Imperfect Gaps in Gap-ETH and PCPs

by

Nikhil Vyas

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2019

Signature redacted

Certified by .....
Richard Ryan Williams
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Imperfect Gaps in Gap-ETH and PCPs

by

Nikhil Vyas

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

In this thesis we study the role of perfect completeness in probabilistically checkable proof systems (PCPs) and give a new way to transform a PCP with imperfect completeness to a PCP with perfect completeness, when the initial gap is a constant. In particular, we show that $\mathrm{PCP}_{c,s}[r,q] \subseteq \mathrm{PCP}_{1,s'}[r+O(1), q+O(r)]$ for $c-s=\Omega(1)$ which in turn implies that one can convert imperfect completeness to perfect in linear-sized PCPs for $NTIME[O(n)]$ with a $O(\log n)$ additive loss in the query complexity q. We show our result by constructing a "robust circuit" using threshold gates. These results are a gap amplification procedure for PCPs (when completeness is imperfect), analogous to questions studied in parallel repetition [21] and pseudorandomness [14].

We also investigate the time complexity of approximating perfectly satisfiable instances of 3SAT versus those with imperfect completeness. We show that the Gap-ETH conjecture without perfect completeness is equivalent to Gap-ETH with perfect completeness; that is, MAX 3SAT$(1-\epsilon, 1-\delta)$ for $\delta > \epsilon$ has $2^{o(n)}$-time algorithms if and only if MAX 3SAT$(1, 1-\delta)$ has $2^{o(n)}$-time algorithms. We also relate the time complexities of these two problems in a more fine-grained way, to show that $T_2(n) \leq T_1(n(\log\log n)^{O(1)})$, where $T_1(n), T_2(n)$ denote the randomized time-complexity of approximating MAX 3SAT with perfect and imperfect completeness, respectively. This is joint work with Mitali Bafna.

Thesis Supervisor: Richard Ryan Williams
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

The result of this thesis is based on a collaboration with Mitali Bafna. I would like to thank Ryan Williams for being such a great advisor and for his guidance on the project.

Contents

Chapter 1

Introduction

The PCP theorem [2] was a breakthrough result showing that NP has proofs verifiable using only $O(1)$ bits and constant probability of error, with a polynomial blow-up in the size of the proof. The theorem led to a flurry of activity in getting the best set of parameters: the soundness (i.e. the probability of error), proof size and queries. PCP constructions were instrumental in showing optimal hardness of approximation results for a host of problems such as k-SAT and 3-LIN [17]. Despite this progress, many important questions have remained wide open. For instance: *do there exist linear-size PCPs for $NTIME[O(n)]$, with constant queries and constant soundness?* Hence we believe it is important to understand the role of all the parameters in PCPs for $NTIME[O(n)]$, and we focus our attention on the completeness of these proof systems.

We investigate the question: *can imperfect completeness help obtain better PCPs?* The size versus query tradeoff in PCPs has been extensively studied: A long line of work culminated in a PCP for $NTIME[O(n)]$ with $O(n \cdot \text{polylog } n)$ size and $O(1)$ queries [10]. On the other hand, Ben-Sasson et al [5] achieved a linear-sized PCP for $NTIME[O(n)]$ with $O(n^\epsilon)$ query size for all constants $\epsilon > 0$.[1] These results are far from what is conjectured: namely, that PCPs exist with $O(1)$ queries and linear size. In this thesis, we show how to transform any PCP with imperfect completeness and

[1] This particular construction is non-uniform. To our knowledge no explicit PCPs with $o(n)$ query complexity, constant soundness and linear size are known.

constant gap (between soundness and completeness) to one with perfect completeness and a mild additive extra number of queries. The loss in query complexity in the transformation from imperfect to perfect completeness in the latter regime (of linear-size) is inconsequential in comparison to the query complexity of [5].

Although in current PCP constructions for $NTIME[O(n)]$, perfect completeness might come for free when one does not care about the verifier's predicate, PCPs with imperfect completeness are very important in showing optimal hardness of approximation for problems like 3LIN [17], where deciding satisfiability is in polynomial time. For other CSPs like Max 1-in-k-SAT one can get substantially better approximation algorithms for perfectly satisfiable instances [16]. The Unique Games Conjecture of Khot [19] asks for a PCP with "unique" queries and imperfect completeness, the latter being necessary due to the tractability of satisfiable instances of Unique Games. Although in some previous cases like 3LIN, imperfect completeness was necessary, but for cases like 2-to-1 games and Max k-CSP one would guess that the same hardness of approximation results should hold with perfect completeness. Unfortunately all the current methods [12, 9] incur a loss in completeness, and it is unclear whether this is because of the nature of the problem or due to the inefficacy of current methods. This leads to the central question: *Given a CSP, how hard is it to approximate instances that are perfectly satisfiable, compared to those that are not?*

We also study this question in a fine-grained way, and compare the time complexities of approximating satisfiable versus imperfectly satisfiable instances of 3SAT. NP-hardness results (while very useful in measuring intractability with respect to poly-time algorithms) do not imply tight or even superpolynomial lower bounds for the running time. The Exponential Time Hypothesis (ETH) [18] states that there are no $2^{o(n)}$ time algorithms for deciding satisfiability of 3SAT. Through the equivalence between PCPs and gap problems, using state of the art PCPs [10, 6] there is a reduction from a 3SAT instance on n variables and clauses to a Gap-3SAT instance with $O(n \cdot \text{polylog } n)$ variables and clauses. This proves that under ETH, Gap-3SAT does not have $O(2^{n/\log^c(n)})$ algorithms for some fixed c, whereas Gap-3SAT has eluded even $2^{o(n)}$ algorithms. To get around precisely this gap, the Gap-ETH hypothesis was

proposed [11, 20]. Gap-ETH states that Gap-3SAT does not have $2^{o(n)}$ algorithms. This hypothesis has led to several tight inapproximability results [8, 13, 7, 1] with respect to the running time required. We study the role of perfect completeness in Gap-ETH, where Gap-ETH without perfect completeness is the hypothesis that there are no $2^{o(n)}$ algorithms for Gap-3SAT without perfect completeness.

Gap amplification is in itself an important problem studied in the context of parallel repetition [22], error reduction and pseudorandomness [14]. We study this problem in PCPs and show a way to transform any PCP into a one-sided error one. Similar questions of gap amplification when completeness is not 1, have been studied for parallel repetition [21], but these results incur a huge blow-up in the alphabet and hence cannot be applied to get perfect completeness in PCPs with constant alphabet. These techniques in parallel repetition have been used in quantum computation, to show instances of multi-player games with large separation between the entangled and classical value and amplification of entangled games [21, 3].

1.0.1 Our contributions

PCPs without perfect completeness: We show a way to boost the completeness of PCPs which makes the completeness 1. Our results go via the construction of "robust circuits" for the approximate threshold function on n bits. These circuits are of depth $O(\log n)$, fan-in $O(1)$ and size $O(n)$, and use successive layers of threshold gates to boost the fraction of ones in inputs that have large Hamming weight, while maintaining the fraction of ones in other inputs below a certain threshold. The circuits are tolerant to some form of adversarial corruptions and this property allows us to prove the soundness of the new PCP. Our main theorem is the following:

Theorem 1.0.1. *Let $c, s \in (0,1), s < c$ be constants then there exists a constant $s' \in (0,1)$ depending only on c, s such that,*

$$PCP_{c,s}[r,q] \subseteq PCP_{1,s'}[r, q + O_{s,c}(r)]$$

furthermore if the original proof size was n then the final proof size in $n + O(2^r)$.

Note that in the above theorem, one can prove inclusion in a PCP class, with arbitrary constant s'' (instead of a fixed constant s') by applying derandomized serial repetition ($\text{PCP}_{1,s'}[r,q] \subseteq \text{PCP}_{1,s''}[r,O(q)]$ with same proof size). This does not blow up the size of the PCP and the query complexity only increases by a constant factor.

As a corollary we show that linear-sized PCPs for $NTIME[O(n)]$ with imperfect completeness, can be converted to a linear-sized PCPs with perfect completeness and $q + O(\log n)$ queries. Current PCP constructions with constant rate and alphabet have query complexity $n^{\Omega(1)}$[5], so we show that for improving upon this, it is enough to get linear sized PCPs with imperfect completeness and better query complexity.

We also consider the notion of "randomized reduction between PCPs", defined below. Bellare et al [4] considered the notion of a randomized reduction R between two promise problems given by sets (A_1, B_1) and (A_2, B_2). A randomized polynomial time reduction R from promise problems $(A_1, B_1) \leq_R (A_2, B_2)$ with error probability p satisfies:

1. if $x \in A_1$ then w.p. $\geq 1 - p$, $R(x) \in A_2$.

2. if $x \in B_1$ then w.p. $\geq 1 - p$, $R(x) \in B_2$.

