Published in ACM TRANSACTIONS ON ALGORITHMS 2:3, 473-489 (2006)

The NP-Completeness Column: The Many Limits on Approximation

DAVID S. JOHNSON
AT&T Labs – Research, Florham Park, New Jersey

Abstract. This is the 25th edition of a column that covers new developments in the theory of NP-completeness. The presentation is modeled on that which M. R. Garey and I used in our book "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman & Co., New York, 1979, hereinafter referred to as "[G&J]." Previous columns, the first 23 of which appeared in *J. Algorithms*, will be referred to by a combination of their sequence number and year of appearance, e.g., "[Col 1, 1981]." Full bibliographic details on the previous columns, as well as downloadable unofficial versions of them, can be found at `http://www.reseach.att.com/~dsj/columns/`. This edition of the column discusses the wide range of lower bounds on approximation guarantees for NP-hard optimization problems, both in their functional forms and in the hypotheses on which they depend.

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices**]: Complexity Classes—*reducibility and completeness*; *relations among complexity classes*; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms: Algorithms, Theory

Additional Key Words and Phrases: approximation algorithms, lower bounds, set cover, clique, label cover, Unique Games Conjecture, probabilistically checkable proofs

1. INTRODUCTION

When [G&J] was written, it seemed sufficient to divide NP-hard optimization problems into four classes as far as their approximability was concerned:

(1) Problems that have a fully polynomial-time approximation scheme (FPTAS): an algorithm that, given an instance I and an $\epsilon > 0$, finds a feasible solution whose value is within a factor of $1 + \epsilon$ of optimal in time polynomial in both $1/\epsilon$ and the size of I.

(2) Problems that do not have an FPTAS unless P = NP, but have a polynomial-time approximation scheme (PTAS): an algorithm that, given an instance I and an $\epsilon > 0$, finds a feasible solution whose value is within a factor of $1 + \epsilon$ of optimal in time that for fixed ϵ is polynomial in the size of I.

Author's address: Room C239, AT&T Labs, Inc. - Research, 180 Park Avenue, Florham Park, NJ 07932, e-mail: dsj@research.att.com.

(3) Problems that do not have a PTAS unless P = NP, but have a polynomial-time approximation algorithm that finds a solution whose value is within a constant factor of optimal.

(4) Problems for which no polynomial-time approximation algorithm guarantees a solution whose value is within a constant factor of optimal unless P = NP.

It seemed reasonable to think that if an algorithm could produce solutions that were an arbitrarily large factor worse than optimum, that was sufficient evidence that the algorithm was a poor performer, at least from a worst-case point of view. Even back in 1978 when [G&J] was being written, however, it was known that finer distinctions were possible. Consider the SET COVER problem: given a collection $\mathcal{C}$ of subsets C_i of a base set U of n elements, find a minimum sized subcollection $\mathcal{C}'$ whose union equals U. For this problem the greedy algorithm that repeatedly picks the set containing the most as-yet-uncovered elements was studied by Johnson [1974] and Lovász [1975]. Although it can be off optimal by an arbitrarily large factor, for a particular instance that factor is at most $\ln n$. In contrast, no polynomial-time approximation algorithm then known for the CLIQUE problem found cliques that were guaranteed to be within a factor of $o(n)$ of optimal for n-vertex graphs. This is exponentially worse than the result for SET COVER, a difference that might well be worth caring about.

Unfortunately, at the time we did not have techniques for proving that such discrepancies were inherent to the problems, even assuming that P $\neq$ NP. Such tools arrived with the discovery of probabilistically checkable proofs and their relation to approximation in the early 1990s [Feige et al. 1996; Arora and Safra 1998; Arora et al. 1998] (see also [Col 23, 1992]). These tools made possible proofs that for various problems X and functions $f(n)$, no polynomial-time algorithm could have a worst-case ratio that was $o(f(n))$ assuming P $\neq$ NP or some other likely hypothesis. For instance, Lund and Yannakakis [1994] showed that no polynomial-time algorithm for SET COVER could guarantee a solution that was within a factor $(1/4)\log n$ of optimal assuming NP $\not\subseteq \cup_{k>0}\text{DTIME}(n^{\log^k n})$, where "DTIME" refers to deterministic running time. This latter class is often called "quasi-polynomial time" and we shall typically abbreviate it as $\text{DTIME}(n^{\text{polylog}\, n})$. Note that NP $\not\subseteq \text{DTIME}(n^{\text{polylog}\, n})$ implies P $\neq$ NP but not vice versa, so the former is a stronger (and hence possibly less likely) hypothesis. This result was subsequently tightened by Feige [1998], who showed that no guarantee better than $(1 - o(1))\ln n$ was possible assuming NP $\not\subseteq \text{DTIME}(n^{O(\log\log n)})$. Note that this lower bound agrees with the Greedy algorithm upper bound except possibly for lower order terms, and the hypothesis is weaker than that needed by Lund and Yannakakis [1994], although still stronger than P $\neq$ NP.

Given these lower bound results, the greedy algorithm for SET COVER goes from being a poor performer to an optimal one (at least in a sense). Results such as these have made the whole idea of logarithmic growth rates, and even polylogarithmic ones, respectable. They may be inherent to the problem we are studying, and even if not, proving such a guarantee typically requires insight into the problem, and such insights may eventually lead to algorithms with constant worst-case bounds.

Moreover, for those problems where worst-case ratio growth is unavoidable, polylogarithmic growth seems much less objectionable than the kind we saw for CLIQUE, where the poor algorithmic behavior observed back when [G&J] was written has now also been proved unavoidable. After a long sequence of improving lower bounds, we now know that unless P = NP no polynomial-time algorithm for CLIQUE can provide an $O(n^{1-\epsilon})$ guarantee for n-vertex graphs for any $\epsilon > 0$. This bound comes from a recent derandomization by Zuckerman [2006] of a result of Håstad [1999] and, although not as tight as the results we saw above for SET COVER, is tight in the sense that one can view the function $f(n) = n$ as the limit of the functions $f(n) = n^{1-\epsilon}$ as $\epsilon \to 0$, and a guaranteed ratio of n can be obtained simply by taking a single vertex as one's output clique.

One can make finer distinctions, however, if one is willing to make the stronger assumption that NP $\not\subseteq$ BPTIME($n^{\text{polylog}\, n}$), the class of problems solvable by randomized algorithms that run in quasi-polynomial time and for all instances give the correct answer with probability at least 2/3. Under this assumption, CLIQUE cannot in polynomial time be approximated to within a factor better than $n/2^{(\log n)^{3/4+\epsilon}}$ for any $\epsilon > 0$ [Khot and Ponnuswami 2006]. There is still a gap, however, between these lower bounds and the best we currently can achieve in polynomial time. The currently best polynomial-time approximation algorithm for CLIQUE is only guaranteed to find a clique within a factor of $n(\log \log n)^2/(\log n)^3$ of optimal [Feige 2004]. This is not a very strong guarantee, but the algorithm is a distinct improvement over the straightforward greedy heuristics studied in Johnson [1974], all of which could be off by $\Omega(n)$ factors.

SET COVER and CLIQUE are just two out of a wide variety of problems for which strong lower bounds on polynomial-time approximability have been proved in the flowering of research that followed the initial breakthroughs of the 1990s, and the abovementioned conjectures are just a few of those on which such bounds rely. In this column I hope to give a brief indication of the breadth and variety of results that have been proved for problems now known to be in Class 4.

