Clique Is Hard to Approximate within $n^{1-o(1)}$

Lars Engebretsen and Jonas Holmerin

Royal Institute of Technology
Department of Numerical Analysis and Computing Science
SE-100 44 Stockholm, SWEDEN
Fax: +46 8 790 09 30
{enge,joho}@nada.kth.se

Abstract. It was previously known that Max Clique cannot be approximated in polynomial time within $n^{1-\epsilon}$, for any constant $\epsilon > 0$, unless **NP** = **ZPP**. In this paper, we extend the reductions used to prove this result and combine the extended reductions with a recent result of Samorodnitsky and Trevisan to show that clique cannot be approximated within $n^{1-O(1/\sqrt{\log\log n})}$ unless $\mathbf{NP} \subseteq \mathbf{ZPTIME}(2^{O(\log n (\log\log n)^{3/2})})$.

1 Introduction

The Max Clique problem, i.e., the problem of finding in a graph $G = (V, E)$ the largest possible subset C of the vertices in V such that every vertex in C has edges to all other vertices in C, is a well-known combinatorial optimization problem. The decision version of Max Clique was one of the problems proven to be **NP**-complete in Karp's original paper on **NP**-completeness [10], which means that we cannot hope to solve Max Clique efficiently, at least not if we want an exact solution. Thus, attention has turned to algorithms producing solutions which are at most some factor from the optimum value. It is trivial to approximate Max Clique in a graph with n vertices within n—just pick any vertex as the clique—and Boppana and Halldórsson [5] have shown that Max Clique can be approximated within $O(n/\log^2 n)$ in polynomial time. It is an astonishing, and unfortunate, result that it is hard to do substantially better than this. In fact, the Max Clique problem cannot be approximated within $n^{1-\epsilon}$, for any constant $\epsilon > 0$, unless **NP** = **ZPP**. The first to explore the possibility of proving strong lower bounds on the approximability of Max Clique were Feige et al. [8], who proved a connection between Max Clique and probabilistic proof systems. Their reduction was then improved independently by Bellare, Goldreich, and Sudan [3] and Zuckerman [12]. As the final link in the chain, Håstad [9] constructed a probabilistic proof system with the properties needed to get a lower bound of $n^{1-\epsilon}$.

Since the hardness result holds for any arbitrarily small constant ϵ, the next logical step to improve the lower bound is to show inapproximability results for non-constant ϵ. However, Håstad's proof of the existence of a probabilistic proof system with the needed properties is very long and complicated. This has, until now, hindered any advance in this direction, but recently, Samorodnitsky and

U. Montanari et al. (Eds.): ICALP 2000, LNCS 1853, pp. 2–12, 2000.

Trevisan [11] constructed another probabilistic proof system with the needed properties, but where the proof of correctness is much simpler. Armed with this new construction, new results are within reach.

In this paper, we show that it is indeed impossible to approximate Max Clique in polynomial time within $n^{1-\epsilon}$ where $\epsilon \in O(1/\sqrt{\log \log n})$, given that **NP** does not admit randomized algorithms with slightly super-polynomial expected running time. To do this we first ascertain that the reductions from probabilistic proof systems to Max Clique [8,3,12] work also in the case of a non-constant ϵ. This has the additional bonus of collecting in one place the various parts of the reduction, which were previously scattered in the literature. We also extend the previously published reductions to be able to use the construction of Samorodnitsky and Trevisan [11], which characterizes **NP** in terms of a probabilistic proof system with so called non-perfect completeness. To our knowledge, such reductions have not appeared explicitly in the literature before.

