Free Bits, PCPs and Non-Approximability – Towards Tight Results

MIHIR BELLARE* ODED GOLDREICH† MADHU SUDAN‡

In honor of Shimon Even's 60th Birthday

Abstract

The first part of this paper presents new proof systems and improved non-approximability results. In particular we present a proof system for NP using logarithmic randomness and two amortized free bits, so that Max Clique is hard within $N^{1/3}$ and Chromatic Number within $N^{1/5}$. We also show hardness of $38/37$ for Max-3-SAT, $27/26$ for Vertex Cover, $82/81$ for Max-Cut, and $94/93$ for Max-2-SAT.

The second part of this paper presents a "reverse" of the FGLSS connection by showing that an NP-hardness result for the approximation of Max Clique to within a factor of $N^{1/(g+1)}$ would imply a probabilistic verifier for NP with logarithmic randomness and amortized free-bit complexity g. We also show that "existing techniques" won't yield proof systems of less than two bits in amortized free bit complexity.

Finally, we initiate a comprehensive study of PCP and FPCP parameters, proving several triviality results and providing several useful transformations.

1 Introduction

The success of the interactive proof based approach to deriving non-approximability results seems beyond question— not only has problem after problem fallen, but results grow succesively stronger. Key to these improvements has been the consideration of new parameters in proof checking complexity such as the number of "free bits" and "amortized free bits." Today, it even seems possible that this approach may lead to *tight* non-approximability results for several popular optimization problems.

This is the topic which this paper investigates. Broadly, we are interested in two things. The first is to exploit as well as possible the existing relations betweens proofs and non-approximation to get improved hardness results. This involves continuing previous work by the construction of new proof systems of improved complexity. The second, which is more novel, is to understand the limits of the relation between proofs and approximation, and the limits to improvements in proof systems. It has lead us to a variety of different kinds of investigations and results. Let us begin with a high level overview.

1.1 Overview of main results

NEW PROOF SYSTEMS AND NON-APPROXIMABILITY RESULTS. We continue previous work [4, 18, 3, 2, 7, 19, 9] by constructing new proof systems of improved complexity. They are based on a new error correcting code called the long code. A central result in this category is a proof system for NP using logarithmic randomness and two amortized free-bits, and directly yielding a $N^{1/3}$ non-approximability factor for Max Clique. We also obtain improved non-approximability results for Chromatic Number and Max-3-SAT, and the first reasonable and explicit constant factor non-approximability results for the Min Vertex Cover problem, Max Cut, and Max-2-SAT. Several of these results are strong enough to indicate that the gap between factors that are attainable by polynomial time algorithms, and those we can indicate are not, is now quite narrow. See Section 3 and Figure 1.

As the above indicates, non-approximability results are getting steadily stronger, especially for Max Clique. How large a Max Clique non-approximability factor can we show? And, in minimizing amortized free-bits, are we on the right track? Are there other ways? The next set of results provides answers to these kinds of questions.

A REVERSE CONNECTION. We essentially show that proof checking is *necessary* to getting non-approximability results for Max Clique. Furthermore, it indicates that not just proof checking, but the minimization of the amortized free-bit complexity is necessary. Roughly, we show that if, for some $f > 0$, Max Clique is NP-hard to approximate within $N^{1/(1+f)}$ then NP has proof systems of (logarithmic randomness and) amortized free-bit complexity f. This result can be viewed as "inverting," in a strong way, the FGLSS-connection. (See Section 4.) So our current efforts are in the right direction.

A LOWER BOUND ON AMORTIZED FREE-BITS. Now that we know

* Department of Computer Science & Engineering, Mail Code 0114, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093. E-mail: mihir@cs.ucsd.edu. This work was done while the author was at the IBM T. J. Watson Research Center.

†Computer Science and Applied Math. Dept., Weizmann Institute of Science, Rehovot, Israel. Supported by grant No. 92-00226 from the Israel–US Binational Science Foundation (BSF), Jerusalem, Israel. Email: oded@wisdom.weizmann.ac.il

‡IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. Email: madhu@watson.ibm.com

we must minimize amortized free-bits, we ask ourselves how low we can take them. Our approach here is to look at current techniques and assess their limitations. We derive lower bounds showing that any proof system using the existing frameworks (of this and previous papers) must use at least two amortized free-bits. Our reverse connection now implies that proving a better than $N^{1/3}$ hardness for Max Clique requires new techniques. See Section 5.

We stress that this last result makes various assumptions about methods, and is intended to show that significantly novel techniques are required to go further. But it does not suggest an *inherent* limitation. Indeed, if we believe Max Clique is hard to approximate within $N^{1-o(1)}$ then our reverse connection says NP has proof systems with arbitrarily small constant amortized free-bit complexity; we are just saying they may be hard to find.

PCP AND FPCP: PROPERTIES AND TRANSFORMS. Probabilistic proofs involve a vast arena of complexity parameters: query complexity, free-bit complexity, amortized free-bit complexity, randomness, and proof sizes to name a few. A better understanding of the basic properties and relations between these parameters would help move us forward. We initiate, accordingly, a systematic investigation of the properties of pcp complexity classes as a function of the parameter values. Besides providing new results we take the opportunity to state and prove a few folklore ones. We focus in particular on "triviality" results. These are results which say that certain parameter combinations yield classes probably not capable of capturing NP. For example, the class of languages recognizable with error $1/2$ and logarithmic randomness using one (non-amortized!) free-bit is in P— so don't expect to prove NP using just one free-bit. (But nothing rules this out when amortization is considered). We also investigate transformations: to reduce the randomness, error or other complexities at various costs. See Sections 6 and 7.

DISCUSSION. The reverse connection does more than guide our choice of parameters. It provides a new conceptual tool because it enables us to reflect, in the language of proof systems, theorems, properties and transformations of graphs, and vice versa. This turns out to be useful and revealing. It also leads, in some cases to new results derived by turning graphs into proof systems via our connection, and then back to graphs via the FGLSS connection, in the process gaining some property.

A related contribution of this work is to distill and formalize the role of randomized reductions. These transforms provide an elegant and concise way of stating connections between proofs and approximability, or just between different kinds of proof systems, and make it easier to manipulate the many connections that exist to derive new results.

VERSIONS. This extended abstract is a very abridged version of our full paper, with almost no proofs. The latest (100+ page) version of our full paper is [8]. It contains complete definitions, proofs, and history.

WHAT FOLLOWS. In later sections we will detail the results sketched here more precisely. First, however, we provide some history, then some definitions.