I begin in Section 2 with a brief survey of the conjectures on which approximation complexity lower bounds have been based, including the recently-introduced *Unique Games Conjecture* of Khot [2002], and the relative strengths (and believability) of these hypotheses. I then move on to look at some of the range of results that have been proved for problems in Class 4 based on these hypotheses. We have already seen two quite different "approximation complexity functions" – $\Theta(\log n)$ for SET COVER and only a little less than $\Theta(n)$ for CLIQUE, where n is the size of the base set underlying the problem (the universe U in one case, the set of vertices V in the other). A natural question is what other functions f serve as approximation complexities for interesting problems. Indeed, several papers in the literature have listed as one of their accomplishments the fact that they have identified the "first" problem to have a particular approximation complexity function.

In Section 3 I discuss the issues involved in formalizing the notion of "approximation complexity function," and note that under plausible definitions, essentially *any* polynomial-time computable function is the approximation complexity for some optimization problem assuming P $\neq$ NP. This is easy to prove by including an arbitrarily large cost function in the problem, as has already been observed by Trevisan [2004]. However, a slightly restricted version of the result continues to hold even if one restricts attention to graph problems in which the objective function is the size of a subset of edges, and I present the relevant constructions. The problems constructed are still (unavoidably) artificial, however, and so these results leave open the question of which functions hold for more "natural" problems.

"Natural" is of course not something one can define formally, although there have been successful attempts at deriving formal restrictions that capture large numbers of (mostly) natural problems. The most successful of these is the work of Khanna et al. [2000], who show that if one restricts attention to "Boolean constraint satisfaction problems," every Class 4 optimization problem has essentially only five options for its approximation complexity function. The function is either n^ϵ for some $\epsilon > 0$, or it is the same as that for one of four specified problems. Unfortunately, this is too restrictive a class of problems to cover everything interesting. In any Boolean constraint satisfaction problem there is a fixed bound k on the number of variables that can be involved in any one constraint, this bound holding for all instances. Thus although CLIQUE can be modeled as such a problem, SET COVER cannot – and indeed has an approximation complexity function that may not fit in any of the five classes.

So suppose we simply restrict attention to problems that are interesting for some reason other than their approximation complexity. In this case the $\log n$ and (roughly) n for SET COVER and CLIQUE are joined by $\log^* n$, $\log^2 n$, and (perhaps) $\log\log n$, $\sqrt{\log n}$, $2^{(\log n)^{1-\epsilon}}$, and many others. Section 4 surveys the upper and lower bounds on approximation complexity that have been proved for problems in Class 4, by listing some of the problems for which these bounds have been proved. In selecting the problems to be covered, special emphasis has been placed on problems with tight (or relatively tight) bounds and on the four key problems from Khanna et al. [2000] mentioned above.

I do not have space to say much about the techniques used to prove the results I cite, many of which involve refinements on the concept of probabilistically checkable proof or sophisticated derandomization constructions. For readers interested in this side of the story, there are many readable surveys and tutorials [Arora and Lund 1997; Khot 2005; Trevisan 2004].

2. PLAUSIBLE (AND SOMEWHAT LESS PLAUSIBLE) HYPOTHESES

As seen in the above discussion and in the coverage of the SHORTEST VECTOR problem in the previous column, the set of hypotheses under which hardness-of-approximation results have been proved is quite large, often with stronger hypotheses yielding faster growing lower bounds (or larger ones in the case of constant guarantees). Figure 1 graphically depicts some of these hypotheses and the implication relationships between them. For definitions of the classes involved, see Johnson [1990] or the Complexity Zoo (`http://qwiki.caltech.edu/wiki/Complexity_Zoo`). Note that some papers list NP $\not\subseteq$ coRP as their hypothesis, but this is equivalent to NP $\not\subseteq$ ZPP: If NP $\subseteq$ ZPP, then NP $\subseteq$ coRP since ZPP = RP $\cap$ coRP. Conversely, if NP $\subseteq$ coRP then NP $\subseteq$ coNP which in turn implies NP = coNP = coRP = R and so NP = ZPP.

The gold standard among all these hypotheses is P $\neq$ NP, which is implied by all the others. Over the years progress has involved not only the proof of larger/faster growing lower bounds, but also the proof of the same bounds while

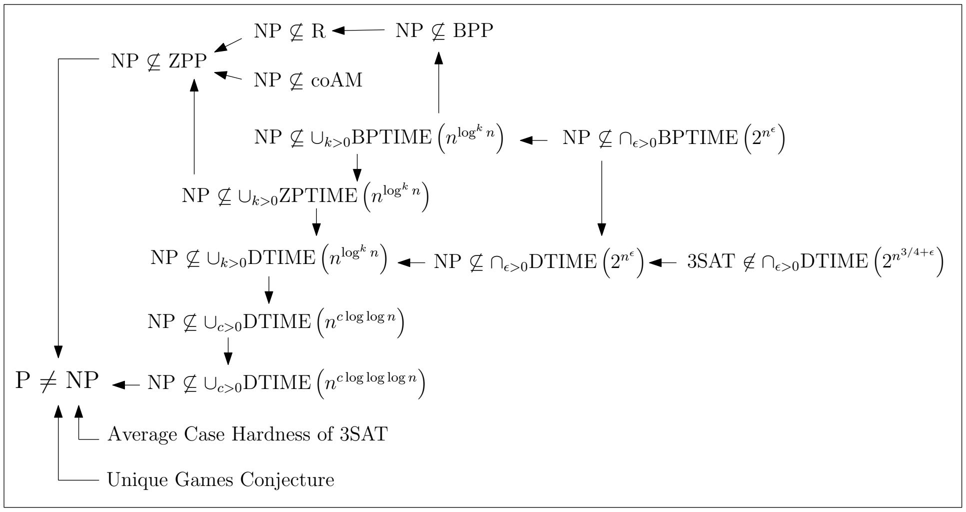

Fig. 1. Some of the hypotheses under which hardness-of-approximation results have been proved and the implication relationships between them.

assuming weaker hypotheses, with the ultimate goal being to get the result assuming only P $\neq$ NP. This can lead to rather lengthy histories – the one for CLIQUE is particularly extensive, starting with the original result of Feige et al. [1996] that it was hard to approximate within a factor of $2^{(\log n)^{1-\epsilon}}$ for any $\epsilon > 0$ unless NP $\not\subseteq$ DTIME($n^{\text{polylog}\, n}$), and then proceeding through multiple upgradings of the bound at the frequent cost of a stronger hypothesis, only to be replaced by a weaker hypothesis later on, and with the story not likely finished yet. See Khot and Ponnuswami [2006] for a detailed survey.

There is not room here to present applications for all the hypotheses included in Figure 1, although a few have been covered already and more will be covered in Section 4. However, some of the stronger hypotheses listed in the figure are worth additional comments. Conveniently, three of them, all apparently independent, have been applied to the same NP-complete problem from [G&J]: the BALANCED COMPLETE BIPARTITE SUBGRAPH problem (BCBS), in which we are given a bipartite graph $G = (V_1, V_2, E)$ and a positive integer K, and are asked if there are two sets $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ such that $|U_1| = |U_2| = K$ and such that $u \in U_1$ and $v \in U_2$ implies that $\{u, v\} \in E$.

