Efficient Holographic Proofs

by

Alexander Craig Russell

B.A., Cornell University (1991)
S.M., Massachusetts Institute of Technology (1993)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1996
[June 1996]

Author .....
Department of Mathematics
May 3, 1996

Certified by .....
Michael Sipser
Professor, Department of Mathematics
Thesis Supervisor

Certified by .....
Richard Stanley
Chairman, Applied Mathematics Committee

Accepted by .....
David Vogan
Chairman, Departmental Committee on Graduate Students

Efficient Holographic Proofs

by

Alexander Craig Russell

Submitted to the Department of Mathematics
on May 3, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The formalization of *interaction* in a complexity-theoretic setting in 1985 began a new and wildly successful era in computational complexity theory. This study has culminated in the development of the theory of *holographic* or *probabilistically checkable* proofs. These are proofs so robust that they may be verified (with high probability) by consideration of a randomly selected *fragment* of the proof text. Such machinery has profoundly recharacterized many of the classical complexity classes. Although these recharacterizations have offered new structural insight into the classes involved, the great triumph of the theory has been an unexpected connection with approximation algorithms. It provides the first general scheme for concluding that even approximating certain problems is difficult.

This thesis constructs probabilistically checkable proof systems for NP more efficient than any previously known and discusses the ramifications of these new systems in the realm of approximation algorithms. Perhaps the most significant advance is the construction of the first $O(\log n)$-communication probabilistically checkable proof system for NP which achieves *any* constant error with a *fixed* number (4) of questions. These advances are applied to improve results on the hardness of approximating SET COVER, MAX CLIQUE, CHROMATIC NUMBER, MAX 3SAT, and QUARTIC PROGRAMMING.

Finally, we explore the relativized behavior of this new machinery. We show that the recharacterizations of PSPACE in this framework are lamentably unstable. These results reinforce the belief that the framework of relativization is incompatible with these new techniques. They can also be seen as a further assault of the (already battered) random oracle hypothesis.

Thesis Supervisor: Michael Sipser
Title: Professor, Department of Mathematics

Acknowledgements

I am grateful to Mike Sipser, my research advisor, for his careful guidance and kind support. His potent intuition and perspective have been unsurpassed guides.

I am grateful also to those other members of the mathematical community with whom I have worked: Mihir Bellare, Joan Feigenbaum, Rosario Gennaro, Michel Goemans, Shafi Goldwasser, Oded Goldreich, Mauricio Karchmer, Marcos Kiwi, Michael Klugerman, Ravi Kumar, Dror Lapidot, Carsten Lund, Leonard Schulman, Silvio Micali, Rina Panigrahy, Dan Spielman, and Shang-Hua Teng.

I am grateful for the steadfast support and encouragement of my family, who I find applauding with bright eyes at every turn. I thank Sally for her bountiful cheer, support, and diversion.

Risking duplication of an entire generation of thesis acknowledgement pages, I humbly thank Be Hubbard for ever inspiring familial comfort in the halls of the MIT Theory Group. I also offer great thanks to David Jones whose patience is exceeded only by his LaTeXwizardry– he is the source of any positive aspects of this document's typesetting. Thanks also to Joanne Talbot for sharing her coffee with me.

I thank Mauricio Karchmer and Adi Pandya for taking my side in the everlasting aesthetics debate.

Finally and most of all, I thank Ravi Sundaram. Much of the richness, mathematical and otherwise, of my graduate career has come about because of my friendship with him. The fates have done me an unthinkable favor in his introduction.

Contents

Chapter 1

Introduction

The field of complexity theory is that of the classification of computational problems according to various resource bounds. It has matured during the last two decades to the point where there is general agreement about both the basic definitions and the selection of central open problems. It is a field that can pride itself at having evolved, rather quickly, from an ad-hoc collection of facts to a well-structured mathematical discipline with life independent of its engineering origins. Indeed, the theory has given rise to widely applicable classification machinery that has driven much research in computer science. The celebrated theory of NP-*completeness* is perhaps the most compelling example of this phenomenon. It shows an enormous collection of well-studied decision problems to be "equally difficult," and offers convenient machinery for concluding that new problems fall into this class (see [49], for example). These problems are the NP-*complete* problems. The existence of efficient algorithms for this class of problems is the most significant open question of the field (see [84] for a graceful discussion of the current status of this problem).

The recent development of *probabilistically checkable* (or *holographic*) proofs provides the first general framework for concluding that even *approximating* the solution to natural decision problems is difficult. For example, applying these tools one can show that approximating the size of the largest clique in a graph to within $n^{1-\epsilon}$ is NP-hard (for all $\epsilon > 0$). The strength of these "hardness of approximation" results is intimately related to certain qualities of the probabilistically checkable proofs invoked to reach the conclusions. The primary topic of this thesis is the fabrication of an holographic proof system more favorable than previously known systems in terms of the qualities related to such lower bounds. The resulting lower bounds are then catalogued and, in the case of SET COVER, explored in some detail.

In a general sense, the theory of probabilistically checkable proofs offers an alternative method for expressing mathematical proofs. Traditionally, proofs have been written as a sequence of propositions, each following from its predecessor(s) by some logical rule. In order to check the validity of such a proof, one must (of course) read through the entire sequence of propositions– a single faulty step invalidates the proof. It may seem silly to complain about this state of affairs since it is *a priori* unclear what more we can ask for.

Indeed, if one wants to be absolutely certain of the truth of a proposition, there is no substitute for the above process. Suppose, however, that we are in the business of verifying proofs and don't mind making a mistake with some small probability. As we shall see, this added flexibility will allow us to check proofs with astounding alacrity. It is in fact possible to "check" a proof by examining an asymptotically vanishing fraction of the proof text.

To realize this goal, one has to abandon the traditional method for writing down proofs and adopt some far more robust dialect for proof expression. Imagine a "proof checker" who, upon receipt of a voluminous proof, decides to base his decision (about the correctness of the text) on, say, 30 randomly selected pages of the proof. If he discovers some blatant error on one of these pages, of course he knows that the proof is incorrect. In the case that he uncovers no errors, he assumes that the proof is correct. If the proof is written in the traditional manner described above, such a proof checker is (very) likely to accept as correct a proof having a single error buried in the text. We shall see that there is a language for expressing proofs so that errors, if they occur at all, occur almost everywhere. This will allow the proof checker above to confidently determine the validity of a proof based on a small, randomly chosen collection of fragments. In fact, one application of the machinery we shall build in Chapter 3 shows that there is a language for expressing proofs so that consideration of 29 randomly chosen *symbols* of the proof is enough to expose any error with probability $\frac{1}{2}$.

In addition to such structural revelations, the theory offers the first general scheme for concluding that even *approximating* the solution to many NP-complete problems is difficult [38]. Consider, as an example, the following problem: given a universe $\mathcal{S}$ and n subsets $S_1, \ldots, S_n \subset \mathcal{S}$, compute the cardinality of the smallest cover of $\mathcal{S}$ by these sets S_i. This is called the SET COVER problem and was one of the first problems shown to be NP-complete. It is suspected that there is no exact polynomial time algorithm for any NP-complete problem. One is naturally led to ask if there is a polynomial time algorithm that offers good *approximate* solutions to the SET COVER problem. This was resolved (positively) in [65] by the construction of a polynomial time algorithm which, for any set system, produces a cover with at most $1 + \ln n$ times the number of sets used by the optimal cover (where $n = |\mathcal{S}|$). It was unclear if this factor could be improved or if there was a boundary here. In §4.1 we shall argue that there is indeed a $\Theta(\log n)$ boundary– we show that if one could approximate SET COVER to within $\frac{1}{8} \ln n$ then the problems in NP could actually be solved deterministically in time $n^{O(\log \log n)}$.

This holographic proof framework has been applied to several complexity classes other than NP including PSPACE [31, 33], NEXP [9, 7, ...], and the classes in the polynomial hierarchy [68]. In each of these cases it has offered a new definition for the given complexity class. In the last section of this thesis we shall argue that some of the these new characterizations differ in a very essential way from the existing definitions. Specifically, we show that in most *relativized* worlds, these new characterizations diverge wildly from their traditional counterparts.

The technique of *relativization* is a method for exaggerating differences between various computational models. It has been used in the past to argue that proving various complexity-theoretic conjectures, like P $\neq$

NP, was likely to be hard. The technique consists of a general method for providing "extra information" to a family of Turing machines, say, in order that their computing capacity become easier to understand. In Chapter 5 we show that some of the new characterizations discussed above are very unstable with respect to relativization. This can be seen both as an attack of the technique of relativization and some indication that these new models of computation are unlike any ever before studied.

Although the theory of holographic proofs has marvelous structure and application, it is young and is only beginning to acquire a standard terminology. It will be necessary for some generous agent to produce a survey of the area which recasts in some uniform and compelling language all of the ultimately relevant machinery. At the time of this writing, the field is still evolving so quickly that such a synthesis is unlikely to be durable. In any case, I do not take up this task– foundational material which can be conveniently encapsulated shall be cited without proof or guilty feelings. Having allowed myself this freedom, I promise the reader as much unity as I can muster. My intention is that the manuscript make no demands on the reader other than ownership of some mathematical sophistication and familiarity with the rudiments of discrete mathematics and theoretical computer science.

This thesis is an expansion of two published articles: "Efficient Probabilistically Checkable Proofs and Applications to Approximation" [18], which is joint work with Mihir Bellare, Shafi Goldwasser, and Carsten Lund, and "The Relativized Relationship between Probabilistically Checkable Debate Systems, IP, and PSPACE" [81], which is joint work with Ravi Sundaram. In Chapter 2, we give a brief history of the development of holographic proof systems and define the basic concepts involved. Chapter 3 develops our improved holographic machinery which is applied, in Chapter 4, to offer improved lower bounds for several approximation algorithms. Chapter 5 is a discussion of the relativized behavior of holographic proof theoretic characterizations of known complexity classes.

Chapter 2

Notations, Definitions and History

Σ shall always denote a finite set (often $\{0,1\}$) which we shall call an *alphabet*. Σ^k shall denote the normal cross product set and members of this set we shall write as *strings* $w_1 w_2 \ldots w_k$ rather than ordered tuples. We reserve the symbol Λ to denote the "empty string" and define $\Sigma^0 = \{\Lambda\}$. A *language over* Σ is any subset L of $\Sigma^* \stackrel{\text{def}}{=} \cup_{k \geq 0} \Sigma^k$. When Σ is understood we shall omit "over Σ" from the previous definition. 1^k denotes the concatenation of k 1s.

We shall adopt the standard notation and foundational material of [63] when Turing machines enter our discussion. The language accepted by a Turing machine M is denoted $L(M)$. An oracle Turing machine M operating with oracles $O_1, \ldots, O_k$ is denoted $M^{O_1,\cdots,O_k}$ and the language so accepted $L(M^{O_1,\cdots,O_k})$. When a Turing machine M computes a function, the result of M running on x is written $M[x]$. The result of a *probabilistic* Turing machine with coin tosses R on input x shall be written $M[x;R]$.

2.1 History

The independent formalization of "interaction" in a complexity-theoretic setting by Babai [6] and Goldwasser, Micali, and Rackoff [52] in 1985 began a new and wildly successful era in theoretical computer science. Although this machinery attracts attention for its intrinsic elegance and expressive power in the realm of cryptography, the great triumph of the theory is an unexpected connection with approximation algorithms discovered by Feige, Goldwasser, Lovász, Safra, and Szegedy [38].

Goldwasser, Micali and Rackoff [52] defined *interactive Turing machines* and the class IP, as follows:

Definition 2.1 *An interactive Turing machine V is a Turing machine with a read-only input tape, a work tape, a random tape, a read-only communication tape, called the* response *tape, and a write-only communication tape, called the* query *tape.*

Appropriately coupling interactive Turing machines with functions $P : \Sigma^* \rightarrow \Sigma^*$ defines a language class in the following way:

Definition 2.2 *Let V be an interactive Turing machine, and $P : \Sigma^* \to \Sigma^*$ a function. Supplying V with an input x, a random string r and the function F naturally gives rise to a computation path by following the usual rules for Turing machine transition except in the case where V writes some special symbol "?" on it's query tape. The contents of the response tape are then replaced with $P(Q)$ where Q is the entire contents of the query tape, and computation proceeds as usual. Notice that the function P is supplied with the entire history of "queries" by V when it "responds." We shall restrict our attention to computation of this sort that is polynomially bounded in $|x|$ (over all possible runs of V). If V accepts with input x, random string r, and P, we write $(V \leftrightarrow F)[x; r] = accept$. Define the class* IP *to consist of those languages L for which there exists a polynomial time interactive Turing machine V, called a "verifier," so that*

- *(Completeness[1]) $x \in L \Rightarrow \exists P, \Pr_R[(V \leftrightarrow P)[x; R] = accept] = 1$, and*
- *(Soundness) $x \notin L \Rightarrow \forall P, \Pr_R[(V \leftrightarrow P)[x; R] = accept] \leq \frac{1}{3}$.*

Clearly, NP $\subset$ IP. It was immediately shown that IP is (probably) more expressive than NP: Babai and Szemerédi [11] placed some matrix group problems into IP and, more significantly[2], Goldreich, Micali, and Wigderson [50] demonstrated that

$$\overline{\text{GRAPH ISOMORPHISM}} = \{(G_1, G_2) \mid G_1 \text{ is not isomorphic to } G_2\} \in \text{IP}.$$

It was known that IP $\subset$ PSPACE [77], and community sentiment was that IP was "just above" NP. This was reinforced when Fortnow and Sipser [47] gave an oracle for which $\text{coNP}^O \not\subset \text{IP}^O$ (see Chapter 5 for a discussion of this topic).

Then, in a breakthrough which pioneered the algebraic methods now dominating the field, Lund, Fortnow, Karloff, and Nisan [73] demonstrated that IP contains the entire polynomial hierarchy. Shamir [82] then completely characterized IP, showing the following:

Theorem 2.1 IP = PSPACE.

It is interesting to note that if one restricts the verifier to a constant number of rounds, the resulting class, called AM (see [10, 53]), actually lies inside Σ_2^P and so is (probably) much weaker.

Study of the above equipment was strongly motivated by the alluring discovery of *zero-knowledge* proofs [52, 50]. (Roughly, a zero-knowledge proof of a proposition is a protocol carried out by a "verifier" with a "prover" (as above) which provides overwhelming evidence for the truth of the proposition without revealing *any other information.*) Such a protocol, for example, was discovered for $\overline{\text{GRAPH ISOMORPHISM}}$. It was natural to ask if such protocols could be given for the languages of NP. When it was found by Fortnow that this was unlikely [43, 25], various relaxations of this goal were studied. Goldreich, Micali, and Wigderson

[1] *The completeness condition here is not the one found in [52]. They just ask that* $\Pr[accept] \geq \frac{2}{3}$. *That this stronger condition yields the same class is shown in [21]*

[2] This result, coupled with [53] and [25], shows that GRAPH ISOMORPHISM is not NP-complete unless the polynomial hierarchy collapses to Σ_2^P.

[51] showed that by assuming the existence of secure encryption functions, such proofs can be constructed for all of NP. In 1988, Ben-Or, Goldwasser, Kilian, and Wigderson [21] demonstrated that strengthening the machinery by adding a "second prover" resulted in a framework where zero-knowledge proofs existed for NP without unproven assumptions. They defined the following language class:

Definition 2.3 *A k-interactive Turing machine V is a Turing machine with a read-only input tape, a work tape, a random tape, k read-only communication tapes, called the* response *tapes, and k write-only communication tapes, called the* query *tapes.*

Definition 2.4 *Define computation by V, a k-interactive Turing machine, with functions $P_1, \ldots, P_k : \Sigma^* \to \Sigma^*$ following definition 2.2. Then define* MIP_k *to consist of those languages L for which there exists a k-interactive Turing machine V so that*

- *(Completeness)* $x \in L \Rightarrow \exists P_1, \ldots, P_k, \Pr_R[(V \leftrightarrow P_1, \ldots, P_k)[x; R] = \mathit{accept}] = 1$,
- *(Soundness)* $x \notin L \Rightarrow \forall P_1, \ldots, P_k \Pr_R[(V \leftrightarrow P_1 \ldots, P_k)[x; R] = \mathit{accept}] \leq \frac{1}{3}$.

The upper bound on the accept probability in case $x \notin L$ is called the error *of the system.*

It is worth noting that the same class is defined if the provers are allowed to be probabilistic. That is, for a verifier V admitting error at most ϵ (on some $x \notin L$) and any distribution on tuples of functions $(P_\omega^1, \ldots, P_\omega^k)$ over a probability space Ω, we have that

$$\Pr_{R,\omega}[(V \leftrightarrow P_\omega^1, \ldots, P_\omega^k)[x; R] = \text{accept}] =$$
$$\operatorname*{Exp}_{\omega}[\Pr_R[(V \leftrightarrow P_\omega^1, \ldots, P_\omega^k)[x; R] = \text{accept}]] \leq \epsilon.$$

As in the case of IP, MIP_k was found to be remarkably expressive. Babai, Fortnow, and Lund [8] completely classified MIP_k as non-deterministic exponential time:

Theorem 2.2 *For $k \geq 2$,* $\mathrm{MIP}_k = \mathrm{MIP}_2 = \mathrm{NEXP}$.

Notice that by increasing the number of rounds, sequential repetition (that is, independently repeating the protocol many times with the same provers) naturally reduces the error exponentially, so that NEXP can be recognized by two-prover systems with $2^{-\mathrm{poly}(n)}$ error. Extrapolating from IP, one might expect that restricting these multi-prover systems to a single round would be crippling. On the contrary, NEXP can be realized in the single round case. There is, in fact, a two-prover single round proof system for NEXP which attains exponentially small error [71, 42]. Since much of the remainder of this thesis shall concern itself with such systems, we define them in some detail:

Definition 2.5 *A* 1-round verifier V *is a pair of (randomized) polynomial time Turing machines $V = (Q, C)$. Intuitively, Q shall be responsible for generating the queries to the functions involved and shall be called the*

"querier." C is responsible for evaluating the answers received and shall be called the "checker." Formally, Q computes a function $Q : \Sigma^* \times \{0,1\}^* \to (\Sigma^*)^k$ *and C computes a function* $C : \Sigma^* \times \{0,1\}^* \times (\Sigma^*)^k \to \{accept, reject\}$. *Both Q and C must run in time polynomial in their first input. Given a collection of functions* $f_1, \ldots, f_k : \Sigma^* \to \Sigma^*$ *and a random string R, V is said to* accept x on R *if* $C(x, R, f_1(q_1), \ldots, f_k(q_k)) = accept$ *where* $(q_1, \ldots, q_k) = Q(x, R)$. *If* $(q_1, \ldots, q_k)$ *is not compatible with the signatures of the functions, C is defined to reject. This value* $C(x, R, f_1(a_1), \ldots, f_k(a_k))$ *is also written* $(V \leftrightarrow f_1, \ldots, f_k)[x; R]$.

