# Zero Knowledge and the Chromatic Number

Uriel Feige *

Dept. of Applied Mathematics
and Computer Science
The Weizmann Institute
Rehovot 76100, Israel
feige@wisdom.weizmann.ac.il

Joe Kilian

NEC Research Institute
4 Independence Way
Princeton, New Jersey 08540
joe@research.nj.nec.com

### Abstract

We present a new technique, inspired by zero-knowledge proof systems, for proving lower bounds on approximating the chromatic number of a graph. To illustrate this technique we present simple reductions from max-3-coloring and max-3-sat, showing that it is hard to approximate the chromatic number within $\Omega(N^{\delta})$, for some $\delta > 0$. We then apply our technique in conjunction with the probabilistically checkable proofs of Håstad, and show that it is hard to approximate the chromatic number to within $\Omega(N^{1-\epsilon})$ for any $\epsilon > 0$, assuming $\mathsf{NP} \not\subseteq \mathsf{ZPP}$. Here, $\mathsf{ZPP}$ denotes the class of languages decidable by a random expected polynomial-time algorithm that makes no errors. Our result matches (up to low order terms) the known gap for approximating the size of the largest independent set. Previous $O(N^{\delta})$ gaps for approximating the chromatic number (such as those by Lund and Yannakakis, and by Furer) did not match the gap for independent set, and do not extend beyond $\Omega(N^{1/2-\epsilon})$.

## 1 Introduction

We present improved hardness results for approximating the chromatic number. To present and discuss our results, we use the following standard notation.

- $\chi(G)$ – the chromatic number (the minimum number of colors that suffice in order to legally color the vertices of a graph $G$ such that adjacent vertices receive different colors).
- $\alpha(G)$ – the maximum number of vertices in an independent set of graph $G$.
- $\omega(G)$ – the maximum number of vertices in a complete subgraph (clique) of $G$.

*Incumbent of the Joseph and Celia Reskin Career Development Chair. Yigal Alon Fellow. Part of this work done while the author was visiting the NEC Research Institute.

The above three quantities are strongly interrelated. The chromatic number can readily be seen to be the minimum number of independent sets that cover all vertices of $G$. Hence $\alpha(G)$ and $\chi(G)$ are "weak duals" of each other, and $\chi(G) \cdot \alpha(G) \geq N$, where $N$ denotes the number of vertices in the graph $G$.

The quantities $\alpha(G)$ and $\omega(G)$ are related by $\omega(G) = \alpha(\bar{G})$, where $\bar{G}$ denotes the complement of graph $G$ (edges are replaced by nonedges, and nonedges are replaced by edges). Because of this strong connection, many results concerning one of these quantities trivially yield results for the other. To avoid unnecessary repetition, results that relate to either $\alpha(G)$ or $\omega(G)$ will be explicitly discussed only in terms of $\alpha(G)$.

The precise calculation of these three quantities has long been known to be intractable, assuming that $\mathsf{P} \neq \mathsf{NP}$ [23]. Subsequent work has focused on the complexity of approximating them. We say that an algorithm approximates a function $f(x)$ within a ratio $q(x)$, if on any input $x$ the algorithm outputs two numbers $\ell(x)$ and $u(x)$ such that $\ell(x) \leq f(x) \leq u(x)$, and $u(x)/\ell(x) \leq q(x)$. For graph properties, the ratio of approximation $q(G)$ is typically expressed only as a function of $N$, the number of vertices in the input graph.

Known polynomial time algorithms for approximating $\chi(G)$ and $\alpha(G)$ do not guarantee a good approximation ratio. The best approximation ratios known to be efficiently achievable are

$$O(N(\log\log N)^2/(\log N)^3)$$

for $\chi(G)$ [20], and $O(N/(\log N)^2)$ for $\alpha(G)$ [9]. As well as computing an approximation for $\chi(G)$ or $\alpha(G)$, these algorithms produce a coloring (or independent set) that is within the guaranteed ratio from optimal.

The theory of NP-completeness can be used in order to provide evidence that approximating the above quantities is hard. An early result of Garey and Johnson [18] showed it NP-hard to approximate $\chi(G)$ to within any constant less than 2. More recently [13], a connection was established between the hardness of approximating $\alpha(G)$ and the construction of probabilistically checkable proofs (PCPs). Using this connection it became possible to prove hardness of approximation results for $\alpha(G)$ under the assumption that $\mathsf{P} \neq \mathsf{NP}$, and even sharper hardness of approximation results under the assumption that $\mathsf{NP} \not\subseteq \mathsf{ZPP}$. After a long line of work [13, 4, 3, 6, 14, 7, 5, 21, 22], it has been shown hard to approximate $\alpha(G)$ to within $N^{1-\epsilon}$ for any constant $\epsilon > 0$.

Lund and Yannakakis [28] reduced approximating $\alpha(G)$ to approximating $\chi(G)$, thereby showing it hard to approximate $\chi(G)$ to within $N^c$ for some constant $c > 0$. Subsequent work [24, 7, 16] has somewhat simplified and improved this reduction. Applying the most efficient reduction [16] to the strongest result known for independent set [22] implies that it is NP-hard (under randomized reductions) to approximate $\chi(G)$ to within $O(N^{1/2-\epsilon})$ for any constant $\epsilon > 0$.

It is well known that any algorithm $B$ for finding independent sets that are within a ratio of $O(N^c)$ from the maximum independent set (where $0 < c < 1$) can be used in order to color within a ratio of $O(N^c)$ from optimal, by repeatedly applying the algorithm $B$ to the input graph and extracting color classes (independent sets). A natural question is whether the constants in the exponents for the hardness of approximating $\alpha(G)$ and $\chi(G)$ are the same up to low-order terms. We note that with previous approaches the constant established for $\chi(G)$ has always been strictly smaller than that established for $\alpha(G)$.

In this paper we develop a new approach, inspired by zero-knowledge proofs, to showing the hardness of computing chromatic numbers. Ignoring many important details, previous approaches can be viewed as a two step reduction: the first step is a reduction from approximating the acceptance probability of a PCP to approximating $\alpha(G)$, the second step is a reduction from approximating $\alpha(G)$ to approximating $\chi(G)$. Our new approach has a different two step structure: the first step is a reduction from approximating the acceptance probability of a PCP to approximating the acceptance probability of a "zero knowledge" PCP, the second step is a reduction from approximating the acceptance probability of a zero knowledge PCP to approximating $\chi(G)$.

After the underlying principles are well understood, the implementation of the new approach turns out to be relatively straightforward. For the first step, we can build upon the vast body of previous work on zero knowledge proof systems. The second step (reduction to $\chi(G)$) closely resembles the first step (reduction to $\alpha(G)$) of previous approaches. Based on three different zero knowledge protocols, we give three different reductions showing that $\chi(G)$ is hard to approximate within $O(N^c)$, for some $c > 0$. The strongest of these reductions gives our main theorem.

**Theorem 1** Unless $\mathsf{NP} \subseteq \mathsf{ZPP}$ it is intractable to approximate $\chi(G)$ to within $N^{1-\epsilon}$ for any constant $\epsilon > 0$.

Frequently in the literature this condition is given as $\mathsf{NP} \not\subseteq \mathsf{coRP}$ instead of $\mathsf{NP} \not\subseteq \mathsf{ZPP}$; the two can be shown to be equivalent. We remark that the proof of Theorem 1 shows that for any $\epsilon > 0$, it is NP-hard (under randomized reductions) to distinguish between graphs with $\alpha(G) \leq N^\epsilon$ and graphs with $\chi(G) \leq N^\epsilon$.

### 1.1 Outline of the paper

In Section 2 we compare our approach for proving hardness of $\chi(G)$ with previous approaches. In Section 3 we present in detail our framework for proving the hardness of approximating the chromatic number. In Section 4 we use this framework to give simple hardness results via reductions from max-3-coloring and from max-3-sat. In Section 5 we develop techniques for randomizing long-code based proofs and prove Theorem 1.

## 2 Putting the new approach in perspective

A relatively simple proof that $\alpha(G)$ is hard to approximate follows ideas of Berman and Schnitger [8].

**Simple hardness results for $\alpha(G)$:**

1. Reduce the problem of approximating max-3SAT to approximating $\alpha(G_0)$ within a constant factor in a graph $G_0$ that has linear size independent sets.

2. Use a technique known as *randomized graph products* [8, 10] to produce a new graph $G$ that only has small independent sets. The problem of approximating $\alpha(G_0)$ within a constant factor is reduced to that of approximating $\alpha(G)$ within a factor of $O(N^\epsilon)$, for some $\epsilon > 0$.

