ that represents the quadratic equations derived from the circuit C. We assume that the first $2\log W$ bits of $\mathbf{c}$ are $\mathbf{u}_1\mathbf{u}_2$.

Let $\pi_1\pi_2\pi_3$ be a PCP proof for the $\texttt{QUADEQ}$ instance, where

- π_1 is supposedly $\texttt{WH}(\mathbf{u}_1)$, π_2 is supposedly $\texttt{WH}(\mathbf{u}_2)$ and π_3 is supposedly $\texttt{WH}(\mathbf{c})$.

Step 3: Alphabet Reduction

The PCP verifier does the following:

1. Check that π_1, π_2 and π_3 are 0.99-close to $\mathtt{WH}(\mathbf{u}_1)$, $\mathtt{WH}(\mathbf{u}_2)$ and $\mathtt{WH}(\mathbf{c})$ respectively.
2. Check that the first $2\log W$ bits of $\mathbf{c}$ are $\mathbf{u}_1\mathbf{u}_2$. This is done by concatenation test:
   - 2.1 Choose randomly $\mathbf{x}, \mathbf{y} \in \{0,1\}^{\log W}$.
   - 2.2 Check that $\pi_3(\mathbf{x}\mathbf{y}0^{|\mathbf{c}|-2\log W}) = \pi_1(\mathbf{x}) + \pi_2(\mathbf{y})$. the error probability is 1/2.

A constant number of checks are done in the Step 2 of the above test.

Step 3: Alphabet Reduction

PCP of Proximity.

There is a PCP verifier for every old constraint.

There is a verifier $\mathbb{V}$ that, given any circuit C with $2k$ input variables, runs in $\mathrm{poly}(|C|)$ time, uses $\mathrm{poly}(|C|)$ random bits, and enjoys the following property.

1. If $\mathbf{u}_1, \mathbf{u}_2 \in \{0,1\}^k$ and $\mathbf{u}_1\mathbf{u}_2$ is a satisfying assignment for C, then there is some $\pi_3 \in \{0,1\}^{2^{\mathrm{poly}(|C|)}}$ such that $\mathbb{V}$ accepts $\mathtt{WH}(\mathbf{u}_1)\mathtt{WH}(\mathbf{u}_2)\pi_3$ with probability 1.
2. For $\pi_1, \pi_2 \in \{0,1\}^{2^k}$ and $\pi_3 \in \{0,1\}^{2^{\mathrm{poly}(|C|)}}$, if $\mathbb{V}$ accepts $\pi_1\pi_2\pi_3$ with probability $> 1/2$, then π_1 and π_2 are 0.99-close to $\mathtt{WH}(\mathbf{u}_1)$ and $\mathtt{WH}(\mathbf{u}_2)$ respectively for some $\mathbf{u}_1, \mathbf{u}_2 \in \{0,1\}^k$, where $\mathbf{u}_1\mathbf{u}_2$ is a satisfying assignment to C.

Step 3: Alphabet Reduction

The PCP verifier can be turned into a $q_0\mathrm{CSP}$ instance. This is the proof of the equivalence between PCP Theorem and PCP Theorem.

Step 3: Alphabet Reduction

Fact. If the value of the old CSP is $\leq 1-\epsilon$, then the value of the new CSP is $\leq 1-\frac{1}{3}\epsilon$.

Suppose an assignment to the new variables satisfied $> 1-\frac{1}{3}\epsilon$ fraction of the new constraints. By decoding an assignment to the old variables satisfied a $1-\delta$ fraction of the old constraints. For each violated old constraint C_s, at least half of the set $\mathcal{C}_s$ of the new constraints is violated. Thus $\frac{1}{2}\delta \leq \frac{1}{3}\epsilon$. So at least $1-\delta \geq 1-\frac{2}{3}\epsilon > 1 - \epsilon$ fraction of the old constraints were violated, contradicting to the assumption.

This finishes the proof of **Alphabet Reduction** as well as the PCP Theorem.

Håstad's 3-Bit PCP Theorem

Hardness of $2\mathtt{CSP}_W$

A $2\mathtt{CSP}$ instance is regular if its constraint graph is regular.