We then define a parameterized multi-prover interactive proof class.

Definition 2.6 *For functions* $p, r, q, a : \mathbb{N} \to \mathbb{N}$ *and* $\epsilon : \mathbb{N} \to [0,1]$, *define the complexity class* $\mathrm{MIP}[p, r, q, a, \epsilon]$ *to consist of those languages L for which there exists a 1-round verifier* $V = (Q, C)$ *so that, on input x with* $|x| = n$,

- *V is provided* $r(n)$ *random bits,*
- *(Completeness) if* $x \in L$ *then* $\exists O_1, \ldots, O_p : \Sigma^q \to \Sigma^a$,

$$\Pr_R[[V \leftrightarrow O_1, \ldots, O_p][x; R] = accepts] = 1,$$

- *(Soundness) if* $x \notin L$ *then* $\forall O_1, \ldots, O_p : \Sigma^q \to \Sigma^a$,

$$\Pr_R[[V \leftrightarrow O_1, \ldots, O_p][x; R] = accept] \leq \epsilon.$$

Lapidot and Shamir [71], focusing on this one-round scenario, showed that with four provers, one can attain exponentially small error in only one round. Their "parallelization" machinery became a central tool in the theory (see §3.2) and was immediately applied by Feige and Lovász [42] to show that two provers suffice:

Theorem 2.3 $\mathrm{NEXP} \subset \mathrm{MIP}[2, \mathrm{poly}(n), \mathrm{poly}(n), \mathrm{poly}(n), 2^{-n}]$.

Fortnow, Rompel, and Sipser [46] found that the study of these proof systems is often simplified by the consideration of verifiers allowed to interact with *oracles* rather than provers– that is, were allowed multiple access to a non-adaptive partner. They provided transformations which convert between the two models. This framework is expressed in the following definition.

Definition 2.7 *For functions* $p, r, q, a : \mathbb{N} \to \mathbb{N}$ *and* $\epsilon : \mathbb{N} \to [0,1]$, *define the complexity class* $\mathrm{PCP}[p, r, q, a, \epsilon]$ *to consist of those languages L for which there exists a 1-round verifier* $V = (Q, C)$ *so that, on input x with* $|x| = n$,

- *V is provided* $r(n)$ *random bits,*
- *(Completeness) if* $x \in L$ *then* $\exists O : \Sigma^q \to \Sigma^a$, $\Pr_R[[V \leftrightarrow \overbrace{O, \ldots, O}^{p}](x; R) = accept] = 1$,

- *(Soundness) if* $x \notin L$ *then* $\forall O : \Sigma^q \to \Sigma^a$, $\Pr_R[[V \leftrightarrow \overbrace{O, \ldots, O}^{p}](x; R) = accept] \leq \epsilon$.

These two frameworks are closely related and we shall allow the phrase "holographic proof system" to refer to either of them. In both cases, consideration of a small fraction of a fixed "holographic" proof is enough to establish the truth or falsity of a given proposition with high probability.

Notice that attaching to each question the name of the prover to which it is intended, one can give a nearly complexity-preserving simulation of a MIP system by a PCP system:

Lemma 2.1 $\text{MIP}[p, r, q, a, \epsilon] \subset \text{PCP}[p, r, q + \log p, a, \epsilon]$

Fortnow, Rompel, and Sipser [46] gave a natural simulation of PCP systems by MIP systems:

Lemma 2.2 $\text{PCP}[p, q, r, a, \epsilon] \subset \text{MIP}[2, r + \log p, q, pa, 1 - \frac{1-\epsilon}{p}]$.

Once it had been discovered (cf. theorem 2.3) that an exponential-time computation could be (probabilistically) verified with only polynomial communication (and computation), the community immediately set to adapting these techniques to NP. Following [7], [38], and [4], Arora, Lund, Motwani, Sudan, and Szegedy [3] showed the following:

Theorem 2.4 $\text{NP} \subset \text{PCP}[O(1), O(\log n), O(\log n), 1, \frac{1}{2}]$.

Notice that independent repetition the above protocol $k(n)$ times yields a low-error characterization for NP with the following complexity.

Theorem 2.5 *For* $k(n)$ *reasonable*[3], $\text{NP} \subset \text{PCP}[O(k), O(k \log n), O(\log n), 1, 2^{-k}]$.

Motivated both by applications to approximation algorithms (see §2.1.1 below) and aesthetic interest, these low-error characterizations became a central topic. Applying the deterministic amplification machinery of [1, 30, 64], one can easily obtain a low-error characterization of NP without undue cost in randomness:

Theorem 2.6 $\text{NP} \subset \text{PCP}[O(\log n), O(\log n), O(\log n), 1, \frac{1}{n}]$.

Both this system and that of theorem 2.5 above suffer from their utilization of $\Theta(\log n)$ questions. The best known result attaining low error with only a constant number of provers was the following theorem obtained by combining [71, 42] and theorem 2.4.

Theorem 2.7 $\text{NP} \subset \text{MIP}[2, O(\log^3 n), O(\log^3 n), O(\log^3 n), \frac{1}{n}]$.

The primary contribution of this thesis is the construction of a low-error proof system for NP attaining the following complexity.

[3] *We use the word* reasonable *to describe polynomial-time computable functions into* $\mathbb{N}$ *or* $\mathbb{Q}$ *the behavior of which may be naturally determined from the context of their use.*

Theorem 2.8 *Let $k(n) = O(\log n)$ be reasonable, and let $\bar{k}(n) = \max(k(n), \log\log n)$. Then*

$$\text{NP} \subseteq \text{MIP}[4, r, q, a, 2^{-k(n)}],$$

where $r = O(k(n)\log n + k(n)^2\bar{k}(n))$, $q = O(r)$, and $a = O(k(n)^2\bar{k}(n))$.

Notice that for any constant $\epsilon > 0$, there is a choice of $k = O(1)$ so that this yields

$$\text{NP} \subset \text{MIP}[4, O(\log n), O(\log n), O(\log\log n), \epsilon] \subset \text{PCP}[4, O(\log n), O(\log n), O(\log n), \epsilon],$$

the first known proof systems for NP simultaneously attaining *any* constant error and $O(\log n)$ communication complexity with a (fixed) constant number of provers. The effects of such a system can be seen both in the realm of lower bounds for approximation algorithms (cf. theorem 4.3) and in other efficient proof systems (cf. theorem 3.2).

For $k = \Theta(\log n)$ this system duplicates the complexity obtained by theorem 2.7. For smaller values of k, however, it provides an asymptotically superior system. Selection of $k = \Theta(\log\log n)$, for example, shall be used to provide the improved results for the hardness of approximating SET COVER in §4.1.

2.1.1 Lower Bounds for Approximation

As mentioned before, the study of efficient holographic proof systems has been strongly motivated by a flourishing connection with approximation algorithms discovered by Feige, Goldwasser, Lovász, Safra, and Szegedy [38]. A detailed dicussion of this connection in the case of the SET COVER problem appears in §4.1. What follows is a primitive, but illustrative, example of this phenomenon in the case of MAX CLIQUE.

Let us first define what we mean by an approximation algorithm. Formally, an α-approximation algorithm for a maximization problem is a polynomial time algorithm which produces, for any instance π, a value A_π so that $\alpha\,\text{opt}_\pi \leq A_\pi \leq \text{opt}_\pi$. An α-approximation algorithm for a minimization problem is defined analogously. The factor α is often written as a function of the input size.

Let us concentrate on the MAX CLIQUE problem. From theorem 2.5 above, for any constant $\epsilon > 0$ we have that SAT $\in$ NP $\subset$ PCP$[p = O(1), O(\log n), O(\log n), 1, \epsilon]$. Let $V_{\text{SAT}} = (Q, C)$ be a verifier for SAT with these parameters and ϕ a formula. One way to determine if $\phi \in$ SAT, then, is to apply V_{SAT} to ϕ and compute the maximum probability that $V_{\text{SAT}}[\phi]$ accepts (over all polynomial size oracles O). Of course, since there is a gap between the maximum acceptance probability of $V_{\text{SAT}}[\phi]$ depending on whether $\phi \in$ SAT, it is actually enough to *approximate* this value. Now, given $V_{\text{SAT}}[\phi]$, construct the graph $G_\phi = (V, E)$ where

$$V = \{\langle R, \vec{a}\rangle \mid R \in \{0,1\}^r, \vec{a} \in \{0,1\}^p, C(\phi, R, \vec{a}_1, \ldots, \vec{a}_p) \text{ accepts}\} \text{ and}$$
$$E = \{(\langle R, \vec{a}\rangle, \langle R', \vec{a}'\rangle) \mid (Q(\phi, R)_i = Q(\phi, R')_j) \Rightarrow \vec{a}_i = \vec{a}'_j\}$$

so that an edge is placed between two vertices $\langle R, \vec{a} \rangle$ and $\langle R', \vec{a}' \rangle$ when these two answers $\vec{a}$ and $\vec{a}'$ are consistent (don't provide different answers to the same question). A clique in this graph, then, corresponds to a consistent selection of answers for a set of random strings of V_{SAT} (and hence a partial determination of an oracle O). In this case we have that the clique number of G, $\omega(G)$, is directly related to the maximum acceptance probability of V_{SAT}. Indeed, we have that

$$\max_{O} \Pr_{R}[(V \leftrightarrow O)[\phi; R] \text{ accepts}] = \frac{\omega(G)}{2^r},$$

so that approximating MAX CLIQUE to within ϵ is enough to determine if $\phi \in$ SAT.

See §§4.2.2 and 4.3 for more discussion of lower bounds on approximating $\omega(G)$.

Results of this form have been developed for many of the well-studied NP optimization problems[4] including MAX 3SAT, MAX 2SAT, CHROMATIC NUMBER, MAX CUT, MIN VERTEX COVER, and those problems in MAX-SNP. Crescenzi and Kann [34] have compiled an comprehensive list of lower (and upper) bounds for approximation problems.

[4] The NP optimization problems are those optimization problems the decision versions of which are in NP.

Chapter 3

Efficient Multi-Prover Proof Systems

3.1 Algebraic Preliminaries

3.1.1 Computation in Finite Fields

We shall frequently work in GF(2^t), the finite field with 2^t elements. It is often convenient to render this field as a quotient of the polynomial ring GF(2)[x]: fixing an irreducible polynomial $p \in$ GF(2)[x] of degree t, one has that GF(2^t) $\cong$ GF(2)[x]/(p). Such polynomials always exists (see [72], for example). Elements of GF(2^t) are then in bijective correspondence with polynomials of degree $t-1$ and both addition and multiplication may be carried out efficiently. Production of an irreducible polynomial of appropriate degree is then sufficient for computing inside GF(2^t). Shoup [83] has shown that this can be done deterministically in time polynomial in t. (Of course, when $t = O(\log n)$ as it shall be for us, such polynomials can be found by exhaustive search in polynomial time.) See [72] for other background on finite fields.

3.1.2 Polynomials and Codes

A principal component of existing constructions of non-trivial holographic proof systems is computationally manageable large-distance codes. The basic tools for constructing such codes are developed below.

Definition 3.1 *We may naturally associate with an element p of $\mathbb{F}[x_1, \ldots, x_m]^n$ a function $f_p : \mathbb{F}^m \to \mathbb{F}^n$. Such functions comprise the class $\mathfrak{P}(\mathbb{F}^m, \mathbb{F}^n)$ of* polynomial functions. *For a polynomial $p \in \mathbb{F}[x_1, \ldots, x_m]$, we define the degree of the polynomial* deg p *to be the maximum sum of the exponents in any monomial of p. We define the* variable degree, vardeg p *to be the maximum exponent on any variable of p. These are likewise defined for elements of $\mathbb{F}[x_1, \ldots, x_m]^n$ as the maximum of the appropriate quantity over all components. We define these quantities for elements $f \in \mathfrak{P}(\mathbb{F}^m, \mathbb{F}^n)$ as the minimum over all polynomial representatives for f, i.e.* $\deg f = \min\{\deg p \mid f_p = f\}$. *Define* $\mathfrak{P}_d(\mathbb{F}^m, \mathbb{F}^n) = \{f \in \mathfrak{P}(\mathbb{F}^m, \mathbb{F}^n) \mid \deg f \leq d\}$.

Since we shall primarily be dealing with these objects in a computational setting, the distinction between $\mathfrak{P}(\mathbb{F}^m, \mathbb{F}^n)$ and $\mathbb{F}[x_1, \ldots, x_m]^n$ shall be frequently blurred.

Definition 3.2 *Let $\mathbb{E} \subset \mathbb{F}$ be (coherent) fields and $f \in \mathbb{E}[x_1, \ldots, x_k]$. The* $\mathbb{F}$-variety *of f is $V_{\mathbb{F}}(f) = \{\vec{x} \in \mathbb{F}^k \mid f(\vec{x}) = 0\}$. When $\mathbb{E} = \mathbb{F}$, $V_{\mathbb{F}}(f)$ shall be written $V(f)$.*

Lemma 3.1 *Let $\mathbb{F}$ be a finite field and $f \in \mathbb{F}[x_1, \ldots, x_k]$ a non-zero polynomial with* vardeg $f \leq d$. *Then*

$$|V(f)| \leq kd\,|\mathbb{F}|^{k-1}.$$

Proof: The proof proceeds by induction on k. The base case is elementary. Assume the statement for polynomials in $\mathbb{F}[x_1, \ldots, x_{k-1}]$, and let $f \in \mathbb{F}[x_1, \ldots, x_k]$ be non-zero. Then there exists polynomials $f_d, \ldots, f_0 \in \mathbb{F}[x_1, \ldots, x_{k-1}]$ so that $f = x_k^d f_d + \cdots + x_k^0 f_0$. Let

$$Z_h = \bigcap_i V(f_i) = \{\vec{z} \in \mathbb{F}^{k-1} \mid \forall i, f_i(\vec{z}) = 0\}$$

and $Z_d = \overline{Z_h}$. Then $|Z_h| \leq (k-1)d\,|\mathbb{F}|^{k-2}$ by induction and each point in Z_h induces $|\mathbb{F}|$ zeros. We also have that $|Z_d| \leq |\mathbb{F}|^{k-1}$ and each point in Z_d can induce at most d zeros. Hence $|V(f)| \leq d\,|\mathbb{F}|^{k-1} + (k-1)d\,|\mathbb{F}|^{k-1} = kd\,|\mathbb{F}|^{k-1}$, as desired. □

Corollary 3.1 *Let $\mathbb{F}$ be a finite field and $f, g \in \mathbb{F}[x_1, \ldots, x_k]$ distinct polynomials with* vardeg $f - g \leq d$. *Then*

$$|V(f-g)| \leq kd\,|\mathbb{F}|^{k-1}.$$

Corollary 3.2 *Let $\mathbb{F}$ be a finite field and $f, g : \mathbb{F}^k \to \mathbb{F}^l$ distinct polynomial functions with* vardeg $f - g \leq d$. *Then*

$$\left|\{\vec{z} \in \mathbb{F}^k \mid f(\vec{z}) = g(\vec{z})\}\right| \leq kd\,|\mathbb{F}|^{k-1}.$$

The following notion of a polynomial extension shall be central to our study.

Definition 3.3 *Let $\mathbb{E} \subset \mathbb{F}$ be finite fields and $f : \mathbb{E}^n \to \{0,1\}$. A polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$ is an* extension *of f when $\forall \vec{e} \in \mathbb{E}^n$, $p(\vec{e}) = f(\vec{e})$ and* vardeg $p \leq |\mathbb{E}|$. *When $\mathbb{F}$ is understood, we shall let $\hat{f}$ denote a canonical extension of f.*

The primary application for the above definition shall be the creation of certain "codes". For two functions $f, g : X \to Y$ on a finite set X define

$$\Delta(f,g) = \frac{|\{x \in X \mid f(x) \neq g(x)\}|}{|X|}.$$

A subset $\mathcal{C}$ of $\mathbb{F}^n$ is a *code* with *distance* δ if for $c, c' \in \mathcal{C}$, $c \neq c' \Rightarrow \Delta(c, c') \geq \delta$ where c and c' are considered functions from $\{1, \ldots, n\} \to \mathbb{F}$. A particularly convenient manner for describing a code $\mathcal{C}$ is to realize it as

the image of a injective *encoding* function $E : \Sigma^n \hookrightarrow \mathbb{F}^m$. This provides a natural correspondence between objects in Σ^n and their "codewords." Often $\Sigma = \mathbb{F} = \{0,1\}$ and this is an error-correcting code in the natural sense.

For a field $\mathbb{F}$, the space of functions $\mathfrak{F}_{m,n} = \{f : \mathbb{F}^m \to \mathbb{F}^n\}$ is a metric space under the metric $d(f,g) = \Delta(f,g)$. Then corollary 3.2 shows that with respect to Δ the subspace of polynomial functions $\mathfrak{P}(\mathbb{F}^m, \mathbb{F}^n) \subset \mathfrak{F}_{m,n}$ form a large-distance code. Combining such polynomial codes with the simple *robust* code defined below, one can give an effective construction of a codes $E : \{0,1\}^n \hookrightarrow 0,1^m$ with constant distance and only polynomial expansion (that is $m \leq \text{poly } n$).

Definition 3.4 *Let $W = \{0,1\}^n$ and let $\iota : \{0,1\}^W \xrightarrow{\cong} \{0,1\}^{2^n}$ be an isomorphism. Let $P : W \to \{0,1\}^W$ be the function given by $P(w)_{\hat{w}} = \hat{w}^T \cdot w \bmod 2$. Then define $E_R : \{0,1\}^n \hookrightarrow \{0,1\}^{2^n}$ to be the function given by $E_R = \iota \circ P(w)$. $E_R(w)$, then, is a list of the parities of each substring of w. Notice that $w_1 \neq w_2 \Rightarrow \Delta(E_R(w_1), E_R(w_2)) \geq \frac{1}{2}$.*

The robust code produces codewords of exponential size. As promised, we combine the above notions to build a code with constant distance and polynomial size. Codes such as these are used to construct the holographic proof system of theorem 2.4 (see §3.3.1).