This notion naturally extends to PCP complexity classes. We give a randomized reduction between PCP classes with imperfect and perfect completeness.

Theorem 1.0.2. *Let $c, s \in (0,1), s < c$ be constants then there exists a constant $s' \in (0,1)$ depending only on c, s such that,*

$$PCP_{c,s}[r,q] \leq_R PCP_{1,s'}[r, q + O_{s,c}(\log r)]$$

with probability $1 - 2^{-\Omega(r)}$. Furthermore if the original proof size was n then the final proof size in $n + O(2^r)$.

Gap-ETH without perfect completeness We study the relation between time complexities of approximating satisfiable instances of MAX 3SAT versus that of approximating unsatisfiable instances. We first show the equivalence of the Gap-ETH

conjecture with perfect and imperfect completeness. We formally state the Gap-ETH conjecture below:

Conjecture 1 (Gap Exponential-Time Hypothesis (Gap-ETH) [11, 20]). *For some constants $\delta, \epsilon > 0$, no algorithm can, given a 3-SAT formula ϕ on n variables and $m = O(n)$ clauses, solve the decision problem MAX 3-SAT$(1, 1-\epsilon)$ in $O(2^{\delta n})$ time.*

There are many versions of the Gap-ETH conjecture that one can consider. Many works study the randomized Gap-ETH conjecture which says that there are not even any randomized algorithms that can decide Max 3-SAT$(1, 1-\epsilon)$. We show the following theorem:

Theorem 1.0.3. *If there exists a randomized (with no false positives) $2^{o(n)}$ time algorithm for MAX 3SAT$(1, 1-\gamma)$ for all constant $\gamma > 0$ then there exists a randomized (with no false positives) $2^{o(n)}$ time algorithm for MAX 3SAT$(s(1+\epsilon), s)$ for all constants $s, \epsilon > 0$.*

Algorithms with no false positives are interesting as:

- Randomized SAT (not MAX-SAT) algorithms can be modified to have no false positives by self-reduction.
- some of the hardness results from Gap-ETH go through reductions which do not produce false positives [7]. This allows us to compose without losing in hardness by assuming Gap-ETH with one sided error.

As the original Gap-ETH hypothesis [11, 20] talks about deterministic algorithms we would prefer to get a deterministic reduction between these two problems.

We can get more fine-grained results relating the time-complexities of approximating MAX 3SAT with perfect and imperfect completeness using the Theorem 1.0.2 stated earlier.

Corollary 1.0.1. *If there exists a $T(n)$ time algorithm for MAX 3SAT$(1, 1-\delta)$ for all $\delta > 0$ then there exists a $T(n(\log\log n)^{O(1)})$ time randomized algorithm for MAX 3SAT$(1-\epsilon, 1-\gamma)$ for all $\epsilon, \gamma, 0 < \epsilon < \gamma$.*

1.0.2 Previous work

Bellare et al [4] also studied the problem transforming probabilistically checkable proofs with imperfect completeness to those with perfect completeness. Their techniques do not yield any inclusions for PCP classes. They proved the following randomized reduction between PCP classes:

$$\mathrm{PCP}_{c,s}[r,q] \leq_R \mathrm{PCP}_{1,rs/c}[r,qr/c]$$

For constant c and $r = \omega(1)$, they lose a multiplicative factor of r in the soundness, which makes the theorem non-trivial only when $s = o(1)$.

1.1 Preliminaries

We will use the following notation:

Notation: $\mathrm{Thr}_\delta(x_{i_1}, \ldots, x_{i_r})$ = threshold at δ-fraction taken on the set of bits $\{x_{i_1}, \ldots, x_{i_r}\}$. We also use $\mathrm{Thr}_\delta(x|_S)$ to mean that the threshold is with respect to the bits of x restricted to $S \subseteq [n]$ and sometimes drop the x and δ to use $\mathrm{Thr}(S)$, when the input/fraction being used is clear from context. $exp(x)$ refers to e^x. For a string $x \in \{0,1\}^n$, let $\bar{x} = \frac{1}{n}\sum_i x_i$, denote the average number of 1's in x.

MAX k-CSP(c,s) - the promise problem of deciding whether there exists an assignment satisfying more than c-fraction given k-clauses or every assignment satisfies at most s fraction of clauses. When the CSP is a 3SAT instance, it is denoted by MAX 3SAT(c,s).

Firstly we discuss some standard probability results like the Chernoff bound and the Lovász local lemma.

Chernoff Bounds:

1. Multiplicative Chernoff bound 1: Let $X = \frac{1}{n}X_i$, where $X_1, \ldots, X_n$ are random

variables in $\{0,1\}$, with $E[X] = \mu$. Then for all $\delta \geq 1$,

$$\Pr[X > (1+\delta)\mu] \leq exp(-\Omega(\delta\mu))$$

for $\delta \leq 1$,

$$\Pr[X > (1+\delta)\mu] \leq exp(-\delta^2\mu/3))$$

$$\Pr[X < (1-\delta)\mu] \leq exp(-\delta^2\mu/2))$$

2. Multiplicative Chernoff bound 2: Let $X = \frac{1}{n}X_i$, where $X_1, \ldots, X_n$ are random variables in $\{0,1\}$, with $E[X] = \mu$. Then for all $\delta \geq 2$,

$$\Pr[X > (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \exp(-\Omega(\delta(\log(1/\delta))\mu)).$$

Lemma 1.1.1 (Lovász local lemma). *Let $E_1, E_2, \ldots, E_n$ with $\Pr[E_i] = p$ be events such that any E_i is independent of all but d other events. Then if $pe(d+1) \leq 1$ then*

$$\Pr\left[\bigcap E_i\right] \geq (1-1/d)^n$$

Let us now define probabilistic proof systems. Firstly, we define the notion of an (r,q)-restricted verifier: For integer valued functions $r(\cdot)$ and $q(\cdot)$, a verifier is said to be (r,q)-restricted if on every input of length n, it tosses at most $r(n)$ coins and queries the proof in at most $q(n)$ bits non-adaptively.

Definition 1.1.1 (PCP). *For integer-valued functions $r(\cdot), q(\cdot)$ and functions $c(\cdot), s(\cdot)$ mapping to $[0,1]$, the class $PCP_{c,s}[r,q]$ consists of all languages for which there exists an (r,q)-restricted non-adaptive verifier V with the following properties:*

1. *Completeness: For all $x \in L$, there exists a proof π such that $V^\pi(x)$ accepts with probability at least c (over the coin tosses of V).*

2. *Soundness: For all $x \notin L$, for all proofs π, $V^\pi(x)$ accepts with probability at most s.*

We now go to the notion of averaging samplers. Averaging samplers are used to derandomize the process of random sampling to estimate the average number of ones in a string $x = \{0,1\}^n$, see survey of [15]. We use the following sampler therein:

Lemma 1.1.2. *The expander sampler with parameters (δ, ϵ, N) is an expander graph on N vertices, such that the neighbors of a vertex i, specify a sample $S_i \subseteq [N]$. The set family $\mathcal{ES}(\delta, \epsilon, N) = \{S_i\}_{i=1}^N$ satisfies the following properties:*

1. *For all i, $|S_i| = \frac{1}{\delta\epsilon^2}$*
2. *For every S_i the number of sets S_j which intersect with it are $O\left(\frac{1}{\delta^2\epsilon^4}\right)$.*
3. *For any string $x \in \{0,1\}^N$, $\Pr_{S \sim \mathcal{ES}(\delta,\epsilon,N)}[|\overline{(x|_S)} - \overline{x}| > \epsilon] \leq \delta$, where $\overline{(x|_S)}$ denotes the average of x taken over the positions specified by S.*

We analyse the standard expander sampler given above and prove that one can get a sampler with the following properties.