1.2 History of non-approximability

Early work in non-approximability includes that of Garey and Johnson [22] showing that it is NP-hard to approximate the chromatic factor within a factor less than two. The indication of higher factors, and results for other problems, had to wait for the interactive proof approach which began with [18].

The works of [18, 3, 2] culminated in the proof that there is a constant $\epsilon > 0$ for which approximating Max Clique within N^{ϵ} is NP-hard. This was based on the characterization $\mathrm{NP} = \mathrm{PCP}_{1,1/2}[\log, O(1)]$. The work on improving the value of ϵ progressed by looking a new complexity parameters and constructing new proof systems to minimize them. Thus [7] looked at the average number of query bits; [19] looked at free bits; and finally [9] looked at amortized free bits. The last built new proof systems achieving amortized free-bit complexity three, implying a $N^{1/4}$ hardness for Max Clique assuming $\mathrm{NP} \not\subseteq \mathrm{co}\tilde{\mathrm{R}}\tilde{\mathrm{P}}$.

Arora et. al. [2] showed that there exists a constant $\epsilon > 1$ such that approximating Max-3-SAT within ϵ is NP-hard. (This implied the same for all of Max-SNP [31].) The above mentioned works [7, 19, 9] have found increasingly larger values for the Max-3-SAT non-approximability factor ϵ.

1.3 Related work

Following the presentation of our results of Sections 3 and 4, Arora has also investigated the limitations of proof checking techniques in proving non-approximability results [1]. Like in our free-bit lower bound result, he tries to assess the limitations of current techniques by making some assumptions about these techniques and then showing a lower bound. His focus is on the reductions, which he assumes are "code like." In this setting he can show that one should not expect to prove non-approximability of Max Clique within $N^{1/2}$. In contrast we have a larger lower bound of $N^{1/3}$, but we make different kinds of assumptions about the way proof systems are designed. (The assumptions made by us and by Arora do not seem to be comparable: neither implies the other.)

2 Background and definitions

PROOF SYSTEMS AND PARAMETERS. A probabilistic proof system [20, 18][1] is described by a probabilistic, polynomial time *verifier* V. It takes an input x of length n and tosses coins R. It has oracle access to a $\mathrm{poly}(n)$ length string σ describing the proof: to access a bit it writes a $O(\log n)$ bit address and is

[1] An analogous discussion can be carried out also using the model of transparent proofs due to Babai et. al. [4].

returned the corresponding bit of the proof. Following its computation it will either accept or reject its input x. The accepting probability is the maximum, over all σ, of the probability (over R) that V accepts x on coins R and proof string σ. While the task is typically language recognition, we will, more generally, consider promise problems (A, B) consisting of a set A of "positive" instances and a set B of "negative" instances [15]. (Languages are a special case of promise problems; a language L is represented by the promise problem $(L, \overline{L})$.)

Of interest in the applications are various parameters of the system. The completeness probability $c = c(n)$ and the soundness probability $s = s(n)$ are defined in the usual ways. In case $c = 1$ we say that the system has perfect completeness. The gap is $g = c/s$. The query complexity is the maximum (over all coin tosses and proof strings) of the number of bits of the proof that are examined by the verifier. The free-bit complexity, roughly speaking, is the logarithm of number of possible accepting configurations of V on coins R and input x. (This omits a technical constructivity condition.) For example a verifier which makes 3 queries and accepts iff the parity of the answers is odd has 4 accepting configuration and thus free-bit complexity 2. Either the query or the free-bit complexity may be considered in amortized form: e.g. the amortized free-bit complexity is the free-bit complexity (of a proof system with perfect completeness) divided by the logarithm of the gap. (That is, the number of free-bits needed per factor of 2 increase in the gap.) Also, either the query or free-bit complexity may be considered on the average, the average being over the random string of the verifier.

We use the notation $\mathrm{PCP}_{c,s}[r, q]$ to denote the class of promise problems recognized by verifiers tossing r coins, having query complexity q, and achieving completeness probability c and soundness probability s. $\mathrm{FPCP}_{c,s}[r, f]$ is defined analogously with f being the free-bit complexity. $\overline{\mathrm{PCP}}[r, q]$ is defined analogously with q being the amortized query complexity, and $\overline{\mathrm{FPCP}}[r, f]$ is defined analogously with f the amortized free-bit complexity. Also we sometimes use a more generic notation in which parameters are specified by name; this is pretty self-explanatory.

APPROXIMATION. Recall that an approximation algorithm A for a maximization problem achieves a ratio of $\alpha \in [1, \infty)$ if the output $A(w)$ of the algorithm on instance w is at least the optimum divided by α and at most the optimum. For a minimization problem it is required that $A(w)$ is at most α times the optimum and at least the optimum.

MAX CLIQUE APPROXIMATION. Recall the best known polynomial time approximation algorithm for Max Clique achieves a factor of only $N^{1-o(1)}$ [14], scarcely better than the trivial factor of N. (Throughout the paper, when discussing the Max Clique problem, N denotes the number of vertices in the graph.) There is not even a heuristic algorithm that is conjectured to do better. (The Lovász Theta function had been conjectured to approximate the Max Clique size within $N^{1/2}$, but this conjecture was disproved by Feige [16].) The same situation holds for the chromatic number.

An additional motivation for investigating whether there exist even "weak" approximation algorithms for Max Clique was suggested by Blum [12]: he shows that a factor $N^{1-\epsilon}$ factor approximation algorithm for some constant $\epsilon > 0$ would imply an algorithm for coloring a 3-colorable graph with $O(\log N)$ colors, which is significantly fewer colors than known algorithms use.

GAPS IN CLIQUE SIZE. Hardness of approximation (say of Max Clique) is typically shown via the construction of promise problems with gaps in max clique size. Specifically, let $\mathsf{MaxClique}(G)$ denote the Max Clique size of a graph G, and let $\mathsf{Gap\text{-}Clique}_{c,s}$ be the promise problem (A, B) defined as follows: A is the set of all graphs G with $\mathsf{MaxClique}(G)/N \geq c(N)$, and B is the set of all graphs G with $\mathsf{MaxClique}(G)/N \leq s(N)$. The gap is defined as c/s. Now, a hardness result will typically specify a value of the gap $g(N) = c(N)/s(N)$ for which $\mathsf{Gap\text{-}Clique}_{c,s}$ is NP-hard under a (possibly randomized) Karp reduction. This means that there is no polynomial time algorithm to approximate the Max Clique size of an N node graph within $g(N)$ unless NP has randomized polynomial time algorithms.