Example 3.1 *Fix n and let $\mathbb{B} \subset \mathbb{F}$ be finite fields with $|\mathbb{B}| = \log n$ and $|\mathbb{F}| = \log^2 n$. Let c be a constant large enough that $(\log n)^{\frac{c \log n}{\log \log n}} \geq n$ and define $m \stackrel{def}{=} \frac{c \log n}{\log \log n}$. We may then fix a (structureless) injection $\alpha : \{0, \ldots, n\} \hookrightarrow \mathbb{B}^m$. For $w \in 0,1^n$, we may naturally define $w^{\mathbb{B}} : \mathbb{B}^m \to \{0,1\}$ so that $w^{\mathbb{B}}(\alpha(i)) = w_i$. As defined above, we may form the polynomial extension of this function $w^{\mathbb{F}} : \mathbb{F}^m \to \mathbb{F}$ which has $\text{vardeg } w^{\mathbb{F}} \leq |\mathbb{B}|$. From lemma 3.2, for $v \neq w \in \{0,1\}^n$,*

$$\Delta(w^{\mathbb{F}}, v^{\mathbb{F}}) \geq 1 - \frac{c}{\log \log n} \to 1.$$

Then, define the code

$$\hat{E} : w \mapsto \left(E_R(w^{\mathbb{F}}(x))\right)_{x \in \mathbb{F}^m}$$

where $w^{\mathbb{F}}(x)$ is written as an element of $\{0,1\}^{2 \log \log n}$. Notice that $\hat{E}$ has distance approaching $\frac{1}{2}$ as $n \to \infty$ and $\left|\hat{E}(x)\right| \leq \text{poly}(|x|)$.

3.2 Reducing Randomness

The goal of this Chapter is to produce efficient MIP systems for NP. (Specifically, we shall build the proof system of theorem 3.1.) We begin by producing a low-error system which is desirably efficient in terms of prover multiplicity, randomness, and question length, but requires extravagant answer length. In §3.3 we provide a transformation which rectifies this.

Lemma 3.2 *Let $k = k(n) = O(\log n)$ be reasonable and $b(n) = \max(k(n), \log\log n)$. Then*

$$\mathrm{NP} \subset \mathrm{MIP}[2, O(k\log n), O(k\log n), O(k\log n 2^{b(n)}), 2^{-k}].$$

Proof: This proof is an adaptation of the the algebraic parallelization machinery of [71] in order that it may be combined with the extended base field technique of [7]. Let $L \in \mathrm{NP}$. Applying theorem 2.5, we may place $L \in \mathrm{MIP}_1[p, r, q, a, \epsilon]$, with $p = O(k)$, $r = O(k\log n)$, $q = O(\log n)$, $a = O(1)$, and $\epsilon = 2^{-k-1}$. Let $V = (Q, C)$ be the 1-round verifier for L with these parameters. We construct a new 1-round verifier $\mathbf{P}(V) = (Q', C')$ which possesses the desired parameters. Define $l \stackrel{\mathrm{def}}{=} \lceil \frac{q}{b} \rceil$ and let $\mathbb{B} \subset \mathbb{F}$ be a finite fields with $|\mathbb{B}| = 2^b$ and $|\mathbb{F}| = 2^{4b+3\log p+\log l+9}$.

Q' begins by generating a random string R of length r and simulating Q to derive $Q(x, R) = (q_1, \ldots, q_p)$, the vector of questions that V would ask on this random string. We may consider each q_i as an element of $\mathbb{B}^l \subset \mathbb{F}^l$ (the inclusion $\{0,1\}^q \hookrightarrow \mathbb{B}^l$ is structureless). Q' now chooses $z_1, \ldots, z_p$ independently and uniformly in $\mathbb{F}^l$. For two points $x, y \in \mathbb{F}^l$, let $\ell[x, y] : \mathbb{F} \to \mathbb{F}^l$ be a (canonical) parameterization of the line through x and y. Then define $\ell_i \stackrel{\mathrm{def}}{=} \ell[q_i, z_i]$. The (2-question) result of Q' is $(\langle \ell_1, \ldots, \ell_p \rangle, \langle z_1, \ldots, z_p \rangle)$.

C' expects to receive two replies, the first a vector $(A_1, \ldots, A_p)$ of polynomials, $A_i : \mathbb{F} \to \mathbb{F}^a$, with $\deg A_i \leq l\,|\mathbb{B}|$ and the second a vector $(\zeta_1, \ldots, \zeta_p)$ of elements of $\mathbb{F}^l$. (If the results fail to have this form, C' rejects.) Define $s_i \stackrel{\mathrm{def}}{=} \ell_i^{-1}(q_i)$ and $t_i \stackrel{\mathrm{def}}{=} \ell_i^{-1}(z_i)$. C' accepts if $\forall i, A_i(t_i) = \zeta_i$ and $C(x, r, A_1(s_1), \ldots, A_p(s_p))$ accepts. (Again, if $A_i(s_i) \notin \{0,1\}^a$, C' rejects.)

V' is easily seen to satisfy all of the required parameters excluding, perhaps, the error ϵ:

- V' requires $r + pl(4b + \log\frac{q}{b} + 3\log p + 9) = O(k\log n) + O(k)O(\log n)(1 + \frac{(\log\frac{q}{b} + 3\log p + 9)}{b}) = O(k\log n)$ because $b = \Omega(\log\log n)$,

- the queries produced by Q' are bounded in size by a constant function of the randomness used by Q', and so are $O(k\log n)$,

- the replies expected by C' have size $p(l\,|\mathbb{B}|)(4b + \log\frac{q}{b} + 3\log p + 9)a = O(k\log n)2^b$.

To check completeness, suppose that $x \in L$ so that there exist $P_i : \{0,1\}^q \to \{0,1\}^a$ for $i \in \{1, \ldots, k\}$ which satisfy V with probability 1. Considering P_i as a function from $\mathbb{B}^l \to \{0,1\}^a$, define $\hat{P}_i$ to be an extension of P_i to $\mathbb{F}^l$ so that $\hat{P}_i : \mathbb{F}^l \to \mathbb{F}^a$ and vardeg $P_i \leq |\mathbb{B}| - 1$. Then the two functions $F_1 : \ell_1, \ldots, \ell_p \mapsto \hat{P}_1 \circ \ell_1, \ldots, \hat{P}_p \circ \ell_p$ and $F_2 : z_1, \ldots, z_p \mapsto \hat{P}_1(z_1), \ldots, \hat{P}_p(z_p)$ can be seen to satisfy $\mathbf{P}(V)$ with probability 1, as desired.

To prove soundness, we adapt [71]. Suppose $x \notin L$ and fix two functions

$$F_1 : \mathfrak{P}_1(\mathbb{F}, \mathbb{F}^l)^p \to \mathfrak{P}_{l|\mathbb{B}|}(\mathbb{F}, \mathbb{F}^a)^p, \text{ and}$$
$$F_2 : (\mathbb{F}^l)^p \to (\mathbb{F}^l)^p.$$

One would like, at this point, to demonstrate that frequent acceptance by V induces "near-functionality" on the part of F_1– that is, p functions $f_1, \ldots, f_p : \mathfrak{P}_1(\mathbb{F}, \mathbb{F}^l) \rightarrow \mathfrak{P}_{l|\mathbb{B}|}(\mathbb{F}, \mathbb{F}^a)$ so that F_1 is closely approximated by $f_1 \times \cdots \times f_p$. This would allow us to conclude a (natural) upper bound on the error probability. Unfortunately, it is unclear that maximal strategies have this form and we shall have to settle for something less. We demonstrate that over the possible values of $(z_1, \ldots, z_p)$, F_1 is closely approximated by a *convex combination* such function tuples. That is, we shall demonstrate the existence of a collection of functions $\{(F^1_\alpha, \ldots, F^p_\alpha) \mid \alpha \in (\mathbb{F}^t)^p\}$ so that

$$\Pr_{q,z}\left[F_1(\ell[q_1, z_1], \ldots, \ell[q_p, z_p]) \neq F^1_z \times \ldots \times F^p_z(\ell[q_1, z_1], \ldots, \ell[q_p, z_p]) \text{ but } C' \text{ accepts}\right] \leq \delta$$

for some (small) δ. The remarks after definition 2.4 concerning convex combinations of "prover strategies" applies here and the result will follow.

Our goal, then, is to show that for most $z = (z_1, \ldots, z_n)$, F_1 is roughly "functional" in each coordinate on the family of lines $\{(\ell_1, \ldots, \ell_p) \mid z_i \in \ell_i\}$ associated with z. For each $z \in (\mathbb{F}^t)^p$ and $\ell_1 \in \mathfrak{P}_1(\mathbb{F}, \mathbb{F}^l)$ let

$$\Im(\ell_1; z) \stackrel{\text{def}}{=} \{A \mid \exists q_2, \ldots, q_p, F_1(\ell_1, \ell[q_2, z_2], \ldots, \ell[q_k, z_k])_1 = A\}$$

denote the set of possible answers (offered by F_1) to ℓ_1 for this z. For given ℓ_1 and z, associate with each element $A \in \Im(\ell_1; z)$ the probability

$$p^{\ell_1}_z(A) = \Pr_{q_2, \ldots, q_r}[F_1(\ell_1, \ell[q_2, z_2], \ldots, \ell[q_p, z_p])_1 = A \text{ and } C \text{ accepts}].$$

Then, define $F^1_z : \ell \mapsto A_{\max}$ where $A_{\max} \in \Im(\ell, z)$ is a canonical element maximizing $p^\ell_z(A_{\max})$ for ℓ. These "majority" functions will be shown to closely approximate F_1. Finally, define the *deformity at* z to be the function

$$D_z(\ell) = \sum_{A \neq F^1_z(\ell)} p^\ell_z(A).$$

We would like to show that $D_z(\ell_1)$, the probability that an answer *other* than $F^1_z(\ell_1)$ is given, is likely to be small. Expressing this as a Ky Fan distance (see [35] for example), we show the following:

Lemma 3.3 *For $\ell_1 \in \mathfrak{P}_1(\mathbb{F}, \mathbb{F}^l)$, $\Pr_z[D_z(\ell_1) > \delta] \leq \delta$ for all $\delta \geq 3\sqrt[3]{\frac{l|\mathbb{B}|}{|\mathbb{F}|}}$.*

Proof: Fix $\ell_1 \in \mathfrak{P}_1(\mathbb{F}, \mathbb{F}^l)$ and suppose that

$$\Pr_z[D_z(\ell_1) > \delta] > \delta$$

for some $\delta \in [0, 1]$. We show that $\delta < 3\sqrt[3]{\frac{l|\mathbb{B}|}{|\mathbb{F}|}}$, which proves the lemma. We should like to focus on an event which naturally permits application of corollary 3.2, which expresses the relevant coding nature of polynomials. Since $\Pr_z[D_z(\ell_1) > \delta] > \delta$, we may fix a tuple $(z_2, \ldots, z_p)$ for which $\Pr_{z_1}[D_z(\ell_1) > \delta] > \delta$. Now

consider the probability space induced by independent selection of $q_2, \ldots, q_p, q_2', \ldots, q_p'$ and $z_1 \in \ell_1$. Over this space, we consider the event E consisting of those triples for which

$$F_1(\ell_1, \ell[q_2, z_2], \ldots, \ell[q_p, z_p])_1 \neq F_1(\ell_1, \ell[q_2', z_2], \ldots, \ell[q_p', z_p])_1$$

and V accepts both answers. Then

$$\begin{aligned}\Pr[E] \leq & \Pr[V \text{ accepts both} \mid F_1(\ell_1, \ell[q_2, z_2], \ldots, \ell[q_p, z_p])_1 \neq F_1(\ell_1, \ell[q_2', z_2], \ldots, \ell[q_p', z_p])_1] \\ \leq & \frac{l\,|\mathbb{B}|}{|\mathbb{F}|}\end{aligned}$$

from corollary 3.2. We shall compute a lower bound for $\Pr[E]$ in terms of δ which will yield the statement of the lemma. Let $\beta < \frac{\delta}{2}$ (we shall fix this quantity later). We consider the following two cases depending on the density of the most likely convincing answer, $F_z^1(\ell_1)$:

1. Suppose that $\Pr_{z_1}\left[p_z^{\ell_1}(F_z^1(\ell_1)) \geq \beta \mid D_z(\ell_1) > \delta\right] \geq \frac{1}{2}$. Then $\Pr[E] \geq \frac{\delta^2\beta}{2}$.

2. Otherwise $\Pr_{z_1}\left[p_z^{\ell_1}(F_z^1(\ell_1)) < \beta \mid D_z(\ell_1) > \delta\right] \geq \frac{1}{2}$. In this case where the probability of the most convincing answer is small, we may partition $\Im(\ell_1, z)$ into two disjoint sets $T_1 \cup T_2 = \Im(\ell_1, z)$ so that for each $i \in \{1, 2\}$, $\sum_{t \in T_i} p_z^{\ell_1}(t) \geq \frac{\delta}{2} - \frac{\beta}{2}$. Then

$$\Pr[E] \geq \frac{\delta(\delta + \beta)(\delta - \beta)}{8}.$$

Selecting $\beta = (\sqrt{5} - 2)\delta$ equates these two bounds and gives

$$\frac{\sqrt{5} - 2}{2}\delta^3 \leq \frac{l\,|\mathbb{B}|}{|\mathbb{F}|}$$

so that $\delta \leq 3\sqrt[3]{\frac{l|\mathbb{B}|}{|\mathbb{F}|}}$. □

This yields the functionality we sought: fixing ℓ_1 we have that

$$\Pr_{z, q_2, \ldots, q_p}[F_z^1(\ell_1) \neq F_1(\ell_1, \ell[z_2, q_2], \ldots, \ell[z_p, q_p])_1 \text{ and } C' \text{ accepts}] \leq 2\delta \leq 6\sqrt[3]{\frac{l\,|\mathbb{B}|}{|\mathbb{F}|}} \tag{3.1}$$

where δ is the minimum value rending true the statement of lemma 3.3. There is, of course, nothing special about the first coordinate, and one may define functions F_z^i for each $i \in \{2, \ldots, p\}$ as we have defined F_z^1. The inequality 3.1 above shall hold analogously for each of these coordinates, and we have that

$$\Pr_{z,q}[\exists i, F_z^i(\ell[q_i, z_i]) \neq F_1(\ell[z_1, q_1], \ldots, \ell[z_p, q_p])_i \text{ but } C' \text{ accepts}] \leq 6p\sqrt[3]{\frac{l\,|\mathbb{B}|}{|\mathbb{F}|}}. \tag{3.2}$$

Notice that

$$6p\sqrt[3]{\frac{l\,|\mathbb{B}|}{|\mathbb{F}|}} = 6p\sqrt[3]{\frac{l2^b}{2^{4b+3\log p+\log l+9}}} = 6p\sqrt[3]{\frac{1}{2^{3b+3\log p+9}}} = \frac{6p}{2^{b+\log p+3}} \leq \epsilon.$$

Finally, since $x \notin L$, for any $(F_1^z, \ldots, F_p^z)$, $\Pr_R[(V \leftrightarrow F_1^z, \ldots, F_p^z)[x; R] \text{ accepts}] \leq \epsilon$ so that

$$\Pr_R[(V' \leftrightarrow F_1, F_2)[x; R] \text{ accepts}] \leq 2\epsilon \leq 2^{-k}.$$

□

Since we shall need this parallelization machinery again in the next section, we isolate it in the following lemma.

Lemma 3.4 *Let $L \subset \text{MIP}[p, r, q, a, \epsilon]$ for reasonable functions p, r, q, a and ϵ. Let V be a verifier accepting L with these parameters. Let $\mathbf{P}_f(V)$ denote the verifier resulting from the above parallelization process with a primary field ($\mathbb{F}$) of size 2^f and a base field ($\mathbb{B}$) of size 2. Then $\mathbf{P}_f(V)$ is a*

$$\text{MIP}[2, r + pqf, 2pqf, 2pqaf, \epsilon + 6p\sqrt[3]{\frac{q}{2^f}}]$$

verifier.

3.3 Reducing Answer Sizes

Lemma 3.2 gives a MIP system for any $L \in$ NP all the parameters of which are desirable save the large answer sizes. In light of this, we would like to reduce the answer size of a given MIP system with minimal cost in terms of the other parameters. We achieve this in two stages. First, we give a recursive simulation of the MIP system and repeat this simulation enough times in parallel to maintain low error. This recursive simulation results in a MIP system which has desirable complexity in terms of randomness, communication, and error, but uses too many provers. We then demonstrate that this resulting system can be parallelized without significant cost to produce a system with appropriate complexity and a constant number of extra provers. Application of this entire transformation to the system of lemma 3.2 yields a system with similar complexity in terms of prover multiplicity, randomness, query size, and error, but with appropriately short answers.

For the recursive simulation mentioned above we shall use the framework of probabilistically checkable proofs, outlined next.

3.3.1 Probabilistically Checkable Proofs

We shall consider a variant of the PCP proof systems discussed in definition 2.7 and theorem 2.4. A verifier of this new sort shall differ in the two ways described below and shall be called a *proof checker*.

1. We consider the input to be comprised of a constant number of strings $x_1, \ldots, x_k$, each of which is offered to the verifier separately. For this reason, we shall consider proof checkers for sets $S \subset (\Sigma^*)^k$ rather than languages.

2. The proof checker may expect its k inputs to be provided by oracles according to an appropriate error-correcting code E. During the course of the proof checker's computation, it may elect only to examine some (perhaps small) portion of these encoded inputs.

These two alterations can be found in [7, 4, 3].

Naturally, we shall be interested in sets $S \subset (\Sigma^*)^k$ corresponding to easily computable languages. Those corresponding to languages in P and NP are defined below.

Definition 3.5 *A set $S \subset (\Sigma^*)^k$ is an* P*-relation if* $\{\langle \vec{s}_1, \ldots, \vec{s}_k \rangle \mid \vec{s} \in S\} \in \mathrm{P}$.

Definition 3.6 *A set $S \subset (\Sigma^*)^k$ is an* NP*-relation if* $\{\langle \vec{s}_1, \ldots, \vec{s}_k \rangle \mid \vec{s} \in S\} \in \mathrm{NP}$.

As before, we shall treat oracles as functions. In particular, if O is an oracle and $x \in \{0,1\}^*$ a word, the notation $O = x$ shall mean that $O : \{0,1\}^{\lceil \log x \rceil} \to \{0,1\}$ is the function $O : i \mapsto x_i$. We shall again use the notation $(V \leftrightarrow O_1 \ldots O_k)[1^n; R]$ to denote the behavior of V with oracles $O_1, \ldots, O_k$ on random string R and input 1^n. (In these cases where we are interpreting oracles as purveyors of "input," the 1^n appearing in this expression is just a convenient tool for expressing the running time of the machine.) When no confusion can arise concerning oracle identity, we shall also use the notation $(V \leftrightarrow \{O_i \mid i \in I\})[1^n; R]$.