A $2\mathtt{CSP}$ instance satisfies projection property if for each constraint $\varphi(x_1, x_2)$ and each value u of x_1 there is a unique value v of x_2 such that $\varphi(u, v) = 1$.

In other words there is a function $h : [W] \rightarrow [W]$ such that $\varphi(u, v)$ iff $h(u) = v$.
We may identify φ to h.

Hardness of $\mathtt{2CSP}_W$

Corollary. There are some ν and some W such that ν-$\mathtt{GAP2CSP}_W$ is NP-hard.

Raz Theorem. There is a $c > 1$ such that for every $t > 1$, ϵ-$\mathtt{GAP2CSP}_W$ is NP-hard for $\epsilon = 2^{-t}$, $W = 2^{ct}$. This is true also for $\mathtt{2CSP}_W$ instances that are regular and have the projection property.

In Raz Theorem one may choose ϵ as small as possible while keeping W not too large.

1. R. Raz. A parallel repetition theorem. SIAM J. Comput., 27(3):763–803, 1998, and in STOC '95.

Håstad Theorem.

For every $\delta \in (0, 1/2)$ and every $L \in \mathbf{NP}$, there is a PCP verifier $\mathbb{V}$ for L that makes three binary queries and satisfies completeness with parameter $1 - \delta$ and soundness with a parameter at most $\frac{1}{2} + \delta$.

Moreover, given a proof $\pi \in \{0, 1\}^m$, $\mathbb{V}$ chooses $i_1, i_2, i_3 \in_{\mathtt{R}} [m]$ and $b \in_{\mathtt{D}} \{0, 1\}$ according to some distribution $\mathtt{D}$ and accepts iff $\pi_{i_1} + \pi_{i_2} + \pi_{i_3} = b \ (mod\ 2)$.

[1] J. Håstad. Some Optimal Inapproximability Results. J. ACM, 48:798-859, 2001, also in STOC'97.

Threshold Result by Håstad Theorem

An instance of MAX-E3LIN consists of finitely many equations of the form

$$x_{i_1} + x_{i_2} + x_{i_3} = b \tag{4}$$

taking values in $\mathbf{F}_2$. One looks for the size of the largest set of satisfiable equations.

Håstad Theorem reveals that an NP-complete problem L has a PCP verifier that checks if a proof validates equations of the form (4). It also tells us how to reduce L to MAX-E3LIN.

Therefore it is NP-hard to compute a $\frac{1/2+\delta}{1-\delta}$-approximation to MAX-E3LIN for each $\delta \in (0, 1/2)$.

In other words $(\frac{1}{2}+\epsilon)$-approximation to MAX-E3LIN is NP-hard for all $\epsilon \in (0, 1/2)$.

It is straightforward to give a $\frac{1}{2}$-approximation algorithm for MAX-E3LIN. Just assign a truth value to a variable such that it validates at least as many constraints as it invalidates.

Threshold Result for MAX-3SAT

Fact. For all $\epsilon \in (0, 1/8)$ computing $(\frac{7}{8}+\epsilon)$-approximation to MAX-3SAT is NP-hard.

Convert $x + y + z = 0$ to four clauses $\overline{x} \vee y \vee z$, $x \vee \overline{y} \vee z$, $x \vee y \vee \overline{z}$, $\overline{x} \vee \overline{y} \vee \overline{z}$ and $x + y + z = 1$ to four clauses $x \vee \overline{y} \vee \overline{z}$, $\overline{x} \vee y \vee \overline{z}$, $\overline{x} \vee \overline{y} \vee z$, $x \vee y \vee z$.

If an assignment to x, y, z satisfies the equation, it satisfies all the 4 clauses. Otherwise it satisfies 3 of the clauses.

By the previous reduction, we see that in the yes case at least a $1 - \epsilon$ fraction of the clauses are satisfied, and in the no case at most a $1 - (\frac{1}{2} - \epsilon) \times \frac{1}{4} = \frac{7}{8} + \frac{\epsilon}{4}$ fraction of the clauses are satisfied.

There is a $\frac{7}{8}$-approximation algorithm to MAX-3SAT.