Given the similarity of BCBS to CLIQUE, one would expect similar hardness-of-approximation results for it. Until recently, however, one could not even rule out a PTAS, and even now our hardness results are much weaker than those for CLIQUE and require some of our strongest hypotheses. Currently the fastest growing lower bound we know is due to Khot [2004], who shows that if one assumes that NP $\not\subseteq \cap_{\epsilon>0}$BPTIME$\left(2^{n^{\epsilon}}\right)$, then there exists a $\delta > 0$ such that the maximization version of this problem is hard to approximate within a factor of n^{δ}. In comparison, as we have already seen, CLIQUE cannot be approximated to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ unless P = NP.

Although the above lower bound for BCBS is the best we can currently get, we can at least obtain it in two ways. Feige [2002] shows that same conclusion holds if we assume a hypothesis about the following randomized model for 3SAT. For any fixed $\Delta > 0$, let Δ3SAT consist of those instances in which the number m of clauses equals $\lceil \Delta n \rceil$, where n is the number of variables. A random n-variable instance of Δ3SAT is one in which each clause is picked uniformly (and independently) from the set of all possible 3-literal clauses based on the literals $x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n$. It is well known that for large enough Δ, an instance of Δ3SAT is highly likely to be unsatisfiable. Feige's hypothesis is the following.

Average-Case Hardness of 3SAT: For any $\epsilon > 0$ there is a $\Delta > 0$ such that no polynomial-time algorithm exists that for a random n-variable instance of Δ3SAT outputs "unsatisfiable" with probability at least 1/2, but never says "unsatisfiable" when the instance has a truth assignment satisfying $(1 - \epsilon)m$ or more clauses.

The fact that the same hardness-of-approximation follows from two such different hypotheses perhaps lends it more credence, even though both hypotheses are relatively strong. The third hypothesis that implies that BCBS cannot be approximated to within any constant bound is also quite strong. It states that 3SAT $\notin$ DTIME $\left(2^{n^{3/4+\epsilon}}\right)$ for any $\epsilon > 0$. Under this hypothesis, Feige and Kogan [2004] show that there is some $\delta > 0$ such that BCBS cannot be approximated to within a factor of $2^{(\log n)^\delta}$. This is not as strong a lower bound as the one implied by the previous two hypotheses, although there is always the possibility that this hypothesis might hold while the previous two do not.

On the other hand, this last hypothesis seems if anything more suspect. There are many well-known NP-hard problems that are contained in the class $\cap_{\epsilon>0}$DTIME $\left(2^{n^{3/4+\epsilon}}\right)$. For example, consider the Traveling Salesman Problem on weighted planar graphs. Currently the best exact algorithm for this problem has running time $O\left(2^{6.987\sqrt{n}}n^{3/2}\right)$ [Dorn et al. 2005]. Thus although an $O\left(2^{n^{3/4}}\right)$ algorithm for 3SAT would be a major theoretical breakthrough, it wouldn't upset any of our fundamental assumptions about NP-complete problems.

To conclude this section I will discuss perhaps the "hottest" of the strong hypotheses in Figure 1, the *Unique Games Conjecture*. This has been the topic of a rapidly expanding body of research since it was first proposed by Khot [2002]. Much of the excitement about it comes from the fact that it implies tight bounds on the approximability of several famous problems in our Class 3 of problems that have no PTAS but can be approximated to within a constant factor. However, based on it one can also show that certain fundamental problems are in Class 4, and stronger versions of it have been shown to imply $\Omega(\sqrt{\log \log n})$ growth rates.

Although this hypothesis was originally formulated in terms of 2-prover interactive proofs, from which the "game" terminology arises, it is more easily stated in combinatorial terms as a special case of the well-known LABEL COVER problem introduced by Arora et al. [1997]. This problem comes in several variants, all given the same name in the literature. As defined by Khot [2002] it is the following.

LABEL COVER

INSTANCE: Bipartite graph $G = (V_1, V_2, E)$, positive integers N and M, a map $\pi_{u,v} : \{1, 2, \ldots, M\} \to \{1, 2, \ldots, N\}$ for each edge $\{u, v\} \in E$ where $u \in V_1$ and $v \in V_2$, and a bound $B \le |E|$.

QUESTION: Are there label assignments L_1 and L_2, where $L_1 : V_1 \to \{1, 2, \ldots, M\}$ and $L_2 : V_2 \to \{1, 2, \ldots, N\}$ such that at least B edges are *covered*, where an edge $\{u, v\} \in E$ is covered by a labeling if $\pi_{u,v}(L_1(u)) = L_2(v)$?

LABEL COVER and its variants have proved a key tool in proving many hardness-of-approximation results, e.g., see Arora and Lund [1997]. The maximization version of the problem (find an assignment that maximizes the number of edges covered) is known to be hard to approximate within any constant factor assuming P $\neq$ NP. More specifically, for any ϵ, $0 < \epsilon < 1/2$, there is a constant k_ϵ such that for instances with $M, N \le k_\epsilon$, it is NP-hard to distinguish between the case where there is a labeling that covers all edges in E and the case where no labeling covers more than $\epsilon|E|$ edges, a result that follows from the parallel repetition theorem of Raz [1998] as pointed out by Khot [2002]. The Unique Games Conjecture claims that something similar holds in a restricted special case.

Unique Games Conjecture: For any $\epsilon, \delta \in (0, 1/2)$ there is a constant $k = k_{\epsilon,\delta}$ such that, for LABEL COVER instances with $|M| = |N| = k$ in which the maps $\pi_{u,v}$ are all permutations, no polynomial-time algorithm can distinguish between the case where there is a labeling that covers at least $(1 - \delta)|E|$ edges and the case where no labeling covers more than $\epsilon|E|$ edges.

There does not seem to be any *a priori* reason to believe this conjecture is true. It is known that for any $\epsilon > 0$ there is a $\lambda < 1$ and a positive integer k such that for instances with $M = N = k$ it is NP-hard to distinguish between the case where there is a labeling that covers at least $\lambda|E|$ edges and the case where no labeling covers more than $\epsilon\lambda|E|$ edges [Feige and Reichman 2004]. This is not what the conjecture requires, however, since λ might be much smaller than 1. Moreover, the conjecture is false if there is a polynomial-time algorithm that distinguishes between instances for which labelings exist that can cover $(1 - \epsilon)|E|$ edges and those where no labeling can cover more than $(1/k)^{\epsilon/(2-\epsilon)}|E|$ edges [Khot et al. 2005], and a polynomial-time algorithm is already known that nearly does this. The algorithm is due to Charikar et al. [2006] and distinguishes between the first case and the case where a labeling exists that covers $\Omega\left(\frac{(1-\epsilon)^2}{\sqrt{\epsilon \log k}}\right)\left(\frac{\sqrt{\log k}}{k}\right)^{\epsilon/(2-\epsilon)}|E|$ edges.

Thus, many researchers are skeptical about the Unique Games Conjecture. If the conjecture *is* true, however, it will resolve several major open problems about the hardness of approximation. For example, consider the classic VERTEX COVER problem. At present the best polynomial-time approximation algorithm for this problem has worst-case performance ratio $2 - \Theta(1/\sqrt{\log n})$ [Karakostas 2005]. The current best lower bound under a standard hypothesis says that no polynomial-time algorithm can have a worst-case ratio better than 1.36067 unless P = NP [Dinur and Safra 2005]. This leaves a substantial gap. Assuming the Unique Games Conjecture, VERTEX COVER cannot be approximated to within any constant factor less than 2 [Khot and Regev 2003], and the gap essentially disappears.