The above reduction is randomized, due to its second step, and relies on the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$. It can be derandomized [31, 1], though the value of $\epsilon$ obtained by using known derandomization techniques is not as good as the one obtained by the randomized reduction. For this reason, hardness of approximation results for $\alpha(G)$ (and $\chi(G)$) often assume $\mathsf{NP} \not\subseteq \mathsf{ZPP}$, as will be assumed throughout in this paper (See also the remark in Section 3.6).

The fact that max-3SAT does not have approximation schemes follows from the PCP theorem [3] (in fact, it also implies the PCP theorem). Stronger hardness of approximation results for $\alpha(G)$ can be proven by direct reduction from PCPs. Specifically, one needs a PCP that uses as few *amortized free bits* as possible [7, 5].

**Strong hardness results for $\alpha(G)$:**

1. Reduce the problem of approximating the acceptance probability for a PCP with $f$ amortized free bits to approximating $\alpha(G_0)$ within a constant factor in a graph $G_0$ that has linear size independent sets.

2. Use randomized graph products to produce a new graph $G$ that only has small independent sets. The problem of approximating $\alpha(G_0)$ within a constant factor is reduced to that of approximating $\alpha(G)$ within a factor of $O(N^{1/(1+f)})$.

The quality of the hardness result depends on $f$, the number of amortized free bits in the PCP. Following a long line of work [6, 14, 7, 5, 21], Håstad [22] shows how to obtain $f$ arbitrarily close to 0, implying that $\alpha(G)$ is hard to approximate within a factor $O(N^{1-\epsilon})$, for any $\epsilon > 0$.

To prove hardness of approximation results for $\chi(G)$, a direct reduction from $\alpha(G)$ to $\chi(G)$ was used. Known reductions do not preserve the ratio of approximation. Improving upon earlier reductions [28, 24, 7], Furer [16] gives a randomized reduction that shows that if independent set cannot be approximated to within $N^{1/(f+1)}$ then chromatic number cannot be approximated to within $N^{min(1/2,1/(2f+1))-o(1)}$. Hence it is hard to approximate $\chi(G)$ within a factor of $O(N^{1/2-\epsilon})$, for any $\epsilon > 0$. Furer also proposes a simpler reduction from $\alpha(G)$ to $\chi(G)$ that suffices in order to show that it is hard to approximate $\chi(G)$ within a factor of $O(N^{\epsilon})$, for some $\epsilon > 0$.

Our new approach for proving hardness results for approximating $\chi(G)$ does not provide a direct reduction from $\alpha(G)$ to $\chi(G)$. Instead, it is based on randomized versions of PCPs (RPCPs), introduced in [11]. Zero-knowledge PCPs are a subfamily of randomized PCPs. Indeed, some of the randomized PCPs we use are in fact zero-knowledge PCPs. Similar PCPs were previously used in [25] to show the hardness of certain probabilistic inference problems. In its simpler form, our reduction has the following structure:

**Simple hardness results for $\chi(G)$:**

1. Reduce the problem of approximating max-3SAT to approximating the acceptance probability in a randomized PCP for max-3SAT.

2. Reduce the problem of approximating the acceptance probability in a randomized PCP for max-3SAT to approximating $\chi(G_0)$ within a constant factor in a graph $G_0$ that can be colored with a constant number of colors.

3. Use randomized graph products to produce a new graph $G$ that has a large chromatic number. The problem of approximating $\chi(G_0)$ within a constant factor is reduced to that of approximating $\chi(G)$ within a factor of $O(N^{\epsilon})$, for some $\epsilon > 0$.

The first step of the reduction uses standard techniques for constructing zero knowledge proof systems (specifically, we use protocols from [19, 14]). The second step is identical to the usual reduction to $\alpha(G_0)$ [13]. The fact that the same reduction now also applies to $\chi(G_0)$ is a consequence of the weak duality relation between $\alpha(G)$ and $\chi(G)$, and the particular structure of zero knowledge PCPs. The third step uses a version of randomized graph products that is suitable for amplifying gaps in chromatic number [12], rather than the randomized graph products that are used in the context of independent sets [8, 10].

To obtain stronger hardness of approximation results for $\chi(G)$, we start from a PCP with few amortized free bits. We transform it to a randomized PCP with few amortized free bits. Thereafter, the reduction proceeds exactly as in the simple hardness result described above, though the analysis is somewhat more involved.

**Strong hardness results for $\chi(G)$:**

1. Reduce the problem of approximating the acceptance probability for a PCP with $f$ amortized free bits to approximating the acceptance probability for an RPCP with a certain covering parameter $\rho$.

2. Reduce the problem of approximating the acceptance probability for an RPCP with parameter $\rho$ to approximating the *fractional* chromatic number $\chi_f(G_0)$ within a constant factor in a graph $G_0$ with bounded fractional chromatic number.

3. Use randomized graph products to produce a new graph $G$ that has a large chromatic number. The problem of approximating $\chi_f(G_0)$ within a constant factor is reduced to that of approximating $\chi(G)$ within a factor of $O(N^\delta)$, for some $\delta > 0$ that depends on $\rho$.

The quality of hardness result that we obtain depends on how well $\rho$ reflects $f$ in the first step of the reduction. Fortunately, it is possible to transform the strongest known PCPs (those of [22]) to randomized PCPs with essentially no quantitative loss in the relevant parameters. Hence we obtain that $\chi(G)$ is hard to approximate within a ratio of $O(N^{1-\epsilon})$, for any $\epsilon > 0$, under the assumption that $\mathsf{NP} \not\subseteq \mathsf{ZPP}$.

# 3 Our approach

In this section we formally define the main ingredients used by our approach, explain how they fit together, and identify the parameters that determine the quality of the hardness of approximation result that is obtained.

## 3.1 PCPs

A *probabilistically checkable proof system* (PCP) for an NP language $L$ is a procedure for providing polynomial size witnesses for inputs in the NP language, coupled with a very efficient probabilistic method of checking the validity of witnesses. The verifier $V$ in a PCP proof system receives a pair $(x, w)$ and uses a probabilistic procedure to verify with high probability that $w$ is a PCP witness that $x \in L$. In the verification procedure, $V$ selects at random (though not uniformly) a subset of the bits of $w$, evaluates a polynomial time predicate on these bits, and decides whether to accept or reject. We require that if $x \in L$, then for an appropriate choice of $w$ the verifier always accepts (perfect completeness), and if $x \notin L$, then regardless of the choice of $w$, the probability that $V$ accepts (taken over the random choices of subsets of bits of $w$) is at most $e$, where $e < 1$ is a parameter of the PCP system.

The number of possible random choices made by $V$ is denoted by $R$ (that is, $V$ uses $\log_2(R)$ random bits). We identify other parameters of interest. Let $S_1, \ldots S_R$ denote the $R$ subsets of bits of $w$ that $V$ may possibly request to see. (The question of whether all these subsets are distinct is irrelevant for our purpose.) For any set $S_i$, let $n_i$ denote the number of settings of the corresponding bits in $w$ that cause $V$ to accept. Clearly, $n_i \leq 2^{|S_i|}$. We let $\ell = \max_i[n_i]$, and $A = \sum_{i=1}^{R} n_i$. Note that $A \leq \ell R$. The average number of *amortized free bits* is defined as $f = \log_{1/e} A/R$ ($e$ denotes the error parameter of the PCP, not the base of the natural logarithm!).

The PCP theorem [3] states that 3SAT (and hence any NP language) has a PCP in which $R$ is polynomial and all $S_i$ are of constant size. Subsequent work [14, 7, 5] identified $f$ as the most relevant parameter for proving hardness of approximating $\alpha(G)$. Håstad [21, 22] showed how to design PCPs with $f < \epsilon$, for any $\epsilon > 0$.

## 3.2 The FGLSS reduction

The FGLSS reduction [13] has as its starting point a PCP with parameters $(e, R, \ell, A)$ for an NP-complete language $L$. For any input $x$, the question of whether $x \in L$ is reduced to approximating $\alpha(G_0)$ for some graph $G_0$ that depends on $x$. The vertices of $G_0$ are all pairs $(S_i, v)$, where $S_i$ is a subset of bits of $w$ that $V$ may request on input $x$, and $v$ is a setting to these bits that causes $V$ to accept. Hence $G_0$ has $A$ vertices. Two vertices $(S_i, v)$ and $(S_j, v')$ are adjacent if $S_i \bigcap S_j \neq \emptyset$ and $v$ and $v'$ differ in the setting that they give to the bits in the intersection. In particular, vertices $(S_i, v)$ and $(S_i, v')$ are adjacent. As shown in [13], if $x \in L$ then $\alpha(G_0) = R$, and if $x \notin L$ then $\alpha(G_0) \leq eR$. This implies that it is NP-hard to approximate $\alpha(G_0)$ to within $1/e$ for some $e < 1$.