Gap problems can be similarly defined for all the other optimization problems we consider. From now on, we discuss approximation in terms of these gap problems.

THE FORWARD CONNECTION. To explain our "reverse" connection for Max Clique it will help to recall the "forward" connection. It relates the non-approximability factor of Max Clique to proof checking complexity. The basic reduction is that of [18], used today in slightly tighter randomized form due to [11, 34]. The sequence of works [18, 3, 2, 7, 19, 9] lead us through a sequence of parameters: query complexity, free bit complexity and, finally, for the best known results, amortized free bit complexity. The final relation is that if NP is in $\overline{\mathrm{FPCP}}[\log, f]$ then the Max Clique size of an N-vertex graph is NP-hard to approximate (under randomized Karp reductions) within a factor of $N^{1/(1+f+\epsilon)}$, for any $\epsilon > 0$.

3 New proof systems and applications

This section is about our non-approximability results and proof systems obtained via the long code. (This is our new error corrcting code.)

3.1 Statements of results

We provide the following new proof systems.

Theorem 3.1 Let $\epsilon > 0$ be arbitrary. Then NP is contained in each of the following–

(1) $\overline{\mathrm{FPCP}}[\log, 2+\epsilon]$.

(2) $\mathrm{FPCP}_{1,s}[\log, 2]$ for $s{=}0.884464$.
(3) $\mathrm{PCP}_{1,1/2}[$ coins$=\log$; queries$=19$; queries$_{\mathrm{av}}=15.58]$ where queries$_{\mathrm{av}}$ is the average number of queries.
(4) $\mathrm{PCP}_{1,s}[\log, 3]$ for $s{=}0.8999$.

The first result above improves the result of [9] showing that $\mathrm{NP} \subseteq \overline{\mathrm{FPCP}}[\log, 3{+}\epsilon]$ for every $\epsilon > 0$. The second minimizes the soundness error one can get using only two free (non-amortized) bits. Both these have applications to approximation. The last two results although not used to obtain any of our non-approximation results, are of some intrinsic interest. Specifically, we look at the problem of how few bits of the proof one need query to detect an error $1/2$ of the time, and at how low an error one can get using only three queries.

For the applications to Max-SNP we construct a special verifier. It is a simple verifier for NP which achieves soundness error of about 86% while performing one of two very simple tests.

Proposition 3.2 (The MaxSNP Verifier): For any $\gamma > 0$ and for any language $L \in \mathrm{NP}$, there exists a verifier V_{SNP} for L such that

- V_{SNP} uses logarithmic randomness and is perfectly complete;
- V_{SNP} has soundness error $1 - \frac{6144}{44969} + \gamma$; and
- on access to an oracle π, the verifier V_{SNP} performs one of the following actions:
  (1) Parity check: V_{SNP} makes three queries q_1, q_2 and q_3 to the proof π and rejects if $\pi(q_1) \oplus \pi(q_2) \neq \pi(q_3)$.
  (2) RMB check: V_{SNP} makes four queries q_1, q_2, q_3 and q_4 to the proof π and rejects if $\pi(q_1) \cdot \pi(q_2) \neq \pi(q_3) \oplus \pi(q_4)$.

  Furthermore, the probability (over its coin tosses) that V_{SNP} performs a parity check is $q \stackrel{\mathrm{def}}{=} \frac{28585}{44969} \approx 0.6356$ and the probability that V_{SNP} performs a RMB check is $1 - q$.

The next theorem states our non-approximability results. Figure 1 summarizes the results and compares them to past work and the factors achieved by the best known approximation algorithms.

Theorem 3.3 Let $\epsilon > 0$ be an arbitrary constant. Assuming $\mathrm{NP} \neq \mathrm{coRP}$ there is no polynomial time approximation algorithm achieving any of the following:
(1) A factor of $N^{\frac{1}{3}-\epsilon}$ for Max Clique
(2) A factor of $N^{\frac{1}{5}-\epsilon}$ for Chromatic Number.
Assuming $\mathrm{P} \neq \mathrm{NP}$ there is no polynomial time approximation algorithm achieving any of the following:
(3) A factor of $27/26$ for Min Vertex Cover
(4) A factor of $38/37$ for Max-3-SAT
(5) A factor of $38/37$ for Max-Exact-3-SAT[2]
(6) A factor of $82/81$ for Max CUT
(7) A factor of $94/93$ for Max-2-SAT.

The Max Clique hardness result is of course a direct consequence of the first part of Theorem 3.1. Via the recent reduction of Furer [21], which in turn builds upon the previous reductions presented in [29, 28, 9], we get the improved Chromatic number result. The result on Vertex Cover is a consequence of the second part of Theorem 3.1. The other Max-SNP results exploit Proposition 3.2.

Even though we do not know if the "pcp approach" allows to get the best possible non-approximability results for these problems, we feel that the current results are not ridiculously far from the known upper bounds.

3.2 Discussion of techniques

The construction and analysis of the proof systems based on the long code is the most technical part of our work. We will now provide only some very brief intuition.

The starting point for all our proof systems is a two-prover proof system achieving arbitrarily small but fixed constant error with logarithmic randomness and constant answer size, as provided by Raz [32]. This proof system has the property that the answer of the second prover is supposed to be a predetermined function of the answer of the first prover. Thus, verification in it amounts to checking that the first answer satisfies some predicate and that the second answer equals the value obtained from the first answer. Following the "proof composition" paradigm of Arora and Safra [3], we will "encode" the answers of the two provers under a suitable code and then, "recursively", check these encodings. As usual, we will check both that these encodings are valid and that they correspond to answers which would have been accepted by the original verifier. Specifically, the first answer needs to satisfy some fixed predicate and the second answer needs to be a function (e.g., a projection) of the first answer.

Our main technical contribution is a new code, called the *long code*, and means to check it. The long code of an l-bit information word a is the sequence of 2^{2^l} bits consisting of the values of all possible boolean functions at a. The long code is certainly a disaster in terms of coding theory, but it has big advantages in the context of proof verification, arising from the fact that it carries enormous amounts of data about a. The difficulty will be to check that a prover claiming to write the long code of some string a is really doing so.