As a second and perhaps more surprising example, consider the problem MAX CUT in which we are given a graph $G = (V, E)$ and asked for a subset $S \subset V$ that maximizes the number of edges in E with precisely one endpoint in S. The best polynomial-time approximation algorithms known for this problem are due to Goemans and Williamson [1995]. They use semidefinite programming and rounding to find solutions that are guaranteed in the limit to be within a factor of $\alpha_{GW} = \min_{0<\theta\leq\pi}((\pi(1-\cos(\theta))/(2\theta))$ of optimal, where $\alpha_{GW} \sim 1.138$. More precisely, for any $\epsilon > 0$ there is an algorithm that guarantees a solution of size at least OPT$/(\alpha+\epsilon)$. The best lower bound on the closeness-to-optimality ratio that has so far been proved based on a standard hypothesis is that of Håstad [2001], which says that assuming P $\neq$ NP no polynomial-time approximation algorithm is guaranteed to be within a factor of $17/16 - \epsilon$ of optimal for any $\epsilon > 0$. Note, however, that $17/16 = 1.0625$ and is significantly smaller than α_{GW}. Surprisingly, the Unique Games Conjecture allows us to close this gap completely. It implies that no polynomial-time algorithm is guaranteed to be within a factor of $\alpha_{GW} - \epsilon$ of optimal for any $\epsilon > 0$, as follows from a combination of results in Khot et al. [2005] and Mossel et al. [2005].

The fact that the Unique Games Conjecture implies tight bounds for two such disparate problems would seem to add to its credibility, and it has had many other applications [Khot 2002; Khot and Regev 2003; Khot et al. 2005; Khot and Vishnoi 2005; Chawla et al. 2006]. The strongest results that can be based on the conjecture are ones that say some given constant factor cannot be achieved in polynomial time. In some cases, however, it has been successfully used to show that this is true for *all* constant factors, thus proving that the relevant problem must be in our Class 4. In particular, this has been proved for the nonuniform version of the famous SPARSEST CUT problem [Khot and Vishnoi 2005]. In this version, we are given a graph $G = (V, E)$ with a positive integer weight $w(e)$ for each edge $e \in E$ and a nonnegative integer demand $d(u, v)$ for each pair of vertices. For a given subset $S \subset V$, let $cutweight(S)$ be the total weight of edges with precisely one endpoint in S and let $cutdemand(S)$ be the total demand for all vertex pairs with precisely one member in S. Our goal is to find an $S \subset V$ that minimizes $cutweight(S)/cutdemand(S)$. By itself, the Unique Games Conjecture cannot yield a lower bound on how the approximation complexity function grows with n. If a stronger version of the conjecture is true, however, Chawla et al. [2006] have shown that there is a $c > 0$ such that SPARSEST CUT cannot be approximated to within a ratio of $c\sqrt{\log\log n}$. In this version of the conjecture, δ and ϵ are no longer constants; δ must be $O(1/\sqrt{\log\log n})$, ϵ must be bounded by $1/(\log n)^{\Omega(\delta)}$, and $k_{\epsilon,\delta}$ must be $O(\log n)$. (The proceedings version of this paper proved a better lower bound $(\Omega(\log\log n))$, but to do so assumed a still-stronger version of the Unique Games Conjecture that has since been disproved [Charikar et al. 2006].) I shall have a bit more to say about SPARSEST CUT in Section 4.

3. APPROXIMATION COMPLEXITIES AND HOW TO REALIZE THEM

In this section I formalize the notion of the approximation complexity of an optimization problem. First I need to specify what we mean by an "optimization problem" and by "n." As in [G&J], problems are defined in terms of combinatorial objects like names, sets, lists, and functions, assuming a "reasonable encoding" into the subsets of Σ^* that are the actual subject of complexity theory. For our purposes, an *optimization problem* X has five components: (1) a polynomial-time recognizable set I_X of instances (2) a polynomial-time recognizable set of $(instance,\ feasible\ solution)$ pair, (3) a polynomial-time computable cost function that assigns a positive integer cost to each valid $(instance,\ feasible\ solution)$ pair, and (4) the specification of whether we want a solution of maximum or minimum cost. (5) a polynomial-time algorithm that, given an instance x, produces a feasible solution for x (so that every instance has a feasible solution and the problem has at least one polynomial-time approximation algorithm).

For the problems in which we are interested, each instance will include some key underlying set (vertices, elements, cities, etc.), and we will take the parameter n to be the cardinality of that set, as this is what is typically done in the literature. We assume that n is polynomially-related to the length of the encoding of the instance (the classical "size" of an instance), although note that it can be significantly smaller. For example, with graph instances $G = (V, E)$ most authors take $n = |V|$, whereas for dense graphs the actual length of the encoding can be $\Theta(n^2 \log n)$.

Suppose A is an approximation algorithm for problem X, with $A(x)$ denoting the solution value when A is applied to instance x (the expected solution value in the case of randomized algorithms), and $\mathrm{OPT}(x)$ denoting the optimum solution value for x. The *approximation ratio function* f_A for A is then

$$f_A(n) = \begin{cases} \min \left\{ \frac{A(x)}{\mathrm{OPT}(x)} : x \in I_X \text{ has parameter } n \right\}, & \text{for minimization problems} \\ \max \left\{ \frac{\mathrm{OPT}(x)}{A(x)} : x \in I_X \text{ has parameter } n \right\}, & \text{for maximization problems} \end{cases}$$

So far everything has been straightforward, but now things get a bit more complicated. Our lower bound results are often not a single lower bound but a sequence of stronger and stronger lower bounds, for instance $(1 - \epsilon)f(n)$, $f(n)^{1-\epsilon}$, or even $2^{f(n)^{1-\epsilon}}$ for all $\epsilon > 0$. Moreover, our upper bounds may similarly come from a sequence of better and better algorithms. So we need to have some notion of a limiting function or, more accurately, a limiting set of functions, given the imprecision of the most natural limiting processes.

There are many possible options, but for the purposes of this discussion, let us choose the following one. Say that an optimization problem X has *approximation complexity* $f(n)$ under hypothesis H if (A) for every function $g(n)$ such that $f(n) = o(g(n))$, there is a polynomial-time approximation algorithm for X that guarantees a solution within a factor $g(n)$ of optimal, and (B) for every function $h(n) = o(f(n))$, no polynomial-time approximation algorithm for X can guarantee a solution within a factor of $h(n)$ of optimal if H is true. Under this definition constant factors don't matter, and so if $f(n)$ is an approximation complexity function for X under H, then so is any function $g(n) = \Theta(f(n))$. Even so, the definition is perhaps more restrictive than we might want, since for instance it does not allow us to claim that lower bounds of the form $(\log n)^{2-\epsilon}$ for all $\epsilon > 0$ match a $\log^2 n$ upper bound. On the other hand, there are interesting functions in the gap between these bounds, such as $\log n / \log \log n$, and there will always be theorists who want to explore things in more detail, so perhaps this is not a defect.

Even this restrictive definition, however, does not significantly limit the set of approximation complexity functions that can occur, even assuming just P $\neq$ NP. For a given problem X, denote by $ACF(X)$ the family of approximation complexity functions for X under the hypothesis that P $\neq$ NP.