Theorem 1.1.1 (Sampler). *For all constants ϵ, δ, γ, there exists a constant C such that, there is a set family $\mathcal{S}(\epsilon, \delta, \gamma, N) = (S_i)_{i=1}^{N/2}$ on $[N]$ with the following properties:*

1. *For any string $x \in \{0,1\}^N$, $\Pr_{S \sim \mathcal{S}}[|\overline{(x|_S)} - \overline{x}| > \epsilon] \leq \delta$.*
2. *For all $\eta < (1-\gamma)/2$, for any string $x \in \{0,1\}^N$, where $\overline{x} \geq 1 - \eta$, we get that, $\Pr_{S \sim \mathcal{S}}[\overline{(x|_S)} < \gamma] \leq \eta/2$.*
3. *For all i, $|S_i| = C = O_{\epsilon,\delta,\gamma}(1)$.*
4. *The number of sets in $\mathcal{S}$ is $N/2$.*

Proof. We first consider an expander sampler from Lemma 1.1.2 with parameters $(\epsilon, \delta/2, N)$. This is an expander graph G over N nodes, with the set family $\mathcal{ES} = (S_v)_{v \in [N]}$, where S_v = set of neighbors of v in G. From the proof given in the sampler survey [15], one can check that if one takes the second eigenvalue λ to be small enough, $\leq \text{poly}(\epsilon, \delta)$, then the following holds:

For any string $x \in \{0,1\}^N$,

$$\Pr_{S \sim \mathcal{ES}}[|\overline{(x|_S)} - \overline{x}| > \epsilon] \leq \delta/2. \tag{1.1}$$

Now let us see, how to achieve property (2). We use the Expander Mixing Lemma, and show (proof deferred to later in this section) that if λ is small enough ($\leq \text{poly}(\gamma)$) then the following holds: For all $\eta < (1-\gamma)/2$, for any string $x \in \{0,1\}^N$, where $\overline{x} \geq 1 - \eta$, we get that,

$$\Pr_{S \sim \mathcal{ES}}[\overline{(x|_S)} < \gamma] \leq \eta/4. \tag{1.2}$$

Taking the second eigenvalue λ less than the minimum required in both proofs above, we get that both the above statements hold, for some $\lambda = O_{\epsilon,\delta,\gamma}(1)$. Note that the degree of an expander, which is also the sample complexity, is $\text{poly}(1/\lambda) = O_{\epsilon,\delta,\gamma}(1) = C$, hence property (3) holds.

To get property (4), we arbitrarily take $N/2$ of the samples and define this as the set family $\mathcal{S}$ given by the sampler. This hurts the probabilities in equations 1.1 and 1.2 by a factor of at most 2 and hence we get properties (1), (2) for the new set family.

□

1.1.1 Organization

In Chapter 2 we study deterministic reductions which imply inclusion of PCP classes with imperfect completeness in PCP classes with perfect completeness. In Chapter 3 we study similar randomized reductions. In Chapter 4 we prove the equivalence of Gap-ETH with and without perfect completeness.

Chapter 2

PCPs without perfect completeness

In this chapter we prove that PCPs with imperfect completeness can be converted to ones with perfect completeness with a mild blow-up in queries.

2.0.1 Reductions with minimal Query Blow-up

We first show a reduction that preserves the randomness complexity while losing an additive factor in the queries.

Reminder of Theorem 1.0.1 *For all constants $c, s \in (0,1), s < c$, there exists a constant $s' \in (0,1)$, such that for all integer-valued functions $r(\cdot), q(\cdot)$, the following is true:*

$$PCP_{c,s}[r,q] \subseteq PCP_{1,s'}[r, q + O_{s,c}(r)].$$

Furthermore if the original proof size was n, then the final proof size will be $n + O(2^r)$.

For notational simplicity we will prove that:

$$\mathrm{PCP}_{9/10,6/10}[r,q] \subseteq \mathrm{PCP}_{1,9/10}[r, q + O(r)],$$

with proof size $n + O(2^r)$. All constants that follow are universal constants, although in full generality, they only depend on c, s that we have fixed to $(9/10, 6/10)$.

The rest of this section is devoted to the proof of this theorem. The main idea

here is to build a "robust circuit" of small depth, using threshold gates of small fan-in, over the proof oracle of the original PCP. We then ask the new prover to provide the original proof and along with that, also ask for what each gate in the circuit evaluates to, when provided the original clause evaluations as input. As discussed earlier, the circuit boosts the fraction of ones in every layer, for inputs x that satisfy $\overline{x} \geq 9/10$, while maintaining the fraction of ones for inputs that satisfy $\overline{x} \leq 7/10$. We need to do this boosting step by step so that the fan-in does not blow up, and also need to use threshold gates that take "random" subsets of inputs from the previous layer, so that the ones in the input get distributed across all the gates. We get rid of the random subsets, by using any standard sampler over the gates of the previous layer.

Let us now describe the circuit more formally. Later we will give a way to get complete PCPs from incomplete ones using this circuit.

Description of Circuit $\Gamma_m(\cdot)$:

- The circuit has $d = \log m$ layers, $L_1, \ldots, L_d$, with layer i composed of $w_i = m/2^i$ gates denoted by $L_{i1}, \ldots, L_{iw_i}$. The zeroth layer L_0 is the m inputs to the circuit.
- Every gate $L_{(i+1)j}$ is a threshold gate $\mathrm{Thr}_{0.8}$. Let the set family given by the sampler from Theorem 1.1.1 on w_i nodes with parameters $\mathcal{S}(1/10, 6/10, 8/10, w_i) = (S_{(i+1)j})_{j=1}^{w_{i+1}}$. Let $L_{(i+1)j} = \mathrm{Thr}_{0.8}(L_i|_{S_{(i+1)j}})$. By property 3 of expander sampler fan-in $= |S_{(i+1)j}| = O(1)$.

We now use this circuit to give our main reduction.

Proof of Theorem 1.0.1. Let $L \subseteq \{0,1\}^*$ be a language in $\mathrm{PCP}_{9/10,6/10}[r, q]$ via the proof system $\mathcal{P} = (\Pi, Q)$, where Π and Q denote the proof and the set of queries. We can now use the equivalence between MAX q-CSP(c, s) and PCPs, to get a set of clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ of width q, for $m = 2^r$, such that $L \leq$ MAX q-$\mathcal{C}(9/10, 6/10)$. (When $y \in L$, then there exists an assignment x, such that 9/10-fraction of the clauses when evaluated on x output 1, whereas when $y \notin L$, for every assignment x, at most 6/10 of the clauses evaluate to 1.)

To prove the theorem, we will give a new proof system $\mathcal{P}' = (\Pi', Q')$ for L, that has perfect completeness and soundness equal to 9/10. We will transform $\mathcal{P}$ using the circuit $\Gamma_m(\cdot)$ described above, to get $\mathcal{P}'$. We consider the circuit $\Gamma_m(C_1(\Pi), \ldots, C_m(\Pi))$ and ask the new prover to give one bit for every gate of the circuit. More precisely, we ask the new prover to give bits of Π (interpreted as an assignment $x \in \{0,1\}^n$ for the MAX q-CSP: $\mathcal{C}$) and in addition gives bits for every layer in the circuit Γ_m:

$$\ell_i = \{\ell_{i1}, \ldots, \ell_{iw_i}\}, \forall i \in \{0, 1, \ldots, d\}.$$

These bits are supposed to correspond to a correct evaluation of the circuit Γ_m when given $(C_1(x), \ldots, C_m(x))$ $(\Pi = x)$ as input. That is, ideally the prover should give us, $\ell_{0j} = C_j(x), \forall j \in [m]$ and $\ell_{(i+1)j} = L_{(i+1)j}(\ell_i), \forall i \in [d], j \in w_i$, where $L_{(i+1)j}(\ell_i)$ denotes the gate $L_{(i+1)j}$ evaluated on the output bit vector ℓ_i of the previous layer. We probabilistically test this using a new set of queries Q', described below.

Verifier Checks (Q'): For notational simplicity in describing the queries of the new verifier, we will do the following. For each layer i (that has $m/2^i$ gates), consider 2^i copies of the set of gates L_i, and let this new set be denoted by $L'_{i1}, \ldots, L'_{im}$ with corresponding proof bits by $\ell'_i = \{\ell'_{i1}, \ldots, \ell'_{im}\}$ and each gate having its set of inputs $(S'_{i1}, \ldots, S'_{im})$. Note that this duplication of bits/gates is only for description of the queries, and the prover will only give $m/2^i$ bits for every layer i.

Intuitively, we will check whether every gate is correct with respect to its immediate inputs (from the layer below it) and whether the final gate (on the topmost layer) evaluates to 1. To do so, the verifier tosses $\log m$ random coins and on random string $j \in [m]$, it checks whether the following is true:

$$Q'_j := (C_j(x) \stackrel{?}{=} \ell'_{0j}) \wedge (L'_{1j}(\ell_0) \stackrel{?}{=} \ell'_{1j}) \ldots \wedge (L'_{dj}(\ell_{d-1}) \stackrel{?}{=} \ell'_{dj}) \wedge \ell'_{dj},$$

where the clause $(L'_{ij}(\ell_{i-1}) \stackrel{?}{=} \ell'_{ij})$ outputs 1 iff $(L'_{ij}(\ell_{i-1})$ equals $\ell'_{ij})$. As explained earlier, each of the clauses, checks whether the gate L'_{ij} is correct, with respect to its input layer $\ell_{(i-1)}$. Notice here that each check Q_j, checks one gate in every layer

and furthermore these checks are uniform across a layer, i.e. every gate in a layer is checked with the same probability.