When we combine the new reductions with the probabilistic proof system of Samorodnitsky and Trevisan [11], we obtain the following concrete result regarding the approximability of Max Clique:

Theorem 1. *Unless* $\mathbf{NP} \subseteq \mathbf{ZPTIME}(2^{O(\log n (\log \log n)^{3/2})})$, *Max Clique on a graph with n vertices cannot be approximated within $n^{1-O(1/\sqrt{\log \log n})}$ in polynomial time.*

As a comparison, the best known polynomial time approximation algorithm [5], approximates Max Clique within $n^{1-O(\log \log n / \log n)}$. We omit several proofs from this extended abstract. They are contained in the full version of the paper, available from the authors' home page.[1]

2 Preliminaries

Definition 1. *Let P be an* **NP** *maximization problem. For an instance x of P let* $\mathrm{opt}(x)$ *be the optimal value. A solution y with weight $w(x, y)$, is* c-approximate *if it is feasible and $w(x, y) \geq \mathrm{opt}(x)/c$.*

Definition 2. *A* c-approximation algorithm *for an* **NP** *optimization problem P is a polynomial time algorithm that for any instance x and any input y outputs a c-approximate solution.*

We use the wording *to approximate within c* as a synonym for *to compute a c-approximate solution.*

Definition 3. *Max Clique is the following maximization problem: Given a graph $G = (V, E)$ find the largest possible $C \subseteq V$ such that if v_1 and v_2 are vertices in C, then (v_1, v_2) is an edge in E.*

[1] http://www.nada.kth.se/~enge/

Definition 4. *G-gap E3-Sat-5 is the following decision problem: We are given a Boolean formula ϕ in conjunctive normal form, where each clause contains exactly three literals and each literal occurs exactly five times. We know that either ϕ is satisfiable or at most a fraction G of the clauses in ϕ are satisfiable and are supposed to decide if the formula is satisfiable.*

We know from [7] that G-gap E3-Sat-5 is **NP**-hard.

2.1 Previous Hardness Results

A language L is in the class **NP** if there exists a polynomial time Turing machine M, with the following properties:

- For instances $x \in L$, there exists a proof π, of size polynomial in $|x|$, such that M accepts (x, π).
- For instances $x \notin L$, M does not accept (x, π) for any proof π of size polynomial in $|x|$.

Arora and Safra [2] used a generalization of the above definition of **NP** to define the class $\mathbf{PCP}[r, q]$, consisting of a probabilistically checkable proof system (PCP) where the verifier has oracle access to the membership proof, is allowed to use $r(n)$ random bits and query $q(n)$ bits from the oracle.

Definition 5. *A probabilistic polynomial time Turing machine V with oracle access to π is an* (r, q)-restricted verifier *if it, for every oracle π and every input of size n, uses at most $r(n)$ random bits and queries at most $q(n)$ bits from the oracle. We denote by V^π the verifier V with the oracle π fixed.*

Definition 6. *A language L belongs to the class $\mathbf{PCP}[r, q]$ if there exists a (r, q)-restricted verifier V with the following properties:*

- *For instances $x \in L$, $\Pr_\rho[V^\pi \text{ accepts } (x, \rho)] = 1$ for some oracle π.*
- *For instances $x \notin L$, $\Pr_\rho[V^\pi \text{ accepts } (x, \rho)] \leq 1/2$ for all oracles π.*

Above, ρ is the random string of length r.

The connection between the approximability of Max Clique and PCPs was first explored by Feige et al. [8], who showed that

$$\mathbf{NP} \subseteq \mathbf{PCP}[O(\log n \log\log n), O(\log n \log\log n)] \tag{1}$$

and used this characterization of **NP** and a reduction to show that Max Clique cannot be approximated within any constant unless

$$\mathbf{NP} \subseteq \mathbf{DTIME}(n^{O(\log\log n)}). \tag{2}$$

The assumption on **NP** needed to prove hardness result on the approximability of Max Clique is closely related to the connection between the classes **NP** and $\mathbf{PCP}[r, q]$ for various values of r and q. This connection was the subject of intensive investigations leading to the following result of Arora et al. [1]:

Theorem 2. $\mathbf{NP} = \mathbf{PCP}[O(\log n), O(1)]$.

A consequence of this result is that the abovementioned assumptions in the proof of Feige et al. [8] could be weakened to $\mathbf{P} = \mathbf{NP}$.

A technical tool in the proof of Feige et al. [8] is the construction of a graph $G_{V,x}$, corresponding to a verifier in some proof system and some input x.