THEOREM 3.1. *Suppose $f : Z^+ \rightarrow Z^+$ is a polynomial-time computable function from the positive integers to themselves. Then there is an optimization problem X such that $f \in ACF(X)$ under the hypothesis that* P $\neq$ NP.

PROOF. Our proof uses a special case of the HAMILTONIAN CIRCUIT problem that was proved NP-complete by Papadimitriou and Steiglitz [1976]. In this special case we are given a graph $G = (V, E)$ where, as indexed, the vertices $v_1, v_2, \ldots, v_n$ constitute a Hamiltonian path, and are asked whether G contains a Hamiltonian circuit. Call this problem HCGP (where GP stands for "given a path").

Our optimization problem X will be the variant of the Traveling Salesman Problem in which all distances lie in the range $[1, nf(n) - n + 1]$ and $d(v_i, v_{i+1}) = 1$, $1 \leq i \leq n - 1$. Note that instances of this problem are polynomial-time recognizable since $f(n)$ is polynomial-time computable, so X is indeed an "optimization problem" according to our definition. We transform an instance of HCGP to X by letting V be the set of cities, and letting the distance between u and v be 1 if $\{u, v\}$ is an edge of the HCGP instance and otherwise be $nf(n) - n + 1$. Note that there is a trivial polynomial-time approximation algorithm for this problem that yields a guarantee of $f(n)$ – simply add the edge $\{v_n, v_1\}$ to the given Hamiltonian path. The length of the resulting tour will be at most $nf(n)$ and any tour must have length at least n. On the other hand if there were a polynomial-time algorithm that guaranteed any bound less than $f(n)$ (much less one that is $o(f(n))$ then it could be used to tell whether G has a Hamiltonian circuit, implying P = NP. □

One might well object to the above result because the objective function is a sum of numbers rather than the size of a set, as in SET COVER, CLIQUE, and many other popular graph problems. However, one can get something almost as strong even for "nonnumeric" graph problems in which the objective function is simply the size of some subset of the edges.

THEOREM 3.2. *Suppose $f : Z^+ \rightarrow Z^+$ is a polynomial-time computable function that is $O(n^{1-\epsilon})$ for some $\epsilon > 0$. Then there is a non-numeric graph problem X such $f \in ACF(X)$ under the hypothesis that* P $\neq$ NP.

PROOF. We once again exploit the HCGP problem, although in this case X will be a special case of the MINIMUM STEINER SIMPLE CYCLE problem (MSSC): Given a graph $H = (U, F)$ and a subset of the vertices $U' \subseteq U$, find a minimum length simple cycle that contains all the vertices in U'. (If one omits the requirement that the cycle be simple and allows arbitrary edge lengths, one gets what is called the "Steiner TSP problem" in the literature [Cornuéjols et al. 1985].) In our construction we let U' correspond to the vertex set V in the original HCGP instance with an edge between two members of U' if and only if they were adjacent in G. We then simulate the long edges of our previous construction by long paths of new vertices all having degree 2.

Unfortunately, this complicates the construction because adding new vertices increases n, which is why we need to have an added restriction on f. If the long paths have $nf(n)+n-1$ vertices, the number of vertices (the "n" for the constructed graph) can become as large as roughly $n^3 f(n)$. Things will work out, however, so long as we target this number at a value x such that $f(x)/x \leq 1/n^3$. Such an x exists and is easy to find: By assumption $f(x) = O(x^{1-\epsilon})$ and $f(x) \geq 1$ for all $x > 0$, so there exist constants $c > 0$ and $\epsilon > 0$ such that $f(x) \leq cx^{1-\epsilon}$ for all $x \geq 1$, and hence $f(x)/x \leq 1/n^3$ so long as $x \geq (cn^3)^{1/\epsilon}$. To bring the total number of vertices up to x, we add a long path attached to v_1 containing enough additional vertices to bring the overall total up to precisely x. Interested readers should readily be able to fill in the remaining details. □

The above proof technique does not seem to extend to functions that are not $O(n^{1-\epsilon})$ for some $\epsilon > 0$. If we are willing to settle for a weaker hypothesis, however, we can extend it up through the types of functions involved in the best current lower bounds for CLIQUE, an interesting coincidence.

THEOREM 3.3. *Suppose* $f : Z^+ \rightarrow Z^+$ *is a polynomial-time computable function that is* $O(n/2^{(\log n)^\epsilon})$ *for some* $\epsilon > 0$. *Then there is a non-numeric graph problem* X *such that* $f \in ACF(X)$ *under the hypothesis that* NP $\subseteq$ DTIME$(n^{\text{polylog}\, n})$.

PROOF. The proof is analogous to that of the previous theorem, except this time we allow the number of vertices to grow by a "quasi-polynomial" $n^{\text{polylog}\, n}$ factor roughly on the order of $2^{(3 \log n)^{1/\epsilon}}$). Thus if there were a polynomial time algorithm for X with a sufficiently good guarantee it could be used in a quasi-polynomial-time algorithm for solving HCGP and we would have NP $\subseteq$ DTIME$(n^{\text{polylog}\, n})$. □

4. APPROXIMATION COMPLEXITY FUNCTIONS SEEN IN THE WILD

Let us now look at some of the functions (or bounds on functions) that have been observed for interesting problems in Class 4. The results are presented roughly in order of increasing growth rate of the functions involved. For this discussion, we use the best lower bounds under any of the hypotheses listed in Figure 1. If an ϵ is included in the bound stated in a subsection heading, the claimed bound holds for all $\epsilon > 0$, If a δ is included, the claimed bound holds for *some* $\delta > 0$.

For some of the problems I cheat a little (as the literature often does itself), since the upper bounds are derived from randomized algorithms and the lower bounds are for deterministic algorithms. If past experience is a guide, most randomized algorithms can be derandomized with just a polynomial blow-up in the running time, so this is probably not a major issue. Moreover, any lower bound on deterministic algorithms that assumes NP $\not\subseteq$ DTIME$(f(n))$ can be turned into a lower bound on randomized algorithms if we assume NP is not contained in the corresponding randomized class BPTIME$(f(n))$. Nevertheless, when the upper bound is for a randomized algorithm for which no derandomization has been explicitly claimed, I point this out and also mention the current best deterministic bound. For space reasons, only the current best lower and upper bounds in terms of n will typically be mentioned. Readers interested in the history of the problems should see the references, which also often contain related and/or more general results.

4.1 $f(n) = \Theta(\log^* n)$ – ASYMMETRIC k-CENTER

In this problem an instance consists of a directed graph $G = (V, A)$, for each arc $(u, v) \in A$ an integer length $d(u, v) > 0$, and an integer k. A solution is a subset $U \subseteq V$ with $|U| = k$ (the *centers*). The goal is to minimize

$$\max_{v \notin U} \left(\min_{u \in U} d(v, u) \right).$$

The problem has a polynomial-time approximation algorithm with an $O(\log^* n)$ performance guarantee [Panigrahy and Vishwanathan 1998], and there is a constant $\delta > 0$ such that it cannot be approximated to within a factor of $\log^* n + \delta$ unless $\mathrm{NP} \subseteq \mathrm{DTIME}\left(n^{\mathrm{polylog}\, n}\right)$ [Chuzhoy et al. 2005].