To perform the check above, we query the proof bits $\ell_{i-1}|_{S'_{ij}}$, making a constant number of queries, since the fanin of every gate is a fixed constant, i.e. L_{ij} has fanin $|S'_{ij}| = O(1)$. We then evaluate the threshold gate L_{ij} on these bits and take the $\wedge$ across the layers. The check $(C_j(x) \stackrel{?}{=} \ell'_{0j})$ needs to query q queries to x, hence the total number of queried proof bits is $q + O(\log m) = q + O(r)$. Further note that the randomness complexity of the verifier remains the same as before i.e. $= r = \log m$.

We now prove the completeness and soundness of the protocol $\mathcal{P}'$.

Completeness: If the original proof system $\mathcal{P}$ had completeness 9/10, then there exists a proof $\Pi = x$ which satisfies 9/10 of the clauses $\mathcal{C}$. The new prover can give us the bit vectors, x and in addition the evaluations of the circuit $\Gamma(C_1(x), \ldots, C_m(x))$, i.e. $x, \ell_0 := (C_j(x))_{j=1}^m$ and $\ell_i := (L_{ij}(\ell_{i-1}))_{j=1}^m$.

In Lemma 2.0.1, we prove that, $\overline{\ell_i} \geq 1 - \frac{2^{-i}}{10}$. Since $d = \log m$ and the number of gates on level d is $O(1)$, we get that the fraction of 1s in z_d is $\geq 1 - 1/m$, which gets rounded to 1, since there is only one gate in the topmost layer. Since every query Q'_j checks the consistency of a set of gates and if the bit $\ell_{dj} = 1$, we get completeness equals 1.

Soundness: If the original proof system $\mathcal{P}$ had soundness 6/10, then for all proofs Π that the prover might give, Π satisfies $\leq 6/10$ of the clauses $\mathcal{C}$. Let $\Pi' = (x, \ell_0, \ldots, \ell_d)$ be the proof provided by the new prover.

Let $z_0 := (C_j(x))_{j=1}^m$ and $z_{i+1} := (L_{(i+1)j}(\ell_i))_{i=1}^{w_i}$ be the true local evaluations. Note here that, z_{i+1} is the evaluation bits of layer L_{i+1} evaluated on the bits that the prover provides in the previous layer, ℓ_i. By the soundness of $\mathcal{P}$ we get that x satisfies at most 6/10 of $\mathcal{C}$ which means that $\overline{z_0} \leq 6/10$.

Now we have two cases:

1. The prover provided the bit vectors ℓ_i such that they agree with the true eval-

uations z_i in most places, i.e.

$$\forall i, \Pr_{j\sim[w_i]}[\ell_{ij} \neq z_{ij}] \leq 1/10.$$

Hence we have that $\overline{\ell_0} \leq \overline{z_0} + 1/10 \leq 7/10$. Lemma 2.0.2 gives us that for $\overline{\ell_i} \leq 7/10$, $\overline{L_{i+1}(\ell_i)} \leq 6/10$ and therefore $\overline{z_{i+1}} \leq 6/10$. Hence we get that by induction, for all i, $\overline{z_i} \leq 6/10$ and $\overline{\ell_i} \leq 7/10$, and more importantly $\overline{\ell_d} \leq 7/10$. Recall that our verifier checks are uniform over the every layer, and since $\ell_{dj} = 1$ is required for verifier's j^{th} check, Q_j to succeed, we get that soundness is $\leq 7/10$.

2. There exists a layer $i \in \{0, \ldots, d\}$ such that:

$$\Pr_{j\sim[w_i]}[\ell_{ij} \neq z_{ij}] > 1/10.$$

Since z_{ij}'s are the correct evaluations, the above implies that, the prover's proof will fail the local checks in 1/10-fraction of the gates of layer i. Since the verifier checks are uniform over the gates of every layer, (i.e. they check the gate of each layer with the same probability), the verifier checks the incorrect gates with probability at least 1/10. Hence the soundness in this case is $\leq 9/10$.

Note that one of these cases has to occur, hence the overall soundness is the maximum of the two cases, i.e. $\leq 9/10$.

Proof Length: Every layer L_i has width $m/2^i$. Thus the total number of gates in the circuit is $m + m/2 + \ldots = O(m) = O(2^r)$. Since Π' consists of the original proof appended with the circuit evaluations, the proof length is $n + O(2^r)$. □

We now complete the proofs of completeness and soundness in Theorem 1.0.1.

Lemma 2.0.1 (Completeness). *Let $y_0 \in \{0,1\}^m$ be such that $\overline{y_0} \geq 9/10$. Let $y_i \in \{0,1\}^{w_i}$ denote the output string of layer i, when $\mathcal{C}$ is evaluated with the zeroth layer set to y_0. Then we have that for all i, y_i satisfies $\overline{y_i} \geq 1 - \frac{2^{-i}}{10}$.*

Proof. We will prove the lemma by induction on i. Note that the base case $i = 0$, holds trivially. Now consider the $(i+1)^{th}$ layer of the circuit and the gates $L_{(i+1)j}$ that take as input the set $S_{(i+1)j}$ corresponding to the expander sampler on w_i bits. By the induction hypothesis we have that y_i is such that $\overline{y_i} \geq 1 - \frac{2^{-i}}{10}$. By the expander sampler property 2 with parameters $(1/10, 6/10, 8/10, w_i)$ we get that,

$$\Pr_{j\sim[w_{i+1}]}[L_{(i+1)j}(y_i) = \mathrm{Thr}(y_i|_{S_{(i+1)j}}) = 0] \leq \Pr_{j\sim[w_{i+1}]}[\overline{(y_i|_{S_{(i+1)j}})} < 0.8] \leq \left(\frac{1}{2}\right)\left(\frac{2^{-i}}{10}\right).$$

Which directly implies

$$\Pr_{j\sim[w_{i+1}]}[L_{(i+1)j}(y_i) = 1] \geq 1 - \frac{2^{-i-1}}{10} \Leftrightarrow \overline{y_{i+1}} \geq 1 - \frac{2^{-i-1}}{10}$$

which completes the induction. □

Lemma 2.0.2 (Soundness). *Let $y_i \in \{0,1\}^{w_i}$ denote an instantiation of the output gates of layer i with $\overline{y_i} \leq 7/10$. Let $y_{i+1} = L_{i+1}(y_i)$ denote the output of layer $i+1$ when evaluated on the string y_i. Then we have that y_{i+1} satisfies $\overline{y_{i+1}} \leq 6/10$.*

Proof. Recall that in the circuit, the gate $L_{(i+1)j}$ took as input the set $S_{(i+1)j}$ corresponding to the sampler on w_i bits. By the expander sampler property 1, with parameters $(1/10, 6/10, 8/10, w_i)$ we get that, for any string $y_i \in \{0,1\}^{w_i}$ with $\overline{y_i} \leq 7/10$:

$$\Pr_{j\sim[w_{i+1}]}[|\overline{(y_i|_{S_{(i+1)j}})} - 7/10| > 1/10] \leq \Pr_{j\sim[w_{i+1}]}[L_{(i+1)j}(y_i) = \mathrm{Thr}(y_i|_{S_{(i+1)j}}) = 1] \leq 6/10$$

which directly implies $\overline{y_{i+1}} \leq 6/10$ completing the proof.

□

Theorem 1.0.1 implies the following transformation from linear sized PCPs with imperfect completeness to linear sized PCPs with perfect completeness.

Corollary 2.0.1. *If $NTIME[O(n)] \subseteq PCP_{9/10,6/10}[\log n + O(1), q]$ then $NTIME[O(n)] \subseteq PCP_{1,9/10}[\log n + O(1), q + O(\log n)]$.*

Chapter 3

Randomized reductions between PCPs

In this chapter we prove that PCPs with imperfect completeness can be reduced using randomness to ones with perfect completeness with a lesser blow-up in queries compared to Chapter 2. We construct a circuit similar to the one in the previous section, but this time we use a randomized circuit to get better parameters and show that our reduction works with high probability.

3.0.1 Randomized Reductions with minimal Query Blow-up

Reminder of Theorem 1.0.2 *For all constants $c, s \in (0,1), s < c$, there exists a constant $s' \in (0,1)$, such that for all integer-valued functions $r(\cdot), q(\cdot)$, the following is true:*

$$PCP_{c,s}[r,q] \leq_R PCP_{1,s'}[r, q + O_{s,c}(\log r)].$$

Furthermore if the original proof size was n, then the final proof size will be $n + O(2^r)$.

For notational simplicity we will prove that:

$$\mathrm{PCP}_{9/10,6/10}[r,q] \leq_R \mathrm{PCP}_{1,9/10}[r, q + O(\log r)],$$

with proof size $n + O(2^r)$. All constants that follow are universal constants, although in full generality, they only depend on c, s that we have fixed to $(9/10, 6/10)$.