We formalize the framework described above:

Definition 3.7 *Let $S \subset (\Sigma^*)^k$. A* $(t : \mathbb{N} \to \mathbb{N}, \epsilon)$-proof checker *for S is a tuple (V, E) satisfying the following criteria.*

- *V is a $t(n)$-time bounded, probabilistic $k+1$-oracle Turing machine. The first k oracles we shall call the* input *oracles and the last the* proof *oracle.*
- *V queries each of its oracles $O(1)$ times, receiveing a single bit in response for each query.*
- *E is a polynomial-time computable encoding function with constant distance: $\exists \delta > 0, \forall x, y, x \neq y \Rightarrow \Delta(E(x), E(y)) > \delta$.*
- *For $\vec{s} \in S$, $\exists \Pi, \Pr_R \left[(V \leftrightarrow \sigma_1, \ldots, \sigma_k, \Pi)[1^{|\vec{s}|}; R] \text{ accepts}\right] = 1$ where $\sigma_i = E(\vec{s}_i)$.*
- *$\forall \vec{w} \in (\Sigma^*)^k$, if 1. $\vec{w} \notin S$ or*

  2. there exists i such that $\min_z \Delta(O_i, E(z)) > \frac{\delta}{2}$

  then $\forall \Pi$, $\Pr_R \left[(V \leftrightarrow O_1, \ldots, O_k, \Pi)[1^{|s|}; R] = \text{accept}\right] \leq \epsilon$.

From [7, 4, 3] we have the following two lemmas.

Lemma 3.5 *Let $S \subset (\Sigma^*)^k$ be a* NP*-relation. Then for all $\epsilon > 0$, there is a* $(\text{poly} \log n, \epsilon)$*-proof checker for* S.

Lemma 3.6 *Let $R \times S \subset (\Sigma^*)^l \times (\Sigma^*)^k$ be a* P*-relation. Then for all $\epsilon > 0$, there is a polynomial time computable function $\mathfrak{S} : r \mapsto V_r$ defined on R so that V_r is a* $(\text{poly} \log n, \epsilon)$*-proof checker for*

$$S_r \stackrel{def}{=} \{s \in S \mid r \times s \in R \times S\}.$$

3.3.2 Recursive Answer Size Reduction

Lemma 3.7 *Let p be constant. For appropriate $2^f = \Omega(\log(a + r) \cdot 2^{3k})$ we have*

$$\text{MIP}[p, r, q, a, 2^{-k}] \subseteq \text{MIP}[p + 2, r', q', a', 3 \cdot 2^{-(k+1)}],$$

where $r' = O(r + kf \log(pa + r))$, $q' = O(q + r + kf \log(pa + r))$, and $a' = O(kf \log(pa + r))$.

Proof: The proof will be presented in two steps:

1. *(Recursive Simulation)* A recursive simulation of any constant-prover MIP system is given:

   Let p be a constant and $r, q, a, m : \mathbf{N} \to \mathbf{N}$ and $\epsilon : \mathbf{N} \to (0, 1)$ reasonable functions. Then there exists a constant $c \in (0, 1)$ so that

   $$\text{MIP}(p, r, q, a, \epsilon) \subseteq \text{MIP}(m(2p + 2), r + O(m \log(pa + r)), q + r + O(\log(pa + r)), O(1), \epsilon + c^m)$$

   Notice that the size of the answers in this system is a constant and that the error can be reduced with a commensurate increase in prover multiplicity, communication complexity, and randomness.

2. *(Parallelization)* We show how to efficiently parallelize a certain class of MIP proof systems. In particular, the system generated in step 1 is efficiently parallelized:

   Let $L \in \text{MIP}(p = p_{\text{common}} + p_{\text{rest}}, r, q, a, \epsilon)$ and let V be a verifier accepting L with these parameters. Assume further that there is some subcollection of p_{common} provers $\{P_\alpha | \alpha \in \Pi_{\text{common}}\}$ to which the questions sent by V are (always) of form $Q_{\text{common}} \circ \hat{Q}_i$, where Q_{common} is the same across these p_{common} provers. Let $|Q_{\text{common}}| = q_{\text{common}}$ and $|\hat{Q}_i| = q_{\text{vary}}$. Then for appropriate f with $2^f = \Omega(\frac{q_{\text{vary}}}{\epsilon^3})$ we have that

   $$L \in \text{MIP}[p_{\text{rest}} + 2,\ r + p_{\text{common}} q_{\text{vary}} f,\ q + p_{\text{common}} q_{\text{vary}} f,\ a + a p_{\text{common}} q_{\text{vary}} f,\ 3\epsilon/2].$$

Appropriate coupling of step 2 and step 1 yields the statement of the lemma.

Proof of step 1: We begin by showing that for constant p, there exists $c \in (0,1)$ so that

$$\text{MIP}(p, r, q, a, \epsilon) \subseteq \text{MIP}(2p+2, r + O(\log(pa+r)), q + r + O(\log(pa+r)), O(1), \epsilon + c). \tag{3.3}$$

Let $L \in \text{MIP}(p, r, q, a, \epsilon)$, and $V = (Q, C)$ be a verifier which accepts L with this complexity. The relation

$$\left\{(x, \hat{R}, a_1, \ldots, a_p) \mid C(x, \hat{R}, a_1, \ldots, a_p)\right\}$$

is a P-relation. For a fixed x and $\hat{R}$ lemma 3.6 above yields (in polynomial time) a proof checker $V_{x,\hat{R}}$ for $\left\{(a_1, \ldots, a_p) \mid C(x, \hat{R}, a_1, \ldots, a_p)\right\}$ using $O(\log(a+r))$ random bits to select $l = O(1)$ bits from each of its p input oracles and its proof oracle. We may assume that this machine admits error ϵ_c with $\epsilon_c \ll \frac{1}{l}$. (One may just use sequential repetition, as in theorem 2.5, to drive the error down exponentially suffering but a linear increase in the number of questions.)

We are now ready to construct a $2(p+1)$ prover system for L with verifier $V' = (Q', C')$. V' generates $\hat{R}$ as V would have and constructs $V_{x,\hat{R}}$, a proof checker requiring $p+1$ oracles. V' will then use its $2(p+1)$ provers to simulate the $p+1$ oracles with which $V_{x,\hat{R}}$ wishes to interact. Set $\Xi \stackrel{\text{def}}{=} \{1, \ldots, p, \Pi\}$. The $2(p+1)$ provers with which V' interacts are denoted $(P_\xi)_{\xi \in \Xi}$ and $(P_\xi^{\text{check}})_{\xi \in \Xi}$, two for each oracle O_i of the proof checker $V_{x,\hat{R}}$.

V' is defined the following way.

1. V' generates $\hat{R}$ at random such that $\left|\hat{R}\right| = r$ and constructs the proof checker $V_{x,\hat{R}}$. Let $Q_i = Q(x,R)_i$ for each $i \in \{1, \ldots, p\}$. For convenience, let $Q_\Pi = \hat{R}$.

2. V' generates R_1 at random with which it simulates $V_{x,\hat{R}}$ so that $|R_1| = O(\log(a+r))$. This simulation results in the (constant number) of queries of $V_{x,\hat{R}}$ to its oracles: let q_ξ^j denote the jth query of $V_{x,\hat{R}}$ to oracle O_ξ. (For concreteness assume that $V_{x,\hat{R}}$ requires exactly l bits in response from each oracle.)

3. V' generates a random string R_2 (of constant length) which it uses to uniformly select $\vec{t} \in \{1, \ldots, l\}^{|\Xi|}$. V' then sends

   - $\hat{R}$ and $(q_\xi^j)_{j \in \{1,\ldots,l\}}$ to P_ξ, and
   - Q_ξ and $q_\xi^{\vec{t}_\xi}$ to P_ξ^{check}

   so that

$$Q'(x, \hat{R} \circ R_1 \circ R_2) = (\hat{R} \circ q_1^1 \circ \cdots \circ q_1^l, Q_1 \circ q_1^{t_1}, \ldots, \hat{R} \circ q_p^1 \circ \cdots \circ q_p^l, Q_p \circ q_p^{t_p}, \hat{R} \circ q_\Pi^1 \circ \cdots \circ q_\Pi^l, \hat{R} \circ q_\Pi^{t_\Pi})$$

   where the provers are ordered $P_1, P_1^{\text{check}}, \ldots, P_p, P_p^{\text{check}}, P_\Pi, P_\Pi^{\text{check}}$.

4. V' expects provers P_ξ for $\xi \in \Xi$ to respond with a vector of answers $(a_\xi^1, \ldots, a_\xi^l)$, one for each question they were asked. The provers P_ζ^{check} are naturally expected to answer with a single bit we denote $\tilde{a}_\xi$. If for any $\xi \in \Xi$, $a_\xi^{t_\xi} \neq \tilde{a}_\xi$, V' rejects, displeased that P_ξ disagreed with the function P_ξ^{check}. Otherwise, V' simulates $V_{x,\hat{R}}$ with answers a_ξ^j and accepts when $V_{x,\hat{R}}$ accepts. (This defines C'.)

For a specified $\hat{R}, R_1$, we say that prover P_ξ^{check} *induces* P_ξ if they are consistent in the sense that for any R_2, the answer received from P_ξ^{check} is identical to the interpreted answer to q_ξ^j according to the response of P_ξ. Notice that if P_ξ^{check} does *not* induce P_ξ then V' discovers this fact with probability at least $\frac{1}{l}$ so that $\forall \hat{R}, R_1$,

$$\Pr_{R_2}[(V' \leftrightarrow \{P_\xi, P_\xi^{\text{check}} \mid \xi \in \Xi\})[x; \hat{R} \circ R_1 \circ R_2] = \text{accept} \mid \exists \xi \in \Xi, P_\xi^{\text{check}} \text{ does not induce } P_\xi] \leq \frac{l-1}{l}.$$

Clearly, if $x \in L$, for any $\hat{R}$ there are oracles that convince $V_{x,\hat{R}}$ with probability 1 and hence provers that convince V' with probability 1.

Suppose $x \notin L$, then we show that V' accepts with probability bounded above by a constant $\epsilon + c$. To begin with,

$$\begin{aligned} \forall P_i \ \Pr_{\hat{R}}[(V \leftrightarrow P_1, \ldots, P_p)[x; \hat{R}] = \text{accept}] &\leq \epsilon \quad \Longrightarrow \\ \forall O_\xi \ (\xi \in \Xi) \ \Pr_{\hat{R},R_1}[(V_{x,\hat{R}} \leftrightarrow \{O_\xi \mid \xi \in \Xi\})[1^{pa}; R_1] = \text{accept}] &\leq \epsilon + \epsilon_c \end{aligned}$$

where ϵ_c is the constant error probability of the proof checker (see definition 3.7). In this case, $\forall P_\xi, P_\xi^{\text{check}}$,

$$\begin{aligned} \Pr_{R=\hat{R}\circ R_1 \circ R_2}[(V' \leftrightarrow \{P_\xi, P_\xi^{\text{check}} \mid \xi \in \Xi\})[x; R] = \text{accept}] &\leq \\ \Pr_{R=\hat{R}\circ R_1 \circ R_2}[(V' \leftrightarrow \{P_\xi, P_\xi^{\text{check}} \mid \xi \in \Xi\})[x; R] = \text{accept} \mid \forall \xi \in \Xi, P_\xi^{\text{check}} \text{ induces } P_\xi] &+ \\ \Pr_{R=\hat{R}\circ R_1 \circ R_2}[(V' \leftrightarrow \{P_\xi, P_\xi^{\text{check}} \mid \xi \in \Xi\})[x; R] = \text{accept} \mid \exists \xi \in \Xi, P_\xi^{\text{check}} \text{ does not induce } P_\xi] &\leq \\ \Pr_{R=\hat{R}\circ R_1 \circ R_2}[(V' \leftrightarrow \{P_\xi, P_\xi^{\text{check}} \mid \xi \in \Xi\})[x; R] = \text{accept} \mid \forall \xi \in \Xi, P_\xi^{\text{check}} \text{ induces } P_\xi] + \frac{l-1}{l} &= \\ \Pr_{\hat{R},R_1}[(V_{x,\hat{R}} \leftrightarrow \{P_\xi^{\text{check}} \mid \xi \in \Xi\})[1^{pa}; R_1] = \text{accept}] + \frac{l-1}{l} &\leq \\ \epsilon + \epsilon_c + \frac{l-1}{l} & \end{aligned}$$

The randomness and communication required are $r + O(\log(a + r))$. Only $O(1)$ answer bits are required. Although the total error $\epsilon + \epsilon_c + \frac{l-1}{l}$ may be larger than one, the error *introduced by this simulation* is at most $\epsilon_c + \frac{l-1}{l}$, and so is less than one by our assumption that ϵ_c is small. This proves the containment (3.3) above.

By repeating steps (2) and (3) of the above protocol in parallel m times (with new provers and new R_1, R_2 for every repetition) we obtain the statement of step 1. This parallelization results in m provers for each prover P of the original protocol. In the sequel, these provers shall be called the provers *associated with* P. □

Proof of step 2:

If we apply the [71, 42] construction (cf. lemma 3.4) in order to reduce the number of provers of the system constructed in step 1, the resulting complexity is too high. Fortunately, the protocol used in step 1 has a nice property: except for $O(\log(a+r))$ bits, the questions to associated provers *across* the m repetitions are identical. (In the language of step 1, $\hat{R}$, and so Q_ξ, remains fixed across the repetitions while R_1 and R_2 vary.) We exploit this property by applying the transformation of lemma 3.4 to the associated collections of provers and noting that the transformation machinery need only apply to the $O(\log(pa+r))$ varying bits.

Let V be a MIP$[p = p_{\text{common}} + p_{\text{rest}}, r, q = q_{\text{common}} + q_{\text{vary}}, a, \epsilon]$ verifier for $L \subseteq \{0,1\}^*$ which interacts with provers $\{P_\pi | \pi \in \Pi\}$ ($|\Pi| = p$) in such a way that V asks some subcollection of provers $\{P_\alpha \mid \alpha \in \Pi_{\text{common}} \subseteq \Pi\}$ (we have $|\Pi_{\text{common}}| = p_{\text{common}}$) questions with a common prefix of length q_{common}. Let $\Pi_{\text{rest}} = \Pi - \Pi_{\text{common}}$. For the purpose of analysis, we consider two tertiary machines V_{rest} and V_{common} which execute certain portions of the interaction process of V:

- V_{rest} expects as input $\langle x, R\rangle$ and demands connection to p_{rest} provers (which should be thought of as $\{P_\beta \mid \beta \in \Pi_{\text{rest}}\}$). It computes $Q(x,R)_\beta$ for $\beta \in \Pi_{\text{rest}}$, sends one to each of its p_{rest} provers, and returns their answers as a result.

- V_{common} expects as input $\langle x, R\rangle$ and demands connection to p_{common} provers (which should be thought of as $\{P_\alpha | \alpha \in \Pi_{\text{common}}\}$). It computes the *varying* portion of the questions $Q(x,R)_\alpha$ for $\alpha \in \Pi_{\text{common}}$, sends one to each of its p_{common} provers, and returns their answers as a result.

We create a new MIP$[p, r, q, a, \epsilon]$ verifier called $V[V_{\text{common}}, V_{\text{rest}}]$ which interacts with provers $\{P_\pi | \pi \in \Pi = \Pi_{\text{common}} \cup \Pi_{\text{rest}}\}$ and accepts the same language as V (but has different conceptual structure). The procedure for $V[V_{\text{common}}, V_{\text{rest}}]$, on input x, is as follows:

1. generate R at random so that $|R| = r$.

2. generate the *fixed* (common) portion Q_{common} of the questions $Q(x,R)_\alpha$ (for $\alpha \in \Pi_{\text{common}}$) and sends Q_{common} to each P_α for $\alpha \in \Pi_{\text{common}}$.

3. run $V_{\text{rest}}(R, x)$ (with provers $\{P_\beta \mid \beta \in \Pi_{\text{rest}}\}$) and collect the returned prover responses.

4. run $V_{\text{common}}(R, x)$ (with provers $\{P_\alpha | \alpha \in \Pi_{\text{common}}\}$) and collect the returned prover responses.

5. accept if V would have accepted with input x, random string R, and these returned answers.

It is clear that the external behavior of $V[V_{\text{common}}, V_{\text{rest}}]$ is identical to that of V.

We now substitute for V_{common} (inside the machine $V[V_{\text{common}}, V_{\text{rest}}]$) the machine $\mathbf{P}_f(V_{\text{common}})$ to produce a new verifier which we call $V[\mathbf{P}_f(V_{\text{common}}), V_{\text{rest}}]$. (We shall select f presently.) The machine $\mathbf{P}_f(V_{\text{common}})$ interacts with two provers, P_{curves} and P_{points} rather than the p_{common} provers $\{P_\alpha | \alpha \in \Pi_{\text{common}}\}$, but still

(naturally) returns p_{common} answers.[1] That the application of the parallelization transformation $V \mapsto \mathbf{P}_f(V)$ in this situation is valid depends on the fact that if the provers (P_α) responses to V_{common} are functional with respect to the the $O(\log(a+r))$ varying bits of the questions, then they are function with respect to the entirety of the questions. With appropriate choice of f, the parallelization protocol of [42], recorded in lemma 3.4 provides answers to these questions which are non-functional with probability at most $\frac{\epsilon}{2}$. The questions of V_{common} are of size q_{vary} so that $V[\mathbf{P}_f(V_{\text{common}}), V_{\text{rest}}]$ is a

$$\text{MIP}[p_{\text{rest}} + 2,\ r + p_{\text{common}} q_{\text{vary}} f,\ q + p_{\text{common}} q_{\text{vary}} f,\ a + a p_{\text{common}} q_{\text{vary}} f,\ 3\epsilon/2]$$

verifier for the language accepted by V, as desired.

□

We now apply the result of step 2 to the system produced by step 1:

1. For each $\xi \in \Xi - \{\Pi\}$, the m provers associated with P_ξ^{check} $(\xi \in \Xi)$ are parallelized, resulting in 2 provers. (Recall that the questions to these provers have the common prefix Q_i.)
2. The provers associated with P_Π^{check} and P_ξ, for $\xi \in \Xi$, are parallelized together. (Their questions have the common prefix $\hat{R}$.)