Observe that using the FGLSS reduction, if $x \notin L$ then $\chi(G_0) \geq A/eR$. However, if $x \in L$, the definition of PCPs does not imply any useful upper bound on $\chi(G_0)$.

## 3.3 Fractional chromatic numbers

The vertices $V(G)$ of a $k$-colorable graph $G$ can be covered by the union of $k$ independent sets $I_1, \ldots, I_k$. Put probabilistically, choosing $I$ uniformly from $\{I_1, \ldots, I_k\}$ covers any $v \in V(G)$ with probability at least $1/k$. The fractional chromatic number $\chi_F(G)$ can be defined by considering arbitrary distributions $\mathcal{I}$ on $G$'s independent sets.

**Definition 1** We define $\chi_f(G)$ as the smallest real $k$ such that for some distribution $\mathcal{I}$ on $G$'s independent sets, choosing $I \leftarrow \mathcal{I}$ covers any $v \in V(G)$ with probability at least $1/k$.

(This is not the standard definition for $\chi_f(G)$, but is equivalent.)

Clearly, $\chi_f(G) \leq \chi(G)$. Lovasz [27] has shown that for any graph $G$,

$$\frac{\chi(G)}{1 + \ln \alpha(G)} \leq \chi_f(G) \leq \chi(G). \tag{1}$$

Thus, $\chi_f(G)$ approximates $\chi(G)$ to within a logarithmic factor, so strong hardness results for approximating $\chi_f(G)$ apply to $\chi(G)$ as well. Finally, by a straightforward argument we have

$$\chi_f(G)\alpha(G) \geq |V(G)| \tag{2}$$

Thus, an upper bound on $\alpha(G)$ implies a corresponding lower bound on $\chi_f(G)$.

## 3.4 Randomized graph products

Graph products amplify gaps in sizes of independent sets [17] and in chromatic number [26]. Randomized graph products were introduced by Berman and Schnitger [8] as a way of enhancing the effect of graph products. Applying the method of randomized graph products [8, 10] to the graph $G_0$ obtained from the FGLSS reduction, one generates a graph $G$ in which the $1/e$ gap in the size of the independent set is amplified to

$$N^{\frac{1}{1+f}-\epsilon},$$

where $N = |V(G)|$, $f$ is the average number of amortized free bits of the underlying PCP, and $\epsilon$ is a positive constant that can be made arbitrarily small. In this section we survey the randomized graph products that were used in [12], which somewhat differ from those of [8, 10].

We first define the inclusive graph product. Linial and Vazirani [26] showed how inclusive graph products could be used to amplify hardness of approximation results for chromatic number.

**Definition 2** Given graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, The *inclusive graph product* $G \times H$ has vertex set $V_G \times V_H$ and edges $((x_G, x_H), (y_G, y_H))$ whenever $(x_G, y_G) \in E_G$ or $(x_H, y_H) \in E_H$.

We denote by $G^k$ the $k$-wise inclusive graph product of $G$ with itself. This product is well behaved with respect to $\alpha, \chi$ and $\chi_f$. It is not hard to show that $\alpha(G^k) = \alpha(G)^k$ and $\chi(G^k) \leq \chi(G)^k$. Lovasz [27] shows that $\chi_f(G^k) = \chi_f(G)^k$. One can amplify constant gaps in $\chi_f(G)$ to gaps of the form $n^c$ by setting $k = O(\lg n)$. Note however that the resulting graph $G^k$ has superpolynomially many nodes, and hence the reduction is neither polynomial time, nor will it give a hardness of approximation gap of the form $N^c$ where $N = |V(G^k)|$. This difficulty can be overcome by randomly selecting a polynomial-size vertex induced subgraph $G'$ of $G^k$.

We summarize some known properties of graph amplification in the following lemma.

**Lemma 1** Vertex induced subgraphs $G'$ of $G^k$ have the following properties:

1. $\chi_f(G') \leq (\chi_f(G))^k$.
2. If $\alpha(G) \leq C$ then $\alpha(G') \leq k|V(G)|$, for nearly all subgraphs $G'$ with $(|V(G)|/C)^k$ vertices.
3. $\chi_f(G') \geq \frac{|V(G')|}{k|V(G)|}$, for all $G'$ satisfying $\alpha(G') \leq k|V(G)|$.

Property 1 follows from multiplicity of $\chi_f(G)$ under graph products, and monotonicity when taking subgraphs. That is, $\chi_f(G^k) \leq \chi_f(G)^k$ and for any $G'$ that is a subgraph of $G^k$, $\chi_f(G') \leq \chi_f(G^k)$. Property 2 of Lemma 1 is proven (with slightly stronger parameters) in [12]. Property 3 follows immediately from Equation 2.

### 3.5 Randomized PCPs

Randomized PCPs fix a distribution over PCP witnesses for any input $x \in L$. (In ordinary PCPs, this distribution may be concentrated on only one witness.) For simplicity we assume that all PCP witnesses in the support have the same size. This notion (denoted RPCP) was implicitly used in [11] for the purpose of generating communication-efficient zero-knowledge two-prover proofs for $\mathsf{NP}$. It was used in [14] to show the hardness of certain probabilistic inference problems.

It is convenient to view an RPCP as follows: A prover has an initial witness $W$ to the NP-statement and a source of randomness $R_P$. Based on $W$ and $R_P$ the prover generates a witness $w$ for the PCP verifier to check. If $x \in L$ then $w$ will have perfect completeness regardless of $R_P$. If $x \notin L$, there is no legal witness $w$, and the verifier accepts with probability at most $e$.

**Remark:** Since we consider computationally unbounded provers, $W$ is technically superfluous. However, all of our constructions use witness $W$ as input, and in fact run in polynomial time given $W$ and $R_P$ as input.

As with ordinary PCPs, an RPCP has parameters $e, R$ and $A$, but instead of parameter $\ell$ we use a new parameter $\rho$, defined as follows.

**Definition 3** For an RPCP system and input $x \in L$, let $S_i$, for $1 \leq i \leq R$ denote a subset of the bits that the verifier may request to see and let $v$ denote a setting to those bits. We limit ourselves only to those settings that if read by $V$ would cause $V$ to accept. The covering parameter $\rho$ is the minimum over all pairs $(S_i, v)$ of the probability, taken over the random choice of RPCP witness $w$, that the bits $S_i$ are set to $v$.

Intuitively, having a large covering parameter means that there are no rare accepting views. Note that with an ordinary PCP for $x \in L$, in which only one PCP witness is used, on any subset $S_i$ there is exactly one view that is observed with probability 1. Thus, most of the possible ways that $V$ might accept $x$ will never be observed and hence $\rho$ is typically equal to 0. Note also that $\rho$ can be at most $1/\ell$, so requiring a large $\rho$ is a stronger condition than requiring a small $\ell$.

As we shall later see, known zero knowledge proof systems coincidently happen to give RPCPs with large $\rho$.

### 3.6 Putting the pieces together

Our reduction for proving hardness of approximating $\chi(G)$ has the following steps:

1. Construct an RPCP for an NP-complete problem (such as 3-SAT), with parameters $(e, R, \rho, A)$, where $A$ (and hence $R$) is polynomial in the input length, and $\rho > eR/A$.

2. On input $x$, apply the FGLSS reduction to the RPCP, and obtain a graph $G_0$. Observe that the FGLSS reduction is indifferent to whether we start with a PCP or an RPCP. Due to the parameter $\rho$, the FGLSS reduction now establishes a gap not only in $\alpha(G_0)$, but also in $\chi_f(G_0)$.

3. Use randomized graph products to randomly obtain a new graph $G$. As we shall see, if $x \in L$ then $\chi_f(G) \leq N^\delta$, where $N = |V(G)|$ and $\delta = \log(1/\rho)/\log(A/eR)$, and if $x \notin L$ then $\chi_f(G) \simeq N$, with high probability. Essentially the same bounds hold also for $\chi(G)$. Hence if

$\chi(G)$ can be approximated within a factor of $N^{\frac{\lg a\rho}{\lg a}-\epsilon}$, where $a = A/eR$, then $x \in L$ can be decided with high probability in random polynomial time.

In Sections 4 and 5 we present various ways of implementing the first step of the reduction. The effect of the second step is summarized in the following lemma.

**Lemma 2** Let $G_0$ be the graph obtained by the FGLSS reduction from an RPCP that on input $x$ has parameters $(R, e, \rho, A)$. Then $G_0$ has the following properties:

1. $|V(G_0)| = A \leq \frac{R}{\rho}$.
2. If $x \in L$, $\chi_f(G_0) \leq 1/\rho$.
3. If $x \notin L$, $\alpha(G_0) \leq eR$. Hence $\chi_f(G_0) \geq A/eR$.