Definition 7. *From a verifier V and some input x, we construct a graph $G_{V,x}$ as follows: Every vertex in $G_{V,x}$ corresponds to an accepting computation of the verifier. Two vertices in $G_{V,x}$ are connected if they correspond to* consistent computations*. Two computations Π_1 and Π_2 are consistent if, whenever some bit b is queried from the oracle, the answers are the same for both Π_1 and Π_2.*

In the original construction, the number of vertices in $G_{V,x}$ was bounded by $2^{r(n)+q(n)}$, where $r(n)$ is the number of random bits used by the verifier and $q(n)$ is the number of bits the verifier queries from the oracle. Feige et al. suggest in their paper that the bound on the number of vertices in $G_{V,x}$ could be improved, and it was later recognized that the number of vertices can be bounded by $2^{r(n)+f(n)}$, where $f(n)$ is the *free bit complexity*.

Definition 8. *A verifier has* free bit complexity f *if the number of accepting computations is at most 2^f for any outcome of the random bits tossed by the verifier.*

Definition 9. *A language L belongs to the class $\mathbf{FPCP}_{c,s}[r, f]$ if there exists verifier V with free bit complexity f that given an input x and oracle access to π tosses r independent random bits ρ and has the following properties:*

- *for instances $x \in L$, $\Pr_\rho[V^\pi$ accepts $(x, \rho)] \geq c$ for some oracle π.*
- *for instances $x \notin L$, $\Pr_\rho[V^\pi$ accepts $(x, \rho)] \leq s$ for all oracles π.*

We say that V has completeness c *and* soundness s.

To understand the intuition behind the free bit complexity of a proof system, it is perhaps best to study the behavior of a typical verifier in a typical proof system. Such a verifier first reads a number of bits, the free bits, from the oracle. From the information obtained from those bits and the random string, the verifier determines a number of bits, the non-free bits, that it should read next from the oracle and the values these bits should have in order for the verifier to accept. Finally, the verifier reads these bits from the oracle and check if they have the expected values.

Theorem 3. *Suppose that $L \in \mathbf{FPCP}_{c,s}[r, f]$. Let x be some instance of L, and construct the graph $G_{V,x}$ as in Definition 7. Then, there is a clique of size at least $c2^r$ in $G_{V,x}$ if $x \in L$, and there is no clique of size greater than $s2^r$ if $x \notin L$.*

Proof. First suppose that $x \in L$. Then there exists an oracle such that a fraction c of all random strings make the verifier accept. The computations corresponding to the same oracle are always consistent, and thus there exists a clique of size at least $c2^r$ in $G_{V,x}$.

Now suppose that $x \notin L$ and that there is a clique of size greater than $s2^r$ in $G_{V,x}$. Since vertices corresponding to the same random string can never represent consistent computations, the vertices in the clique all correspond to different random strings. Thus, we can use the vertices to form an oracle making the verifier accept with probability larger than s. This contradicts the assumption that the PCP has soundness s.

Corollary 1. *Suppose that* $\mathbf{NP} \subseteq \mathbf{FPCP}_{c,s}[O(\log n), f]$ *for some constants* c, s, *and* f. *Then it is impossible to approximate Max Clique within* c/s *in polynomial time unless* $\mathbf{P} = \mathbf{NP}$.

Proof. Let L be some **NP**-complete language and x be some instance of L. Let B be some polynomial time algorithm approximating Max Clique within c/s.

The following algorithm decides L: Construct the graph $G_{V,x}$ corresponding to the instance x. Now run B on $G_{V,x}$. If B determines that $G_{V,x}$ has a clique containing more than $s2^r$ vertices, where $r \in O(\log(n))$ is the number of random bits used by the verifier, accept x, otherwise reject.

Since the number of random bits used by the verifier is logarithmic and the number of free bits is a constant, the graph $G_{V,x}$ has polynomial size. Since B is a polynomial time algorithm, the above algorithm also runs in polynomial time.

It is possible to improve on the above result by *gap amplification*. The simplest form of gap amplification is to simply run a constant number of independent runs of the verifier. If any of the rounds causes the verifier to reject, we reject, otherwise we accept. This shows that, for any constant k,

$$\mathbf{FPCP}_{c,s}[r, f] \subseteq \mathbf{FPCP}_{c^k,s^k}[kr, kf], \tag{3}$$

for any functions c, s, r, and f, which strengthens Corollary 1 to

Corollary 2. *Suppose that* $\mathbf{NP} \subseteq \mathbf{FPCP}_{c,s}[O(\log n), f]$ *for some constants* c, s, *and* f. *Then it is impossible to approximate Max Clique within any constant in polynomial time unless* $\mathbf{P} = \mathbf{NP}$.