In this case, where we are applying step 2 many times to the same system, the provers P_{points} for each of these parallelization steps may be combined into one prover. □

Finally, applying the result of lemma 3.7 to the containment of lemma 3.2 yields our main theorem:

Theorem 3.1 *Let* $k(n) = O(\log n)$ *be reasonable, and let* $\bar{k}(n) = \max(k(n), \log\log n)$. *Then* NP $\subseteq$ MIP$[4, r, q, a, 2^{-k(n)}]$, *where* $r = O(k(n)\log n + k(n)^2\bar{k}(n))$, $q = O(r)$, *and* $a = O(k(n)^2\bar{k}(n))$.

3.4 Improved Efficiency Probabilistically Checkable Proofs

An important tool in the proof of theorem 2.4 is the recursive application of proof machinery adapted from [42]. Carsten Lund shows that by using the more efficient machinery of theorem 3.1 along with some improved analysis of the combinatorial core of [3], one can dramatically improve the constants in the statement of theorem 2.4. It is shown in [3] that NP $\subset$ PCP$[t, O(\log n), O(\log n), 1, \frac{1}{2}]$ for some t on the order of 10^4. Phillips and Safra [78] brought about some improvement in this value of t. To express the advance described above, we define the following "amortized" PCP class.

Definition 3.8 PCP$^{av}[p, r, q, a, \epsilon]$ *denotes those languages with probabilistically checkable proof systems (as in definition 2.7) using* r *randomness, asking queries of size* q, *demanding answers of size* a, *and achieving error at most* ϵ *so that the* average *number of queries (over the random strings) is* p.

[1] This change in the number of provers also changes the number of provers to which $V[\mathbf{P}_f(V_{\text{common}}), V_{\text{rest}}]$ must send the *common* portion of the queries in step 2 (now Q_{common} is sent to only two provers).

Lund [18], using [3], the material of the previous sections, and improved "testing" equipment (cf. [24]), shows the following theorem:

Theorem 3.2 NP $\subset$ PCP$^{av}[29, O(\log n), O(\log n), 1, \frac{1}{2}]$.

This shall have ramifications for approximation algorithms (see §4.3).

3.5 Recent Improvements

Perhaps the most striking advance since the above developments is the (positive) resolution by Raz [80] of the PARALLEL REPETITION CONJECTURE. We have above discussed *sequential repetition* of a MIP system. This is essentially the containment

$$\text{MIP}[p, r, q, a, \epsilon] \subset \text{MIP}[kp, kr, q, a, \epsilon^k]$$

obtained by repeating a given MIP system k times with both independent provers and random strings (cf. theorem 2.5). The PARALLEL REPETITION CONJECTURE is that similar exponential error decay can be effected by repeating an MIP system k times (with independent randomness for each repetition) but using only p provers: the k questions to prover i generated across the k repetitions of the MIP system are asked, as a vector, to a single prover which is responsible for answering them all. Following [86, 89, 36, 39], Raz elegantly closed the subject, demonstrating that

$$\text{MIP}[2, r, q, a, \epsilon] \subset \text{MIP}[2, kr, kq, ka, \epsilon_R^{\frac{k}{s}}]$$

where $s = \log |\mathcal{A}_1| |\mathcal{A}_2|$ and ϵ_R is some constant in $(0, 1)$. This is a potent and widely applicable tool. It shows, for example, that

$$\text{NP} \subset \text{MIP}[2, O(k \log n), O(k \log n), O(k), 2^{-k}],$$

subsuming theorem 3.1 above. Specifically, with $k = \Theta(\log n)$, this yields

$$\text{NP} \subset \text{MIP}[2, O(\log^2 n), O(\log^2 n), O(\log n), \frac{1}{n}]. \tag{3.4}$$

It is natural to ask if pseudo-random techniques, like those used to improve theorem 2.5 to theorem 2.6, can be applied here to prove the following conjecture.

Conjecture 3.1 NP $\subset$ MIP$[2, O(\log n), O(\log n), O(\log n), \frac{1}{n}]$.

Feige and Kilian [40], however, offer evidence that such attempts are unlikely to be fruitful. Notice that realization of $\frac{1}{n}$ error is incompatible with randomness or answers of size $o(\log n)$. Furthermore, if the ques-

tion length is $o(\log n)$ an easy recursive argument[2] shows that NP $\subset$ P. Hence such a proof system (if it exists) is optimal.

[2]Such an argument requires that the other parameters are not allowed to escape from the $O(\log n)$ envelope.

Chapter 4

Lower Bounds for Approximation Algorithms

4.1 Set Cover is Hard to Approximate

As an example of the application of this holographic proof machinery to approximation algorithms, we study the SET COVER problem. A *set system* $\mathcal{S}$ is a tuple $(S; \{S_1, \ldots, S_k\})$ where each $S_i \subset S$. The SET COVER problem is that of determining, given a set system $\mathcal{S}$ and a natural c if there is a cover of S using c sets from $\{S_i\}$. Formally,

$$\text{SET COVER} \stackrel{\text{def}}{=} \left\{ \langle S, \{S_i \mid i \in I\}, l\rangle \mid (S; \{S_i \mid i \in I\}) \text{ is a set system and } \exists J \subset I, |J| \leq l, \bigcup_{j \in J} S_j = S \right\}.$$

SET COVER was among the first problems shown NP-complete [66]. For a set system $\mathcal{S} = (S, \{S_1, \ldots, S_k\})$, let $\text{opt}_{\mathcal{S}}$ denote the smallest cardinal number for which there exists a cover C of size $\text{opt}_{\mathcal{S}}$. In 1974, Johnson [65] gave a $\ln n + 1$-approximation algorithm for SET COVER, that is, a polynomial time algorithm which, given a set system $\mathcal{S}$, produces a cover $(S_j)_{j \in J}$ where $|J|$ is at most $\text{opt}_{\mathcal{S}}(\ln n + 1)$. Lund and Yannakakis [74], using [71, 42], demonstrate the following hardness results for SET COVER:

Theorem 4.1 ([74]) *There exists $c > 0$ so that* SET COVER *cannot be approximated to within c unless*

$$\text{P} = \text{NP}.$$

Theorem 4.2 ([74]) *For $c < \frac{1}{4}$,* SET COVER *cannot be approximated to within $c \log_2 N$ unless*

$$\text{NP} \subset \text{DTIME}[n^{\text{poly} \log n}].$$

We shall adapt the proof of [74] in order that we may apply it to our (more efficient) *four* prover framework (they work with two provers). We shall also investigate the combinatorial core of their argument and (as they suggest) offer a tighter (but probabilistic) construction of their key element. This study will yield strengthened versions of theorems 4.1 and 4.2. Specifically we shall prove the four theorems below. Theorem 4.3 improves theorem 4.1 above. Theorems 4.4 through 4.6 are a sequence of hardness results demonstrating increasingly strong containments (for NP) based on increasingly stringent antecedents. None of these are directly comparable to theorem 4.2 (except for 4.6 in the range $c \in (0, \frac{1}{8})$). Of course, if one is convinced that NP requires exponential time even when randomness is available, there is no distinction between these (equally false) consequents and one concludes that SET COVER cannot be approximated to within $\frac{1}{2}\log_2 n$. It is worth noting that this is remarkably close to the $(\ln n + 1) \approx .7 \log_2 n$ upper bound cited earlier[1].

Theorem 4.3 *For all $c > 0$,* SET COVER *cannot be approximated to within c unless*

$$\mathrm{P} = \mathrm{NP}.$$

Theorem 4.4 *For all $c < \frac{1}{2}$,* SET COVER *cannot be approximated to within $c \log_2 n$ unless*

$$\mathrm{NP} \subset \mathrm{RTIME}[n^{\mathrm{poly}\log n}].$$

Theorem 4.5 *For all $c < \frac{1}{4}$,* SET COVER *cannot be approximated to within $c \log_2 n$ unless*

$$\mathrm{NP} \subset \mathrm{RTIME}[n^{O(\log\log n)}].$$

Theorem 4.6 *For all $c < \frac{1}{8}$,* SET COVER *cannot be approximated to within $c \log_2 n$ unless*

$$\mathrm{NP} \subset \mathrm{DTIME}[n^{O(\log\log n)}].$$

The proofs are reductions from "computing the acceptance probability of a MIP system" to "computing an optimal set cover." That is, given an appropriate MIP system, we show how to build a set system so that the size of the minimum cover reflects the acceptance probability of the MIP system. Of course, since $\mathrm{NP} \subset \mathrm{MIP}[\cdot,\cdot,\cdot,\cdot,\cdot]$ (for appropriate parameters), closely approximating the acceptance probability of an MIP system is NP-hard. This will allow us to conclude that closely approximating the size of the minimum set cover is hard.

4.1.1 (m, l) Set Systems

The reduction we use shall require some combinatorial machinery, which we choose to develop first. The basic object is the following:

[1] See §4.3 for recent work in this area.

Definition 4.1 *A set system $\mathcal{B} = (B, \{B_1, \ldots, B_m\})$ is said to be a* (m, l) set system *if for any cover $(A_j)_{j \in J}$, $J \subset I$ and $A_j \in \{B_j, \overline{B_j}\}$ we have that $|J| \geq l$.*

Lund and Yannakakis [74] give a deterministic construction of (m, l) set systems with universes of size $O(2^{2l}m^2)$.

Theorem 4.7 ([74]) *For all $m, l \in \mathbb{N}$, there exists an (m, l) set system with universe of size $O(2^{2l}m^2)$ computable in time $O(\text{poly}(2^{2l}m^2))$.*

Applying the probabilistic method (see [2] for a beautiful exposition on this subject), we demonstrate the existence of (m, l) set systems with universes of size at most $2^l + l(\ln m + O(1))$.

Theorem 4.8 *For all m, l, there exists a (m, l) set system with universe of size at most $2^l + l(O(1) + \ln m)$. Furthermore, there is a constant c so that for $|B| \geq 2^l + l(\ln m + c)$, selecting each B_i independently and uniformly at random among the sets in 2^B,*

$$\Pr_{\{B_i | i \in I\}}[(B; B_1, \ldots, B_m) \text{ is a } (m, l) \text{ set system}] \geq \frac{3}{4}.$$

Proof: Notice that independently placing each $b \in B$ inside B_i with probability $\frac{1}{2}$ induces the uniform distribution on 2^B. For a subset $A \subset B$ and $s \in \{0, 1\}$, define

$$A^s \stackrel{\text{def}}{=} \begin{cases} A & \text{if } s = 0 \\ \overline{A} & \text{if } s = 1 \end{cases}.$$

For each $J \subset I \stackrel{\text{def}}{=} \{1, \ldots, m\}$ of size l and $s : J \to \{0, 1\}$, let $E(J, s)$ be the event that $\cup_{j \in J} B_j^{b(j)} = B$. Then, for any J and s,

$$\Pr_{\{B_i | i \in I\}}[E(J; s)] \leq (1 - 2^{-l})^{|B|}.$$

There are $2^l \binom{m}{l}$ such events, so

$$\begin{aligned} \Pr_{\{B_i | i \in I\}}[(B, \{B_i \mid i \in I\}) \text{ is a } (m, l) \text{ set system}] &\geq 1 - \Pr_{\{B_i | i \in I\}}[\bigvee_{J, s} E(J, s)] \\ &\geq 1 - \binom{m}{l} 2^l (1 - 2^{-l})^{|B|} \geq \frac{3}{4} \end{aligned}$$

when $|B| \geq 2^l + l(\Omega(1) + \ln m)$. (Coarsely approximate $\binom{m}{l}$ by m^l.) □

We shall use the following technical lemma about (m, l) set systems.

Lemma 4.1 *Let $\mathcal{B} = (B, \{B_i | i \in I\})$ be a (m, l) system (so that $|I| = m$). Fix $s \in \mathbb{N}$ and define $\mathcal{J} = \{J \mid J \subset \{1, \ldots, m\}, |J| \leq s\}$. Let $(C_k)_{k \in K}$ be a cover of B so that for each set C_k we have either $C_k \in \{B_i \mid i \in I\}$ or $C_k = \overline{\cup_{j \in J} B_j}$ for some $J \in \mathcal{J}$. Suppose that in addition, the cover $\{C_k\}$ is non-trivial in*

the sense that if $\overline{\cup_j B_j}$ appears in $\{C_k\}$ then at least one of these B_j is absent. (That is, $\{C_k\}$ does not cover by simply containing $B_{i_1}, \ldots, B_{i_s}$ and $\overline{\bigcup_{j=1}^{s} B_{i_j}}$.) Then $|K| \geq l$.

Proof: Notice that the case when $s = 1$ is just a restatement of definition of a (m, l) set system. The case $s = 0$ is immediate. Consider $s \geq 2$, and let $(C_k)_{k \in K}$ be a cover as in the statement of the lemma. We shall make a new cover, $(C'_k)_{k \in K}$, so that each $C'_k \in \{B_i^s \mid i \in I, s \in \{0,1\}\}$ and for all i, $\{B_i, \overline{B_i}\} \not\subseteq \{C_k \mid k \in K\}$. Then we shall have that $|K| \geq l$, as desired. Since $(C_k)_{k \in K}$ is non-trivial, with every element C_k of form $C_k = \overline{\cup_{j \in J} B_j}$, there is an "excluded" element $e_k \in J$ so that $B_{e_k} \notin \{C_k \mid k \in K\}$. Define

$$C'_k \stackrel{\text{def}}{=} \begin{cases} C_k & \text{if } C_k \in \{B_i^s \mid i \in I, s \in \{0,1\}\} \\ \overline{B_{e_k}} & \text{if } C_k = \overline{\cup_{j \in J} B_j} \end{cases}.$$

Notice that $(C'_k)_{k \in K}$ is a non-trivial cover with elements drawn from $\{B_i^s\}$ and hence has size at least l so that $|K| \geq l$, as desired. □

4.1.2 Canonical Proof Systems

As a second technical preliminary step, we shall massage the proof system of theorem 3.1 into a particularly convenient form.

Definition 4.2 (Canonical Form) *For a* MIP$[p, r, q, a, \epsilon]$ *verifier $V = (Q, C)$, define, for each $i \in \{1, \ldots, p\}$, $\mathcal{Q}_i$ to be the space of possible questions to the ith prover and $\mathcal{A}_i$ to be the space of possible answers from the ith prover. V is said to be* canonical *if*

- *(Functionality) for each random string r and answer a_1, there is a unique vector $(a_2, \ldots, a_p)$ so that $C(x, r, a_1, \ldots, a_p)$ accepts,*
- *(Uniformity) for all $i \in \{1, \ldots, p\}$, the distribution induced by Q on $\mathcal{Q}_i$ is uniform,*
- *(Question space equality) for $i \in \{1, \ldots, p\}$, the sets $\mathcal{Q}_i$ of possible questions to the ith prover are identical, and*
- *(Answer space disjointness) for $i, j \in \{1, \ldots, p\}$ and $i \neq j$, $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$.*

Theorem 4.9 *Let $L \in$ NP and $k(n)$ reasonable. Let $h(n) = \max(k(n), \log\log n)$. Then L has a* MIP$[4, r, q, a, 2^{-k}]$ *canonical verifier where $r = O(k(\log n) + \text{poly}(h)))$, $q = O(r)$, and $a = \text{poly}(h)$.*

Proof: We work with the system of theorem 3.1, which already satisfies conditions (Functionality) and (Uniformity). To achieve (Question Space Equality) we simply inflate each question space to $\mathcal{Q}_1 \times \cdots \times \mathcal{Q}_p$. Formally , define $V' = (Q', C')$ to be a verifier which generates R_1, R_2, R_3 and R_4, independent random

strings of length r, so that

$$C' = C$$
$$Q'(x, R_1R_2R_3R_4)_i = \langle Q(x, R_{\pi^i(1)})_1, Q(x, R_{\pi^i(2)})_2, Q(x, R_{\pi^i(3)})_3, Q(x, R_{\pi^i(4)})_4 \rangle$$

where $\pi = (1234) \in S_4$. To achieve (Answer Space Disjointness), simply require that each prover appends its name to its answer. □

4.1.3 The Set Cover Reduction

What follows is an adaptation of the reduction in [74] to our four prover framework.

Lemma 4.2 *Let* SAT $\in$ MIP$[p, r, q, a, \epsilon]$. *Define* $m(n) = \sum_i |\mathcal{A}|$ *and let* $l : \mathbb{N} \to \mathbb{N}$ *be reasonable. For each* m *and* l, *let* $\mathcal{B}_{m,l} = (B, \{B_i \mid i \in I\})$ *be a specific* (m, l) *set system. Then we may associate with each instance* ϕ *of* SAT *an instance* $\mathcal{S}_\phi$ *of* SET COVER *so that*

1. *if* $\phi \in$ SAT, *then* $\text{opt}_{\mathcal{S}_\phi} \leq \sum_i |\mathcal{Q}_i|$, *and*
2. *if* $\phi \notin$ SAT, *then* $\text{opt}_{\mathcal{S}_\phi} \geq (1 - \epsilon l^p)\frac{l}{p} \sum_i |\mathcal{Q}_i|$

where the number of sets in $\mathcal{S}_\phi$ *is* $p2^{O(r+a+l)}$. *This transformation, modulo the construction of* $\mathcal{B}_{m,l}$, *is computable in time* $\text{poly}(n, p2^{O(r+a+l)})$. *Notice that the definition of* $\mathcal{S}_\phi$ *depends both on* l *and* $\{\mathcal{B}_{m,l}\}$, *the set system family selected for the mapping.*

Proof: Let V_{SAT} be the canonical MIP$[p, r, q, a, \epsilon]$ verifier for SAT. Fix a formula ϕ of size n. Let $\mathcal{A}_i \subseteq \{0,1\}^a$ denote the set of possible answers from prover i. These are the disjoint sets promised by the (Answer space disjointness) condition in definition 4.2. Let $R \stackrel{\text{def}}{=} \{0,1\}^r$ be the space of random strings for V_{SAT}. For each (random string) $\rho \in R$ and each $a_1 \in \mathcal{A}_1$ we let $UA(\rho, A_1)$ denote the unique vector of answers $(A_2, \ldots, A_p)$ so that $C(\phi, r, A_1, \ldots, A_p) =$ accepts, if this vector exists (otherwise $UA(r, A_1)$ is undefined). Define $m \stackrel{\text{def}}{=} 2^a \geq \sum_{i=1}^p |\mathcal{A}_i|$.