**Proof:** Properties 1 and 3 are standard properties of the FGLSS reduction. It remains to show that if $x \in L$ then $\chi_f(G_0) \leq 1/\rho$. Recall that each vertex of $G_0$ can be written as $(S_i, v)$ where $S_i$ is a subset of bits requested by the verifier, and $v$ is a view that will cause $V$ to accept. Furthermore, each PCP witness $w$ that causes $V$ to always accept corresponds to an independent set with $R$ vertices, each having a different $S_i$ component. Let $\mathcal{I}$ be the probability measure induced on independent sets of $G_0$ by choosing $w$ with measure as specified by the RPCP. By the definition of $\rho$, the probability that $I \leftarrow \mathcal{I}$ contains $(S_i, v)$ is at least $1/\rho$. By the definition of $\chi_f$, $\chi_f(G_0) \leq 1/\rho$. $\square$

For two of the RPCPs for NP that we present, it happens that $|V(G_0)| = R\ell$, and that $\rho = 1/\ell$ whenever $x \in L$. This implies that if $x \in L$ then $\chi_f(G_0) \leq \ell$, and if $x \notin L$ then $\chi_f(G_0) \geq \ell/e$, thus giving a hardness gap of $1/e$ for approximating $\chi_f(G_0)$.

Note that Lemma 2 does not directly imply any gap with respect to $\chi(G)$. However, for the above two cases, $|\mathcal{I}| = \ell$, implying that $\chi(G_0) = \ell$ whenever $x \in L$. Since $\chi(G) \geq \chi_f(G)$ for any graph $G$ (and in particular for $G_0$ when $x \notin L$), for these very well behaved RPCPs the gap of $1/e$ is also established for $\chi(G_0)$.

To analyze the third step of our reduction, we invoke Lemma 1. Let $a = A/eR$. Let $G$ be a vertex induced subgraph obtained by randomly selecting $N = a^k$ vertices from $G_0^k$, where $k = c \log n$, and $c$ is a sufficiently large constant. If $x \in L$, then $\chi_f(G) \leq 1/\rho^k$. If $x \notin L$, then with high probability, $\chi_f(G) \geq N/kA$. Expressing the multiplicative gap between $1/\rho^k$ and $N/kA$ as $N^\alpha$ we have

$$\begin{aligned}
\alpha &= \log_N \frac{N/kA}{1/\rho^k} \\
&= \frac{\log a^k \rho^k - \log kA}{\log a^k} \\
&= \frac{\log a\rho}{\log a} - \frac{\log kA}{\log a^k}.
\end{aligned}$$

By making $c$ and hence $k$ sufficiently large, the last term can be made arbitrarily small. Using Equation 1 we obtain similar bounds for $\chi(G)$. The following lemma summarizes these bounds in terms of $N = |V(G)|$, when $c$ is sufficiently large.

**Lemma 3** Suppose that one can generate an RPCP for $\mathsf{NP}$ with parameters $(e, R, \rho, A)$, where $A$ is polynomially bounded. Then it is hard to approximate $\chi(G)$ to within

$$N^{\frac{\lg a\rho}{\lg a} - \epsilon},$$

where $N = |V(G)|$, $a = A/eR$ and $\epsilon$ is an arbitrarily small positive constant, assuming $\mathsf{NP} \not\subseteq \mathsf{ZPP}$.

The most naive use of a randomized reduction would seem to give a $\mathsf{NP} \not\subseteq \mathsf{BPP}$ condition. However, we achieve one-sided error by noting that when $x \in L$ then the graph $G$ generated will always have a small chromatic number. Thus, if the randomized reduction results in a large chromatic number, this serves as a witness that $x \notin L$. This gives a $\mathsf{NP} \not\subseteq \mathsf{coRP}$ condition. Finally, by a standard result, $\mathsf{NP} \subseteq \mathsf{coRP}$ is equivalent to $\mathsf{NP} \subseteq \mathsf{ZPP}$.

**Remark:** In the context of independent set, the use of randomized graph products (as in [8]) gives the condition $NP \notin BPP$. To obtain hardness of approximation results for $\alpha(G)$ under the $NP \notin ZPP$ assumption, Zuckerman [31] proposes a different randomized amplification technique that works directly on the PCP rather than on graphs.

This result can be compared to the corresponding theorem for $\alpha(G)$ by setting $\rho = R/A$. This value corresponds to the optimal case where each accepting view occurs equally often. Then, by Lemma 3, it is hard to approximate $\chi(G)$ to within

$$\begin{aligned} N^{\frac{\log 1/e}{\log A/eR} - \epsilon} &= N^{\frac{1}{1 + log_{1/e} A/R} - \epsilon} \\ &= N^{\frac{1}{1+f} - \epsilon}, \end{aligned}$$

where $f$ is the number of asymptotic free bits of the PCP. This is the same as the equation obtained in [5] for relating asymptotic free-bit complexity and the hardness of computing $\alpha(G)$. Hence, if we can randomize a PCP so as to maintain its free-bit efficiency while making the accepting views occur with close to the same probability, then we can obtain a hardness result for $\chi(G)$ that matches the hardness result for $\alpha(G)$ obtained from the original PCP.

# 4 Simple reductions to chromatic number

In this section we give RPCPs that, using the methodology of the previous section, imply new reductions to chromatic number.

## 4.1 A reduction from max-3-coloring

Petrank's [29] reduction from max-3-sat to max-3-coloring implies the following: For any NP-complete language $L$ and input $x$ one can produce a graph $H$ such that

1. if $x \in L$, then $H$ is 3-colorable, and
2. if $x \notin L$ any coloring $\chi$ of $H$ will miscolor at least a $q$-fraction of the edges of $G$, for some constant $q > 0$.

P$(H, \chi)$ /* Prove that $H$ is 3-colorable */

1. Uniformly select a permutation $\pi$ on $\{0, 1, 2\}$ and output

$$\chi' = \pi(\chi) = \pi(\chi(1)), \ldots, \pi(\chi(n)).$$

V$(H, \chi')$ /* verify that $\chi'$ 3-colors $H$ */

1. $V$ uniformly selects an edge $(x, y) \in H$ and accepts iff $\chi'(x) \neq \chi'(y)$.

Figure 1: A reduction from max-3-coloring [19].

We now give an RPCP for the statement that $H$ is 3-colorable, isomorphic to the standard zero-knowledge proof for 3-coloring [19]. Let $H$ have $n$ nodes (labeled $1, \ldots, n$) and $m$ edges, and view a coloring $\chi$ as a map from $\{1, \ldots, n\}$ to $\{0, 1, 2\}$. We define our RPCP by a randomized prover $P$ that, when given $H$ and a 3-coloring $\chi$ for $H$, produces a PCP $\chi'$, and by a randomized verifier $V$ that checks $\chi'$. We describe $P$ and $V$ in Figure 1.

It is not hard to verify that the RPCP and $V$ has perfect completeness. We now analyze the parameters $(R, e, \rho, A)$. Clearly, $R = m$ and $A = 6R$. The error $e$ is at most $1 - q$, since $V$ will pick a miscolored edge with probability at least $q$. To determine $\rho$, note that for any edge $(x, y) \in H$, $(\chi'(x), \chi'(y))$ is distributed uniformly over

$$\{(0, 1), (0, 2), (1, 0), (1, 2), (2, 0), (2, 1)\}.$$

Hence $\rho = 1/6$. Computing $a = A/eR = 6/(1 - q)$, we have by Lemma 3 that it is hard to approximate $\chi(G)$ to within $N^c$, where $c = \lg[1/(1 - q)]/\lg[6/(1 - q)] - \epsilon$ for any $\epsilon > 0$. Note that $c > 0$ in this case.

In fact, the graph $G_0$ is simple enough that it can be understood without resorting to our machinery. $G_0$ has $6m$ vertices of the form $(e, c)$ where $e$ ranges over the $m$ edges of $H$ and $c$ ranges over the 6 valid 3-colorings of an edge. Two vertices $(e_1, c_1)$ and $(e_2, c_2)$ are connected by an edge if $e_1$ and $e_2$ intersect at a vertex of $H$ and $c_1$ and $c_2$ disagree on the coloring of this vertex. A 3-coloring $\chi$ of $H$ induces an independent set of size $m$ in $\mathcal{P}(G)$. As $\pi$ ranges over its 6 possibilities, $\chi' = \pi(\chi)$ induces 6 independent sets of size $m$. Furthermore, these independent sets are disjoint and cover $G_0$. Hence, $G_0$ has a chromatic number of 6. If $H$ is not 3-colorable (and hence $qm$ of its edges are miscolored) then the largest independent set is of size $m(1 - q)$.