Long Code

Let $W \in \mathbb{N}$.

A function $f\colon \{\pm 1\}^W \to \{\pm 1\}$ is a coordinate function if $f(x_1, \ldots, x_W) = x_w$ for some $w \in W$. In other words $f = \chi_{\{w\}}$.

The long code for $[W]$ encodes each $w \in [W]$ by all the values of $\chi_{\{w\}}$.

Local Test for Long Code

Let $\delta \in (0,1)$.

---

1. Choose $\mathbf{x}, \mathbf{y} \in_{\mathtt{R}} \{\pm 1\}^W$.
2. Choose a noise vector $\mathbf{z} \in_{\mathtt{D}} \{\pm 1\}^W$ using distribution $\mathtt{D}$ defined as follows:

   For $i \in [W]$, choose $z_i = +1$ with probability $1 - \rho$ and $z_i = -1$ with probability ρ.
3. Accept if $f(\mathbf{x})f(\mathbf{y}) = f(\mathbf{xyz})$, reject otherwise.

If $f = \chi_{\{w\}}$, then $f(\mathbf{x})f(\mathbf{y})f(\mathbf{xyz}) = x_w y_w (x_w y_w z_w) = z_w$. The test accepts iff $z_w = 1$, which happens with probability $1 - \rho$.

Local Test for Long Code

Lemma. If the test accepts with probability $\frac{1}{2}+\delta$, then $\sum_\alpha \widehat{f}^3_\alpha(1-2\rho)^{|\alpha|} \geq 2\delta$.

If the test accepts with probability $\frac{1}{2}+\delta$, then $\mathrm{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}[f(\mathbf{x})f(\mathbf{y})f(\mathbf{xyz})] = 2\delta$. Hence,

$$\begin{aligned}
2\delta &\leq \mathrm{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}\left[\left(\sum_\alpha \widehat{f}_\alpha \chi_\alpha(\mathbf{x})\right)\left(\sum_\beta \widehat{f}_\beta \chi_\beta(\mathbf{y})\right)\left(\sum_\gamma \widehat{f}_\gamma \chi_\gamma(\mathbf{xyz})\right)\right] \\
&= \mathrm{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}\left[\sum_{\alpha,\beta,\gamma} \widehat{f}_\alpha \widehat{f}_\beta \widehat{f}_\gamma \chi_\alpha(\mathbf{x})\chi_\beta(\mathbf{y})\chi_\gamma(\mathbf{x})\chi_\gamma(\mathbf{y})\chi_\gamma(\mathbf{z})\right] \;=\; \sum_\alpha \widehat{f}^3_\alpha \mathrm{E}_{\mathbf{x}}[\chi_\alpha(\mathbf{z})] \\
&= \sum_\alpha \widehat{f}^3_\alpha \mathrm{E}_{\mathbf{x}}\left[\prod_{w\in\alpha} z_w\right] \;=\; \sum_\alpha \widehat{f}^3_\alpha \prod_{w\in\alpha} \mathrm{E}_{\mathbf{x}}[z_w] \;=\; \sum_\alpha \widehat{f}^3_\alpha (1-2\rho)^{|\alpha|}.
\end{aligned}$$

The smaller α is, the more significant $\widehat{f}_\alpha$ is. See the next corollary.

Local Test for Long Code

Corollary. If f passes the long code test with probability $\frac{1}{2}+\delta$, then for $k=\frac{1}{2\rho}\log\frac{1}{\epsilon}$, there is α with $|\alpha|\leq k$ such that $\widehat{f}_{\alpha}\geq 2\delta-\epsilon$.