Let $\Sigma = \{0, 1\} = \mathrm{GF}(2)$. Let $\mathcal{F}_l$ denote the set of all functions mapping Σ^l to Σ. Regarding a 2^{2^l}-bit string as a function A of $\mathcal{F}_l$ to Σ, we prove a characterization of codewords saying that such an A is a codeword (of the long code) if and only if it is linear (i.e., satisfies $A(f+g) = A(f) + A(g)$, $\forall f, g \in \mathcal{F}_l$) and "respects the monomial basis" (i.e., satisfies $A(\chi_\emptyset) = 1$ and $A(\chi_S) \cdot A(\chi_T) = A(\chi_{S\cup T})$, for all $S, T \subseteq [l]$, where $\chi_S(x)$ equals the product of the bits of x in locations S). We then show that checking that the oracle is "close"

[2] Max-Exact-3-SAT is Max-3-SAT with exactly three literals per clause.

Problem	Approx		Non-Approx		
	Factor	Due to	New Factor	Previous Factor	Assumption
Max-3-SAT	1.319	[33, 24, 25]	1.027	$1+\frac{1}{72}$ [9]	$\mathrm{P} \neq \mathrm{NP}$
Max-E3-SAT	$1+\frac{1}{7}$	folklore	$1+\frac{1}{37}$	unspecified [2]	$\mathrm{P} \neq \mathrm{NP}$
Max-2-SAT	1.075	[25, 17]	1.010	$1+\frac{1}{504}$ (implied [9])	$\mathrm{P} \neq \mathrm{NP}$
MAX CUT	1.139	[25]	1.012	unspecified [2]	$\mathrm{P} \neq \mathrm{NP}$
Min-VC	$2-o(1)$	[5, 30]	$1+\frac{1}{26}$	unspecified [2]	$\mathrm{P} \neq \mathrm{NP}$
Max-Clique	$N^{1-o(1)}$	[14]		$N^{\frac{1}{4}}$ [9]	$\mathrm{NP} \not\subseteq \mathrm{co}\tilde{\mathrm{RP}}$
			$N^{\frac{1}{3}}$	$N^{\frac{1}{5}}$	$\mathrm{coRP} \neq \mathrm{NP}$
			$N^{\frac{1}{4}}$	$N^{\frac{1}{6}}$ [9]	$\mathrm{P} \neq \mathrm{NP}$
Chromatic	$N^{1-o(1)}$	[14]		$N^{\frac{1}{10}}$ [9]	$\mathrm{NP} \not\subseteq \mathrm{co}\tilde{\mathrm{RP}}$
Number			$N^{\frac{1}{5}}$	$N^{\frac{1}{13}}$	$\mathrm{coRP} \neq \mathrm{NP}$
			$N^{\frac{1}{7}}$	$N^{\frac{1}{14}}$ [9]	$\mathrm{P} \neq \mathrm{NP}$

Figure 1: *Approximation factors attainable by polynomial-time algorithms (Approx) versus factors we show are hard to achieve (Non-Approx).*

to a codeword amounts to two tests – a *linearity* test and a *multiplication*, or *RMB* (respect of monomial basis) test. For the former we can use the standard linearity test of [13], and benefit from the recent analysis of [6]. The latter is new, and, although the the test itself is simple, its analysis is not. We also need a projection test to make sure that the oracle encodes an answer of the second prover which matches the enconding of the answer of the first prover. Finally, an additional idea of "folding" of functions is used to eliminate the need for a "circuit" test (i.e., testing that the encoded answer of the first prover would be accepted by the original verifier).[3]

The tests are put together in a variety of ways to yield the different proof systems stated above. In order to obtain the hardness results for the Max-SNP problems we define "gadgets" which encode the computation of the verifier of Proposition 3.2. In what follows, we illustrate this for Max-3-SAT.

Let $\mathsf{MaxSAT}(\varphi)$ denote the maximum number of clauses in S that are simultaneously satisfiable. A Parity Check (PC) gadget $\mathrm{PC}(a,b,c,x_1,x_2,\ldots,x_n)$ is a set of clauses over three *distinguished* variables a,b,c and n auxiliary variables $x_1,\ldots,x_n$. It is an (α,β)-PC gadget if the following is true: If $a+b=c$ then $\mathsf{MaxSAT}(\mathrm{PC}(a,b,c,x_1,x_2,\ldots,x_n))=\alpha$; else it is at most $\alpha-\beta$. Similarly a Respect-Monomial-Basis Check (RMBC) gadget $\mathrm{RMBC}(a,b,c,d,x_1,\ldots,x_n)$ is a set of clauses over four *distinguished* variables a,b,c,d and n auxiliary variables $x_1,\ldots,x_n$. It is an (α,β)-RMBC gadget if the following is true: If $a\cdot b=c+d$ then $\mathsf{MaxSAT}(\mathrm{RMBC}(a,b,c,d,x_1,x_2,\ldots,x_n))=\alpha$; else it is at most $\alpha-\beta$. We stress that in both cases the maximum number of clauses which are simultaneously satisfied is at most α. The result for MAX 3-SAT follows from the existence of a $(4,1)$-PC gadget and a $(7,1)$-RMBC gadget and the following lemma.

Lemma 3.4 (Max-3-SAT implementation of a verifier): Let V be a verifier for L of logarithmic randomness, with perfect completeness and soundness s, such that V performs either a single Parity Check (with probability q) or a single RMB check (with probability $1-q$). If there exists an (α_1,β)-Parity-Check gadget and an (α_2,β)-RMBC gadget then L reduces to approximating Max-3-SAT to within a factor of $1+\frac{(1-s)\beta}{\alpha_1 q+\alpha_2(1-q)-(1-s)\beta}$.

4 FPCP and Clique Approximation

Our result essentially "inverts" the forward connection discussed in Section 2.

Theorem 4.1 For every constant $f>0$, the following two statements are equivalent.

(1) For all $\epsilon>0$, approximating the Max Clique size of an N-vertex graph to within a factor of $N^{1/(1+f+\epsilon)}$ is NP-hard via a randomized Karp reduction.

(2) For all $\epsilon>0$, the class NP is random Karp reducible to $\overline{\mathrm{FPCP}}[O(\log n), f+\epsilon]$.

[3] This additional idea is inessential for the construction of a pcp system with amortized free-bit complexity 2.

The same holds for Cook reductions.

Thus if there is some (any) way to show hardness of Max Clique approximaiton to within $N^{1/(1+f+\epsilon)}$ then NP has (via a randomized reduction) a proof system with logarithmic randomness and amortized free bit complexity $f+\epsilon$. We stress both the "qualitative" and the "quantitative" aspects of this result. Qualitatively, it provides an answer to the following kind of a question: "What do proofs have to do with approximating clique size, and can we not prove the non-approximability result without using proof checking?" The result indicates that proofs are inherent, and explains, perhaps, why hardness results avoiding the proof connection have not appeared. However, at this stage it is the quantitative aspect that interests us more. It says that to get tighter results on Max Clique hardness, we must construct proof systems to minimize the amortized free bit complexity.