## 4.2 A reduction from max-3-sat

While the previous reduction was very simple, it builds upon the hardness of approximating max-3-coloring, which involves a further reduction from max-3-sat. Here we give a direct reduction from max-3-sat to approximate chromatic number. While slightly more complicated, this reduction again has the property that $G_0$ is highly structured and easy to understand.

Zero-knowledge PCPs based on 3-sat follow from the construction in [11]. We review a simpler (though weaker) zero-knowledge PCP from [14]. As well as being simpler it is in some ways better

P$(F, \vec{a}, Q)$ /* Prove that $F$ is satisfiable by $\vec{a}$ */

1. Compute $s_{i,j} = a_{ij} \oplus t_{ij}$. Uniformly choose a masking bit $b$ and rotation $r \in Z_3$. Compute $\vec{a}'$, $S'$ and $Q'$ by

$$\begin{aligned} a_i' &= a_i \oplus b, \\ s_{ij}' &= s_{i,j+r} \oplus b, \\ q_i' &= q_i + r. \end{aligned}$$

2. Output $(b, r, \vec{a}', S', Q')$.

V$(F, b, r, \vec{a}', S', Q')$ /* verify that $F$ is satisfiable */

1. Uniformly select $i \in \{1, \ldots, m\}$ and $j \in \{0, 1, 2\}$.

2a. (Check that $s_{i,q_i} = 1$) With probability $\frac{1}{2}$, $V$ first looks at $q_i'$ and $b$, then looks at $s'_{i,q_i'}$. He accepts if and only if $s'_{i,q_i'} \oplus b = 1$.

2b. (Check that $s_{ij} = a_{ij} \oplus t_{ij}$) Otherwise, $V$ first looks at $r$ and then looks at $s'_{i,j-r}$ and $a'_{i,j}$. He accepts if and only if $s'_{i,j-r} = a'_{ij} \oplus t_{ij}$.

Figure 2: Zero-knowledge PCP based on max-3-sat [14].

behaved than the older construction. It's weaknesses, being two rounds instead of one and being restricted to honest verifiers, are irrelevant to our application.

First, some preliminary notation. Let

$$F = C_1 \wedge \cdots \wedge C_m$$

be a 3-sat formula over $x_1, \ldots, x_n$. Let $v_{ij}$ denote the index of the $j$th variable of $C_i$, where $1 \le i \le m$ and $0 \le j \le 2$, and let $t_{ij}$ be 1 if this variable is negated, 0 otherwise. Let $\vec{a} = a_1, \ldots, a_n$ denote an assignment for $x_1, \ldots, x_n$. We define $x_{ij}$ as notational shorthand for $x_{v_{ij}}$ and define $a_{ij}$ analogously. For example, if $C_4 = (x_2 \vee \bar{x}_5 \vee x_7)$ then $v_{4,1} = 5$ and $x_{4,1}$ is "shorthand" for $x_5$; it is not a new variable.

We define $S = \{s_{ij}\}$ by $s_{ij} = a_{ij} \oplus t_{ij}$. Intuitively, $s_{ij}$ indicates whether $\vec{a}$ satisfies the $j$th literal in $C_i$. The assignment $\vec{a}$ satisfies $F$ if and only if for every $1 \le i \le m$ there exist $q_i$ such that $s_{i,q_i} = 1$. That is, $Q = q_1, \ldots, q_m \in \{0, 1, 2\}$ specifies one way in which $\vec{a}$ satisfies $F$. Figure 2 describes the zero-knowledge PCP of [14]. In this discussion, $\oplus$ denotes addition mod 2 and $+/-$ denotes addition/subtraction mod 3.

We briefly review the analysis of this protocol. By a tedious but straightforward case analysis, if $(\vec{a}, Q)$ represents a correct witness that $F$ is satisfiable, then $(P, V)$ will always accept. To analyze the soundness of the protocol, note that the values of $(b, r, \vec{a}', S', Q')$ implicitly define a witness $(\vec{a}, S, Q)$ by $a_i = a_i' \oplus b$, $s_{ij} = s'_{i,j-r} \oplus b$ and $q_i = q_i' - r$. If for some clause $C_i$, $\vec{a}$ does not satisfy $C_i$, then either $s_{i,q_i} \neq 1$ ($q_i$ points to a variable that doesn't really satisfy $C_i$ according to $S$) or $s_{i,q_i} \neq a_{i,q_i} \oplus t_{i,q_i}$ ($S$ is incorrect). In the former case, $V$ will reject if it executes Step 2a, since $s_{i,q_i} = s'_{i,q_i'} \oplus b$. In the latter case, $V$ will reject if it chooses $j = q_i$ and executes Step 2b. Thus, $V$

will reject with probability at least $1/6$.

To analyze the zero-knowledge properties of this protocol, we note that if $V$ executes Step 2a, then its view is determined by $(q_i', b)$ (since the view of $s'_{i,q_i'}$ is determined by these values). For any setting of $\vec{a}$ and $Q$ (and hence $q_i$), there is a bijection between the 6 possible values of $(r, b)$ and the observed values of $(q_i', b)$. If $V$ executes Step 2b, then its view is determined by $(r, s'_{i,j-r})$ (since $a'_{ij}$ is then determined). Similarly, for any setting of $\vec{a}$ and $Q$ there is a bijection between the 6 possible values of $(r, b)$ and the observed values of $(r, s'_{i,j-r})$.

The PCP theorem [3] implies for input $x$ and NP-complete language $L$, we can generate $F$ so that

- if $x \in L$ then $F$ is satisfiable, and
- if $x \notin L$ then at most $(1-q)m$ of $F$'s clauses can be simultaneously satisfied for some constant $q > 0$.

Summarizing the above analysis, the above protocol would then have the following properties:

1. (perfect completeness) If $F$ is satisfiable then $(P, V)$ will always accept.
2. (soundness) Otherwise, $V$ will reject with probability at least $q/6$.
3. (zero-knowledge) If one fixes the coin tosses of $V$, there will always be exactly 6 views that will cause him to accept. If one then fixes the satisfying witness $(\vec{a}, S, Q)$ used by $P$, there will be a bijection between the 6 possible values of $(b, r)$ and the satisfying view observed by $V$.

We can compute the parameters $(R, e, \rho, A)$ as follows. By inspection of $V$, $R = 6m$ and $A = 36m$. By the soundness property given above, $e = 1 - q/6$. Thus, $a = 6/(1-q)$. By the zero-knowledge property, $\rho = 1/6$. Hence, one can use Lemma 3 to obtain that it is hard to approximate $\chi(G)$ to within $N^c$, where $c = \lg[1/(1-q)]/\lg[6/(1-q)] - \epsilon$ for any $\epsilon > 0$.

We note that the intermediate graph $G_0$ again behaves more nicely than is strictly required by our analysis. If $x \in L$ then not only is $\chi_f(G_0) = 6$ but in fact $G_0$ is disjointly covered by 6 independent sets of size $6m$. If $x \notin L$ then $G_0$ has no independent set larger than $(1 - q/6)6m$. Thus, as before there is actually no need to consider fractional chromatic numbers in our analysis. This run of luck ends here.

# 5 Randomizing Long-Code Based Proofs

In this section we develop techniques for randomizing the long-code based PCPs of [5, 21, 22]. These results may be viewed as transformations that take a PCP (or other witness) and produce a new PCP $(P, V)$ whose verification requires relatively few free bits. We give a method for creating proof systems $(P', V')$ that achieve a near-optimal value for $\rho$ without requiring more free bits than $(P, V)$. We apply them to the protocol of [22]; they may also be (and in some cases more simply) applied to the earlier protocols of [5, 21].

## 5.1 The long code

There are $2^{2^k}$ boolean functions on $k$ bits. The long code $LC(x)$ for a $k$-bit string $x$ consists of the values of all functions $f$ applied to $x$. Typically, proofs using the long code check random locations in the long code, and hence they are checking the values of random functions on certain strings. In the PCP of [22], these strings consist of all $k$-bit subsets of the original witness. For each such string, the PCP contains its long code.

## 5.2 Preliminary lemmas

Let $D$ be a probability distribution over a finite set of strings. For a string $x$, let $P_D(x)$ denote the probability given to $x$ under $D$. For a function $f$, define $bias_D(f)$ by

$$bias_D(f) = |Pr(f(x) = 1) - Pr(f(x) = 0)| = 2|E(f(x)) - \frac{1}{2}|,$$

where $x$ is chosen according to $D$.