By the previous lemma,

$$\begin{aligned}
2\delta &\leq \sum_{\alpha}\widehat{f}^3_{\alpha}(1-2\rho)^{|\alpha|} = \sum_{|\alpha|\leq k}\widehat{f}^3_{\alpha}(1-2\rho)^{|\alpha|}+\sum_{|\alpha|>k}\widehat{f}^3_{\alpha}(1-2\rho)^{|\alpha|} \\
&\leq \max_{|\alpha|\leq k}\widehat{f}_{\alpha}+\sum_{|\alpha|>k}\widehat{f}^2_{\alpha}(1-2\rho)^{|\alpha|} \\
&\leq \max_{|\alpha|\leq k}\widehat{f}_{\alpha}+(1-2\rho)^{k} \\
&\leq \max_{|\alpha|\leq k}\widehat{f}_{\alpha}+\epsilon.
\end{aligned}$$

Bifolded Long Code

A function f: $\{\pm 1\}^W \to \{\pm 1\}$ is bifolded if $f(-\mathbf{v}) = -f(\mathbf{v})$ for all $\mathbf{v} \in \{\pm 1\}^W$.

The coordinate function is bifolded because $\chi_{\{w\}}(-\mathbf{v}) = -\chi_{\{w\}}(\mathbf{v})$.

We may assume without loss of generality that the long code in Håstad's verifier is bifolded. Define $f(\mathbf{v})$ if the most significant bit of $\mathbf{v}$ is 1.

Bifolded Long Code

If f: $\{\pm 1\}^W \rightarrow \{\pm 1\}$ is bifolded and $\widehat{f}_\alpha \neq 0$ then $|\alpha|$ must be an odd number (and in particular, nonzero).

If $|\alpha|$ is even, then

$$\widehat{f}_\alpha = \langle f, \chi_\alpha \rangle = \mathrm{E}_{\mathbf{v}}[f(\mathbf{v}) \textstyle\prod_{i \in \alpha} \mathbf{v}_i] = 0$$

since $\prod_{i \in \alpha} \mathbf{v}_i = \prod_{i \in \alpha} (-\mathbf{v}_i)$.

Bifolded function has only odd size set indexed Fourier coefficients.

Håstad's Verifier $\mathbb{V}_H$

1. $\mathbb{V}_H$ expects a proof $\widetilde{\pi}$ of length $n2^W$ coding n functions $f_1, \ldots, f_n : \{\pm\}^W \rightarrow \{\pm\}$, which are the bifolded long codes of an assignment to the variables.
2. $\mathbb{V}_H$ randomly chooses a constraint $\varphi_r(i,j)$ described by a function $h : [W] \rightarrow [W]$.
3. $\mathbb{V}_H$ verifies that f_i, f_j code up w and u respectively such that $h(w) = u$.

Håstad's Verifier $\mathbb{V}_H$

The Basic Håstad Test.

1. Input. Two functions $f, g : \{\pm\}^W \rightarrow \{\pm\}$ and a function $h : [W] \rightarrow [W]$.
2. Goal. Check that f, g are the long codes of w, u respectively such that $h(w) = u$.
3. Test.
   - 3.1 $\mathbb{V}_\mathrm{H}$ chooses $\mathbf{v}, \mathbf{y} \in_\mathrm{R} \{\pm 1\}^W$.
   - 3.2 $\mathbb{V}_\mathrm{H}$ chooses noise vector $\mathbf{z} \in \{\pm 1\}^W$ by letting $z_i = +1$ with probability $1 - \rho$ and $z_i = -1$ with probability ρ.
   - 3.3 $\mathbb{V}_\mathrm{H}$ accepts if
   $$f(\mathbf{v})g(\mathbf{y}) = f(h^{-1}(\mathbf{y})\mathbf{v}\mathbf{z})$$
   and rejects otherwise.

For $u \in [W]$ let $h^{-1}(u) = \{w \colon h(w) = u\}$.

For $\mathbf{y} \in \{\pm 1\}^W$ let $h^{-1}(\mathbf{y}) \in \{\pm 1\}^W$ be such that $\left(h^{-1}(\mathbf{y})\right)(w) = y_{h(w)}$ for all $w \in [W]$.