Can we hope to do so? It is a feature of the measure that so far this seems entirely possible. In particular it seems possible that the amortized free bit complexity of a pcp verifier for NP can be ϵ for any $\epsilon > 0$. Indeed, the theorem says that if Max Clique is indeed hard to approximate to within $N^{1-\epsilon}$ as we believe, then such a system will exist.

Let us now see how this reverse connection is obtained. Denote by $\mathcal{G}(V,x)$ the graph constructed from verifier V and input x by the FGLSS reduction. The vertices correspond to possible accepting transcripts in V's computation and edges corresponding to consistent/non-conflicting computations. The maximum clique size in the constructed graph is proportional to the accepting probability of V. Here we "reverse" the process; given a graph we constract a verifier such that the same proportion holds. Furthermore, applying the FGLSS-construction to our verifier retreives the original graph. We stress that by the term *graph* we mean an undirected simple graph (i.e., no self-loops or parallel edges).

Theorem 4.2 (Clique verifier of ordinary graphs): There exists a verifier, denoted W, of logarithmic randomness-complexity, logarithmic query-length and zero free-bit complexity, that, on input a N-vertex graph G, satisfies $\max_\pi \Pr[W^\pi \text{ accepts } G] = w(G)/N$. Furthermore, $\mathcal{G}(W,G)$ is isomorphic to G where the isomorphism is easily computable.

THE CONSTRUCTION OF W. On input a graph G on N nodes, the verifier W works with proofs of length $\binom{N}{2} - |E(G)|$. The proof π is indexed by the edges in $\overline{G}$ (i.e., non-edges in G). For clarity of the proof we assume that the binary value $\pi(\{u,v\})$ is either u or v. On input G and access to oracle π, the verifier W picks uniformly a vertex u in the vertex set of G and queries the oracle at each $\{u,v\} \in E(\overline{G})$. The verifier accepts if and only if all answers were u. Clearly, W tosses $\log_2 N$ coins. Also, once W picks a vertex u, the only pattern it may accepts is $(u,u,\ldots,u)$. Thus the free-bit complexity of W is 0. To analyze the probability that W accepts the input G, when given the best oracle access, one may merely prove that the graphs $\mathcal{G}(W,G)$ and G are isomorphic.

We now generalize the above construction to get verifiers which indicate the existence of large cliques in layered graphs. An (L,M,N)*-layered graph* is an N-vertex graph in which the vertices are arranged in L layers so that there are no edges between vertices in the same layer and each layer has at most M vertices. We use a convention by which, whenever a layered graph is given to some algorithm, a partition into layers is given along with it.

Theorem 4.3 (clique verifier for layered graphs): There exists a verifier, denoted W, of logarithmic randomness-complexity and logarithmic query-length so that, on input an (L,M,N)-layered graph G, the free-bit complexity of W is $\log_2 M$ and it satisfies $\max_\pi \Pr[W^\pi \text{ accepts } G] = w(G)/L$.

THE GENERALIZED CONSTRUCTION OF W. On input a (L,M,N)-layered graph G, the verifier W works with proofs consisting of two parts. The first part assigns every layer (i.e., every integer $i \in [L]$) a vertex in the layer (i.e., again we use a redundant encoding by which the answers are vertex names rather then an index between 1 and the number of vertices in the layer). The second part assigns pairs of non-adjacent (in G) vertices, a binary value, which again is represented as one of the two vertices. On input G and access to oracle π, the verifier W picks uniformly a layer i in $\{1,\ldots,L\}$ and queries π at i obtaining as answer a vertex u. If u is not in the i^{th} layer of G then the verifier rejects. Otherwise, it continues as in Theorem 4.2 (i.e., queries the oracle at each $\{u,v\} \in E(\overline{G})$ and accepts iff all answers equal u). (Actually, it is not needed to query the oracle on pairs of vertices belonging to the same layer.) The properties of W are established as before; in particular, observe that once a vertex u is specified, the only accepting pattern is $(u,u,\ldots,u)$. The free-bit complexity is determined by the number of vertices in layer i, which is upper bounded by M (and on the average equals N/L).

Main Consequences

We are interested in problems exhibiting a gap in Max-Clique size between positive and negative instances. It is convenient to let $\overline{\mathsf{MaxClique}}(G) = \mathsf{MaxClique}(G)/N$ be the fraction of nodes in a maximum clique of G. As a direct consequence of Theorem 4.2, we get

Corollary 4.4 For functions c, s mapping $\mathcal{Z}^+$ to $[0,1]$ we have Gap-Clique$_{c,s} \in \mathrm{FPCP}_{c,s}[\log, 0]$.

This corollary transforms the gap in the promise problem into a gap in a pcp system. However, the accepting probabilities in this pcp system are very low (also on yes-instances). Below, we use Theorem 4.3 to obtain a pcp system with almost perfect completeness for this promise problem. We start by presenting a randomized reduction of the promise problem to a layer version. An alternative method is presented in Section 7 (cf., Proposition 7.5).

Proposition 4.5 (Layering the clique promise problem): There exists a polynomial-time randomized transformation, T, of graphs into layered graphs so that, on input a graph G, integers C and $L \leq C/(3\log_2 N)$, outputs a subgraph $H = T(G,C,L)$ of G in L layers such that with high probability H has at most $2N/L$ vertices per layer and if $w(G) \geq C$ then $w(H) = L$.

We remark that there exist an alternative transformation,[4] using only logarithmically many coins, and guranteeing only $\Pr[\, w(H) \leq (1-\epsilon)\cdot L\,] < L/(\epsilon C)$, for every $\epsilon \in [0,1]$, provided $w(G) \geq C$. Combining Theorem 4.3 and Proposition 4.5, we obtain

Proposition 4.6 (Inverse of FGLSS-reduction): For any polynomial-time computable functions c, s, ϵ of $\mathcal{Z}^+$ to $[0,1]$, the promise problem Gap-Clique$_{c,s}$ is randomly (Karp-) reducible to $\mathrm{FPCP}_{1,s'}[\log, f']$, where

$$\begin{aligned} f'(N) &= \log_2(1/c(N)) + \log_2\log_2(N) + 2 \\ s'(N) &= 2\log_2(N)\cdot\frac{s(N)}{c(N)}\,. \end{aligned}$$

Namely, Gap-Clique$_{c,s}$ has a pcp system with logarithmic randomness and free-bit complexity f' in which YES instances are always accepted and NO instances are accepted with probability at most $s'(N)$.