**Lemma 4** Let $D$ be a distribution on $\{0,1\}^n$ such that for all $x \in \{0,1\}^n$, $P_D(x) < 2^{-5n/6}$. Then for all but a $e^{-2^{n/3-1}}$ fraction of functions $f : \{0,1\}^n \longrightarrow \{0,1\}$,

$$bias(f) < 2^{-n/6}.$$

**Proof:** If we choose $f$ uniformly, then for each $x \in D$ we choose $f(x)$ uniformly from $\{0,1\}$. Denote the $N = 2^n$ elements of $\{0,1\}^n$ by $x_1, \ldots, x_N$ and define $S$ by

$$S = \sum_{i=1}^{N} P_D(x_i) X_i,$$

where the $X_i$ are independent random variable distributed uniformly over $\{-1,1\}$. The value of $bias_D(f)$, where $D$ is fixed and $f$ is uniformly distributed, has the same distribution as $|S|$.

Clearly $E(S) = 0$. We can bound the tails of $S$ using a standard Chernoff bound. If $P_D(x_i) \leq \Delta$ for all $i$, we can define the random variables $Y_i = (P_D(x_i)/\Delta)X_i$. Then, noting that $|Y_i| \leq 1$ and that $S = \Delta \sum_{i=1}^{N} Y_i$, we can use ([2], Theorem A.16, p. 240) to conclude that

$$Pr(S > \Delta\lambda) < e^{-\lambda^2/2N}$$

for $\lambda > 0$. We can set $\Delta = 2^{-5n/6}$ by the condition of the lemma; setting $\lambda = 2^{2n/3}$ proves the conclusion of the lemma. □

In the long-code based PCPs, the verifier queries the values of random functions $f_1, \ldots, f_m$ on sets of variables $S_1, \ldots, S_k$. Our randomizing process ensures that many of the underlying variables are random (uniformly and independently distributed). We argue that if $f_1, \ldots, f_m$ are uniformly chosen, and the input sets have sufficiently many truly random bits, their outputs will be nearly uniformly and independently distributed. A slight technical complication arises because $S_1, \ldots, S_k$ overlap, and these shared variables may cause dependencies in the function outputs.

Let $n_i = |S_i|$, let $r_i$ denote the number of truly random inputs in $S_i$ and let $a_i = n_i - r_i$ denote the number of nonrandom inputs, which we assume are chosen adversarially. Let $s(i)$ denote the index of the set that $f_i$ is being applied to. That is, $f_i(S_{s(i)})$ is one of the functions being computed.

**Definition 4** Given $S_1, \ldots, S_k$ and $s(1), \ldots, s(m)$, we say that functions $f_1, \ldots, f_m$ are *well balanced* if for all setting of the nonrandom bits, and for all $y_1, \ldots, y_m \in \{0, 1\}$,

$$Pr(f_i(S_{s(i)}) = y_i, 1 \leq i \leq m) > 2^{-m}/2,$$

where the probability is taken over the setting of the random bits.

We now show that if the sets have enough random inputs, most choices of $f_1, \ldots, f_m$ will be well balanced.

**Lemma 5** Suppose that for all $i$, $n_i > 6 \lg m + 12$, $r_i - m - 1 \geq 5n_i/6$ and

$$2^{\frac{1}{3}n_i - 1} - 2^{\frac{1}{6}n_i} \geq m + a_1 + \cdots + a_i.$$

Then all but a

$$m \sum_{i=1}^{k} e^{-2^{n_i}}$$

fraction of the choices of $f_1, \ldots, f_m$ will be well balanced.

**Remark:** The bounds above are far from optimal, but are sufficient for our purposes. The condition that $r_i - m - 1 \geq 5n_i/6$ is the significant one; the others are for technical reasons to make the calculation go through, and will be trivially satisfied in the lemma's applications. Similarly, $m \sum_{i=1}^{k} e^{-2^{n_i}}$ should be read as "small" for our purposes.

**Proof:** W.l.o.g. we assume that if $i < j$ then $s(i) \leq s(j)$. We say that $f_i$ is *low-biased* if for all $y_i \in \{0, 1\}$,

$$Pr(f_i(S_{s(i)}) = y_i) > \frac{1}{2} - 2^{-n_{s(i)}/6},$$

even when conditioned on all possible settings of the nonrandom inputs to sets $S_1, \ldots, S_{s(i)}$ and all possible values of $f_1(S_{s(1)}), \ldots, f_{i-1}(S_{s(i-1)})$.

We note that if $f_1, \ldots, f_i$ are low-biased then for any $y_1, \ldots, y_i \in \{0, 1\}$, the probability that $f_j(S_{s(j)}) = y_j$ for all $1 \leq j \leq i$ is at least

$$\begin{aligned}
\prod_{j=1}^{i} \left( \frac{1}{2} - 2^{-n_{s(j)}/6} \right) &\geq \prod_{j=1}^{i} \frac{1}{2}(1 - 1/2m) \\
&\geq \left( \frac{1}{2} \right)^m (1 - 1/2m)^m \\
&\geq 2^{-m}/2
\end{aligned}$$

regardless of the setting of the nonrandom variables. In particular, if $f_1, \ldots, f_m$ are low-biased they will also be well behaved.

Conceptually, we randomly choose $f_1, f_2, \ldots$, and hope that each $f_i$ is low-biased. If we ever choose a function that isn't low-biased, we give up. However, we show that at each step, assuming that $f_1, \ldots, f_{i-1}$ are low-biased, nearly all $f_i$ are low-biased.

We argue that whatever the settings of the (relevant) nonrandom variables and $y_1, \ldots, y_{i-1}$, the conditional distribution on the inputs to $f(i)$ (i.e. the variables in $S_{s(i)}$) meets the conditions of Lemma 4. By Bayes law, the probability of any event $A$ condition on an event $B$ is at most the *a priori* probability of $A$ divided by the probability of $B$. The *a priori* probability of a given setting of variables in $S_{s(i)}$ is initially at most $2^{-r_{s(i)}}$, since $r_{s(i)}$ of the bits are uniform. Assuming that $f_1, \ldots, f_{i-1}$ are low-biased, the probability that $f_j(S_{s(j)}) = y_j$ for $1 \leq j < i$ is at least

$$\prod_{j=1}^{i-1} \left( \frac{1}{2} - 2^{-\frac{1}{6}n_i} \right)$$

regardless of the setting of the nonrandom variables. Since $n_{s(j)} > 6 \lg m + 12$,

$$\left( \frac{1}{2} - 2^{-\frac{1}{6}n_i} \right) \geq \frac{1}{2} \left( 1 - \frac{1}{2m} \right).$$

Thus, the above product is at least

$$\begin{aligned} (1/2)^{i-1}(1-1/2m)^{i-1} &\geq (1/2)^m(1-1/2m)^m \\ &\geq 2^{-m-1}. \end{aligned}$$

Thus, the probability of any given setting of the variables in $S_{s(i)}$, conditioned on $y_1, \ldots, y_{i-1}$, is at most

$$2^{-r_{s(i)}+m+1} \leq 2^{-5n_{s(i)}/6},$$

when $r_{s(i)}$ and $m$ are as in the statement of the lemma.

Hence, for any setting of the nonrandom variables in $S_1, \ldots, S_{s(i)}$ and $y_1, \ldots, y_{i-1}$, the conditional distribution on the inputs to $f_i$ meets the requirements of Lemma 4. (The setting of the other nonrandom variables do not affect this conditional distribution). Thus, for all but at most a $e^{-2^{(\frac{1}{3}n_{s(i)}-1)}}$ fraction of the possible choices of $f_i$, the conditional probability that $f_i(S_{s(i)}) = y_i$ is at least $\frac{1}{2} - 2^{-\frac{1}{6}n_i}$. Now, there are at most

$$2^{i-1}2^{a_1+\cdots+a_{s(i)}}$$

possible settings for $y_1, \ldots, y_{i-1}$ and the relevant nonrandom variables. Hence, at most a

$$\left(2^{i-1}2^{a_1+\cdots+a_{s(i)}}\right) e^{-2^{(\frac{1}{3}n_{s(i)}-1)}} \leq e^{-2^{\frac{1}{6}n_i}}$$

fraction of the possible choices of $f_i$ fails for any of these settings.

Finally, we can bound the probability of ever picking an $f_i$ that is not low-biased by

$$\sum_{i=1}^{m} e^{-2^{\frac{1}{6}n_{s_i}}} \quad \leq \quad m \sum_{i=1}^{k} e^{-2^{\frac{1}{6}n_i}},$$

which implies the lemma. $\square$

## 5.3 Overview of Håstad's PCP.

We review here the PCP described in [22]. Håstad starts with a redundant 3CNF formula $F$ (pedantically, a family of formulas) with $n$ variables and $2n$ clauses in which each variable $x_i$ appears exactly 6 times. The construction requires that for some positive constant $e$, either $F$ is satisfiable or at most a $(1-e)$ fraction of $F$'s clauses may be satisfied simultaneously.