Håstad's Verifier $\mathbb{V}_H$

Notice that $f, \mathbf{v}$ constitute a location in $\widetilde{\pi}$. It follows from the equivalence

$$f(\mathbf{v})g(\mathbf{y}) = f(h^{-1}(\mathbf{y})\mathbf{vz}) \text{ iff } f(\mathbf{v})g(\mathbf{y})f(h^{-1}(\mathbf{y})\mathbf{vz}) = 1$$

that, by translating back from $\{\pm 1\}$ to $\{0, 1\}$, the verifier $\mathbb{V}_{\mathrm{H}}$ accepts iff

$$\tilde{\pi}[i_1] + \tilde{\pi}[i_2] + \tilde{\pi}[i_3] = 0 \pmod 2$$

for the corresponding $i_1, i_2, i_3 \in [n2^W]$.

Main Claim. If φ is satisfiable, there is a proof $\mathbb{V}_{\rm H}$ accepts with probability $1-\rho$. If $\texttt{val}(\varphi) \leq \epsilon$, no proof is accepted by $\mathbb{V}_{\rm H}$ with probability $> 1/2+\delta$, where $\delta = \sqrt{\epsilon/\rho}$.

Håstad's 3-Bit PCP Theorem is proved by taking $\rho = \epsilon^{1/3}$, which makes the soundness parameter at most $1/2 + \epsilon^{1/3}$ and the completeness parameter at least $1-\epsilon^{1/3}$.

Proof of Completeness Part

If φ_r is satisfiable, the proof $\widetilde{\pi}$ contains the bifolded long code encodings of the n values in $[W]$. Suppose f, g are the long codes of $w, u \in [W]$ satisfying $h(w) = u$.

$$\begin{aligned} f(\mathbf{v})g(\mathbf{y})f(h^{-1}(\mathbf{y})\mathbf{vz}) &= v_w y_u (h^{-1}(\mathbf{y})_w v_w z_w) \\ &= v_w y_u (y_{h(w)} v_w z_w) \\ &= v_w^2 y_u^2 z_w \\ &= z_w. \end{aligned}$$

Thus $\mathbb{V}_{\mathrm{H}}$ accepts iff $z_w = +1$, which happens with probability $1 - \rho$.

Proof of Soundness Part

For each $\alpha \subseteq [W]$, we define a set

$$h_2(\alpha) = \{u \in [W] \mid \{w \in W \mid h(w) = u\} \cap R \text{ is of odd size}\}.$$

Notice that for every $v \in h_2(\alpha)$ there is at least one $w \in \alpha$ such that $h(w) = v$.

Proof of Soundness Part

Lemma. Let $f, g\colon \{\pm 1\}^W \to \{\pm 1\}$ be bifolded functions and $h\colon [W] \to [W]$ be such that they pass the Basic Håstad's Test with probability at least $1/2 + \delta$. Then

$$\sum_{\alpha \subseteq [W], \alpha \neq \emptyset} \widehat{f}^2_{\alpha} \widehat{g}_{h_2(\alpha)} (1 - 2\rho)^{|\alpha|} \geq 2\delta.$$

If the Basic Håstad's Test accepts f, g, then f and g are correlated.

Proof of Soundness Part

By hypothesis $\mathrm{E}[f(\mathbf{v})g(\mathbf{y})f(h^{-1}(\mathbf{y})\mathbf{vz})] \geq 2\delta$. Using the Fourier expansions of f, g one gets