Proposition 4.6 shows that the well-known method of obtaining clique-approximation results from efficient pcp systems (cf., [18, 19, 9]) is "complete" in the sense that if clique-approximation to within some factor can be shown NP-hard then this can be done via the "pcp method". This concludes the sketch of the proof of Theorem 4.1.

5 Limitations of Common Approaches

The basic tasks of a verifier constructed by recursion are to check that a given string is close to a codeword, and that another given string encodes the projection of the data word of the first string. These tasks are central to all existing ("low-complexity") pcps. In this section we provide lower bounds on the free-bit complexity of these tasks. Specifically, we consider the task of checking that a string (given by oracle access) is close[5] to a valid codeword and the task of checking that one oracle is an encoding of a projection of a string encoded by a second oracle. Loosely speaking, we show that each of these tasks has amortized free-bit complexity of at least one (and this is tight by the long code and the tests we present for it). Furthermore, we show that the amortized free-bit complexity of performing both tasks (with respect to the same given oracles) is at least two (which is also tight). We consider these bounds as an indication that one will have to depart significantly from the known techniques in order to obtain lower (than two) amortized free-bit complexity for NP. One possible avenue which may lead to a amortized free-bit complexity of 1 is to perform a relaxed form of the codeword test (see footnote above) at free-bit complexity less than one.

Definition 5.1 The *absolute distance* between two words $w, u \in \{0,1\}^n$, denoted $d(w,u)$, is the number of bits on which w and u disagree. We say that the code $E : \{0,1\}^m \to \{0,1\}^n$ has *absolute distance* d if for every $x \neq y \in \{0,1\}^m$ the absolute distance between $E(x)$ and $E(y)$ is at least d. The absolute distance between a word w and a code E, denoted $d_E(w)$, is defined as the minimum absolute distance between w and a codeword of E. A *codeword test (with respect to E)* is an oracle machine, T, such that $T^{E(a)}(R)$ accepts for all a, R. The error probability of T is defined to be

$$\max \Pr_R \left[T^A(R) \text{ accepts}\right]\,,$$

the maximum being taken over all $A \in \{0,1\}^n$ such that $d_E(A) \geq \lfloor d/2 \rfloor$.

Lemma 5.2 Let $E : \{0,1\}^m \mapsto \{0,1\}^n$ be a code of absolute distance $d > 1$, and let T be a codeword test with respect to E which uses $f_{\rm av}$ free-bits on the average. Then, T has error probability at least $\frac{1}{F} - \frac{1}{M}$, where $F = 2^{f_{\rm av}}$ and $M = 2^m$, and its amortized free bits complexity is at least $1 - F/M$.

Definition 5.3 Let E_1, mapping $\{0,1\}^m$ to $\{0,1\}^n$, and E_2, mapping $\{0,1\}^k$ to $\{0,1\}^{n'}$, be two codes, and let $\sigma : \{0,1\}^m \to \{0,1\}^k$ be a function. A *projection test (with respect to the above)* is a two-oracle machine, T, such that $T^{E_1(x),E_2(\sigma(x))}(R)$ accepts for all x, R. The error probability of T is defined to be

$$\max \Pr_R \left[T^{E_1(a),E_2(b)}(R) \text{ accepts}\right]\,,$$

the maximum being taken over all a, b such that $\sigma(a) \neq b$.

Lemma 5.4 Let E_1, E_2 and σ be as above, and T be a projection test with respect to them, which uses $f_{\rm av}$ bits on the average. Then, T has error probability at least $\frac{1}{F} - \frac{1}{K}$, where $K = |\{\sigma(a) | a \in \{0,1\}^m\}|$ and $F = 2^{f_{\rm av}}$, and thus its amortized free-bit complexity is at least $1 - F/K$.

Finally, we define a tester which combines the two tests above: i.e., T takes two oracles A and B and performs a codeword test on A and a projection test on the pair A, B.

[4] This alternative transformation is used for presenting an alternative inverse of the FGLSS-reduction. See our technical report [8].

[5] Here 'close' means closer than half the distance of the code. Indeed, our Max Clique result may use a verifier that makes such a test, but other verifiers in the paper perform a weaker form of a codeword test which is required to detect only strings that are at distance almost equal to the distance of the code.

Definition 5.5 Given a code $E_1 : \{0,1\}^m \to \{0,1\}^n$ of absolute distance d a code $E_2 : \{0,1\}^k \to \{0,1\}^{n'}$ and a function $\sigma : \{0,1\}^m \to \{0,1\}^k$, a *Combined Test* for (E_1, E_2, σ) is a two-oracle machine T such that $T^{E_1(a), E_2(\sigma(a))}(R)$ accepts on all a, R. The error of the test is defined to be

$$\max \Pr_R \left[T^{A,B}(R) \text{ accepts}\right] ,$$

where the maximum is over all $(A, B) \in S$, and where S contains (A, B) iff either $d_{E_1}(A) \geq \lfloor d/2 \rfloor$ or $A = E_1(a)$ and $B = E_2(b)$ for some $b \neq \sigma(a)$.

Lemma 5.6 Let E_1 (of distance > 1), E_2 and σ be as above and T be a combined codeword and projection test with respect to them. Suppose that, given access to a pair of oracles, of length n and n' respectively, T accepts at most $F^2 = 2^{2f}$ possible patterns for each possible sequence of coin tosses. Then, T has error probability at least $\frac{1}{64F} - \frac{1}{4K} - \frac{1}{2M}$, where $K = 2^k$ and M is the minimum, over all $b \in \{0,1\}^k$, of the number of $a \in \{0,1\}^m$ projected by σ to b. Thus, the amortized free-bit complexity of T is at least $2 - O(\frac{1}{f} + \frac{F}{\min\{K,M\}})$.

6 Complexity of PCP and FPCP

In the rest of this extended abstract, unless stated differently, r, l, p, q and k are integer functions, $c, s, \epsilon : \mathcal{Z}^+ \to [0,1]$ and $f : \mathcal{Z}^+ \to \mathcal{R}^+$

Proposition 6.1 (pcp systems with at most 3 queries):

(1) $\forall\, c, s$ so that s is strictly less than c, $\mathrm{PCP}_{c,s}[\log, 1] = \mathrm{P}$.