Given a set $S = \{s_1, \ldots, s_c\}$ where $s_1 \leq \cdots \leq s_c$ and a function $f : \{0,1\}^c \longrightarrow \{0,1\}$, we use the notation $f(S)$ to denote the function $f$ applied to the variables indexed by $S$. That is, $f(S)$ is shorthand notation for $f(x_{s_1}, \ldots, x_{s_c})$. For some constant $c$, the PCP simply consists of a listing of the values of $f(S)$ where $f$ ranges over all boolean functions over $\{0,1\}^c$, and $S$ ranges over all subsets of $\{1, \ldots, n\}$ of size $c$. Note that the PCP implicitly includes all functions on all subsets of size less than $c$.

The verifier's view of running this PCP may be summarized as follows. In this description, all variables are either sets or constants whose relative sizes will be discussed shortly.

*[Set selection phase]*

1. The verifier first uniformly chooses a set $U$ of size $u$.
2. The verifier uniformly chooses sets $V_1, \ldots, V_{k_1}$, subject to $|V_i| = v$ and $U \subset V$.
3. For $1 \leq i \leq k_1$, the verifier chooses sets $W_{i,j}$, where $1 \leq j \leq k_2$, as follows. For each variable in $V_i$, the verifier chooses uniformly one of the 6 clauses this variable appears in, and includes all of this clause's variables in $W_{i,j}$.

*[function query phase]*

4. The verifier queries (from the PCP) the values of $\delta s$ random functions on $U$.
5. The verifier queries the values of $\delta k_1^{-1} s$ random functions on each set $V_i$.
6. The verifier queries the values of $\delta(k_1^{-1} k_2^{-1})s$ random functions on each set $W_{i,j}$.
7. The verifier makes consistency checks on the PCP. That is, it makes queries whose answers can be inferred from the answers to previous queries, and rejects if the actual answers differ from the inferred answers.

Step 7 is crucial to the soundness of the PCP, and indeed is the heart of the checking procedure. However, since we are only interested in the view of the verifier when the PCP is correctly constructed, the answers to these queries are for our purposes irrelevant to the view of the verifier, since they are completely dependent on the answers to previous queries.

We now discuss the relationship between the constants mentioned above. The parameter $\delta > 0$ may be set arbitrarily low. The resulting PCP will have error at most $c_0 2^{-s}$ for some constant $c_0 > 0$ and may be checked with $3\delta s$ free bits, thus requiring approximately $3\delta$ amortized free bits. Once $\delta$ is selected, $k_1, k_2, s, u$ and $v$ are chosen in succession. Each constant must be chosen sufficiently large based on the values of the previous constants and possibly on the soundness $e$ of the redundant 3SAT formula (i.e. the formula is either satisfiable or only a $(1-e)$ of the clauses are simultaneously satisfiable). We exploit the fact that $u$ may be chosen as large as desired, after which $v$ may be chosen as large as desired.

## 5.4 Randomizing Håstad's PCP

To randomize Håstad's PCP, we use the fact that the verifier initially queries random locations on the long code. That is, it chooses subsets of bits and then queries the value of random functions on these subsets. Summarizing Section 5.2, we observe that random functions are with high probability very sensitive to all their inputs. Thus, if we can ensure that many bits in each subset are randomly distributed, the values of these functions, and hence the verifier's views, will be randomly distributed.

Recall that the [22] construction begins with a redundant 3CNF formula. We randomize this formula by appending to it dummy clauses of the form $(q_i \vee r_j \vee r_k)$. The $q$ and $r$ variables are dummy variables that do not appear in the original formula. We also require that, as with the original formula, each variable appears exactly 6 times. We augment the satisfying assignment of the original formula by setting each $q_i$ variable to be 1 and setting each $r_i$ variable at random.

Suppose that the original formula $S$ has $n$ variables and the randomized formula $S'$ has $kn$ variables. If $S$ is satisfiable then $S'$ is as well. If at most a $(1-e)$ fraction of the clauses are simultaneously satisfiable, then at most a $(1-e/k)$ fraction of the clauses are simultaneously satisfiable. Hence $k$ can be an arbitrarily large constant, and $S'$ will remain a redundant 3CNF formula in which each variable appears exactly 6 times. Thus, $S'$ will be a valid starting point for the construction of [22]. Note that by reducing $e$ to $e/k$ the values of the other constants must be chosen to be larger. However, this is not a concern for us, since for other reasons we want $u$ and $v$ to be made quite large anyway.

## 5.5 Analyzing the randomized PCP.

We argue that by randomizing the original 3SAT formula and setting $u$ and $v$ appropriately, then for most of the verifier's coin tosses, each possible view will occur reasonably often.

**Lemma 6** Suppose that we randomize the original 3CNF formula, expanding it from $n$ variables to $1000n$ variables. Consider the PCP for this 3CNF, with parameters $\delta, k_1, k_2, s, u, v$, as defined in [22] and reviewed in Section 5.3. Then for any constant $c < 1$ (that may depend on $\delta, k_1, k_2$, and $s$), one can make $u$ and $v$ sufficiently large, such that for a fraction $c$ of the verifier's coin tosses, each of the $2^m$ possible accepting views (where $m = 3\delta s$) will appear with probability at least $2^{-m}/2$.

We state more precisely what is means for $u$ and $v$ to be sufficiently large. The upper bound on the error probability $(c_0 2^{-s})$ in Håstad's PCP holds whenever $u$ is chosen sufficiently large, depending on $\delta, k_1, k_2$ and $s$, and $v$ is chosen sufficiently large, depending on $\delta, k_1, k_2, s$ and $u$. We must make $u$ and $v$ sufficiently large so as to satisfy an additional requirement (making $c$ arbitrarily close to 1). To do this we may choose $u$ to be larger than is required by Håstad's PCP, and then choose $v$ to be larger than is required by Håstad's PCP, even given the larger $u$. However, this does not not affect the soundness analysis of the PCP.

**Proof:** Recall that the view of Håstad's PCP consists of random boolean functions $f_1, \ldots, f_m$ applied to the sets $U, V_i$ and $W_{i,j}$ (for $1 \leq i \leq k_1$ and $1 \leq j \leq k_2$). The verifier's coin tosses dictate both the sets and the functions. We show that with high probability, the verifier's choice of sets $U, \{V_i\}, \{W_{i,j}\}$ meet the condition of Lemma 5. Then we show that the under these conditions, most choices of the functions give balanced outputs.

First, we observe that the $q_i$ variables are irrelevant since they are always 1. Choosing a random function, then setting a prespecified set of the input bits to 1 is equivalent to first deleting these inputs and choosing a random function on the remaining inputs. Hence, we will conceptually delete these variables from our variable sets for the rest of the proof.

Next, we note that with high probability, each function will have a large fraction of random inputs. Of the $1000n$ variables in the satisfying assignment, only $n$ are from the original satisfying assignment and $666n$ are randomly assigned (the remaining $333n$ are fixed at 1, and are ignored). By a standard Chernoff bound, $U$ will have more than $u/2$ randomly assigned variables and less than $u/900$ original (nonrandom) variables, with probability at least $1 - e^{-\Omega(u)}$. Similarly, with probability at least $1 - k_1 e^{-\Omega(v)}$, each $V_i$ will have more than $v/2$ random variables and less than $v/900$ nonrandom variables. Finally, if the condition on the $V$ sets holds, then each $W_{i,j}$ will have more than $v/2$ random variables and less than $v/300$ nonrandom variables.

(Note that we implicitly ignore issues such as the possibility that randomly chosen variables and clauses might be the same or intersect. Such events occur with probability $o(1)$, since $n$ is nonconstant.)

We can now use Lemma 5 to show that the outputs of $f_1, \ldots, f_m$ will be roughly uniform with probability at least $c$. Let $U$ correspond to $S_1$ and the $V$ and $W$ sets correspond to $S_2, \ldots, S_k$ (here, $k = 1 + k_1 + k_1 k_2$). By the above argument, with probability $1 - e^{-\Omega(u)} - k_1 e^{-\Omega(v)}$, $n_1 \geq r_1 > u/4$ and $a_1 < u/900$ and for $i > 1$, $n_i \geq r_i > v/4$ and $a_i < v/300$. When this occurs, for $u$ and $v$ sufficiently large, $n_i > 6 \lg m + 12$, $r_i - m - 1 \geq 5n_i/6$ and

$$2^{n_i/3-1} - 2^{n_i/6} \geq m + a_1 + \cdots + a_i.$$

(The last inequality is where we need $U$ to correspond to set $S_1$ and not some later set. We observe that $a_1 < u/900$ and $a_1 \cdots a_i < kv/300$.)