$$\begin{aligned}
2\delta &\leq \mathrm{E}_{\mathbf{v},\mathbf{y},\mathbf{z}}\left[\left(\sum_{\alpha}\widehat{f}_{\alpha}\chi_{\alpha}(\mathbf{v})\right)\left(\sum_{\beta}\widehat{g}_{\beta}\chi_{\beta}(\mathbf{y})\right)\left(\sum_{\gamma}\widehat{f}_{\gamma}\chi_{\gamma}(\mathbf{v}h^{-1}(\mathbf{y})\mathbf{z})\right)\right] \\
&= \sum_{\alpha,\beta,\gamma}\widehat{f}_{\alpha}\widehat{g}_{\beta}\widehat{f}_{\gamma}\mathrm{E}_{\mathbf{v},\mathbf{y},\mathbf{z}}[\chi_{\alpha}(\mathbf{v})\chi_{\beta}(\mathbf{y})\chi_{\gamma}(\mathbf{v})\chi_{\gamma}(h^{-1}\mathbf{y})\chi_{\gamma}(\mathbf{z})] \\
&= \sum_{\alpha,\beta}\widehat{f}^{2}_{\alpha}\widehat{g}_{\beta}\mathrm{E}_{\mathbf{y},\mathbf{z}}[\chi_{\alpha}(h^{-1}\mathbf{y})\chi_{\alpha}(\mathbf{z})\chi_{\beta}(\mathbf{y})] \\
&= \sum_{\alpha,\beta}\widehat{f}^{2}_{\alpha}\widehat{g}_{\beta}\mathrm{E}_{\mathbf{z}}[\chi_{\alpha}(\mathbf{z})]\mathrm{E}_{\mathbf{y}}[\chi_{\alpha}(h^{-1}(\mathbf{y}))\chi_{\beta}(\mathbf{y})] \\
&= \sum_{\alpha,\beta}\widehat{f}^{2}_{\alpha}\widehat{g}_{\beta}(1-2\rho)^{|\alpha|}\mathrm{E}_{\mathbf{y}}[\chi_{\alpha}(h^{-1}(\mathbf{y}))\chi_{\beta}(\mathbf{y})].
\end{aligned}$$

Proof of Soundness Part

To continue one has

$$\begin{aligned}
\mathrm{E}_{\mathbf{y}}[\chi_\alpha(h^{-1}(\mathbf{y}))\chi_\beta(\mathbf{y})] &= \mathrm{E}_{\mathbf{y}}[\prod_{w\in\alpha} h^{-1}(\mathbf{y})_w \prod_{u\in\beta} y_u] && \text{(Definition of Fourier basis } \chi\_\text{)} \\
&= \mathrm{E}_{\mathbf{y}}[\prod_{w\in\alpha} y_{h(w)} \prod_{u\in\beta} y_u] && \text{(Definition of } h^{-1}\text{)} \\
&= \mathrm{E}_{\mathbf{y}}[\prod_{v\in h_2(\alpha)} y_v \prod_{u\in\beta} y_u] && (\prod_{w\in\alpha} y_{h(w)} = \prod_{v\in h_2(\alpha)} y_v) \\
&= \langle \chi_{h_2(\alpha)}, \chi_\beta \rangle \\
&= \begin{cases} 1 & \text{if } h_2(\alpha) = \beta \\ 0 & \text{otherwise} \end{cases} && \text{(Fourier basis is orthonormal)}
\end{aligned}$$

Proof of Soundness Part

Hence

$$\begin{aligned} 2\delta &\leq \sum_{\alpha,\beta} \widehat{f}_{\alpha}^{2} \widehat{g}_{\beta} (1-2\rho)^{|\alpha|} \mathrm{E}_{\mathbf{y}}[\chi_{\alpha}(h^{-1}(\mathbf{y}))\chi_{\beta}(\mathbf{y})] \\ &= \sum_{\alpha} \widehat{f}_{\alpha}^{2} \widehat{g}_{h_2(\alpha)} (1-2\rho)^{|\alpha|} \\ &= \sum_{\alpha\neq\emptyset} \widehat{f}_{\alpha}^{2} \widehat{g}_{h_2(\alpha)} (1-2\rho)^{|\alpha|}. \end{aligned}$$

The last equality holds since all even Fourier coefficients of an bifolded function are 0.

Proof of Soundness Part

Lemma. Suppose φ is an instance of $2\mathrm{CSP}_W$ such that $\mathtt{val}(\varphi) < \epsilon$. If ρ, δ satisfy $\rho\delta^2 > \epsilon$, the verifier $\mathbb{V}_\mathrm{H}$ accepts any proof with probability at most $1/2 + \delta$.

The Lemma completes the proof of **Main Claim** and of Håstad's 3-bit PCP Theorem.

Proof.

If $\mathbb{V}_\mathrm{H}$ accepts a proof $\tilde{\pi}$ with probability $> 1/2 + \delta$, we can construct an assignment π probabilistically such that the expected fraction of the satisfying constraints is $\geq \rho\delta^2$. But then $\mathtt{val}(\varphi) \geq \rho\delta^2 > \epsilon$, contradicting to the assumption. □

The proof of the lemma is given on the next few slides.