This immediately implies the following corollary using the query reduction[1] result by Dinur [10],

Corollary 3.0.1. *If there exists a $T(n)$ time algorithm for MAX 3SAT$(1, 1-\delta)$ for all $\delta > 0$ then there exists a $T(n(\log\log n)^{O(1)})$ time randomized algorithm for MAX 3SAT$(1-\epsilon, 1-\gamma)$ for all $\epsilon, \gamma, 0 < \epsilon < \gamma$.*

The rest of this section is devoted to the proof of theorem 1.0.2. The main idea as in Theorem 1.0.1 is to build a "robust circuit" of small depth, using threshold gates of small fan-in, over the proof oracle of the original PCP. We then ask the new prover to provide the original proof and along with that, also ask for what each gate in the circuit evaluates to, when provided the original clause evaluations as input. As discussed earlier, the circuit boosts the fraction of ones in every layer, for inputs x that satisfy $\overline{x} \geq 9/10$, while maintaining the fraction of ones for inputs that satisfy $\overline{x} \leq 7/10$. We need to do this boosting step by step so that the fan-in does not blow up, and also need to use threshold gates that take random subsets of inputs from the previous layer, so that the ones in the input get distributed across all the gates.

Let us now describe the circuit more formally. Later we will give a way to get complete PCPs from incomplete ones using this circuit.

Description of Circuit $\Gamma_m(\cdot)$:

- The circuit has $d = \log\log m$ layers, $L_1, \ldots, L_d$, with layer i composed of $w_i = m/2^i$ gates denoted by $L_{i1}, \ldots, L_{iw_i}$. The zeroth layer L_0 is the m inputs to the circuit.
- Every gate L_{ij} is a threshold gate $\mathrm{Thr}_{0.8}$. A gate L_{ij} takes as inputs a random set of f gates from the previous layer L_{i-1}, i.e. we pick a uniformly

[1]This result reduces queries to a constant but blows-up the proof size.

random set S_{ij} of size f, (sampled with replacement) from $[m/2^{i-1}]$ and connect gate L_{ij} with gates $L_{(i-1)k}, \forall k \in S_{ij}$.

We now use this circuit to give our main reduction.

Proof of Theorem 1.0.2. Let $L \subseteq \{0,1\}^*$ be a language in $\mathrm{PCP}_{9/10,6/10}[r,q]$ via the proof system $\mathcal{P} = (\Pi, Q)$, where Π and Q denote the proof and the set of queries. We can now use the equivalence between MAX q-CSP(c,s) and PCPs to get a set of clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$ of width q, for $m = 2^r$, such that $L \leq$ MAX q-$\mathcal{C}(9/10, 6/10)$. (When $y \in L$, then there exists an assignment x, such that 9/10-fraction of the clauses when evaluated on x output 1, whereas when $y \notin L$, for every assignment x, at most 6/10 of the clauses evaluate to 1.)

To prove the theorem, we will give a new proof system $\mathcal{P}' = (\Pi', Q')$ for L, that has perfect completeness and soundness equal to 9/10. We will transform $\mathcal{P}$ using the circuit $\Gamma_m(\cdot)$ described above, to get $\mathcal{P}'$. We consider the circuit $\Gamma_m(C_1(\Pi), \ldots, C_m(\Pi))$ and ask the new prover to give one bit for every gate of the circuit. More precisely, we ask the new prover to give bits of Π (interpreted as an assignment $x \in \{0,1\}^n$ for the MAX q-CSP: $\mathcal{C}$) and in addition gives bits for every layer in the circuit Γ_m:

$$\ell_i = \{\ell_{i1}, \ldots, \ell_{iw_i}\}, \forall i \in \{0, 1, \ldots, d\}.$$

These bits are supposed to correspond to a correct evaluation of the circuit Γ_m when given $(C_1(x), \ldots, C_m(x))$ $(\Pi = x)$ as input. That is, ideally the prover should give us, $\ell_{0j} = C_j(x), \forall j \in [m]$ and $\ell_{(i+1)j} = L_{(i+1)j}(\ell_i), \forall i \in [d], j \in w_i$, where $L_{(i+1)j}(\ell_i)$ denotes the gate $L_{(i+1)j}$ evaluated on the output bit vector ℓ_i of the previous layer. We probabilistically test this using a new set of queries Q', described below.

Verifier Checks (Q'): For notational simplicity in describing the queries of the new verifier, we will do the following. For each layer i (that has $m/2^i$ gates), consider 2^i copies of the set of gates L_i, and let this new set be denoted by $L'_{i1}, \ldots, L'_{im}$ with corresponding proof bits by $\ell'_i = \{\ell'_{i1}, \ldots, \ell'_{im}\}$ and each gate having its set of inputs $(S'_{i1}, \ldots, S'_{im})$. Note that this duplication of bits/gates is only for description of the

queries, and the prover will only give $m/2^i$ bits for every layer i.

Intuitively, we will check whether every gate is correct with respect to its immediate inputs (from the layer below it) and whether the final gate (on the topmost layer) evaluates to 1. To do so, the verifier tosses $\log m$ random coins and on random string $j \in [m]$, it checks whether the following is true:

$$Q'_j := (C_j(x) \stackrel{?}{=} \ell'_{0j}) \wedge (L'_{1j}(\ell_0) \stackrel{?}{=} \ell'_{1j}) \ldots \wedge (L'_{dj}(\ell_{d-1}) \stackrel{?}{=} \ell'_{dj}) \wedge \ell'_{dj},$$

where the clause $(L'_{ij}(\ell_{i-1}) \stackrel{?}{=} \ell'_{ij})$ outputs 1 iff $(L'_{ij}(\ell_{i-1})$ equals $\ell'_{ij})$. As explained earlier, each of the clauses, checks whether the gate L'_{ij} is correct, with respect to its input layer $\ell_{(i-1)}$. Notice here that each check Q_j, checks one gate in every layer and furthermore these checks are uniform across a layer, i.e. every gate in a layer is checked with the same probability.

To perform the check above, we query the proof bits $\ell_{i-1}|_{S'_{ij}}$, making a constant number of queries, since the fanin of every gate is a fixed constant, i.e. L_{ij} has fanin $|S'_{ij}| = O(1)$. We then evaluate the threshold gate L_{ij} on these bits and take the $\wedge$ across the layers. The check $(C_j(x) \stackrel{?}{=} \ell'_{0j})$ needs to query q queries to x, hence the total number of queried proof bits is $q+O(\log\log m) = q+O(\log r)$. Further note that the randomness complexity of the verifier remains the same as before, $= r = \log m$.

We now prove the completeness and soundness of the protocol $\mathcal{P}'$. Since the reduction is randomized, this boils down to proving that, 1) Completeness: given a Max q-CSP that was c-satisfiable, with high probability it gets mapped to a Max q'-CSP that is perfectly satisfiable and 2) Soundness: given a Max q-CSP that was at most s-satisfiable, with high probability it gets mapped to a Max q'-CSP that is at most s'-satisfiable.

Completeness: If the original proof system $\mathcal{P}$ had completeness 9/10, then there exists a proof $\Pi = x$ which satisfies 9/10 of the clauses $\mathcal{C}$. The new prover can give us the bit vectors, x and in addition the evaluations of the circuit $\Gamma(x)$, i.e. $x, \ell_1 := (C_j(x))_{j=1}^m$ and $\ell_i := (L_{ij}(\ell_{i-1}))_{j=1}^m$. In Lemma 3.0.1, we prove that with

probability $\geq 1 - 1/m^{1/4}$, $\overline{\ell_d} = 1$. Since every query Q'_j checks the consistency of a set of gates and if the bit $\ell_{dj} = 1$ we get that with probability $1 - 1/m^{1/4} = 1 - 2^{-\Omega(r)}$, completeness equals 1.

Soundness: We will call a circuit $\Gamma_m(\mathcal{C})$ "good" if the following property holds:

For all layers i, $\forall \ell_i \in \{0,1\}^{w_i}$ such that $\overline{\ell_i} \leq 7/10$, the circuit is such that $\overline{L_{i+1}(\ell_i)} \leq 6/10$. (Recall that $L_{i+1}(z)$ denotes the output of layer L_{i+1} when evaluated on the string z.)

Lemma 3.0.1 gives us that,

$$\Pr[\forall \ell_i \text{ with } \overline{\ell_i} \leq 7/10, \overline{L_{i+1}(\ell_i)} \leq 6/10] \geq 1 - 2^{-m/2^i}$$

Taking a union bound over the layers of the circuit, we get that,

$$\begin{aligned}\Pr[\Gamma_m(\mathcal{C}) \text{ is good }] &= \Pr[\forall i, \forall \ell_i \text{ with } \overline{\ell_i} \leq 7/10, \overline{L_{i+1}(\ell_i)} \leq 6/10] \\ &\geq 1 - (\log\log m) 2^{-m/2^d} \\ &\geq 1 - 2^{-\sqrt{m}} = 1 - 2^{-\Omega(r)}\end{aligned}$$

We will now show that if the randomized circuit $\Gamma_m(\mathcal{C})$ is good then the new PCP is sound. Since the circuit is good with high probability, showing this is enough to complete the randomized reduction claimed in Theorem 1.0.2.