4.2 $f(n) \in \left[\text{no constant}, O(\sqrt{\log n})\right]$ – MIN 2CNF DELETION

This is the first of the four canonical problems from Khanna et al. [2000]. In it we are given an instance of 2SAT, and are asked for a truth assignment that minimizes the number of unsatisfied clauses (as opposed to maximizing the number of satisfied clauses, which is MAX 2SAT). It cannot be approximated within any constant factor assuming the Unique Games Conjecture [Khot 2002]. If one is willing to assume the strengthened version of that (already strong) hypothesis described in Section 2, then there is a $c > 0$ such that the problem cannot be approximated to within a factor $c\sqrt{\log \log n}$ of optimal [Chawla et al. 2006]. The upper bound derives from a randomized algorithm described in Agarwal et al. [2005]. The best bound for a deterministic algorithm is currently $O(\log n)$ [Garg et al. 1996].

4.3 $f(n) \in \left[\text{no constant}, O(\sqrt{\log n})\right]$ – MIN UNCUT

This is the second of the four canonical problems from Khanna et al. [2000]. In it we are given a graph and asked for the minimum number of edges that need to be deleted to yield a bipartite subgraph. It is equivalent to the problem in which we are given a conjunction of logical clauses, each of which is the exclusive-or of two literals, and are asked for a truth assignment that minimizes the number of unsatisfied clauses. It cannot be approximated within any constant bound assuming the Unique Games Conjecture [Khot and Vishnoi 2005; Chawla et al. 2006]. If one assumes the above strengthened version of the conjecture, it too cannot be approximated to within a factor of $c\sqrt{\log \log n}$ of optimal for some constant $c > 0$ [Vishnoi 2006]. The algorithmic upper bounds are the same (and from the same references) as for MIN 2CNF DELETION.

4.4 $f(n) \in \left[\text{no constant}, O(\sqrt{\log n} \log \log n)\right]$ – SPARSEST CUT

This problem was defined in Section 2. For the lower bound one must assume the Unique Games Conjecture, and if one is willing to assume the strengthened version of that hypothesis, then there is again $c > 0$ such that the problem cannot be approximated to within a factor $c\sqrt{\log \log n}$ of optimal [Chawla et al. 2006]. Until recently the best upper bound was the $O(\log n)$ deterministic guarantee of the classic algorithm of Leighton and Rao [1999]. The improvement to $O(\sqrt{\log n} \log \log n)$ is due to Arora et al. [2005] and involves a randomized algorithm, building on the recent $O(\sqrt{\log n})$ algorithm of Arora et al. [2004] for the uniform case in which all demands are equal. (Note that neither of the above two lower bound results have yet been extended to the uniform case, which thus may not be in Class 4 even if the Unique Games Conjecture is true.)

4.5 $f(n) \in \left[\Omega(\log^{\delta} n), O(\sqrt{\log n})\right]$ – POINT SET WIDTH

In this problem we are given a set of n points in a d-dimensional space where d is part of the input. Our goal is to find the minimum, over all $(d-1)$-dimensional "planes", of the maximum Euclidean distance from any point in the set to the plane. Assuming NP $\not\subseteq$ DTIME $\left(n^{\mathrm{polylog}\, n}\right)$, there exists a $\delta > 0$ such that not even a quasi-polynomial-time algorithm can guarantee a ratio of $O(\log^{\delta} n)$ [Varadarajan et al. 2007]. A (derandomized) polynomial-time algorithm exists that is guaranteed to be within $O(\sqrt{\log n})$ of optimal [Nemirovsky et al. 1999; Varadarajan et al. 2007].

4.6 $f(n) \in \left[(\log\log n)^{1-\epsilon}, O(\log n/\log\log n)\right]$ – UNDIRECTED CONGESTION MINIMIZATION

In this problem we are given an undirected graph $G = (V, E)$ and a set S of source-sink vertex pairs $\{(s_i, t_i)\}$. A *routing* consists of a set of paths in G, one linking each source-sink pair in S. The *congestion* on an edge in a routing is the number of paths of the routing that use that edge, and the congestions of the routing is the maximum edge congestion. The goal of the problem is to find a routing with the minimum congestion. This problem can be approximated to within a factor of $O(\log n/\log\log n)$ using the randomized rounding techniques of Raghavan and Thompson [1987], and this approach can be derandomized. The problem cannot be approximated to within $(\log\log n)^{1-\epsilon}$ for any $\epsilon > 0$ assuming NP $\not\subseteq$ ZPTIME $\left(n^{\mathrm{polylog}\, n}\right)$, as shown by Andrews and Zhang [2005].

4.7 $f(n) \in \left[(\log n)^{1-\epsilon}, O(\log n/\log\log n)\right]$ – DIRECTED CONGESTION MINIMIZATION

This is the directed version of the previous problem. The upper bound is the same since essentially the same algorithm works. The lower bound, due to [Andrews and Zhang 2006], once again holds assuming NP $\not\subseteq$ ZPTIME $\left(n^{\mathrm{polylog}\, n}\right)$. Under the weaker hypothesis that NP $\not\subseteq \cup_{c>0}$DTIME $\left(n^{c \log\log\log n}\right)$, one can still show that the problem cannot be approximated to within $O(\log\log n)$ [Chuzhoy and Naor 2004].

4.8 $f(n) = \Theta(\log n)$ – SET COVER and DOMATIC NUMBER

We have already seen the results for the minimization problem SET COVER. These are mirrored by those for the maximization problem DOMATIC NUMBER. In this problem we are given a graph $G = (V, E)$ and wish to partition V into a maximum number of disjoint subsets such that each such subset V' is a dominating set for G, i.e. is such that every vertex in $V - V'$ is adjacent to some vertex in V'. The problem is shown to have a polynomial-time approximation algorithm with a $(1 + o(1)) \ln n$ guarantee in Feige et al. [2002], which also shows that it cannot be approximated to within a factor $(1-\epsilon)\ln n$ of optimum for any $\epsilon > 0$ if NP $\not\subseteq \cup_{c>0}$DTIME $\left(n^{c \log\log n}\right)$. Note that this is the same hypothesis for the corresponding SET COVER result and the proofs are indeed related.

4.9 $f(n) \in \left[(\log n)^{2-\epsilon}, O(\log^2 n)\right]$ – GROUP STEINER TREE ON TREES

In this problem we are given a tree $T = (V, E)$ and a collection $\mathcal{C} = \{V_1, V_2, \ldots, V_k\}$ of subsets of V and are asked for a minimum length subtree of T that contains at least one vertex from each subset in $\mathcal{C}$. If one assumes that k is polynomially bounded in $n = |V|$, then there is a randomized algorithm that with arbitrarily high probability yields a solution within a factor $O(\log^2 n)$ of optimal [Garg et al. 2000]. The best deterministic algorithms are almost as good, providing $O(\log^{2+\epsilon} n)$ for any $\epsilon > 0$ [Chekuri et al. 2006]. On the other hand, assuming NP $\not\subseteq$ ZPTIME $\left(n^{\text{polylog}\, n}\right)$ no polynomial-time approximation algorithm can be guaranteed to get within a factor of $(\log n)^{2-\epsilon}$ [Halperin and Krauthgamer 2003]. If we allow general graphs as input, rather than just trees, the upper bound increases to $O(\log^3 n \log\log n)$ [Garg et al. 2000] but no improvement in the lower bound is known.