$\mathcal{S}_\phi$, the SET COVER instance associated with ϕ is defined as follows. The base set S of the instance is $\mathcal{S}_\phi = (S, \vec{S})$ associated to ϕ is defined by $S \stackrel{\text{def}}{=} R \times B$. Recalling the notation introduced in lemma 4.1, define $\mathcal{J} = \{J \mid J \subset \{1, \ldots, m\}, |J| \leq p-1\}$. The sets of the system shall be the following. First, for each $q_1 \in \mathcal{Q}_1$ and each $a_1 \in \mathcal{A}_1$ we have the set

$$S(1, q_1, a_1) \stackrel{\text{def}}{=} \{\langle R, b\rangle \in S \mid q_1 = Q_{\text{SAT}}(\phi, \rho)_1, UA(\rho, a_1) = (a_2, \ldots, a_p) \text{ is defined, and } b \in \cup_{i=2}^p B_{a_i}\}.$$

Second, for $i \in \{2, \ldots, p\}$ and $q_i \in \mathcal{Q}_i$, $a_i \in \mathcal{A}_i$ we have the set

$$S(i, q_i, a_i) \stackrel{\text{def}}{=} \{\langle R, b\rangle \in S \mid q_i = Q_{\text{SAT}}(\phi, r)_i \text{ and } b \in B_{a_i}\}.$$

Proof of 4.2(1). Suppose $\phi \in$ SAT. Let $(F_1, \ldots, F_p)$ be prover strategies with which V accepts with probability 1. Consider the collection of sets

$$\mathcal{C} = \{S(i, q_i, F_i(q_i)) \mid i \in \{1, \ldots, p\}, q_i \in \mathcal{Q}_i\}.$$

Then $\mathcal{C}$ covers and $|\mathcal{C}| = \sum_{i=1}^{p} \mid |\mathcal{Q}_i|$, as desired.

Proof of 4.2(2). Suppose $\phi \notin$ SAT, and let $\mathcal{C} = \{C_k \mid k \in K\}$ be a cover of S. We begin by summarizing some notation and definitions that will be used in the proof.

$$\begin{aligned}
\mathcal{A}(i, q_i) &= \{a_i \in \mathcal{A}_i \mid S(i, q_i, a_i) \in \mathcal{C}\} && \text{for } i \in \{1, \ldots, p\}, q_i \in \mathcal{Q}_i \\
\text{weight}(i, q_i) &= |\mathcal{A}(i, q_i)| && \text{for } i \in \{1, \ldots, p\}, q_i \in \mathcal{Q}_i \\
\text{weight}(\rho) &= \textstyle\sum_{i=1}^{p} \text{weight}(i, Q_{\text{SAT}}(\phi, \rho)_i) && \text{for } \rho \in R \\
G &= \{\rho \in R \mid \text{weight}(\rho) \leq l\} \\
\delta &= \frac{|G|}{|R|}
\end{aligned}$$

Those random strings $\rho \in R$ appearing in G are called *good.*

Intuitively, $\mathcal{A}(i, q_i)$ is the set of answers to question q_i which are indicated by the cover $\mathcal{C}$. Then weight(i, q_i) is the number of different answers specified by $\mathcal{C}$ to q_i, and weight(ρ) is the number of strings which are answers to some question specified by R. It is important to note that weight(i, q_i) could be more than 1, so that $\mathcal{C}$ does not specify a unique answer to each question, and thus does not directly define "strategies" for the provers. A random string ρ is *good* if the number of answers to the questions it specifies is at most l.

Lemma 4.3 *Fix a good random string $\rho \in G$. Let $q_i = Q_{\text{SAT}}(\phi, \rho)_i$ for $i \in \{1, \ldots, p\}$. Then there exist $a_1 \in \mathcal{A}(1, q_1), \ldots, a_p \in \mathcal{A}(p, q_p)$ such that $C_{\text{SAT}}(\phi, \rho, a_1, \ldots, a_p) =$ accept.*

Proof:

Let $\mathcal{A}(1) = \{a \in \mathcal{A}(1, q_1) \mid UA(\rho, a)$ is defined$\}$. For $i \in \{2, \ldots, p\}$ let $\mathcal{A}(i) = \mathcal{A}(i, q_i)$. $\mathcal{C}$ is a cover of S, so the sets $S(i, a_i, a)$, for $i \in \{1, \ldots, p\}$ and $a \in \mathcal{A}(i)$, must cover $\{\langle R, b\rangle \in S \mid b \in B\}$. We now "project" this cover onto the second coordinate. That is, let $\mathcal{D}$ consist of the sets $\{b \in B\}$ $\langle \rho, b\rangle \in S(i, q_i, a)$, for $i \in \{1, \ldots, p\}$ and $a \in \mathcal{A}(i)$. $\mathcal{D}$ is a cover of B all the sets of which are drawn from

$$\{B_i \mid i \in I\} \bigcup \{\overline{\cup_{j \in J} B_j} \mid |J| = p - 1, J \subset I\}$$

so that we may apply lemma 4.1. Since $|\mathcal{D}| \leq l$, there is a sequence $(a_2, \ldots, a_p)$ so that $\overline{\cup_{j=2}^{p} B_{a_i}}, B_{a_2}, \ldots, B_{a_p} \in \mathcal{D}$. The (Answer space disjointness) condition implies that

$$a_2 \in \mathcal{A}(2), \ldots, a_p \in \mathcal{A}(p).$$

Moreover, there must be an $a_1 \in \mathcal{A}(1)$ such that $UA(\rho, a_1) = (a_2, \ldots, a_p)$.

□

Claim 4.1 *There exist provers $F_1, \ldots, F_p$ which make V accept ϕ with probability $\geq \delta \cdot l^{-p}$.*

Proof: For $i \in \{1, \ldots, p\}$ and $q_i \in \mathcal{Q}_i$ order the elements of $\mathcal{A}(i, q_i)$ in some canonical fashion. For each $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, l\}$ we then define a prover strategy $F_{i,j} : \mathcal{Q}_i \to \{0,1\}^a$ as follows: $F_{i,j}(Q)$ is the j-th element of $\mathcal{A}(i, q_i)$ if this set has size at least j, and undefined otherwise.

If $\rho \in G$ then $\text{weight}(i, Q_{\text{SAT}}(\phi, \rho)_i) = |\mathcal{A}(i, Q_{\text{SAT}}(\phi, \rho)_i)| \leq l$ for each $i \in \{1, \ldots, p\}$. By lemma 4.3 it follows that for each $\rho \in G$, there exist $j_1, \ldots, j_p \in \{1, \ldots, l\}$ so that

$$C_{\text{SAT}}(\phi, \rho, F_{1,j_1}(Q_{\text{SAT}}(\phi, \rho)_1) \ldots F_{p,j_p}(Q_{\text{SAT}}(\phi, \rho)_p)) = \text{accept}.$$

Thus there exist $j_1, \ldots, j_p \in \{1, \ldots, l\}$ so that

$$\left|\{R \in G \mid C_{\text{SAT}}(\phi, \rho, F_{1,j_1}(Q_{\text{SAT}}(\phi, \rho, 1)) \ldots F_{p,j_p}(Q_{\text{SAT}}(\phi, \rho)_p)) = \text{accept}\}\right| \geq l^{-p} \cdot |G|\,,$$

and thus $F_{1,j_1}, \ldots, F_{p,j_p}$ are yield the desired acceptance probability. □

Lemma 4.4 $|\mathcal{C}| \geq (1 - \delta) \cdot \frac{l}{p} \cdot \sum_{i=1}^{p} |\mathcal{Q}_i|$.

Proof: We know that the sets $\mathcal{Q}_1, \ldots, \mathcal{Q}_p$ are all of the same size. Let α be this common size. Making use of the (Uniformity) condition (cf. definition 4.2) we have

$$\begin{aligned}
\sum_{\rho \in R} \text{weight}(\rho) &= \sum_{\rho \in R} \sum_{i=1}^{p} \text{weight}(i, Q_{\text{SAT}}(\phi, \rho)_i) \\
&= \frac{|R|}{\alpha} \cdot \sum_{i=1}^{p} \sum_{q_i \in \mathcal{Q}_i} \text{weight}(i, q_i) \\
&= \frac{|R|}{\alpha} \cdot |\mathcal{C}|\,.
\end{aligned}$$

On the other hand, $\sum_{\rho \in R} \text{weight}(\rho) \geq \sum_{\rho \in G} \text{weight}(\rho) = (1 - \delta) \cdot |R| \cdot l$. Thus $|\mathcal{C}| \geq [1 - \delta]\alpha l$. But $\alpha = \frac{1}{p} \sum_{i=1}^{p} |\mathcal{Q}|_i$, which proves the claim. □

The bound $\text{opt}_{\mathcal{S}_\phi} \geq (1 - \epsilon l^p) \cdot \frac{l}{p} \cdot \sum_{i=1}^{p} |\mathcal{Q}_i|$ now follows because $(\vec{S}') \cdot l^{-p}$ is at most the error probability ϵ. □

Application of lemma 4.2 to the (m, l) set systems of §4.1.1 yields theorems 4.3 through 4.6:

Proof of Theorem 4.3: Fix $c > 0$ and set $l = 10c$. Then select k so that $2^{-k} l^4 < \frac{1}{2}$. Invoking theorem 4.9,

$$\text{SAT} \in \text{NP} \subset \text{MIP}[4, O(\log n), q, O(\log n), 2^{-k}]$$

where the verifier V_{SAT} may be assumed to be canonical. Now, consider a formula ϕ. Using the (m, l) set system construction given in [74] (cf. theorem 4.7), the set system $\mathcal{S}_\phi$ may be constructed in time polynomial in $|\phi|$. We have that $c < (1 - \epsilon l^p)\frac{l}{p}$ so that approximation of SET COVER to within c is enough to determine if $\phi \in$ SAT, as desired. □

Proof of Theorem 4.4: (Lund and Yannakakis [74] suggest this randomized construction.) Apply the proof of [74] using the randomized (m, l) set system construction of theorem 4.8. Since one cannot be certain that the set system so constructed is indeed an (m, l) system, this plan yields a probabilistic algorithm for SAT with two-sided error (BP $\cdot$ $\mathcal{C}$ type error). Since SAT is self reducible (see [13, 14], for example), this may be (naturally) altered to yield a one-sided probabilistic algorithm by incrementally instantiating the variables of ϕ and checking each new instantiation with the two-sided randomized algorithm. Specifically, the two-sided algorithm may be "pumped-up" to yield error at most 2^{-n}. Given $\phi(x_1, \ldots, x_n)$, one may apply this new, robust algorithm to both $\phi(0, x_2, \ldots, x_n)$ and $\phi(1, x_2, \ldots, x_n)$. If $\phi \in$ SAT, at least one of these is likely to be accepted and the the process can be repeated with the next variable. When $\phi \in$ SAT, this instantiation process provides a witness with high probability. Clearly, when $\phi \notin$ SAT, the process does not provide a witness. This gives a one-sided probabilistic algorithm for SAT. □

Proof of Theorem 4.5: Fix $c \in (0, \frac{1}{4})$. From theorem 4.9,

$$\text{SAT} \in \text{MIP}[4, O(\log n \log\log n), q, O(\log n \log\log n), \log^{-7} n]$$

where the verifier may be assumed to be canonical. Let $\beta \in (0, \frac{1}{4} - c)$ and set $l = c(r + 2a)\beta$. Using the probabilistic (m, l) set system construction given in theorem 4.8), the set system $\mathcal{S}_\phi$ may be constructed (with high probability) in time $n^{O(\log\log n)}$. Recall that the *size* of the set system involved is $N = O(2^{l+2a+r})$. Notice that when $c \log_2 N < (1 - \epsilon l^p)\frac{l}{p}$, we may use the hypothesized $c \log n$-approximation algorithm to solve SAT in randomized time $n^{O(\log\log n)}$. It remains to check that $c \log_2 N < (1 - \epsilon l^5)\frac{l}{5}$. We have that

$$c \log_2 N = cl + c(2a + r) + O(1) = cl + \beta l + O(1) = \frac{l}{4} + O(1).$$

But $\epsilon l^4 = o(1)$ so that $(1 - \epsilon l^4)\frac{l}{4} = (1 - o(1))\frac{l}{4} > c \log_2 N$, as desired. As above, this may be altered to provide one-sided error and we conclude that NP $\subset$ RTIME$[n^{O(\log\log n)}]$, as desired. □

Proof of Theorem 4.6: Fix $c \in (0, \frac{1}{8})$. From lemma 4.9,

$$\text{SAT} \in \text{MIP}[4, O(\log n \log\log n), q, O(\log n \log\log n), \log^{-7} n]$$

where the verifier may be assumed to be canonical. Let $\beta \in (0, \frac{1}{4} - 2c)$ and set $l = c(r + 2a)\beta$. Using the deterministic (m, l) set system construction given in [74] (cf. theorem 4.7), the set system $\mathcal{S}_\phi$ may be constructed in time $n^{O(\log\log n)}$. Recall that the *size* of the set systems involved is $N = O(2^{2l+2a+r})$. Notice

that when $c \log_2 N < (1 - \epsilon l^p)\frac{l}{p}$, we may use the hypothesized $c \log n$-approximation algorithms to solve SAT in time $n^{O(\log \log n)}$. It remains to check that $c \log_2 N < (1 - \epsilon l^p)\frac{l}{p}$. We have that

$$c \log_2 N = 2cl + c(2a + r) + O(1) = \beta l + 2cl + O(1) = \frac{l}{4} + O(1).$$

But $\epsilon l^4 = o(1)$ so that $(1 - \epsilon l^4)\frac{l}{4} = (1 - o(1))\frac{l}{4} > c \log_2 N$, as desired.

□

4.2 Other Lower Bounds

The holographic proof machinery of Chapter 3 may be applied to improve a number of other lower bounds for approximation algorithms. To begin with, there are a number of problems directly related to SET COVER (see [74, 69]): HITTING SET, HYPERGRAPH TRANSVERSAL, DOMINATING SET, MINIMUM EXACT COVER. We shall also apply the MIP machinery of theorem 3.1 to the QUARTIC PROGRAMMING problem. We also work with problems the reductions for which naturally operate on PCP system related to the containment NP $\subset$ PCP$[p, O(\log n), O(\log n), 1, \frac{1}{2}]$ for a specific constant p. Theorem 3.2 shall apply in these cases. Problem of this sort are CLIQUE, CHROMATIC NUMBER, and MAX 3SAT (and those problems in MAX-SNP [76]).

4.2.1 Quartic Programming

Quartic programming is the problem of maximizing an element $f \in \mathbb{Q}[x_1, \ldots, x_n]$ over a convex body $\mathfrak{C}$ defined by linear constraints $A\vec{x} = \vec{b}$. For "continuous" optimization problems, we follow [5, 88, 87] and abandon the normal notion of approximating within a *factor* of $\max_{\mathfrak{C}} f$ and adopt the following notion. A $\delta \in [0, 1]$ approximation algorithm for a continuous approximation problem (specifically quartic and quadratic programming in this thesis) is a polynomial time algorithm which produces a number α so that $\|\max_{\mathfrak{C}} f - a\| \leq \delta \|\max_{\mathfrak{C}} f - \min_{\mathfrak{C}} f\|$. Notice that in this framework, an exact algorithm is a 0-approximation algorithm and any algorithm which always outputs an element of $\mathfrak{C}$ achieves factor 1. Since quadratic programming is a special case of quartic programming, we have the following results from [19, 42]:

Theorem 4.10 *There exists $c > 0$ so that no c-approximation algorithms for quartic programming exists unless* P = NP.

Theorem 4.11 *There exists a constant c so that no $(1 - 2^{\log^c n})$-approximation algorithm for quartic programming exists unless* NP $\subset$ DTIME$[n^{\text{poly} \log n}]$.

Using the machinery of Chapter 3 and reworking the proofs in [19] one can improve the first of these results to the following.

Theorem 4.12 *Let $c \in (0,1)$ be constant. Then there is no c-approximation algorithms for quartic programming unless* P = NP.

4.2.2 MAX CLIQUE

Zuckerman [90] shows that NP $\subset$ PCP$^{av}[t, O(\log n), O(\log n), 1, \frac{1}{2}]$ implies that MAX CLIQUE cannot be approximated to within $n^{\frac{1}{t+1}}$ unless NP $\subset$ BPP. Coupling this with theorem 3.2 above yields the following.

Theorem 4.13 *There is no $n^{\frac{1}{30}}$-approximation algorithm for* MAX CLIQUE *unless* NP $\subset$ BPP.

This improves previous results (obtained by applying the reduction of [38] to the proof systems of [3] with pseudo-random error-reduction machinery like that described in [30, 64, 16]) which concluded that approximating MAX CLIQUE to within n^{ϵ} was NP-complete for some (small) ϵ.

4.2.3 CHROMATIC NUMBER

Lund and Yannakakis, in [74], give lower bounds for approximating CHROMATIC NUMBER: the show that for some constant ϵ close to zero, approximating CHROMATIC NUMBER to within n^{ϵ} is NP-complete. Applying their proofs and machinery from [90] to the proof system of theorem 3.2 yields the following results.

Theorem 4.14 *There is no $n^{\frac{1}{140}}$-approximation algorithm for* CHROMATIC NUMBER *unless* NP $\subset$ BPP.

Theorem 4.15 *There is no $n^{\frac{1}{121}}$-approximation algorithm for* CHROMATIC NUMBER *unless*

$$\text{NEXP} \subset \text{BPEXP}.$$

4.2.4 MAX 3SAT and MAX-SNP

Again relying on theorem 3.2, we give improved hardness results for approximating MAX 3SAT. Tracing through the construction of [3, 18], we have the following.

Theorem 4.16 *There is no $\frac{113}{112}$-approximation algorithm for* MAX 3SAT *unless* P = NP.

Theorem 4.17 *There is no $\frac{94}{93}$-approximation algorithm for* MAX 3SAT *unless* EXP = NEXP.

4.3 Recent Improvements

Many of the lower bounds discussed in this Chapter have undergone spectacular improvement since the development of the machinery we have described. As described in §3.5, the basic proof system machinery has been tightened, effecting an improvement in the associated lower bounds. In some cases, new reductions have been discovered which prove stronger conclusions.

Since the SET COVER problem has played a central role in this exposition, let us start there. Uriel Feige, in [37], has actually shown a $\ln n$ threshold for approximability of SET COVER: he proves that there is no $(1-\epsilon)\ln n$-approximation algorithm for SET COVER unless NP $\subset$ DTIME$[n^{O(\log\log n)}]$. Also, new deterministic constructions of (m, l) set systems have been given by Naor, Schulman, and Srinivasan [75].

The observation on the part of Feige that reductions like the one described at the end of §2.1.1 should be moulded around a new parameter, *free bits*, insitgated a second wave of results. In some cases, focus shifted away from the reduction machinery and back to the proof system architecture. Substantial improvements were gleaned by mutating the available PCP and MIP proof systems on a case-by-case basis in order they might engage most favorably with specific reductions. Bellare has written a survey article discussing these advances [15].