From now on, we will assume that the circuit is good. If the original proof system $\mathcal{P}$ had soundness 6/10, then for all proofs Π that the prover might give, Π satisfies $\leq 6/10$ of the clauses $\mathcal{C}$. Let $\Pi' = (x, \ell_0, \ldots, \ell_d)$ be the proof provided by the new prover.

Let $z_0 := (C_j(x))_{j=1}^m$ and $z_{i+1} := (L_{(i+1)j}(\ell_i))_{i=1}^{w_i}$ be the true local evaluations. Note here that, z_{i+1} is the evaluation bits of layer L_{i+1} evaluated on the bits that the prover provides in the previous layer, ℓ_i. By the soundness of $\mathcal{P}$ we get that x satisfies at most 6/10 of $\mathcal{C}$ which means that $\overline{z_0} \leq 6/10$.

Now we have two cases:

1. The prover provided the bit vectors ℓ_i such that they agree with the true evaluations z_i in most places, i.e.

$$\forall i, \Pr_{j\sim[w_i]}[\ell_{ij} \neq z_{ij}] \leq 1/10.$$

Hence we have that $\overline{\ell_0} \leq \overline{z_0} + 1/10 \leq 7/10$. Lemma 3.0.2 gives us that for $\overline{\ell_i} \leq 7/10$, $\overline{L_{i+1}(\ell_i)} \leq 6/10$ and therefore $\overline{z_{i+1}} \leq 6/10$. Hence we get that by induction, for all i, $\overline{z_i} \leq 6/10$ and $\overline{\ell_i} \leq 7/10$, and more importantly $\overline{\ell_d} \leq 7/10$. Recall that our verifier checks are uniform over the every layer, and since $\ell_{dj} = 1$ is required for verifier's j^{th} check, Q_j to succeed, we get that soundness is $\leq 7/10$.

2. There exists a layer $i \in \{0, \ldots, d\}$ such that:

$$\Pr_{j\sim[w_i]}[\ell_{ij} \neq z_{ij}] > 1/10.$$

Since z_{ij}'s are the correct evaluations, the above implies that, the prover's proof will fail the local checks in 1/10-fraction of the gates of layer i. Since the verifier checks are uniform over the gates of every layer, (i.e. they check the gate of each layer with the same probability), the verifier checks the incorrect gates with probability at least 1/10. Hence the soundness in this case is $\leq 9/10$.

Note that one of these cases has to occur, hence the overall soundness is the maximum of the two cases, i.e. $\leq 9/10$.

Proof Length: Every layer L_i has width $m/2^i$. Thus the total number of gates in the circuit is $m + m/2 + \ldots = O(m) = O(2^r)$. Since Π' consists of the original proof appended with the circuit evaluations, the proof length is $n + O(2^r)$. □

We now complete the proofs of completeness and soundness claims used in the proof of Theorem 1.0.2.

Lemma 3.0.1 (Completeness). *Let $y_0 \in \{0,1\}^m$ be such that $\overline{y_0} \geq 9/10$. Let $y_i \in \{0,1\}^{w_i}$ denote the output string of layer i, when $\mathcal{C}$ is evaluated on y_0. Then we have*

that with probability $\geq 1 - 1/m^{1/4}$ *for all* i, y_i *satisfies* $\overline{y_i} \geq 1 - \left(\frac{1}{10}\right)^{2^i}$ *and hence* $\overline{y_d} = 1$.

Notice here that the completeness $1 - \eta$ increases to $1 - (\eta)^2$ at each step, instead of $1 - \eta$ to $1 - \eta/2$, like it did in the previous section. This increase allows us to use only $\log \log m$ layers to get perfect completeness, albeit with high probability. Now we prove the lemma.

Proof. The theorem statement is implied by proving that with probability $\geq 1 - 1/m^{1/4}$ for all i, $(1 - \overline{y_{i+1}}) \leq (1 - \overline{y_i})^2$.
We will prove the lemma by induction on i. Note that the base case $i = 0$, holds trivially. Now consider the $(i+1)^{th}$ layer of the circuit and the gates $L_{(i+1)j}$ that take as input the set $S_{(i+1)j}$ corresponding to random sets of size f from $[w_i]$.

By induction $\overline{y_i} \geq 1 - \left(\frac{1}{10}\right)^{2^i} \geq .9$ and $.2/(1 - \overline{y_i}) \geq 2$. For a fixed $L_{(i+1)j}$, by the Chernoff bound 2 on number of 0's we get,

$$\begin{aligned}
\Pr[L_{(i+1)j}(y_i) = \mathrm{Thr}_{.8}(y_i|_{S_{(i+1)j}}) = 0] &= \Pr[\mathrm{Thr}_{.2}((1 - y_i)|_{S_{(i+1)j}}) = 1] \\
&\leq exp(\Omega((.2/(1 - \overline{y_i})) \log((.2/(1 - \overline{y_i}))(1 - \overline{y_i})f)) \\
&= exp(\Omega(\log((.2/(1 - \overline{y_i}))f)) \\
&\leq (1 - \overline{y_i})^3
\end{aligned}$$

for some large enough constant f.

Chernoff bound 1 over all the gates in L_{i+1} for the number of 0's gives gives that,

$$\begin{aligned}
\Pr[(1 - \overline{y_{i+1}}) \geq (1 - \overline{y_i})^2] &< exp(-\Omega(((1 - \overline{y_i})^2/(1 - \overline{y_i})^3)(1 - \overline{y_i})^3(m/2^i))) \\
&= exp(-\Omega((1 - \overline{y_i})^2(m/2^i)))
\end{aligned}$$

As we have $\log \log m$ layers $m/2^i > m/\log m$, hence

$$\Pr[(1 - \overline{y_{i+1}}) \geq (1 - \overline{y_i})^2] < exp(-\Omega((1 - \overline{y_i})^2(m/\log m)))$$

A Markov bound over all the gates in L_{i+1} for the number of 0's gives gives that,

$$\Pr[(1-\overline{y_{i+1}}) \geq (1-\overline{y_i})^2] \leq (1-\overline{y_i}).$$

Together these bounds imply

$$\Pr[(1-\overline{y_{i+1}}) \geq (1-\overline{y_i})^2] \leq \log^2(m)/\sqrt{m}.$$

Union bound over all $\log m$ layers gives probability $\leq (\log\log m)\log^2(m)/\sqrt{m} \leq 1/m^{1/4}$. Hence with probability $\geq 1-1/m^{1/4}$, $\overline{y_d} \geq 1-\left(\frac{1}{10}\right)^{2^{\log\log(m)}} \geq 1-1/m^2$. As there are $\leq m$ gates at last layer this means with probability $\geq 1-1/m^{1/4}$, $\overline{y_d}=1$. □

Lemma 3.0.2 (Soundness). *Let $y_i \in \{0,1\}^{w_i}$ denote an instantiation of the output gates of layer i with $\overline{y_i} \leq 7/10$. Let $L_{i+1}(y_i)$ denote the output of layer $i+1$ when evaluated on the string y_i. Then with probability $1-2^{-m/2^i}$, for all y_i, $L_{i+1}(y_i)$ satisfies $\overline{L_{i+1}(y_i)} \leq 6/10$. Formally,*

$$\Pr[\forall y_i \text{ with } \overline{y_i} \leq 7/10, \overline{L_{i+1}(y_i)} \leq 6/10] \geq 1-2^{-m/2^i}.$$

Proof. Fix a gate $L_{(i+1)j}$. Given that the fraction of 1s in layer i is at most 7/10, using Chernoff bound 1, we get that,

$$\Pr[\mathrm{Thr}_{0.8}(S_{(i+1)j})=1] = \Pr[\frac{1}{f}\sum_{k\in S_{(i+1)j}} \ell_{ik} - 7/10 > 8/10 = 7/10(1+1/7)] \quad (3.1)$$

$$< exp(-(1/7)^2(7f/10)/3) \quad (3.2)$$

$$< 1/f, \quad (3.3)$$

for large enough constant f.

By applying Chernoff bound 2 (assuming large enough f) over all gates $L_{(i+1)j}$, we get that,

$$\Pr[\overline{L_{i+1}(y_i)} > 6/10] < exp(-\Omega((6f/10)(\log(6f/10))(m/(f2^{i+1}))))$$

$$= exp(-\Omega((\log(6f/10))(m/2^i)))$$
$$< exp(-2m/2^i).$$

for large enough constant f.