I conclude with four problems for which, if we assume one of our hypotheses, no polynomial-time approximation algorithm can be guaranteed to be within a polylogarithmic factor of optimal. Note that for polylogarithmic guarantees it typically doesn't matter what parameter we choose to base our functions on, since the plausible ones are all polynomially related and so the choice can only affect the constant factors in the guarantee. For faster growing functions, however, the choice can matter. For instance, using the number of edges m instead of n will reduce the base of the exponent when the running time is of the form $O(2^{(\log n)^{1-\epsilon}})$, and will affect the exponent itself if the running time is $O(n^\epsilon)$. Thus for consistency I shall where possible use the same parameter as was used for similar problems earlier in the list.

4.10 $f(n) \in \left[2^{(\log n)^{1-\epsilon}}, O(n)\right]$ – MIN HORN DELETION

This is the third of the four canonical problems from Khanna et al. [2000]. It is like MIN 2CNF DELETION, except that now instead of a 2SAT instance we have a set of 3-literal "Horn clauses" (clauses with exactly one positive literal) together with a set of unary clauses. Again the goal it to find a truth assignment that minimizes the number of unsatisfied clauses. Assuming P $\neq$ NP, no polynomial-time algorithm can be guaranteed to get within a factor of $2^{(\log n)^{1-\epsilon}}$ [Khanna et al. 2000], and there is no known nontrivial upper bound.

4.11 $f(n) \in \left[2^{(\log n)^{1-\epsilon}}, n\right]$ – NEAREST CODEWORD

This problem was defined in the previous column [Col 24, 2006], and is the fourth of the four canonical problems from Khanna et al. [2000]. In it, we are given the parity check matrix for a linear code with word length n, together with a target n-bit string x, and are asked to find the codeword with minimum Hamming distance from x. The lower bound depends on the hypothesis that NP $\not\subseteq$ DTIME($n^{\text{polylog}\, n}$) [Arora et al. 1997]. The upper bound is the trivial one based on the observation that every codeword is of distance n or less from x.

4.12 $f(n) \in \left[2^{(\log n)^{1-1/\log\log^\delta n}}, O(n)\right]$ – LABEL COVER

The version of the problem covered here differs from the one defined in Section 2 in that the "maps" $\pi_{u,v}$ are allowed to be arbitrary relations and an edge $\{u, v\}$ is covered by a labeling if $(L_1(u), L_2(v)) \in \pi_{u,v}$. The upper bound is provided by a randomized algorithm of Peleg [2006] which the reference asserts can be derandomized. The lower bound assumes only P $\neq$ NP and holds for all $\delta \leq 1/2$ [Dinur and Safra 2004].

4.13 $f(n) \in \left[n^{1/2-\epsilon}, O(n^{1/2}\text{polylog}\, n)\right]$ – NODE CAPACITATED UNSPLITTABLE FLOW

This is like the UNDIRECTED CONGESTION MINIMIZATION problem above, except that now we are given a capacity for each vertex, and wish to find paths between a maximum number of source-sink pairs subject to the capacity constraints. Both upper and lower bounds are from Guruswami et al. [2003], with the lower bound depending only on the hypothesis that P $\neq$ NP.

If one considers the variant in which the capacities are on the edges rather than the nodes and are all equal to 1, one gets the EDGE DISJOINT PATH problem. This problem can be approximated to within a factor $O(m^{1/2})$ of optimal, where m is the number of edges, even for directed graphs [Kleinberg 1996]. If we assume P $\neq$ NP, however, the directed version cannot be approximated to within a factor of $m^{1/2-\epsilon}$ for any $\epsilon > 0$ [Guruswami et al. 2003]. (Note that we would get the same bounds for CLIQUE if we expressed those as a function of m.)

ACKNOWLEDGMENT. In preparing this column I received valuable advice, pointers, and technical information from David Applegate, Aaron Archer, Sanjeev Arora, Jon Bentley, Shuchi Chawla, Uri Feige, Howard Karloff, Subhash Khot, Guy Kortsarz, Robert Krauthgamer, Mihai Patrascu, David Peleg, Yuval Rabani, Luca Trevisan, Kasturi Varadarajan, Suresh Venkatasubramanian, and Nisheeth Vishnoi.

REFERENCES

Agarwal, A., Charikar, M., Makarychev, K., and Makarychev, Y. 2005. $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 573–581.

Andrews, M., and Zhang, L. 2005. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 284–293.

Andrews, M., and Zhang, L. 2006. Logarithmic hardness of the directed congestion minimization problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*. ACM, New York, 517–526.

Arora, S., Babai, L., Stern, J., and Sweedyk, E. Z. 1997. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci. 54*, 317–331. (Preliminary version in *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, Calif., 1993, 724–733.)

Arora, S., Lee, J. R., and Naor, A. 2005. Euclidean distortion and the sparsest cut. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 553–562.

Arora, S., and Lund, C. 1997. Hardness of approximations. In *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, Ed. PWS Publishing Company, Boston, Mass., 399–446.

Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. 1998. Proof verification and the hardness of approximation algorithms. *J. ACM 45*, 501–555. (Preliminary version in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, Calif., 1992, 14–23.)

Arora, S., Rao, S., and Vazirani, U. 2004. Expander flows, geometric embeddings, and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*. ACM, New York, 222–231.

Arora, S., and Safra, S. 1998. Probabilistically checking of proofs: A new characterization of NP. *J. ACM 45*, 70–122. (Preliminary version in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, Calif., 1992, 2–13.)

Charikar, M., Makarychev, K., and Makarychev, Y. 2006. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*. ACM, New York, 205–214.

Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., and Sivakumar, D. 2006. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity 15*, 94–114. (Preliminary version in Proceedings of the 20th Annual IEEE Conference on Computational Complexity, IEEE Computer Society, Los Alamitos, Calif., 2005, 144-153.)

Chekuri, C., Even, G., and Kortsarz, G. 2006. A greedy approximation algorithm for the group Steiner problem. *Disc. App. Math. 154*, 15–34.

Chuzhoy, J., Guha, S., Halperin, E., Khanna, S., Kortsarz, G., Krauthgamer, R., and Naor, J. 2005. Asymmetric k-center is $\log^* n$-hard to approximate. *J. ACM 52*, 538–551. (Preliminary version in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2004, 21–27.)

Chuzhoy, J., and Naor, J. 2004. New hardness results for congestion minimization and machine scheduling. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*. ACM, New York, 28–34.

Cornuéjols, J., Fonlupt, J., and Naddef, D. 1985. The traveling salesman problem on a graph and some related integer polyhedra. *Math. Prog. 33*, 1–27.

Dinur, I., and Safra, S. 2004. On the hardness of approximating label-cover. *Inform. Proc. Lett. 89*, 247–254.

Dinur, I., and Safra, S. 2005. On the hardness of approximating vertex cover. *Ann. Math. 162*, 439–486. (Preliminary version in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2002, 33–42, under the title "The importance of being biased.").

Dorn, F., Penninkx, E., Bodlaender, H. L., and Fomin, F. V. 2005. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *Algorithms – ESA 2005: 13th Annual European Symposium*. Lecture Notes in Computer Science, vol. 3669. Springer-Verlag, Berlin, 95–106.

Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *J. ACM 45*, 634–652. (Preliminary version in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1996, 314–318.)

Feige, U. 2002. Relations between average case complexity and approximation complexity. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. ACM, New York, 534–543.