(2) $\forall\, s$ strictly less than 1, $\mathrm{PCP}_{1,s}[\log, 2] = \mathrm{P}$. In contrast, for some constants $0 < s < c < 1$, $\mathrm{PCP}_{c,s}[\log, 2] = \mathrm{NP}$.

(3) $\mathrm{PCP}_{1,0.9}[\log, 3] = \mathrm{NP}$. In contrast, $\forall s \leq 0.299$, $\mathbf{na}\mathrm{PCP}_{1,s}[\log, 3] = \mathrm{P}$, where $\mathbf{na}\mathrm{PCP}$ is a restriction of PCP in which the verifier is required to be non-adaptive.

Item (2) is folklore. The bound obtained in the second part of Item (3), let alone that it is restricted to the non-adaptive case, is weaker than what can be proven for MIP proof systems (see below). This contrast may perhaps provide a testing ground to separate PCP from MIP, a question raised by [7]. Below, $\mathrm{MIP}_{\cdot,\cdot}[\cdot, p]$ denotes the class of languages accepted by a one-round p-prover protocol in which is prover answer is a single bit.

Proposition 6.2 (mip systems with at most 3 queries):

(1) $\forall c, s, r, p, \mathrm{MIP}_{c,s}[r, p] \subseteq \mathrm{MIP}_{c,2s}[r, p-1]$.

(2) $\forall s < \frac{1}{2}, \mathrm{MIP}_{1,s}[\log, 3] = \mathrm{P}$.

Lemma 6.3 $\forall c, s, r, q$ such that $\frac{c}{s} > 2^q$, –

$$\mathrm{PCP}_{c,s}[r, q] \subseteq \mathrm{RTIME}(\mathrm{poly}(n, r, q, (c - 2^q s)^{-1}) \ .$$

Furthermore, if r and q are both logarithmically bounded then $\mathrm{PCP}_{c,s}[r, q] = \mathrm{P}$.

Corollary 6.4 $\forall\, c > 0,\ \overline{\mathrm{PCP}}_c[\log,\ 1] = \mathrm{P}$.

Before turning to free-bit complexity, we comment that results analogous to Proposition 6.1, where PSPACE plays the role of P and NEXP the role of NP, can be derived for the classes $\mathrm{PCP}_{\cdot,\cdot}(\mathrm{poly}, \cdot)$. Furthermore, for all functions c, s so that $c(n) > s(n) + 1/\mathrm{poly}(n)$, we obtain $\mathrm{PCP}_{c,s}[\mathrm{poly}, 1] \subseteq \mathrm{AM}$. In light of the last item of Proposition 6.5, this result may be hard to strengthen.

Proposition 6.5 $\forall s$ strictly smaller than 1:

(1) $\mathrm{FPCP}_{1,s}[\log, 1] = \mathrm{P}$. In contrast, NP equals the classes $\mathrm{FPCP}_{0.5,0.45}[\log, 1]$, $\mathrm{FPCP}_{0.25,0.22}[\log, 0]$, and $\mathrm{FPCP}_{1,0.95}[\log, f]$ for $f = \log_2 3$.

(2) $\mathrm{FPCP}_{1,s}[\mathrm{poly}, 0] \subseteq \mathrm{coNP}$ and $\mathrm{FPCP}_{1,s}[\mathrm{poly}, 1] \subseteq \mathrm{PSPACE}$.

(3) Graph Non-Isomorphism and Quadratic Non-Residousity have pcp systems, with perfect completeness, soundness error $\frac{1}{2}$, query complexity 1 and 0 free-bits.

The last item follows from [27, 26].

In consequence of the above, we note that both the freeness and the amortization are key to going as low as two amortized free bits. Such efficiency under query complexity, amortized query complexity, and (non-amortized) free bit complexity is ruled out because $\mathrm{PCP}_{1,1/2}[\log,\ 2] \subseteq P$, $\overline{\mathrm{PCP}}[\log,\ 1] \subseteq P$, and $\mathrm{FPCP}_{1,1/2}[\log,\ 1] \subseteq P$.

7 Transformations of FPCP Systems

In this section we show several useful transformations which can be applied to pcp systems. We concentrate on the free-bit complexity, and introduce an additional parameter into the notation – the proof length (i.e., $\mathrm{FPCP}_{c,s}[r, f, l]$ refers to randomness r, free-bit f and proof length l). We start by stating the simple fact that the ratio between the completeness and soundness bounds (also referred to as gap) is amplified (i.e., raise to the power k) when one repeats the pcp system (k times). Note, however, that if the original system is not perfectly complete then the completeness bound in the resulting system gets decreased.

Proposition 7.1 (gap amplification): $\forall c, s, r, f, l, k$ –

$$\mathrm{FPCP}_{c,s}[r, f, l] \subseteq \mathrm{FPCP}_{c^k, s^k}[kr, kf, l].$$

Next, we show that in some sense the randomness-complexity of a proof system need not be higher than logarithmic in the length of the proofs/oracles employed. The notation $\leq_K^R$ is used to indicate a randomized Karp reduction.

Proposition 7.2 (reducing randomness): $\forall s, r, f, l, \epsilon$ –

$$\mathrm{FPCP}_{1,s}[r, f, l] \leq_K^R \mathrm{FPCP}_{1,s'}[r', f, l] \ ,$$

where $s' = (1 + \epsilon) \cdot s$ and $r' = O(1) + \log_2(l/\epsilon^2 s)$.

An analogous statement for two-sided error pcp is omitted. Combining Propositions 7.1 and 7.2, we obtain the following corollary which plays a central role in deriving clique approximation results via the FGLSS method [6]–

Corollary 7.3 $\forall r, f, k$ –

$$\overline{\mathrm{FPCP}}[\,r,\,f\,] \leq_K^R \mathrm{FPCP}_{1,2^{-k}}[r+k+\log\,,\;kf]\ .$$

An alternative error reduction procedure, which allows to obtain inapproximability results under $\mathrm{P} \neq \mathrm{NP}$, follows (stated here only for the one-sided error case)

Proposition 7.4 $\forall r, f, k$ and all constants $\epsilon, s > 0$ –

$$\mathrm{FPCP}_{1,s}[r,f] \subseteq \mathrm{FPCP}_{1,s^k}[r+(2+\epsilon)\cdot k+\log\,,\;kf].$$

The following transformation is analogous to the randomized layering procedure for the clique promise problem (i.e., Proposition 4.5). In view of the relation between FPCP and the clique promise problem (shown in Section 4), this analogy is hardly surprising. In this transformation the acceptance probability bounds are pushed higher at the expense of increasing the free-bit complexity.