Hence a union bound over all possible $2^{m/2^i}$ strings y_i gives that,

$$\Pr[\exists y_i, \overline{L_{i+1}(y_i)} > 6/10] \leq 2^{m/2^i} e^{-2m/2^i} < 2^{-m/2^i}.$$

□

Chapter 4

Gap-ETH without perfect completeness

In this section we study the relation between the time complexities of approximating MAX 3-SAT with and without perfect completeness. We show that the Gap-ETH conjecture with and without perfect completeness is equivalent by giving an algorithm for approximating MAX 3-SAT without perfect completeness, that uses an algorithm for approximating MAX 3-SAT with perfect completeness as a subroutine and runs in $2^{o(n)}$-time iff the latter does so.

4.0.1 Reduction for two-sided error randomized algorithms

We first prove that Gap-ETH conjecture with and without perfect completeness are equivalent for randomized algorithms with two-sided error. We show this by showing that the Gap-ETH conjecture without perfect completeness is false if the one with perfect completeness is false.

Theorem 4.0.1. *If there exists a randomized (two-sided error) $2^{o(n)}$ time algorithm for MAX 3SAT$(1, 1-\gamma)$, for all constants $\gamma > 0$, then there exists a randomized (two-sided error) $2^{o(n)}$ time algorithm for MAX 3SAT$(s(1+\epsilon), s)$ for all constants s, ϵ.*

We will prove the above in its contrapositive form. Suppose there is a $2^{o(n)}$ algo-

rithm for MAX 3SAT$(1, 1-\gamma)$ for all constants γ. We will then show that for all constants ϵ, s, δ, there exists an algorithm for MAX 3SAT$(s(1+\epsilon), s)$ with running time less than $2^{\delta n}$. Our randomized algorithm for MAX 3SAT$(s(1+\epsilon), s)$ will use the algorithm for satisfiable MAX 3SAT$(1, 1-\gamma)$ as a subroutine and run in time less than $2^{\delta n}$. The following lemma forms the crux of the proof.

Lemma 4.0.1. *For all constant $s, \epsilon > 0$ there exists a large enough constant k, such that there exists a randomized reduction from MAX 3-SAT$(s(1+\epsilon), s)$ on n variables and $O(n)$ clause to MAX $O(k)$-CSP$(1, 1/2)$ on n variables and $O(n)$ variables, such that:*

- *If the original instance was a NO instance, then the reduction produces an instance which is not a NO instance with probability $\leq 2^{-n}$.*
- *If the original instance was a YES instance, then the reduction produces a YES instance with probability $\geq 2^{-n/k}$.*

Proof. Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a MAX 3SAT$(s(1+\epsilon), s)$ instance. We can assume without loss of generality, that $\epsilon < 1/100$, since the result for a smaller gap implies the result for a larger gap.

Let $(S_i)_{i=1}^n$ be a set family in which every set S_i is a random set chosen by sampling with replacement from $[m]$. Consider new clauses B_i such that each clause is a threshold gate: $B_i = \mathrm{Thr}_{s(1+\epsilon/2)}(\mathcal{C}|_{S_i})$, where $\mathcal{C}$ denotes the vector $(C_1(x), \ldots, C_m(x))$.

Our final CSP will be over the original set of variables x_i. We will have a clause for each of the n B_i's. For the i^{th} clause B_i, we will find the values of all C_j such that $j \in S_i$ and then verify that their threshold value is $\geq s(1+\epsilon/2)$. Our query size is $3k$ as we find values for variables in k clauses each of them on 3 variables.

Soundness For a NO instance and a fixed assignment x the fraction of clauses satisfied by x is $\leq s$. By the Chernoff bound 1, the probability that clause B_i is satisfied is $\leq exp(-(\varepsilon/2)^2 sk/3)$. The probability that at least half of the B_i's are satisfied is at most, $\binom{n}{n/2} exp(-\Omega(\varepsilon^2 skn))$ which is less than $exp(-2n)$, when k is taken to be a large enough constant, depending only on ϵ, s. Therefore by a union

bound, the probability that there exists an assignment x that satisfies at least half of the B_i's is $\leq 2^n exp(-2n) \leq 2^{-n}$.

Completeness For a YES instance there exists an assignment x that satisfies $\geq s(1+\epsilon)$-fraction of the clauses. By the Chernoff bound 1 the probability that the clause B_i is unsatisfied is $\leq exp(-(\varepsilon/3)^2 sk/2)$ as $\epsilon < 1/100$. Therefore the probability that all the B_i's are satisfied is $(1 - exp(-\Omega(\varepsilon^2 sk)))^n \geq (1 - 10/k)^n$ which is $\geq 2^{-n/k}$ when k is a large enough constant. □

Proof of Theorem 4.0.1. The randomized algorithm for solving MAX 3SAT$(s(1+\epsilon), s)$ is as follows: We will run the reduction from Lemma 4.0.1 $2^{n/k}n^2$ times and then convert the resulting MAX $O(k)$-CSP$(1, 1/2)$ instances to MAX 3SAT$(1, 1-\gamma)$ instances on $O(k2^k n)$ variables and $O(k2^k n)$ clauses where γ is a constant depending on k. Then we run the $2^{o(n)}$ algorithm for MAX 3SAT$(1, 1-\gamma)$ (still $2^{o(n)}$ as k, γ are constants) on the resulting instances and if any of the outputs is YES we will also output YES.

By repeating the algorithm for MAX 3SAT$(1, 1-\gamma)$ poly(n) times we can assume the the probability that the algorithm errs is $\leq 2^{-n^2}$, hence we will assume this wlog.

$$\begin{aligned}
\Pr[\text{Error on a YES instance}] &\leq \Pr[\text{Algorithm errs on one of the produced instances}] \\
&\quad + \Pr[\text{None of the } 2^{n/k}n^2 \text{ runs produce a YES instance}] \\
&\leq 2^{-n^2} 2^{n/k} n^2 + (1 - 2^{-n/k})^{2^{n/k}n^2} \\
&\leq 2^{-n/2}
\end{aligned}$$

$$\begin{aligned}
\Pr[\text{Error on a NO instance}] &\leq \Pr[\text{Algorithm errs on one of the produced instances}] \\
&\quad + \Pr[\text{On one of the } 2^{n/k}n^2 \text{ runs the output was not a NO instance}] \\
&\leq 2^{-n^2} 2^{n/k} n^2 + 2^{n/k} n^2 2^{-n}
\end{aligned}$$

$$\leq 2^{-n/2}$$

Total running time $= 2^{n/k}n^2 2^{o(n)}$ which for large enough k is $< 2^{\delta n}$. This gives us the desired contradiction. □

4.0.2 Reduction for one-sided error randomized algorithms with no false positives

We now prove that in fact Gap-ETH conjecture with and without perfect completeness are equivalent for randomized algorithms with no false positives.

Reminder of Theorem 1.0.3 *If there exists a randomized (with no false positives) $2^{o(n)}$ time algorithm for MAX 3SAT$(1, 1-\gamma)$ for all constant $\gamma > 0$ then there exists a randomized(with no false positives) $2^{o(n)}$ time algorithm for MAX 3SAT$(s(1+\epsilon), s)$ for all constants $s, \epsilon > 0$.*

As in the proof of Theorem 4.0.1, we will prove the above in its contrapositive form. Suppose there is a $2^{o(n)}$ algorithm (with no false positives) for MAX 3SAT$(1, 1-\gamma)$ for all constants γ. We will then show that for all constants ϵ, s, δ, there exists an algorithm (with no false positives) for MAX 3SAT$(s(1+\epsilon), s)$ with running time less than $2^{\delta n}$. Our randomized algorithm for MAX 3SAT$(s(1+\epsilon), s)$ will use the algorithm for satisfiable MAX 3SAT$(1, 1-\gamma)$ as a subroutine and run in time less than $2^{\delta n}$. The following lemma which is a stronger version of Lemma 4.0.1 with only one-sided error forms the crux of the proof.

Lemma 4.0.2. *For all constant $s, \epsilon > 0$ there exists a large enough constant k, such that there exists a randomized reduction from MAX 3SAT$(s(1+\epsilon), s)$ to MAX $O(k)$-CSP$(1, 1/2)$ with $O(n)$ variables such that:*

- *If the original instance was NO, then the reduction produces a NO instance.*
- *If the original instance was YES, then the reduction produces a YES instance with probability $\geq 2^{-n/k}$..*

We will first prove Theorem 1.0.3 using the lemma given above. This proof is similar to the proof of Theorem 4.0.1.

Proof of Theorem 1.0.3. The randomized algorithm for solving MAX 3SAT$(s(1+\epsilon), s)$ is as follows: We will run the reduction from Lemma 4.0.2 $2^{n/k}n^2$ times and then convert the resulting MAX $O(k)$-CSP$(1, 1/2)$ instances to a MAX 3SAT$(1, 1-\gamma)$ instances on $O(k2^k n)$ variables and $O(k2^k n)$ clauses where γ is a constant depending on k. Then we run the $2^{o(n)}$ algorithm for MAX 3SAT$(1, 1-\gamma)$ algorithm (still $2^{o(n)}$ as k, γ are constants) on them and if any of the outputs is YES we will also output YES.