Following much work [23, 90, 39, 20, 17, 61] Håstad [60] demonstrated the following.

Theorem 4.18 *For all $\epsilon > 0$, there is no $n^{1-\epsilon}$-approximation algorithm for* MAX CLIQUE *unless* NP $=$ CORP.

Notice that this is impressively close to the $\frac{n}{\log^2 n}$- approximation algorithm of Boppana and Halldórsson [26].

CHROMATIC NUMBER has also gathered much attention [39, 48, 67, 17], culminating in [41] where Feige and Kilian show the following.

Theorem 4.19 *For all $\epsilon > 0$, there is no $n^{\frac{1}{3}-\epsilon}$-approximation algorithms for* CHROMATIC NUMBER *unless* NP $=$ CORP.

The best known approximation algorithm for CHROMATIC NUMBER is due Halldórsson [54] and achieves factor $\frac{n(\log\log n)^2}{\log^3 n}$.

MAX 3SAT has also seen improvement. Following [39, 20], Bellare, Goldreich, and Sudan [17] have shown the following.

Theorem 4.20 *There is no* 1.038*-approximation algorithm for* MAX 3SAT *unless* P $=$ NP.

Sorkin, Sudan, Trevisan and Williamson [85] have given a 1.258-approximation algorithms for MAX 3SAT.

Chapter 5

Relativization and the Random Oracle Hypothesis

The notion of relativization was introduced by Baker, Gill, and Solovay [12] in an attempt to explain the difficulty of the famous $P \stackrel{?}{=} NP$ question. The attaching of oracles to different classes of machines, in general, is a method for exaggerating (perhaps small) differences in the computational capacity of these classes. One way to lend credence to a conjectured relationship between two complexity classes is to exhibit an oracle relative to which the conjecture holds. Thus, the presentation of *contradictory relativizations* of a relationship between two complexity classes has been a standard tool for arguing the difficulty of precisely determining that relationship. The notion of relativization was strengthened by the consideration of random oracles [22]. In the words of Bennett and Gill:

> ... random oracles, by their very structurelessness, appear more benign and less likely to distort the relations among complexity classes than the other oracles used in complexity theory and recursive function theory, which are usually designed expressly to help or frustrate some class of computations.

This led them to formulate the RANDOM ORACLE HYPOTHESIS [22]: *the relationship between two natural complexity classes is preserved with probability 1 under relativization by a random oracle.* In this new framework, a conjectured relationship may be supported by showing that it holds with probability 1 relative to a random oracle. Obviously, this framework precludes the existence of contradictory (probability 1) relativizations.

Counter-examples to the random oracle hypothesis have been demonstrated and discussed in [55, 56, 28, 57, 70, 79]. Recently, the random oracle hypothesis suffered a particularly crippling blow: the classes IP and PSPACE were shown to be equal [73, 82] despite separation with probability 1 [47, 28]. This proof that IP = PSPACE relies heavily on algebraic techniques, the cause of this nonrelativizing behavior. The class

PSPACE has recently been given a new characterization in terms of *Probabilistically Checkable Debate Systems* [31, 33] also using such algebraic techniques. We examine the relativized behavior of IP and PSPACE in comparison with the classes defined by these debate systems. We determine a natural boundary (in terms of certain parameters of the debate systems) separating direct-simulability and inequality (with probability 1). In addition to offering more evidence that these algebraic techniques do not relativize, these boundaries indicate that this new characterization of PSPACE is essentially stronger than the characterization of PSPACE by interactive proof systems–i.e., under relativization by a random oracle, the class of languages recognized by these debate systems is strictly smaller than that recognized by interactive proof systems. We also study these relationships at the EXP level.

Oracles are attached to given enumerations of machines. When we speak of $\mathcal{C}^O$ where $\mathcal{C}$ is a complexity (language) class and O an oracle, we shall mean $\{\mathcal{L} \mid \mathcal{L} = L(M_i^O)\}$ where $\{M_i\}$ is an enumeration of machines such that $\{L(M_i)\} = \mathcal{C}$.

Recently, using the machinery of [3], Condon *et. al.* gave a new characterization of PSPACE in terms of *Probabilistically Checkable Debate Systems*, defined below.

Definition 5.1 *For a function* $f : \Sigma^* \to \Sigma^*$, *let* $f\langle x \rangle \stackrel{def}{=} f(x) \cdot x$. *A* k-player *is a function* $P : \Sigma^* \to \Sigma^k$. *Two* k*-players,* P_1 *and* P_2, *define an* l-debate $D_l(P_1, P_2) \stackrel{def}{=} \overbrace{P_1\langle P_2\langle P_1 \ldots \langle}^{l} \Lambda \rangle \ldots \rangle\rangle$.

Definition 5.2 ([32, 33]) *Define* PCDS$[r(n), a(n)]$ *to be the class of languages* L *for which there exists a probabilistic polynomial time Turing machine* V *and polynomials* q *and* l *so that*

- $x \in L \Rightarrow \exists P_1, \forall P_2, \Pr_{R \in coins}\left[V^{D(P_1,P_2)}[x; R] \text{ accepts}\right] = 1$
- $x \notin L \Rightarrow \forall P_1, \exists P_2, \Pr_{R \in coins}\left[V^{D(P_1,P_2)}[x; R] \text{ accepts}\right] < \frac{1}{3}$

where P_1 *and* P_2 *are* $q(n)$*-players,* $D(P_1, P_2) = D_{l(n)}(P_1, P_2)$ *and, in either case, the verifier* V *uses at most* $O(r(n))$ *random bits and examines at most* $O(a(n))$ *bits of* $D(P_1, P_2)$, *the debate generated by the two players* P_1 *and* P_2. *If we change the reject criteria so that the second player acts randomly, that is*

- $x \notin L \Rightarrow \forall P_1, \Pr_{R \in coins, P_2}\left[V^{D(P_1,P_2)}[x; R] \text{ accepts}\right] < \frac{1}{3}$

then we obtain the class of languages with Random Probabilistically Checkable Debate Systems [33] which we denote RPCDS$[r(n), a(n)]$.

As mentioned above, we have the following two theorems relating these debate systems and PSPACE.

Theorem 5.1 ([32]) PSPACE = PCDS$[\text{poly } n, \text{poly } n]$ = PCDS$[\log n, 1]$.

Theorem 5.2 ([33]) PSPACE = RPCDS$[\text{poly } n, \text{poly } n]$ = RPCDS$[\log n, 1]$.

We concentrate on the behavior of these classes with respect to a random oracle $O \in \Omega = 2^{\Sigma^*}$. The probability measure μ on Ω is defined by independently placing each string in the oracle with probability $\frac{1}{2}$. We begin by considering the relationship between PCDS$[r(n), a(n)]$ and PSPACE.

5.1 The Relativized Relationship between PCDS$[r(n), a(n)]$ and PSPACE

Since we are comparing PSPACE with smaller classes we consider PSPACE to be provided with the weak oracle-access mechanism, that is the oracle tape is a work tape.

Theorem 5.3 $\forall O \subseteq \Sigma^*, \text{PCDS}^O[0, \text{poly}\, n] = \text{PSPACE}^O$.

Proof: By simulation. □

Theorem 5.4 $\forall k, \Pr_{O\in\Omega}\left[\text{PSPACE}^O = \text{PCDS}^O[\text{poly}\, n, n^k]\right] = 0$.

Proof:

We prove in the lemma below that with probability 1, NP^O is not even contained in $\text{PCDS}^O[\text{poly}\, n, n^k]$. Since $\forall O, \text{NP}^O \subseteq \text{PSPACE}^O$, this shows that, with probability 1, $\text{PCDS}^O[\text{poly}\, n, n^k]$ and PSPACE^O are different.

Lemma 5.1 $\forall k, \Pr_{O\in\Omega}\left[\text{NP}^O \subseteq \text{PCDS}^O[\text{poly}\, n, n^k]\right] = 0$.

Proof: For an oracle O, define

$$\hat{O} = \left\{x \mid \forall t \in \{0, \ldots, |x| - 1\}, x10^t \in O\right\}.$$

A polynomial-time machine with access to O can efficiently sample from $\hat{O}$. If O is a random oracle, then $\forall x, \Pr_{O\in\Omega}\left[x \in \hat{O}\right] = \frac{1}{2^{|x|}}$ so that $\forall n, \text{Exp}_{O\in\Omega}\left[\left|\hat{O} \cap \Sigma^n\right|\right] = 1$. For an oracle A, define

$$L_\exists(A) = \left\{1^n \mid \exists y \in \Sigma^{n^{2k}} \cap A\right\}.$$

Clearly, $\forall O, L_\exists(\hat{O}) \in \text{NP}^O$. We show that $\Pr_{O\in\Omega}\left[L_\exists(\hat{O}) \in \text{PCDS}^O[\text{poly}\, n, n^k]\right] = 0$. Fix an enumeration of $\text{PCDS}^O[\text{poly}\, n, n^k]$ verifiers $\{V_i \mid i \in \text{N}\}$. Let V_i be a verifier of this collection which, for $n \geq n_0$, takes at most n^i time, queries at most cn^k debate bits and uses some fixed polynomial, $r(n)$, amount of randomness. For $m, i \in \text{N}$, define

$$\Omega_m^{(s)} = \left\{O \in \Omega \mid \left|\hat{O} \cap \Sigma^m\right| = s\right\}.$$

Then $\mu(\Omega_m^{(0)}) = (1 - \frac{1}{2^m})^{2^m} \approx \frac{1}{e}$. Let n_1 be large enough so that $\frac{2 \cdot n_1^i \cdot 2^{cn_1^k}}{2^{n_1^{2k}}} < \frac{2}{3}$. Let $n > \tilde{n} \stackrel{\text{def}}{=} \max(n_0, n_1)$ and consider the behavior of V_i^O on 1^n with an oracle O selected from $\Omega_{n^{2k}}^{(0)}$. One of the following three cases applies:

1. If $\Pr_{O\in\Omega_{n^{2k}}^{(0)}}\left[\exists P_1, \forall P_2, \Pr_{R\in\text{coins}}\left[V_i^{O,D(P_1,P_2)}[1^n; R] \text{ accepts}\right] = 1\right] \geq \frac{1}{4}$, then

$$\Pr_{O\in\Omega}\left[\exists P_1, \forall P_2, \Pr_{R\in\text{coins}}\left[V_i^{O,D(P_1,P_2)}[1^n; R] \text{ accepts}\right] = 1 \wedge 1^n \notin L_\exists(\hat{O})\right] \geq \frac{1}{4}\Pr_{O\in\Omega}\left[O \in \Omega_{n^{2k}}^{(0)}\right] \approx \frac{1}{4e}. \quad (5.1)$$

(Recall that $\mu(\Omega^{(0)}_{n^{2k}}) \approx \frac{1}{e}$.)

2. If

$$\Pr_{O \in \Omega^{(0)}_{n^{2k}}} \left[\exists P_1, \forall P_2, \Pr_{R \in \text{coins}} \left[V_i^{O, D(P_1, P_2)}[1^n; R] \text{ accepts} \right] \in [\frac{1}{3}, 1) \right] \geq \frac{1}{4e} \tag{5.2}$$

then V_i is behaving improperly, and evidently does not accept $L_\exists(\hat{O})$ for this $\frac{1}{4e}$ fraction of oracles.

3. If $\Pr_{O \in \Omega^{(0)}_{n^{2k}}} \left[\forall P_1, \exists P_2, \Pr_{R \in \text{coins}} \left[V_i^{O, D(P_1, P_2)}[1^n; R] \text{ accepts} \right] < \frac{1}{3} \right] \geq 1 - \frac{1}{2e}$, then we show that this set of oracles on which V_i is successful induces a set of oracles on which V_i errs. To begin with, we show that for any oracle O, most questions that V_i asks of O are asked on very few random strings. Fix an oracle O. Let us consider the behavior of V_i on a particular random string R. Considering all of the possible 2^{cn^k} responses to V_i's cn^k queries[1] to $D(P_1, P_2)$ and noting that on any one path V_i may only query n^i strings of O, we have that on R there are a total of at most $n^i \cdot 2^{cn^k}$ strings of O that V_i might query. We then have that

$$\Pr_{q \in \Sigma^{n^{2k}}} \left[V_i^O[1^n; R] \text{ queries } q \right] \leq \frac{n^i \cdot 2^{cn^k}}{2^{n^{2k}}}.$$

Define

$$\mathcal{R}(Q, O) \stackrel{\text{def}}{=} \left\{ R \in \{0,1\}^{r(n)} \mid \exists q \in Q, \exists D \subseteq \Sigma^*, V_i^{O,D}[1^n; R]) \text{ queries } q \right\}.$$

Then

$$\operatorname*{Exp}_{q \in \Sigma^{n^{2k}}} \left[|\mathcal{R}(\{q\}, O)| \right] \leq \frac{n^i \cdot 2^{r(n)} \cdot 2^{cn^k}}{2^{n^{2k}}}.$$

Invoking Markov's inequality yields

$$\forall O, \Pr_{q \in \Sigma^{n^{2k}}} \left[|\mathcal{R}(\{q\}, O)| \geq \frac{2 \cdot n^i \cdot 2^{r(n)} \cdot 2^{cn^k}}{2^{n^{2k}}} \right] \leq \frac{1}{2}.$$

Define $S_q \stackrel{\text{def}}{=} \{q1, q10, \ldots, q10^{|q|}\}$. Then, because $\forall q_1 \neq q_2 \in \Sigma^{2cn^k}, S_{q_1} \cap S_{q_2} = \emptyset$ we have that

$$\forall O, \Pr_{q \in \Sigma^{n^{2k}}} \left[|\mathcal{R}(S_q, O)| > \frac{2 \cdot n^i \cdot 2^{r(n)} 2^{cn^k}}{2^{n^{2k}}} \right] \leq \frac{1}{2}.$$

[1] There are at most 2^{cn^k} responses to V_i's queries even if V_i is *adaptive* (so that the $i+1$st query may depend on the answer to the ith query).

Now, define $\Omega_m^{(1)} \stackrel{\text{def}}{=} \left\{O \in \Omega \mid \left|\hat{O} \cap \Sigma^m\right| = 1\right\}$. Then $\mu(\Omega_m^{(1)}) \approx \frac{1}{e}$. Let $E(O)$ be the event that $\forall P_1, \exists P_2, \Pr_{R \in \text{coins}}\left[V_i^{O,D(P_1,P_2)}[1^n; R] \text{ accepts}\right] < \frac{1}{3}$. Then we may compute

$$\begin{aligned}
\Pr_{O \in \Omega_{n^{2k}}^{(0)}, q \in \Sigma^{n^{2k}}}\left[E(O) \bigwedge |\mathcal{R}(S_q, O)| < \frac{2 \cdot n^i \cdot 2^{r(n)} \cdot 2^{cn^k}}{2^{n^{2k}}}\right] &\geq \\
\Pr_{O \in \Omega_{n^{2k}}^{(0)}}[E(O)] + \Pr_{O \in \Omega_{n^{2k}}^{(0)}, q \in \Sigma^{n^{2k}}}\left[|\mathcal{R}(S_q, O)| < \frac{2 \cdot n^i \cdot 2^{r(n)} \cdot 2^{cn^k}}{2^{n^{2k}}}\right] - 1 &\geq \\
(1 - \frac{1}{2e}) + (1 - \frac{1}{2}) - 1 &\geq \\
\frac{1}{4}.
\end{aligned}$$

When the two above events occur we can conclude that

$$\forall P_1, \exists P_2, \Pr_{R \in \text{coins}}\left[V_i^{O \cup S_q, D(P_1,P_2)}[1^n; R] \text{ accepts}\right] < \frac{1}{3} + \frac{2 \cdot n^i \cdot 2^{cn^k}}{2^{n^{2k}}}.$$

Notice that if O and q are chosen uniformly from $\Omega_m^{(0)}$ and Σ^m, respectively, then $O \cup S_q$ is uniform on $\Omega_m^{(1)}$. Therefore, for $n > \tilde{n}$,

$$\Pr_{O \in \Omega_{n^{2k}}^{(1)}}\left[\forall P_1, \exists P_2, \Pr_{R \in \text{coins}}\left[V_i^{O \cup S_q, D(P_1,P_2)}[1^n; R] \text{ accepts}\right] < 1\right] \geq \frac{1}{4}.$$

Since $O \in \Omega_m^{(1)}$ implies $1^n \in L_\exists(\hat{O})$,

$$\Pr_{O \in \Omega}\left[\forall P_1, \exists P_2 \Pr_{R \in \text{coins}}\left[V_i^{O, D(P_1,P_2)}[1^n; R] \text{ accepts}\right] \neq 1 \wedge 1^n \in L_\exists(\hat{O})\right] \geq \frac{1}{4} \cdot \frac{1}{e}. \tag{5.3}$$

Let Γ_n be the event that $\exists P_1, \forall P_2, V_i^{O,D(P_1,P_2)}[1^n]$ accepts $\iff 1^n \in L_\exists(\hat{O})$. From (5.1), (5.2) and (5.3) it follows that for $n > \tilde{n}$,

$$\Pr_{O \in \Omega}[\Gamma_n] < 1 - \frac{1}{4e}.$$

Furthermore, for $m > n^i$, Γ_n and Γ_m are independent (or use Lemma 1 of [22]). Hence, for any V_i,

$$\begin{aligned}
\Pr_{O \in \Omega}\left[L(V_i^O) = L_\exists(\hat{O})\right] &\leq \\
\prod_{j=\tilde{n}}^{\infty} \Pr_{O \in \Omega}\left[\Gamma_{2^{ij}}\right] &= 0.
\end{aligned}$$

Finally,

$$\Pr_{O \in \Omega}\left[\exists V_i^O, L(V_i^O) = L_\exists(\hat{O})\right] \leq \sum_i \Pr_{O \in \Omega}\left[L(V_i^O) = L_\exists(\hat{O})\right] = 0$$

so that

$$\Pr_{O\in\Omega}\left[\mathrm{NP}^O \subseteq \mathrm{PCDS}^O[\mathrm{poly}\, n, n^k]\right] = 0.$$

□

Reiterating, from the fact that $\forall O, \mathrm{NP}^O \subseteq \mathrm{PSPACE}^O$ and the above lemma we have the desired theorem.

□

5.2 The Relativized Relationship between $\mathrm{PCDS}[r(n), a(n)]$ and IP

Theorem 5.5 *Consider the two classes* IP *and* $\mathrm{PCDS}[\mathrm{poly}\, n, n^k]$. *We have*

1. $\Pr_{O\in\Omega}\left[\mathrm{IP}^O \subseteq \mathrm{PCDS}^O[\mathrm{poly}\, n, n^k]\right] = 0$,
2. $\Pr_{O\in\Omega}\left[\mathrm{PCDS}^O[\mathrm{poly}\, n, n^k] \subseteq \mathrm{IP}^O\right] = 0$.