The above procedure can improve the inapproximability result from a specific constant c/s to any constant, but to improve the inapproximability result from n^α to $n^{\alpha'}$ for some constants α and α', we have to use a more sophisticated form of gap amplification. Also, the concept of free bit complexity needs to be refined. To see why the above procedure fails in this case, suppose that we have some proof system which gives a graph $G_{V,x}$ with $n = 2^{r+f}$ vertices such that we can deduce that it is impossible to approximate Max Clique within n^α in polynomial time. Put another way, this particular proof system has $c/s = n^\alpha$. Now we try to apply the above gap amplification technique. Then we get a new graph $G_{V',x}$ with $2^{kr+kf} = n^k$ vertices and a new inapproximability factor $c^k/s^k = n^{k\alpha}$. Thus, we have failed to improve the lower bound. Obviously, it is not only the free bit complexity of a proof system that is important when it comes to proving lower bounds for Max Clique, but also the *gap*, the quotient of the soundness

and the completeness. We see above that an exponential increase in the gap does not give us anything if the free bit complexity and the number of random bits increase linearly. Bellare and Sudan [4] recognized that the interesting parameter is $f/\log s^{-1}$ in the case of perfect completeness. This parameter was later named the *amortized free bit complexity* and denoted by $\bar{f}$. Note that the above gap amplification does not change $\bar{f}$. Two methods which do improve the lower bound in the case above by keeping down the number of random bits needed to amplify the gap have appeared in the literature [3,12], and both prove the same result: If every language in **NP** can be decided by a proof system with logarithmic randomness, perfect completeness, and amortized free bit complexity $\bar{f}$, then Max Clique cannot be approximated within $n^{1/(1+\bar{f})-\epsilon}$ in polynomial time, unless **NP** = **ZPP**. The constructions are valid for any constant $\bar{f}$ and some arbitrarily small constant $\epsilon > 0$, and they use the same principle as the above gap amplification: They perform consecutive, although not independent, runs of the verifier and accept if all runs accept.

2.2 A New Amortized Free Bit Complexity

For the case of non-perfect completeness, Bellare et al. [3] define the amortized free bit complexity as $f/\log(c/s)$. In this paper, we propose that this definition should be modified.

Definition 10. *The* amortized free bit complexity *for a PCP with free bit complexity f, completeness c and soundness s is*

$$\bar{f} = \frac{f + \log c^{-1}}{\log(c/s)}. \tag{4}$$

Note that both this definition and the previous one reduce to $f/\log s^{-1}$ in the case of perfect completeness, i.e., when $c = 1$. Note also that the above gap amplification does not change the amortized free bit complexity, neither with the original definition nor with our proposed modification of the definition. However, our proposed definition is robust also with respect to the following: Suppose that we modify the verifier in such a way that it guesses the value of the first free bit. This lowers the free bit complexity by one, and halves the completeness and the soundness of the test. With our proposed definition, the amortized free bit complexity does not change, while it decreases with the definition of Bellare et al. [3]. In the case of perfect completeness, the lower bound on the approximability increases as the amortized free bit complexity decreases. This makes it dubious to have a definition in the general case that allows the free bit complexity to be lowered by a process as the above. Using our proposed definition of the free bit complexity, we first establish that the construction of Zuckerman [12] works also in the case of non-constant parameters:

Theorem 4. *If* $\mathbf{NP} \subseteq \mathbf{FPCP}_{1,s}[r, f]$*, then, for any* $r \in \Omega(\log n)$ *and any* $R > r$*, it is impossible to approximate Max Clique in a graph with N vertices within* $N^{1/(1+\bar{f})-r/R}$ *in polynomial time unless*

$$\mathbf{NP} \subseteq \mathbf{coRTIME}\big(2^{\Theta(R+\bar{f}+R\bar{f})}\big). \tag{5}$$