By repeating the algorithm poly(n) times we can assume the the probability that the algorithm errs (one sided error) is $\leq 2^{-n^2}$, hence we will assume this wlog.

For a NO original instance we will always output a NO instance.

$$\begin{aligned}
\Pr[\text{Error on a YES instance}] &\leq \Pr[\text{Algorithm errs on one of the produced instances}] \\
&\quad + \Pr[\text{None of the } 2^{n/k}n^2 \text{ runs produce a YES instance}] \\
&\leq 2^{-n^2}2^{n/k}n^2 + (1-2^{-n/k})^{2^{n/k}n^2)} \\
&\leq 2^{-n}
\end{aligned}$$

Total running time $= 2^{n/k}n^2 2^{o(n)}$ which for large enough k is $< 2^{\delta n}$. This gives us the desired contradiction. □

Proof of Lemma 4.0.2. Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a MAX 3SAT$(s(1+\epsilon), s)$. We can assume without loss of generality, that $\epsilon < 1/100$, since the result for a smaller gap implies the result for a larger gap. Let the number of clauses in $\mathcal{C}$ be $m = \rho n$. We will sample with repetition from $\mathcal{C}$ to produce a list L of clauses of size $t\rho n$, for some $t > 1$. We call a list *balanced* if:

1. For every set $S \subseteq \mathcal{C}, |S| = s\rho n$, total number of occurrences of clauses from S occurs in L is at most $s(1+\epsilon/3)t\rho n$ times.

2. For every set $S \subseteq \mathcal{C}, |S| = s(1+\epsilon)\rho n$, total number of occurrences of clauses from S occurs in L at least $s(1+2\epsilon/3)t\rho n$ times.

It is easy to see that the probability of sampling an unbalanced list is:

$$\begin{aligned}\Pr[\text{L is unbalanced}] &\leq \binom{\rho n}{s\rho n} exp(-\epsilon^2 st\rho n/9) + \binom{\rho n}{s(1+\epsilon)\rho n} exp(-\epsilon^2 s(1+\epsilon)t\rho n/16) \\ &\leq exp(-10\rho n),\end{aligned}$$

when t is a large enough constant and since $\epsilon < 1/100$.

Let $\mathcal{C}'$ be the CSP given by the set of clauses in L (repeated clauses might be present in $\mathcal{C}$). If L is balanced then the soundness of $\mathcal{C}'$ is $\leq s(1+\epsilon/3)$ and completeness is $\geq s(1+2\epsilon/3)$. If our list is not balanced we will reject it and output any NO instance. This can be done in polynomial time as we can check the condition 1 by finding a set of clauses of size $s\rho n$ which occurs the most and checking that it occurs at most $s(1+\epsilon/3)t\rho n$ in L. We can similarly check condition 2.

Let $(S_i)_{i=1}^{|L|}$ be the set family given to us by the expander sampler from Lemma 1.1.2 with parameters $\mathcal{ES}((100/(s^2\epsilon^2 k)), (s\epsilon/6), |L|)$. Consider new clauses B_i such that each clause is a threshold gate, i.e. $B_i = \text{Thr}_{s(1+\epsilon/2)}(\mathcal{C}'|_{S_i})$, where $\mathcal{C}'$ denotes the vector of clauses of L. By the sampler property $|S_i| \leq k$ and the number of B_i's is equal to $|L| = t\rho n$.

Our final CSP will be given by the set of clauses B_i. For the i^{th} clause will find the values of all C'_j such that $j \in S_i$ and then verify that their threshold value is $\geq s(1+\epsilon/2)$. Our query size is $3k$ as we find values for variables in k clauses each of them on 3 variables.

Soundness If L is balanced, in the NO case the soundness is $\leq s(1+\epsilon/3)$. Then we get that,

$$\begin{aligned}\Pr_i[B_i(\mathcal{C}') = 1] &= \Pr\left[\frac{1}{|S_i|}\sum_{j\in S_i} \mathcal{C}'_j \geq s(1+\epsilon/2)\right] \\ &= \Pr\left[\frac{1}{|S_i|}\sum_{j\in S_i} \mathcal{C}'_j - s(1+\epsilon/3) \geq s(\epsilon/6)\right] \\ &\leq 100/(s^2\epsilon^2 k)\end{aligned}$$

where the last inequality follows from the properties of expander sampler in Lemma 1.1.2. Now for large enough k we get $100/(s^2\epsilon^2 k) \leq 1/2$ hence starting from all NO instances gives us NO instances.

If L is unbalanced we always output a NO instance.

Completeness By the property of the expander sampler in Lemma 1.1.2, the number of query sets that intersect with some query set S_i are at most $O(k^2)$ for large enough k. As the original instance was a YES instance there exists an $x = x_c$ which satisfies $s(1+\epsilon)$ fraction of the clauses. As each clause of list L is a random clause from the original set of clauses, the probability that any specific B_i evaluates to 1 is $\geq 1 - exp(-\Omega(-\epsilon^2 ks))$ by the Chernoff bound for assignment x_c.

As each clause of list L is a random clause from the original set of clauses, we get that the random variables (randomness from choosing the list L, after fixing the sets S_i) B_i and B_j are independent, if two query sets S_i and S_j do not intersect. As calculated above, the probability that any clause fails is $\leq exp(-\Omega(-\epsilon^2 ks))$. For large enough constants k,

$$e \cdot O(k^2) exp(-\Omega(-\epsilon^2 ks)) < 1,$$

which allows us to apply the Lovász local lemma as given in Lemma 1.1.1. This gives us that,

$$\Pr_L[\wedge B_i(\mathcal{C}') = 1] \geq (1 - 1/k^3)^{t\rho n} \geq 2^{-n/(2k)}.$$

Taking into account the case where L is unbalanced, the probability of outputting a YES instance is $\geq 2^{-n/(2k)} - 2^{-10\rho n} \geq 2^{-n/k}$ for large enough k. □

Chapter 5

Conclusions

5.1 Summary

In this thesis, we studied the role of completeness in the construction of PCPs. We showed that PCPs with imperfect completeness can be transformed to perfect completeness, with a mild loss in query complexity. We also studied the Gap-ETH conjecture, and showed that $2^{o(n)}$ algorithms for approximate Max 3-SAT with imperfect completeness exist, if and only if, such algorithms exist for approximate Max 3-SAT with perfect completeness.

5.2 Future Directions

The reduction in Section 2 is not useful to get perfect completeness for PCPs, while preserving their query complexity and losing some factor in the randomness complexity. When the construction is composed with query reduction, it only gives us that $\text{PCP}_{c,s}[\log n + O(1), O(1)] \subseteq \text{PCP}_{1,s'}[\log n + O(\log \log n), O(1)]$, which is anyway the blow-up incurred in state of the art PCPs for $NTIME[O(n)]$ [10]. Hence we pose the following problem:

Open Problem 1. *Let $c, s, s' \in (0, 1)$ with $s < c$ be constants. Then is it true that,*

$$PCP_{c,s}[\log n + O(1), O(1)] \subseteq PCP_{1,s'}(\log n + o(\log \log n), O(1))?$$

It would also be interesting to see whether such blackbox reductions can be applied to get hardness of CSPs with perfect completeness, since for many CSPs such as Max k-CSP and 2-1 games, optimal hardness is only known with imperfect completeness.

Bibliography

[1] Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of SVP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 228–238, 2018.

[2] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[3] Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Hardness amplification for entangled games via anchoring. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 303–316, 2017.

[4] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, pcps, and nonapproximability-towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[5] Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate pcps for circuit-sat with sublinear query complexity. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 320–329, 2013.

[6] Eli Ben-Sasson and Madhu Sudan. Short pcps with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

[7] Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized intractability of even set and shortest vector problem from gap-eth. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 17:1–17:15, 2018.

[8] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 743–754, 2017.

[9] Siu On Chan. Approximation resistance from pairwise independent subgroups. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 447–456, 2013.

[10] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[11] Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016.

[12] Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 376–389, 2018.

[13] Irit Dinur and Pasin Manurangsi. Eth-hardness of approximating 2-csps and directed steiner network. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 36:1–36:20, 2018.

[14] David Gillman. A chernoff bound for random walks on expander graphs. *SIAM J. Comput.*, 27(4):1203–1220, 1998.

[15] Oded Goldreich. A sample of samplers - A computational perspective on sampling (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 4(20), 1997.

[16] Venkatesan Guruswami and Luca Trevisan. The complexity of making unique choices: Approximating 1-in- k SAT. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 99–110, 2005.

[17] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[18] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[19] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 767–775, 2002.

[20] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 78:1–78:15, 2017.

[21] Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011.

[22] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.