Proposition 7.5 (Increasing acceptance probabilities): $\forall c, s, r, f, k$ –

$$\mathrm{FPCP}_{c,s}[r,f] \leq_K^R \mathrm{FPCP}_{c',s'}[r, f+\log_2 k]\ ,$$

where $c' = 1-4(1-c)^k$ and $s' = k \cdot s$.

The following transformation has an opposite effect than the previous one, reducing the free-bit complexity at the expense of lowering the bounds on acceptance probability.

Proposition 7.6 (Decreasing acceptance probabilities): $\forall c, s, r, f$ and $\forall k \leq f$ –

$$\mathrm{FPCP}_{c,s}[r,f] \subseteq \mathrm{FPCP}_{\frac{c}{2^k},\frac{s}{2^k}}[r+k, f-k]$$

Acknowledgements

We thank Uri Feige, Marcos Kiwi, and Luca Trevisan for comments on early drafts.

References

[1] S. ARORA. Reductions, Codes, PCPs and Inapproximability. FOCS 1995.

[2] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN AND M. SZEGEDY. Proof verification and intractability of approximation problems. FOCS 1992.

[3] S. ARORA AND S. SAFRA. Probabilistic checking of proofs: a new characterization of NP. FOCS 1992.

[4] L. BABAI, L. FORTNOW, L. LEVIN, AND M. SZEGEDY. Checking computations in polylogarithmic time. STOC 1991.

[5] R. BAR-YEHUDA AND S. EVEN. A local ratio theorem for approximating the weighted vertex cover problem. In *Analysis and Design of Algorithms for Combinatorial Problems* Vol. 25 of Annals of Discrete Math, Elsevier, 1985.

[6] M. BELLARE, D. COPPERSMITH, J. HÅSTAD, M. KIWI AND M. SUDAN. Linearity testing in characteristic two. FOCS 1995.

[7] M. BELLARE, S. GOLDWASSER, C. LUND AND A. RUSSELL. Efficient probabilistically checkable proofs and applications to approximation. STOC 1993.

[8] M. BELLARE, O. GOLDREICH AND M. SUDAN. Free Bits, PCPs and Non-Approximability – Towards Tight Results. August 1995 (replacing previous version of May 1995). Available from *ECCC*, *Electronic Colloquium on Computational Complexity*, via WWW using `http://www.eccc.uni-trier.de/eccc/`.

[9] M. BELLARE AND M. SUDAN. Improved non-approximability results. STOC 1994.

[10] M. BEN-OR, S. GOLDWASSER, J. KILIAN AND A. WIGDERSON. Multi-Prover interactive proofs: How to remove intractability assumptions. STOC 1988.

[11] P. BERMAN AND G. SCHNITGER. On the complexity of approximating the independent set problem. *Information and Computation* **96**, 77–94 (1992).

[12] A. BLUM. Algorithms for approximate graph coloring. Ph. D Thesis, MIT, 1991.

[13] M. BLUM, M. LUBY AND R. RUBINFELD. Self-testing/correcting with applications to numerical problems. *JCSS* Vol. 47, pp. 549–595, 1993.

[14] R. BOPPANA AND M. HALDÓRSSON. Approximating maximum independent sets by excluding subgraphs. *BIT* Vol. 32, No. 2, 1992.

[15] S. EVEN, A. SELMAN AND Y. YACOBI. The complexity of promise problems with applications to public-key cryptography. *Information and Control* Vol. 2, 159–173, 1984.

[16] U. FEIGE. Randomized graph products, chromatic numbers, and the Lovász theta function. STOC 1995.

[6] In a typical application, $r = O(\log n)$ and one sets k to be a large multiple of r. The FGLSS-graph corresponding to the resulting pcp system will have size $N = 2^{(r+k+O(1))+kf}$ and a gap in clique size of factor 2^k, which can be rewritten as $N^{1/(1+f+\epsilon)}$ where $\epsilon = r/k$.

[17] U. FEIGE AND M. GOEMANS. Approximating the value of two prover proof systems, with application to Max-2SAT and Max-DICUT. ISTCS 1995.

[18] U. FEIGE, S. GOLDWASSER, L. LOVÁSZ, S. SAFRA, AND M. SZEGEDY. Approximating clique is almost NP-complete. FOCS 1991.

[19] U. FEIGE AND J. KILIAN. Two prover protocols – Low error at affordable rates. STOC 1994.

[20] L. FORTNOW, J. ROMPEL AND M. SIPSER. On the power of multiprover interactive protocols. Structures 1988.

[21] M. FURER. Improved hardness results for approximating the chromatic number. FOCS 1995.

[22] M. GAREY AND D. JOHNSON. The complexity of near optimal graph coloring. *Journal of the ACM* Vol. 23, No. 1, 43–49, 1976.

[23] M. GAREY, D. JOHNSON AND L. STOCKMEYER. Some simplified NP-complete graph problems. *TCS* **1**, pp. 237–267, 1976.

[24] M. GOEMANS AND D. WILLIAMSON. New 3/4-approximation algorithm for MAX SAT. *Proceedings of the 3rd Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization*, 1993.

[25] M. GOEMANS AND D. WILLIAMSON. $.878$ approximation algorithms for Max-CUT and Max-2SAT. STOC 1994.

[26] O. GOLDREICH, S. MICALI, AND A. WIGDERSON. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. FOCS 1986.

[27] S. GOLDWASSER, S. MICALI, AND C. RACKOFF. The knowledge complexity of interactive proofs. *SIAM J. Computing* Vol 18, No. 1, 186–208, 1989.

[28] S. KHANNA, N. LINIAL AND S. SAFRA. On the hardness of approximating the chromatic number. ISTCS 1993.

[29] C. LUND AND M. YANNAKAKIS. On the hardness of approximating minimization problems. STOC 1993.

[30] MONIEN AND SPECKENMEYER. Some further approximation algorithms for the vertex cover problem. *Proceedings of CAAP 83*, Lecure Notes in Computer Science Vol. 159, Springer-Verlag, 1983.

[31] C. PAPADIMITRIOU AND M. YANNAKAKIS. Optimization, approximation, and complexity classes. *JCSS* **43**, pp. 425–440, 1991.

[32] R. RAZ. A parallel repetition theorem. STOC 1995.

[33] M. YANNAKAKIS. On the approximation of maximum satisfiability. SODA 1992.

[34] D. ZUCKERMAN. NP-Complete Problems have a version that is hard to Approximate. Structures 1993.