In the case where $\bar{f}$ is some constant and $r \in \mathrm{O}(\log n)$, this reduces to the well known theorem that Max Clique cannot be approximated within $n^{1/(1+\bar{f})-\epsilon}$, for any constant $\epsilon > 0$, unless $\mathbf{NP} = \mathbf{ZPP}$. To see this, just choose $R = r/\epsilon$ above. We also investigate the case of non-perfect completeness. By using the same approach as above—performing consecutive, although not independent, runs of the verifier and accepting if all runs accept—we obtain the following theorem, which is implicit in the works of Bellare et al. [3] and Zuckerman [12]:

Theorem 5. *If* $\mathbf{NP} \subseteq \mathbf{FPCP}_{c,s}[r, f]$*, then, for any* $r \in \Omega(\log n)$*, and any* $R > r$ *such that* $c^D 2^R/2 > 2^r$*, where* $D = (R+2)f/\log s^{-1}$*, Max Clique in a graph with* N *vertices cannot be approximated within* $N^{1/(1+\bar{f})-(r+3)/(R+2)}$ *in polynomial time unless*

$$\mathbf{NP} \subseteq \mathbf{BPTIME}\big(2^{\Theta(R+\bar{f}+R\bar{f})}\big) \tag{6}$$

Note that we in our applications choose R such that the term $(r+1)/R$ in the above theorem is small.

When amplifying the gap of a PCP with non-perfect completeness, it seems more natural to use an accept condition different from the above: Instead of accepting when all runs of the verifier accept, accept when some fraction ν of the runs accept. We investigate the consequences of this new condition and show that using that condition we can construct a reduction without two-sided error also in the case of non-perfect completeness. The parameters of interest turns out to be

$$F_\nu = \frac{f + (1-\nu)\log(q-f+1) + 1}{-\mathcal{H}(\nu, s)}. \tag{7}$$

where q is the number of query bits in the verifier, ν is a parameter which is arbitrarily close to c, and

$$\mathcal{H}(\nu, s) = -\nu \log \frac{\nu}{s} - (1-\nu)\log\frac{1-\nu}{1-s}. \tag{8}$$

We can then prove the following theorem:

Theorem 6. *Suppose every language in* $\mathbf{NP}$ *can be decided by a PCP with completeness* c*, soundness* s*, query complexity* q*, and free bit complexity* f*. Let* μ *and* ν *be any constants such that* $\mu > 0$ *and* $s < \nu < c$*. Let* $h = ((1+\mu)c - \mu - \nu)/(1-\nu)$*. Then, for any* $R > r$*, it is impossible to approximate Max Clique in a graph with* $N = 2^{R+(R+2)F_\nu}$ *vertices within*

$$N^{1/(1+F_\nu)-r/R-(\log h^{-1})/R} \tag{9}$$

by algorithms with expected polynomial running time unless

$$\mathbf{NP} \subseteq \mathbf{ZPTIME}(2^{\Theta(R+F_\nu+RF_\nu)}). \tag{10}$$

If F_ν is a constant and $r(n) \in \mathrm{O}(\log n)$, the above theorem says that Max Clique is hard to approximate within $N^{1/(1+F_\nu)-\epsilon-o(1)}$, for ν arbitrarily close to c if we choose $\nu = (1+\mu)c - 2\mu$, μ small enough and $R = r/\epsilon$ and in the above theorem.

This might seem worse than in the case with two-sided error, where the interesting parameter was $\bar{f} = (f + \log c^{-1})/\log(c/s)$ instead of F_ν. When $c = 1/2$, s is small and ν is close to c, this is indeed the case—then F_ν is about $2\bar{f}$. However, when c is close to 1, s is small and the PCP has reasonable low query complexity, we expect $\bar{f}$ and F_ν to be close.

3 Hardness of Approximating Max Clique

In their recent paper, Samorodnitsky and Trevisan [11] give a new PCP for **NP** with optimal amortized query complexity, $1 + \epsilon$ for any constant $\epsilon > 0$.