Hence, for sufficiently large $u$ and $v$, with probability at least $1 - e^{-\Omega(u)} - k_1 e^{-\Omega(v)}$, all but a $m \sum_{i=1}^{k} e^{-2^{n_i}}$ fraction of the $f_1, \ldots, f_m$ will be well balanced, and furthermore, $m \sum_{i=1}^{k} e^{-2^{n_i}}$ can be made arbitrarily small. The lemma follows. □

### 5.6 Achieving a near-optimal value of $\rho$.

Thus far, we have not bounded the covering parameter $\rho$ of our randomized protocol, since for some of the verifier's coin tosses there may be an accepting view that never appears. However, we can easily fix the above protocol to achieve a large value for $\rho$ while maintaining a low error. We observe that the verifier can determine in polynomial time whether its choice of

$$U, \{V_i\}, \{W_{i,j}\}, f_1, \ldots, f_m$$

will guarantee, for all choices of the nonrandom bits in these sets, that each possible accepting view will occur with probability at least $2^{-m}/2$, where the probability is taken over the choice of the random bits. Note that $m, u$ and $v$ are constants, so exhaustive search is feasible. If it does, then the verifier proceeds to check the PCP; otherwise it simply accepts. (More efficiently, the verifier can adjust its coin tosses to "miss" these bad choices, but this optimization is not needed.)

Let us now bound the error and $\rho$ for the resulting PCP. Clearly, $\rho \geq 2^{-m}/2$, where $m = 3\delta s$. The error obtained by Håstad's construction is at most $c_0 2^{-s}$; the error obtained by using the above modification is at most $c_0 2^{-s} + (1-c)$, where $c$ is from Lemma 6 and may be chosen arbitrarily close to 1. Choosing $c > 1 - c_0 2^{-s}$ gives a protocol with error at most $2c_0 2^{-s}$.

We now use our randomized PCP and Lemma 3 to conclude that it is hard to approximate the chromatic number to within $N^{1-\epsilon}$.

**Proof of Theorem 1**

The above RPCP has parameters $(R, e, \rho, A)$ in which $R$ is polynomially bounded and $(e, \rho, A)$ are bounded as follows:

$$\begin{aligned} e &\leq 2c_0 2^{-s} \\ \rho &\geq 2^{-3\delta s}/2, \text{ and} \\ A &\geq R \text{ (trivially, since the protocol has perfect completeness)} \end{aligned}$$

Defining $a = A/Re$, we have $a \geq 2^s/2c_0$. By Lemma 3, it is hard to approximate $\chi(G)$ to within $N^{\frac{\lg a\rho}{\lg a} - \epsilon}$ assuming $\mathsf{NP} \not\subseteq \mathsf{ZPP}$. We have,

$$\begin{aligned} \frac{\lg a\rho}{\lg a} &= 1 + \frac{\lg \rho}{\lg a} \\ &\geq 1 + \frac{-3\delta s - 1}{s - 1 - \lg c_0} \end{aligned}$$

Thus, by setting $\delta$ sufficiently small, and setting $s$ sufficiently large, $\lg a\rho/\lg a$ can be made arbitrarily close to 1. The theorem follows. $\Box$

## 5.7 Final remarks

We presented a method for randomizing PCPs that are based on the long code. We chose to illustrate the use of this method on the PCP of [22]. The reason for this choice is that currently, this PCP is the only one published that has an arbitrarily small number of amortized free bits. We remark however that our techniques are quite general, and can be applied also to earlier PCPs based on the long code (those of [5, 21]), as well as to simplified versions of the PCP of [22], recently announced by Håstad (private communication).

In an earlier version of this paper [15], we used a slightly different method to randomize the PCPs of [5, 21]. This method had the advantage of having the prover use only a constant number of random bits. The RPCPs that we constructed were distributed uniformly over $2^k$ PCPs (for some constant $k$), indexed by a $k$-bit parameter $R$. In each PCP, $R = \{r_1, \ldots, r_k\}$ is appended to all the variable sets the verifier might choose. Thus, instead of querying the value of $f(x_{i_1}, \ldots, x_{i_c})$, the verifier would query the value of

$$f(x_{i_1}, \ldots, x_{i_c}, r_1, \ldots, r_k).$$

We believe that the same approach can be applied essentially without change to the PCP of [22]. Again, the RPCP would consist of $2^k$ PCPs indexed by a $k$-bit string $R$, chosen uniformly. In each

PCP, the long code $LC(x)$ is replaced by a parameterized "long code" $LC_R(x)$ which contains the values of $f(xR)$ for all functions $f : \{0,1\}^{|x|+|R|} \longrightarrow \{0,1\}$ (i.e. it is the long code for $xR$). However, the analysis of this approach requires one to redo the analysis of [22]. The randomization procedure used in the current version of our paper (based on appending dummy clauses to the 3CNF formula) allows us to use the analysis in [22] as a black box.

### Acknowledgments

We would like to thank Sanjeev Arora for stimulating discussions and Johan Håstad for providing us with preliminary writeups of his work.

## References

[1] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman. Derandomized graph products. *Computational Complexity* 5 (1995), 60–75.

[2] N. Alon, J. Spencer. The Probabilistic Method. John Wiley & Sons, Inc., New York, 1992.

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 1992.

[4] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 2–13, 1992.

[5] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and nonapproximability – towards tight results. *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 422–431, 1995.

[6] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient multi-prover interactive proofs with applications to approximation problems. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 113–131, 1993.

[7] M. Bellare and M. Sudan. Improved non-approximability results. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 184–193, 1994.

[8] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. *Information and Computation*, 96(1):77–94, 1992.

[9] R. Boppana, M. Halldorsson. "Approximating maximum independent sets by excluding subgraphs". *BIT 32 (1992), 180–196.*

[10] A. Blum. Algorithms for Approximate Graph Coloring, Ph.D. Thesis, Massachusetts Institute of Technology, 1991. (Also in MIT/LCS/TR-506, June 1991)

[11] C. Dwork, U. Feige, J. Kilian, M. Naor, S. Safra, "Low Communication, 2-Prover Zero-Knowledge Proofs for NP" *Advances in Cryptology: Crypto '92*, pages 217–229.

[12] U. Feige. Randomized graph products, chromatic numbers, and the Lovasz $\vartheta$-function. In *Proceedings of 27th Symposium on the Theory of Computing*, pp. 635-640, 1995.

[13] U. Feige, S.Goldwasser, L. Lovasz, S. Safra, M. Szegedy. "Interactive proofs and the hardness of approximating cliques". JACM, Vol. 43, No. 2, March 1996, 268–292.

[14] U. Feige and J. Kilian. Two prover protocols–low error at affordable rates. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 172–183, 1994.

[15] U. Feige and J. Kilian. "Zero knowledge and the chromatic number (preliminary version)." In *Proc.of 11th IEEE Conference on Computational Complexity, 1996, IEEE Computer Society Press, 278–287.*

[16] M. Furer. Improved hardness results for approximating the chromatic number. *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 414–421, 1995.

[17] M. R. Garey and D. S. Johnson. "Computers and Intractability: A guide to the theory of NP-completeness", W.M.Freeman and Company.

[18] M. R. Garey and D. S. Johnson. The complexity of near-optimal graph coloring. *Journal of the ACM*, 23:43–49, 1976.

[19] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but Their Validity or All Languages in NP have Zero-Knowledge Proofs. *Journal of the ACM*, Vol. 38, No. 3, pages 691–729, July 1991.

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

[21] J. Håstad. Testing of the long code and hardness for clique *Proc. 28th ACM Symp. on Theory of Computing*, pages 11–19, 1996.

[22] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Proc. 37th IEEE Symp. on Foundations of Computer Science*, pages 627–636, 1996.

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

[24] S. Khanna, N. Linial, and S. Safra. On the hardness of approximating the chromatic number. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, ISTCS*, pages 250–260. IEEE Computer Society Press, 1993.

[25] J. Kilian and M. Naor. On the Complexity of Statistical Reasoning. In *Proceedings, Israeli Symposium on Theory of Computing and Systems*, pages 209–217, 1995.

[26] N. Linial and U. Vazirani. Graph products and chromatic numbers. In *Proc. 30th IEEE Symp. on Foundations of Computer Science*, pp. 124–128, 1989.

[27] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[28] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[29] E. Petrank. The hardness of approximation: Gap location. *Computational Complexity*, Vol. 4, 1994. pages 133–157.

[30] R. Raz. A parallel repetition theorem. In *Proceedings of 27th Symposium on the Theory of Computing*, pp. 447-456, 1995.

[31] D. Zuckerman. Simulating BPP using a general weak random source. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 79–89, 1991.