Proof of Soundness Part, Proof of the Lemma

Choosing π randomly.

---

1. For each $i \in [n]$, use the bifolded function f_i given by $\tilde{\pi}$ to define a distribution $\mathcal{D}_i$ over $[W]$ as follows:
   1.1 Firstly select a set $\alpha \subseteq [W]$ with probability $\widehat{f}_{i\alpha}^{2}$.
   1.2 Then select an element w at random from α.

   This is well-defined because $\sum_\alpha \widehat{f}_{i\alpha}^{2} = \langle f_i, f_i \rangle = 1$ and $\widehat{f}_{i\emptyset} = 0$.
2. Pick $\pi[i]$ by drawing a random sample from distribution $\mathcal{D}_i$.

The assignment π is a random element of the product distribution $\prod_{i=1}^{n} \mathcal{D}_i$.

Proof of Soundness Part, Proof of the Lemma

We intend to prove that

$$\mathtt{E}_{\pi}[\mathrm{E}_{r\in[m]}[\pi \text{ satisfies } \varphi_r]] \geq \rho\delta^2. \tag{5}$$

Let $\frac{1}{2}+\delta_r$ be $\Pr[\mathbb{V}_H \text{ accepts } \widetilde{\pi}|\varphi_r \text{ is chosen}]$. It is sufficient to prove

$$\Pr_{\pi}[\pi \text{ satisfies } \varphi_r] \geq \rho\delta_r^2. \tag{6}$$

- $\mathtt{E}_r[\frac{1}{2}+\delta_r] = \frac{1}{2}+\delta$ implies $\mathtt{E}_r[\delta_r] = \delta$.
- $\mathtt{E}_{\pi}[\mathrm{E}_r[\pi \text{ satisfies } \varphi_r]] = \mathtt{E}_r[\mathrm{E}_{\pi}[\pi \text{ satisfies } \varphi_r]] \geq \mathtt{E}_r[\rho\delta_r^2] \geq \rho\mathtt{E}_r[\delta_r]^2 = \rho\delta^2$.

Proof of Soundness Part, Proof of the Lemma

Let $\varphi_r(i,j)$ be the r-th constraint and h be the function describing the constraint. Now

$$\mathrm{Pr}_\pi[\pi \text{ satisfies } \varphi_r] = \mathrm{Pr}_\pi[h(\pi[i]) = \pi[j]].$$

Let $f = f_i$, $g = f_j$. Recall that $\pi[i]$, $\pi[j]$ are picked by choosing α with probability $\widehat{f}^2_\alpha$, β with probability $\widehat{g}^2_\beta$, and then choosing $\pi[i] \in_{\mathrm{R}} \alpha$, $\pi[j] \in_{\mathrm{R}} \beta$ randomly. Hence

$$\begin{aligned}
\mathrm{Pr}_\pi[h(\pi[i]) = \pi[j]] &= \sum_\alpha \widehat{f}^2_\alpha \sum_\beta \widehat{g}^2_\beta \cdot \mathrm{Pr}_\pi \left[h(\pi[i]) = \pi[j] \mid \pi[i] \in \alpha, \pi[j] \in \beta\right] \\
&\geq \sum_\alpha \widehat{f}^2_\alpha \widehat{g}^2_{h_2(\alpha)} \cdot \mathrm{Pr}_\pi \left[h(\pi[i]) = \pi[j] \mid \pi[i] \in \alpha, \pi[j] \in h_2(\alpha)\right] \\
&\geq \sum_\alpha \widehat{f}^2_\alpha \widehat{g}^2_{h_2(\alpha)} \frac{1}{|\alpha|}.
\end{aligned}$$

By the previous lemma, $2\delta_r \leq \sum_\alpha \widehat{f}^2_\alpha \widehat{g}_{h_2(\alpha)} (1-2\rho)^{|\alpha|}$.