Theorem 7 (Implicit in [11]). *For any positive integer k and any constants $\epsilon > 0$ and $\delta > 0$,*

$$\mathbf{NP} \subseteq \mathbf{FPCP}_{(1-\epsilon)^{k^2}, 2^{-k^2}+\delta}[O(\log n), 2k]. \tag{11}$$

This result implies that the test has free bit complexity ϵ, for any constant $\epsilon > 0$. Since the construction is much simpler than the construction of Håstad [9], with reasonable effort it is possible to work through the construction with a non-constant ϵ. This yields the following theorem (we omit the proof):

Theorem 8. *For any increasing function $k(n)$ and any decreasing functions $\epsilon(n) > 0$ and $\delta(n) > 0$, G-gap E3-Sat-5 has a PCP which has query complexity $q = k^2 + 2k$, free bit complexity $f = 2k$, completeness $c \geq (1-\epsilon)^{k^2}$, soundness $s \leq 2^{-k^2} + \delta$, and uses*

$$r \leq C'_G(\log n + 3k)\log\big((2\epsilon^{-1})\delta^{-2}\big) + \big(2k + k^2\log\epsilon^{-1}\big)\big((2\epsilon^{-1})\delta^{-2}\big)^{C''_G} \tag{12}$$

random bits, for some constants C'_G and C''_G.

When we combine the above theorem with Theorem 6, we obtain the proof of Theorem 1

Proof (of Theorem 1). The proof is just a matter of finding suitable choices for the parameters involved: q, f, s, c, k, and R. By putting $\epsilon = k^{-2}$ and $\delta = 2^{-k^2}$ in Theorem 8, we get that $c \geq e^{-1}$, $s \leq 2^{1-k^2}$, and

$$r \leq C'_G(\log n + 3k)\log(2k^2 2^{2k^2}) + (2k + 2k^2\log k)(2k^2 2^{2k^2})^{C''_G}. \tag{13}$$

If we let

$$k(n) = c_0\sqrt{\log\log n}, \tag{14}$$

where $c_0^2 < 1/2C''_G$, we get

$$k^2 < (\log\log n)/2C''_G, \tag{15}$$

$$2^{2k^2} < (\log n)^{1/C''_G}, \tag{16}$$

which implies that r is dominated by the first term. If we substitute our choice of k in this term, we obtain $r = O(\log n \log\log n)$. Now we set $\nu = (1+\mu)c - 2\mu$, where μ is some arbitrary constant such that $s < \nu < c$. Then

$$-\mathcal{H}(\nu, s) = \nu k^2 + O(1), \tag{17}$$

$$F_\nu = \frac{2k + O(\log k)}{\nu k^2 + O(1)} = \frac{2}{\nu k} + o(1/k). \tag{18}$$

If we set $R = r/F_\nu$ in Theorem 6 we get that

$$h = \frac{(1+\mu)c - \mu - \nu}{1-\nu} > \mu \tag{19}$$

and that it is impossible to approximate Max Clique in a graph with $N = 2^{r/F_\nu + r + 2F_\nu}$ vertices within

$$N^{1/(1+F_\nu) - r/R - (\log h^{-1})/R} \geq N^{1-2F_\nu - o(F_\nu)} \tag{20}$$

in polynomial time, unless

$$\mathbf{NP} \subseteq \mathbf{ZPTIME}(2^{\Theta(r/F + F + r)}) = \mathbf{ZPTIME}(2^{\Theta(\log n (\log\log n)^{3/2})}) \tag{21}$$

Now, we want to express this ratio in terms of N. If we insert Eq. 18 in Eq. 20, we get that

$$N^{1-2F_\nu - o(F_\nu)} = N^{1 - 4/\nu k + o(1/k)}, \tag{22}$$

and if we insert Eq. 14 into this we get that

$$N^{1-2F_\nu - o(F_\nu)} = N^{1 - 4/\nu c_0 \sqrt{\log\log n} + o(1/\sqrt{\log\log n})}. \tag{23}$$

Since $\sqrt{\log\log N} = \sqrt{\log\log n}(1 + o(1))$,

$$o\Big(\sqrt{\log\log N}\Big) = o\Big(\sqrt{\log\log n}\Big) \tag{24}$$

and

$$\begin{aligned} \frac{1}{\sqrt{\log\log N}} &= \frac{1}{\sqrt{\log\log n}} \cdot \frac{1}{1 + o(1)} \\ &= \frac{1}{\sqrt{\log\log n}} \big(1 - o(1)\big) \\ &= \frac{1}{\sqrt{\log\log n}} - o\bigg(\frac{1}{\sqrt{\log\log n}}\bigg). \end{aligned} \tag{25}$$

Thus,

$$N^{1-2F_\nu - o(F_\nu)} = N^{1 - c_1/\sqrt{\log\log N} - o(1/\sqrt{\log\log N})}, \tag{26}$$

where $c_1 = 4/\nu c_0$.