Proof:

1. Using Lemma 5.1 and the fact that $\forall O \in \Omega, \mathrm{NP}^O \subseteq \mathrm{IP}^O$ we have the desired statement.
2. This follows from [29] and the fact that $\forall O, \mathrm{coNTIME}^O[n] \subseteq \mathrm{IP}^O \Rightarrow \mathrm{coNP}^O \subseteq \mathrm{IP}^O$.

□

5.3 The Relativized Relationship between $\mathrm{RPCDS}[r(n), a(n)]$ and IP, $\mathrm{PCDS}[r(n), a(n)]$

Theorem 5.6 $\forall O, \mathrm{IP}^O = \mathrm{RPCDS}^O[\mathrm{poly}\, n, \mathrm{poly}\, n] = \mathrm{RPCDS}^O[0, \mathrm{poly}\, n]$.

Proof: By simulation. □

Consider the classes $\mathrm{RPCDS}[\mathrm{poly}\, n, n^k]$ and IP.

Theorem 5.7 $\forall k, \Pr_{O\in\Omega}\left[\mathrm{RPCDS}^O[\mathrm{poly}\, n, n^k] = \mathrm{IP}^O\right] = 0$.

Proof: We have that $\forall O, \mathrm{RPCDS}^O[\mathrm{poly}\, n, n^k] \subseteq \mathrm{PCDS}^O[\mathrm{poly}, n^k]$ so that Lemma 5.1 yields the desired result. □

Theorem 5.8 *For* $a(n) = \omega(\log n)$,

$$\Pr_{O\in\Omega}\left[\mathrm{PCDS}^O[r(n), a(n)] \subseteq \mathrm{RPCDS}^O[\mathrm{poly}\, n, \mathrm{poly}\, n]\right] = 0.$$

Proof: $\forall O, \text{coNTIME}^O[a(n)] \subseteq \text{PCDS}^O[r(n), a(n)]$ but, by argument similar to that of Lemma 5.1, one may show that

$$\Pr_{O \in \Omega} [\exists L \in \text{coNTIME}^O[a(n)] - \text{RPCDS}^O[\text{poly}\, n, \text{poly}\, n]] = 1.$$

□

5.4 The Relativized Relationship between PCDS$[r(n), a(n)]$ and EXP

An oracle equating NP and EXP has been discovered by Heller [62].

Theorem 5.9 ([62]) $\exists O \subseteq \Sigma^*$ *so that* $\text{EXP}^O = \text{NP}^O$.

Fortnow [44, 45] has shown the following theorem relating the existence of an oracle equating EXP and PCP$[O(1), \log, \log n, 1]$ to the P $\stackrel{?}{=}$ NP question.

Theorem 5.10 *If* $\exists O \subseteq \Sigma^*$ *so that* $\text{PCP}^O[O(1), \log n, \log n, 1] = \text{EXP}^O$ *then* P $\neq$ NP.

We prove a similar result for the class PCDS$[\log n, \log n]$.

Theorem 5.11 *If* $\exists O \subseteq \Sigma^*$ *so that* $\text{PCDS}^O[\log n, \log n] = \text{EXP}^O$ *then* P $\neq$ PSPACE.

Proof: Let O be an oracle so that $\text{PCDS}^O[\log n, \log n] = \text{EXP}^O$. Assume, for contradiction that P = PSPACE. Let L be a $\leq_p$-complete language for EXP^O. We show that $L \in \text{P}^O$ and conclude that $\text{P}^O = \text{EXP}^O$, which contradicts the time hierarchy theorem [59]. Let V be a $\text{PCDS}^O[\log n, \log n]$ verifier for L. We construct D^O, a deterministic polynomial time machine so that $L(D^O) = L$. D^O, given input w, writes down the entire computation tree $\mathfrak{T}$ of $V[w]$, answering $V[w]$'s questions to O by actual questions to O and branching at those nodes where $V[w]$ receives debate tape answers. Notice that choice of a pair (P_1, P_2) determines a path in $\mathfrak{T}$. This path is *satisfied* if $V[w]$ accepts with these responses. Because $V[w]$ uses $O(\log n)$ random bits and receives $O(\log n)$ bits back from the debate tape, the total size of $\mathfrak{T}$ is polynomial in $|w|$. $\mathfrak{T}$ contains no queries to O. D^O would now like to determine if $\exists P_1, \forall P_2$, the induced path in $\mathfrak{T}$ is satisfied. Fortunately, this is a PSPACE decision problem, which can be solved in polynomial time because P = PSPACE. Hence, $L \in \text{P}^O$ and $\text{EXP}^O = \text{P}^O$, contradicting the time hierarchy theorem. □

5.5 The Relativized Relationship between MIP$[\cdot, \cdot, \cdot, \cdot, \cdot]$ and EXP

Considering the relationship between NEXP and MIP$[\cdot, \cdot, \cdot, \cdot, \cdot]$ with the same lens we have applied to the relationship between NEXP and PCDS, we obtain the following theorem.

Theorem 5.12 *If there exists an oracle $O \subseteq \Sigma^*$ so that*

$$\mathrm{MIP}^O[O(\log n), O(\log n), O(\log n), O(1), 1 - \frac{1}{\mathrm{poly}\, n}] = \mathrm{EXP}^O$$

then $\mathrm{P} \neq \mathrm{NP}$.

Proof: Let $O \subseteq \Sigma^*$ be so that $\mathrm{MIP}^O[O(\log n), O(\log n), O(\log n), O(1), 1 - \frac{1}{n}] = \mathrm{EXP}^O$. Assume, for contradiction, that $\mathrm{P} = \mathrm{NP}$. We shall conclude that $\mathrm{P}^O = \mathrm{EXP}^O$, contradicting the time hierarchy theorem. Let $L \in \mathrm{EXP}^O$. By hypothesis, $L \in \mathrm{MIP}^O[O(\log n), O(\log n), O(\log n), O(1), 1 - \frac{1}{\mathrm{poly}\, n}]$. Let V^O be a $\mathrm{MIP}^O[O(\log n), O(\log n), O(\log n), O(1), 1 - \frac{1}{\mathrm{poly}\, n}]$ verifier for L. Then consider the polynomial time oracle machine D^O which, on input x, computes the set

$$Q = \{q \mid \exists P_1, \ldots, P_p \exists r, (V^O \leftrightarrow P_1, \ldots, P_p)[x; r] \text{ queries } O \text{ with } q\}.$$

(Notice that $|Q|$ is polynomial since, depending on $P_1, \ldots, P_p$, and R, there are only polynomially many computation paths of $(V^O \leftrightarrow P_1, \ldots, P_p)[x; r]$, each of which may contain some polynomial number of queries.) D^O then queries O to collect the answers $\{a_i\}$ to these questions $Q = \{q_i\}$ and would like to decide if $\exists P_1, P_2$ so that $\Pr_R[(V \leftrightarrow P_1, \ldots, P_p)[x; r] = 1]$ where q_i is answered by a_i. Fortunately this is an NP decision problem, and so is solvable in P. Then $\mathrm{P}^O = \mathrm{EXP}^O$, which contradicts the time hierarchy theorem. Hence $\mathrm{P} \neq \mathrm{NP}$, as desired. □

5.6 Direction for Future Research

The discovery of simulation techniques which do not relativize (with probability 1) is astonishing. This leads us to question the meaning of relativization in general. One would like to distill the essential non-relativizing ingredient of these algebraic techniques. This may be done by presentation of (perhaps contrived) complexity classes with a somehow simpler (algebraic) proof of equality which exhibit this behavior. Alternatively, this may be done by presentation of a new framework (perhaps just a new oracle-access mechanism [44, 45]), analogous to relativization, in which these techniques behave well.

Bibliography

[1] M. Ajtai, J. Komlós, and E. Szemerédi. An $O(n \log n)$ sorting network. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 1–9, Boston, Massachusetts, 25–27 Apr. 1983.

[2] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley & Sons, Inc., 1992.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *33rd Annual Symposium on Foundations of Computer Science*, pages 14–23, Pittsburgh, Pennsylvania, 24–27 Oct. 1992. IEEE.

[4] S. Arora and S. Safra. Probabilistic checking of proofs; a new characterization of NP. In *33rd Annual Symposium on Foundations of Computer Science*, pages 2–13, Pittsburgh, Pennsylvania, 24–27 Oct. 1992. IEEE.

[5] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21:136–153, 1980.

[6] L. Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, pages 421–429, Providence, Rhode Island, 6–8 May 1985.

[7] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 21–31, New Orleans, Louisiana, 6–8 May 1991.

[8] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. In *31st Annual Symposium on Foundations of Computer Science*, volume I, pages 16–25, St. Louis, Missouri, 22–24 Oct. 1990. IEEE.

[9] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[10] L. Babai and S. Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[11] L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *25th Annual Symposium on Foundations of Computer Science*, pages 229–240, Singer Island, Florida, 24–26 Oct. 1984. IEEE.

[12] T. Baker, J. Gill, and R. Solovay. Relativizations of the P = NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[13] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Computer Science*. Springer-Verlag, Berlin, 1988.

[14] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*, volume 22 of *EATCS Monographs on Computer Science*. Springer-Verlag, Berlin, 1990.

[15] M. Bellare. Proof checking and approximation: Towards tight results. *SIGACT News*, 27(1):2–13, March 1996. Guest column in SIGACT News Complexity Theory Column.

[16] M. Bellare, O. Goldreich, and S. Goldwasser. Randomness in interactive proofs. In *31st Annual Symposium on Foundations of Computer Science*, volume II, pages 563–572, St. Louis, Missouri, 22–24 Oct. 1990. IEEE.

[17] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability—towards tight results. In *36th Annual Symposium on Foundations of Computer Science*, pages 422–431, Milwaukee, Wisconsin, 23–25 Oct. 1995. IEEE.

[18] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 294–304, San Diego, California, 16–18 May 1993.

[19] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. Research Report RC 17831, IBM, March 1992.

[20] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 184–193, Montréal, Québec, Canada, 23–25 May 1994.

[21] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 113–131, Chicago, Illinois, 2–4 May 1988.

[22] C. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq$ co-NP^A with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981.

[23] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. *Information and Computation*, 96:77–94, 1992.

[24] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing*, pages 73–83, Baltimore, Maryland, 14–16 May 1990.

[25] R. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1981.

[26] R. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. In J. R. Gilbert and R. Karlsson, editors, *SWAT 90 2nd Scandinavian Workshop on Algorithm Theory*, volume 447 of *Lecture Notes in Computer Science*, pages 13–25, 1990.

[27] J. Y. Cai, A. Condon, and R. Lipton. PSPACE is provable by two provers in one round. *Journal of Computer and System Sciences*, 48, 1994.

[28] R. Chang, B. Chor, O. Goldreich, J. Hartmanis, J. Hastad, D. Ranjan, and P. Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.

[29] B. Chor, O. Goldreich, and J. Håstad. The random oracle hypothesis is false. Manuscript.

[30] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 14–19, Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989. IEEE.

[31] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions (extended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 305–314, San Diego, California, 16–18 May 1993.

[32] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions. In *Proceedings of the Twenty-Fifth ACM Symposium on Theory of Computing*, pages 305–314. ACM, 1993.

[33] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Random debators and the hardness of approximating stochastic functions. DIMACS Technical Report 93-79, Rutgers University, Piscataway, NJ, 1993.

[34] P. Crescenzi and V. Kann. A compendium of NP optimization problems. Available by ftp from `ftp://www.nada.kth.se/Theory/Viggo-Kann/compendium.ps.Z`.

[35] R. M. Dudley. *Real Analysis and Probability*. Chapman and Hall, 1989.

[36] U. Feige. On the success probability of the two provers in one round proof systems. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, University of Chicago, Chicago, Illinois, 30 June–3 July 1991.

[37] U. Feige. A threshold of ln n for approximating set cover. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, Philadelphia, Pennsylvania, 22–24 May 1996. To appear.

[38] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete (preliminary version). In *32nd Annual Symposium on Foundations of Computer Science*, pages 2–12, San Juan, Puerto Rico, 1–4 Oct. 1991. IEEE.

[39] U. Feige and J. Kilian. Two prover protocols—low error at affordable rates (preliminary version). In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 172–183, Montréal, Québec, Canada, 23–25 May 1994.

[40] U. Feige and J. Kilian. Impossibility results for recycling random bits in two-prover proof systems. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 457–468, Las Vegas, Nevada, 29 May–1 June 1995.

[41] U. Feige and J. Kilian. Zero-knowledge and the chromatic number. In *Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity*, Philadelphia, Pennsylvania, 24–26 May 1996.

[42] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems (extended abstract). In *Proceedings of the Twenty Fourth Annual ACM Symposium on Theory of Computing*, pages 733–744, Victoria, British Columbia, Canada, 4–6 May 1992.

[43] L. Fortnow. The complexity of perfect zero-knowledge (extended abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 204–209, New York City, 25–27 May 1987.

[44] L. Fortnow. Oracles, proofs, and checking. Unpublished Manuscript, July 1993.

[45] L. Fortnow. The role of relativization in complexity theory. *Bulletin of the European Association for Theoretical Computer Science*, 52:229–244, February 1994.

[46] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134, 1994.

[47] L. Fortnow and M. Sipser. Are there interactive proofs for co-NP languages? *Information Processing Letters*, 28:249–251, 1988.

[48] M. Fürer. Improved hardness results for approximating the chromatic number. In *36th Annual Symposium on Foundations of Computer Science*, pages 414–421, Milwaukee, Wisconsin, 23–25 Oct. 1995. IEEE.

[49] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, New York, 1979.

[50] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 174–187, Toronto, Ontario, Canada, 27–29 Oct. 1986. IEEE.

[51] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero knowledge proof systems. *Journal of the Association for Computing Machinery*, 38(1):691–729, July 1991.

[52] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, 18(1):186–208, February 1989.

[53] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 59–68, Berkeley, California, 28–30 May 1986.

[54] M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45:19–23, 1993.

[55] J. Hartmanis. Solvable problems with conflicting relativizations. *Bulletin of the EATCS*, 27, 1985.

[56] J. Hartmanis, R. Chang, S. Chari, D. Ranjan, and P. Rohatgi. Relativization: A revisionistic retrospective. *Bulletin of the EATCS*, 47, 1992.

[57] J. Hartmanis, R. Chang, J. Kadin, and S. Mitchell. Some observations about relativization of space bounded computations. *Bulletin of the EATCS*, 35, 1988.

[58] J. Hartmanis, R. Chang, D. Ranjan, and P. Rohatgi. On IP=PSPACE and theorems with narrow proofs. Technical Report 90-1129, Cornell University, May 1990.

[59] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[60] J. Håstad. Clique is indeed hard. Manuscript, 1996.

[61] J. Håstad. Testing of the long code and hardness for clique. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, Philadelphia, Pennsylvania, 22–24 May 1996. To appear.

[62] H. Heller. *Relativized Polynomial Hierarchy Extending Two Levels*. PhD thesis, Universität München, 1981.

[63] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Series in Computer Science. Addison-Wesley, Reading, Massachusetts, 1979.

[64] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *30th Annual Symposium on Foundations of Computer Science*, pages 248–253, Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989. IEEE.

[65] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[66] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[67] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. In *Proceedings of the Second Israel Symposium on Theory and Computing Systems*. IEEE, 1993.

[68] M. Kiwi, C. Lund, A. Russell, D. Spielman, and R. Sundaram. Alternation in interaction. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, Amsterdam, The Netherlands, 28 June–1 July 1994.

[69] P. Kolaitis and M. Thakur. Approximation properties of NP minimization classes. In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, University of Chicago, Chicago, Illinois, 30 June–3 July 1991.

[70] S. Kurtz. On the random oracle hypothesis. *Information and Control*, 57(1):40–47, April 1983.

[71] D. Lapidot and A. Shamir. Fully parallelized multi prover protocols for NEXP-time (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science*, pages 13–18, San Juan, Puerto Rico, 1–4 Oct. 1991. IEEE.

[72] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1983.

[73] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[74] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems (extended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 286–293, San Diego, California, 16–18 May 1993.

[75] M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science*, pages 182–191, Milwaukee, Wisconsin, 23–25 Oct. 1995. IEEE.

[76] C. H. Papadimitiou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[77] C. H. Papadimitriou. Games against nature (extended abstract). In *24th Annual Symposium on Foundations of Computer Science*, pages 446–450, Tucson, Arizona, 7–9 Nov. 1983. IEEE.

[78] S. Phillips and S. Safra. PCP and tighter bounds for approximating MAXSNP. Manuscript, 1992.

[79] M. O. Rabin and D. Scott. Finite automata and their decision problems. In E. F. Moore, editor, *Sequential Machines: Selected Papers*, pages 63–91. Addison-Wesley, 1964.

[80] R. Raz. A parallel repetition theorem. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 447–456, Las Vegas, Nevada, 29 May–1 June 1995.

[81] A. Russell and R. Sundaram. The relativized relationship between probabilistically checkable debate systems, IP, and PSPACE. *Information Processing Letters*, 53:61–68, 1995.

[82] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[83] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. In *29th Annual Symposium on Foundations of Computer Science*, pages 283–290, White Plains, New York, 24–26 Oct. 1988. IEEE.

[84] M. Sipser. The history and status of the P versus NP question. In *Proceedings of the Twenty Fourth Annual ACM Symposium on Theory of Computing*, pages 603–618, Victoria, British Columbia, Canada, 4–6 May 1992.

[85] G. Sorkin, M. Sudan, L. Trevisan, and D. Williamson. In Preparation.

[86] G. Tardos. Multi-prover encoding schemes, and 3-prover interactive proofs. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, Amsterdam, The Netherlands, 28 June–1 July 1994.

[87] S. Vavasis. Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, 57:279–312, 1992.

[88] S. Vavasis. On approximation algorithms for concave programming. In C. A. Floudas and P. M. Pardalos, editors, *Recent Advances in Global Optimization*, pages 3–18. Princeton University Press, 1992. Also TR 90-1172, Department of Computer Science, Cornell University, December, 1990.

[89] O. Verbitsky. Towards the parallel repetition conjecture. In *Proceedings of the Ninth Annual Structure in Complexity Theory Conference*, Amsterdam, The Netherlands, 28 June–1 July 1994.

[90] D. Zuckerman. NP-complete problems have a version that is hard to approximate. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, pages 305–312, San Diego, California, 18–21 May 1993.