Proof of Soundness Part, Proof of the Lemma

Now $1-x \leq e^{-x} \leq x^{-1}$ holds for all $x > 0$. Therefore

$$(1-2\rho)^{|\alpha|} \leq (e^{-2\rho})^{|\alpha|} = (e^{-4\rho|\alpha|})^{1/2} \leq (4\rho|\alpha|)^{-1/2} \leq \frac{2}{\sqrt{\rho|\alpha|}}.$$

Hence

$$\delta_r\sqrt{\rho} \leq \sum_{\alpha} \widehat{f}^2_{\alpha}\widehat{g}_{h_2(\alpha)}\frac{1}{\sqrt{|\alpha|}}.$$

It follows from the Cauchy-Schwartz inequality that

$$\delta_r\sqrt{\rho} \leq \sum_{\alpha} \widehat{f}^2_{\alpha}\widehat{g}_{h_2(\alpha)}\frac{1}{\sqrt{|\alpha|}} \leq \left(\sum_{\alpha}\widehat{f}^2_{\alpha}\right)^{1/2}\left(\sum_{\alpha}\widehat{f}^2_{\alpha}\widehat{g}^2_{h_2(\alpha)}\frac{1}{|\alpha|}\right)^{1/2} \leq \left(\sum_{\alpha}\widehat{f}^2_{\alpha}\widehat{g}^2_{h_2(\alpha)}\frac{1}{|\alpha|}\right)^{1/2}.$$

So $\rho\delta_r^2 \leq \sum_{\alpha}\widehat{f}^2_{\alpha}\widehat{g}^2_{h_2(\alpha)}\frac{1}{|\alpha|} \leq \Pr_{\pi}[h(\pi[i]) = \pi[j]] = \Pr_{\pi}[\pi \text{ satisfies } \varphi_r]$.

Historical Remark

Interactive proof, Zero knowledge, **IP**.

It all started with the introduction of interactive proof systems.

1. Shafi Goldwasser, Silvio Micali, Charles Rackoff. The Knowledge Complexity of Interactive Proofs. STOC'85, SIAM, 1989.
2. László Babai and Shlomo Moran. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. STOC'85, JCSS, 1988.

The authors of the papers shared the first Gödel Prize (1993).

Goldwasser and Sipser. Private Coins versus Public Coins in Interactive Proof Systems. STOC'86.

"1989 was an extraordinary year." László Babai, 1990

1. N. Nisan. Co-SAT has multi-prover interactive proofs, e-mail announcement. Nov. 27, 1989.
2. C. Lund, L. Fortnow, H. Karloff, N. Nisan. The polynomial time hierarchy has interactive proofs, e-mail announcement, Dec. 13, 1989.
3. A. Shamir. IP=PSPACE, e-mail announcement, Dec. 26, 1989.

On Jan. 17, 1990 another email was sent out by L. Babai, L. Fortnow, and L. Lund.

- Non-Deterministic Exponential Time has Two Prover Interactive Protocols. FOCS 1990. CC 1991.

The main theorem of the paper, **MIP** $=$ **NEXP**, inspired almost all future development of PCP theory and a lot of future development in derandomization theory. It can be interpreted as

$$\mathbf{NEXP} = \mathbf{PCP}(\mathrm{poly}, \mathrm{poly}).$$

A profitable shift of emphasis was made that, instead of scaling down the time or space complexity of verifier, scales down the randomness and query complexity.

Babai, Fortnow, Levin, and Szegedy showed **NP** $\subseteq$ **PCP**$(\mathrm{polylog}, \mathrm{polylog})$.

1. L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking Computation in Polylogarithmic Time. STOC, 1991.

2001 Gödel Prize

1. $\mathbf{NP} \subseteq \mathbf{PCP}(\log \cdot \log\log, \log \cdot \log\log)$.
2. $\mathbf{NP} = \mathbf{PCP}(\log, \log)$.
3. $\mathbf{NP} = \mathbf{PCP}(\log, 1)$.

1. U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive Proofs and the Hardness of Approximating Cliques. FOCS'91, JACM, 1996.
2. S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. FOCS'92, JACM, 1998.
3. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems. FOCS'92, JACM, 1998.

2019 Gödel Prize

Irit Dinur. The PCP Theorem by Gap Amplification. J. ACM, 2007. STOC 2006.