Feige, U. 2004. Approximating maximum clique by removing subgraphs. *SIAM J. Disc. Math. 18*, 219–225.

Feige, U., Goldwasser, S., Lovász, L., Safra, S., and Szegedy, M. 1996. Interactive proofs and the hardness of approximating cliques. *J. ACM 43*, 268–292. (Preliminary version in *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computing*, IEEE Computer Society, Los Alamitos, Calif., 1991, 2–12.)

Feige, U., Halldórsson, M. M., Kortsarz, G., and Srinivasan, A. 2002. Approximating the domatic number. *SIAM J. Comput. 32*, 172–195. (Preliminary version in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, ACM, New York, 2000, 134–143.)

Feige, U., and Kogan, S. 2004. Hardness of approximation of the balanced complete bipartite subgraph problem. Tech. Rep. MCS04-04, Department of Computer Science and Applied Mathematics, Weizmann Institute, Rehovot, Israel.

Feige, U., and Reichman, D. 2004. On systems of linear equations with two variables per equation. In *APPROX and RANDOM 2004*. Lecture Notes in Computer Science, vol. 3122. Springer-Verlag, Berlin, 117–127.

Garg, N., Konjevod, G., and Ravi, R. 2000. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms 37*, 66–84. (Preliminary version in *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, Penn., 1998, 253–259.)

Garg, N., Vazirani, V. V., and Yannakakis, M. 1996. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput. 25*, 235–251. (Preliminary version in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1993, 698–707.)

Goemans, M. X., and Williamson, D. P. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM 42*, 1115–1145. (Preliminary version in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1994, 422–431.)

Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B., and Yannakakis, M. 2003. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *J. Comput. Syst. Sci. 67*, 473–496. (Preliminary version in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, ACM, New York, 1999, 19–28.)

Halperin, E., and Krauthgamer, R. 2003. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*. ACM, New York, 585–594.

Håstad, J. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica 182*, 105–142. (Preliminary version in *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computing*, IEEE Computer Society, Los Alamitos, Calif., 1996, 627–636.)

Håstad, J. 2001. Some optimal inapproximability results. *J. ACM 48*, 798–859. (Preliminary version in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1997, 1–10.)

Johnson, D. S. 1974. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci. 9*, 256–278.

Johnson, D. S. 1990. A catalog of complexity classes. In *Handbook of Theoretical Computers Science, Volume A: Algorithms and Complexity*, J. V. Leeuwen, Ed. Elsevier, Amsterdam, 67–161.

Karakostas, G. 2005. A better approximation ratio for the vertex cover problem. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*. Lecture Notes in Computer Science, vol. 3580. Springer-Verlag, Berlin, 1043–1050.

Khanna, S., Sudan, M., Trevisan, L., and Williamson, D. P. 2000. The approximability of constraint satisfaction problems. *SIAM J. Comput. 30*, 1863–1920. (Preliminary versions from which this paper was derived are in *Proceedings of the 12th IEEE Conference on Computational Complexity*, IEEE Computer Society, Los Alamitos, Calif., 1997, 282–296 (Authors 1, 2, and 3) and *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1997, 11–20 (Authors 1, 2, and 4).)

Khot, S. 2002. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. ACM, New York, 767–775.

Khot, S. 2004. Ruling out PTAS for graph min-bisection, densist subgraph and bipartite clique. In *Proceedings of the 45th IEEE Symposium on Foundations of Computing*. IEEE Computer Society, Los Alamitos, Calif., 136–145.

Khot, S. 2005. Guest column: Inapproximability results via long code based PCPs. *SIGACT News 36,* 2, 25–42.

Khot, S., Kindler, G., Mossel, E., and O'Donnell, R. 2005. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? Tech. Rep. TR05-101, Electronic Colloquium on Computational Complexity. (Preliminary version in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computing*, IEEE Computer Society, Los Alamitos, Calif., 2004, 146–154.)

Khot, S., and Ponnuswami, A. K. 2006. Better inapproximability results for MaxClique, Chromatic Number, and Min-3Lin-Deletion. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming.* Lecture Notes in Computer Science, vol. 4051. Springer-Verlag, Berlin, 226–237.

Khot, S., and Regev, O. 2003. Vertex cover might be hard to approximate to within $2-\epsilon$. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity.* IEEE Computer Society, Los Alamitos, Calif., 379–386.

Khot, S. A., and Vishnoi, N. K. 2005. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics in L_1. In *Proceedings of the 46th IEEE Symposium on Foundations of Computing.* IEEE Computer Society, Los Alamitos, Calif., 53–62.

Kleinberg, J. M. 1996. Approximation algorithms for disjoint paths problems. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass.

Leighton, T., and Rao, S. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM 46*, 787–832. (Preliminary version in *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Los Alamitos, Calif., 1988, 422–431.)

Lovász, L. 1975. On the ratio of optimal integral and fractional covers. *Disc. Math. 13*, 383–s 390.

Lund, C., and Yannakakis, M. 1994. On the hardness of approximating minimization problems. *J. ACM 41*, 960–981. (Preliminary version in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1993, 286–293.)

Mossel, E., O'Donnell, R., and Oleszkiewicz, K. 2005. Noise stability of functions with low influences: invariance and optimality. In *Proceedings of the 46th IEEE Symposium on Foundations of Computing.* IEEE Computer Society, Los Alamitos, Calif., 21–30.

Nemirovsky, A., Roos, C., and Terlaky, T. 1999. On maximization of quadratic form over intersection of ellipsoids with common center. *Math. Prog. 86*, 463–473.

Panigrahy, R., and Vishwanathan, S. 1998. On $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem. *J. Algorithms 27*, 259–268. (Preliminary version in *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, Penn., 1996, 1–5.)

Papadimitriou, C. H., and Steiglitz, K. 1976. Some complexity results for the traveling salesman problem. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing.* ACM, New York, 1–9.

Peleg, D. 2006. Approximation algorithms for label-cover$_{MAX}$ and red-blue set cover problems. *J. Disc. Algorithms*, to appear. (Preliminary versions in *SWAT 2000*, Lecture Notes in Computer Science, vol. 1851, Springer-Verlag, Berlin, 2000, 220–231.)

Raghavan, P., and Thompson, C. D. 1987. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica 7*, 365–374.

Raz, R. 1998. A parallel repetition theorem. *SIAM J. Comput. 27*, 763–803. (Preliminary version in *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1995, 447–456.)

Trevisan, L. 2004. Inapproximability of combinatorial optimization problems. E-print arXiv:cs.CC/0409043 (`http://arxiv.org/abs/cs.CC/0409043`).

Varadarajan, K. R., Venkatesh, S., Ye, Y., and Zhang, J. 2007. Approximating the radii of point sets. *SIAM J. Comput.*, to appear. (Preliminary versions from which this paper was derived are in *Proceedings of the 43th IEEE Symposium on Foundations of Computing*, IEEE Computer Society, Los Alamitos, Calif., 2002, 561–569 (authors 1, 2, and 4), and in *APPROX 2003 + RANDOM 2003*, Lecture Notes in Computer Science, vol. 2764, Springer-Verlag, Berlin, 2003, 178–187 (Authors 3 and 4).)

Vishnoi, N. 2006. Personal communication.

Zuckerman, D. 2006. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing.* ACM, New York, 681–690.