Note that we do not gain anything if we use Theorem 5 instead of Theorem 6. In the former case we get

$$\bar{f} = \frac{2k + O(1)}{k^2 + O(1)} = \frac{2}{k} + o(1/k). \tag{27}$$

and to get a reasonable value for r, we need to set $k^2 = O(\log \log n)$. Thus we get the same hardness result, except for the constant, but with a stronger assumption—$\mathbf{NP} \not\subseteq \mathbf{BPTIME}(\cdot)$ instead of $\mathbf{NP} \not\subseteq \mathbf{ZPTIME}(\cdot)$—if we use Theorem 5.

4 Future Work

An obvious way to improve this result would be to weaken the assumptions on **NP** we used in our hardness result. Best of all, of course, would be to construct deterministic reductions, since this would allow us to replace the probabilistic complexity classes with deterministic ones in all our assumptions on **NP**. Until this is done, an interesting open question is to determine the best definition of the amortized free bit complexity. We have proposed that the definition should be

$$\bar{f} = \frac{f + \log c^{-1}}{\log(c/s)}. \tag{28}$$

This definition works well in the sense that a PCP with one-sided error gives a hardness result for Max Clique under the assumption that **NP**-complete problems cannot be decided with one-sided error in probabilistic polynomial time, and similarly a PCP with two-sided error gives a hardness result for Max Clique under the assumption that **NP**-complete problems cannot be decided with two-sided error in probabilistic polynomial time.

However, we have seen in Theorem 6 that if one wants to use a PCP with two-sided error to obtain hardness results under the assumption that **NP**-complete problems cannot be decided with one-sided error in probabilistic polynomial time, the interesting parameter is (close to) F_c, defined in Eq. 7. To establish whether it is possible to improve this to our proposed definition of $\bar{f}$, or if F_c is the best possible in this case is an interesting open question.

Trying to obtain an upper bound is also interesting, especially since it is currently unknown how well the Lovász ϑ-function approximates Max Clique. Feige [6] has shown that it cannot approximate Max Clique within $n/2^{c\sqrt{\log n}}$, but, in light of Håstad's results [9] and the results of this paper, this does not compromise the Lovász ϑ-function. It may very well be that it beats the combinatorial algorithm of Boppana and Halldórsson [5].

References

1. S. Arora, C. Lund, R. Motwani, M. Sudhan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998.

2. S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, Jan. 1998.
3. M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs and non-approximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, June 1998.
4. M. Bellare and M. Sudan. Improved non-approximability results. In *Proc. Twenty-sixth Ann. ACM Symp. on Theory of Comp.*, pages 184–193, Montréal, Québec, May 1994. ACM Press.
5. R. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *Bit*, 32(2):180–196, June 1992.
6. U. Feige. Randomized graph products, chromatic numbers, and the Lovasz ϑ-function. In *Proc. Twenty-seventh Ann. ACM Symp. on Theory of Comp.*, pages 635–640, Las Vegas, Nevada, May 1995. ACM Press.
7. U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
8. U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, Mar. 1996.
9. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *Proc. 37th Ann. IEEE Symp. on Foundations of Comput. Sci.*, pages 627–636, Burlington, Vermont, Oct. 1996. IEEE Computer Society.
10. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.
11. A. Samorodnitsky and L. Trevisan. Notes on a PCP characterization of NP with optimal amortized query complexity. Manuscript, May 1999.
12. D. Zuckerman. On unapproximable versions of NP-complete problems. *SIAM J. Comput.*, 25(6):1293–1304, Dec. 1996.