The Approximability of NP-hard Problems

Sanjeev Arora*
Princeton University

1 Introduction

Many problems in combinatorial optimization are *NP-hard* (see [60]). This has forced researchers to explore techniques for dealing with NP-completeness. Some have considered algorithms that solve "typical" or "average" instances instead of worst-case instances [86, 100]. In practice, however, identifying "typical" instances is not easy. Other researchers have tried to design *approximation algorithms*. An algorithm *achieves an approximation ratio* α for a maximization problem if, for *every* instance, it produces a solution of value at least OPT/α, where OPT is the value of the optimal solution. (For a minimization problem, achieving a ratio α involves finding a solution of cost at most $\alpha\, OPT$.) Note that the approximation ratio is ≥ 1 by definition.

After twenty-five years of research, approximation algorithms is a major research area with deep techniques (see [75] for a detailed survey). Nevertheless, researchers have failed to design good approximation algorithms for a wide variety of NP-hard optimization problems. Recent developments in complexity theory —specifically, in the area of probabilistically checkable proofs or PCPs— suggest a reason for this failure: for many NP-hard problems, including MAX-CLIQUE, CHROMATIC NUMBER, MAX-3SAT, and SET-COVER, achieving certain reasonable approximation ratios is no easier than computing optimal solutions. In other words, approximation is NP-hard. These negative results are described in Section 3.

Interestingly enough, such negative results, instead of discouraging researchers, have fed an increased interest in approximation algorithms. Maybe this was to be expected. PCP-based inapproximability results are known for only some NP-hard problems, not for all. In cases where one has failed to obtain an inapproximability result, one has a reason to hope for an approximation algorithm! Indeed, I feel that the approximation properties of NP-hard problems are best investigated with a two-pronged approach, combining a search for good approximation algorithms with a search for inapproximability results. A failed attempt to prove an inaproximability result can provide insight into how to design an algorithm, and vice versa.

Another important current endeavor is to classify NP-hard problems according to their approximability. The goal is to identify a small number of structural features which explain the variation in approximability among NP-hard optimization problems. Section 4 describes some work along those lines.

Section 5 surveys some approximation algorithms that were recently discovered. Their discovery appears to have been influenced (either directly or indirectly) by the issues opened up by the abovementioned inapproximability results.

2 Definitions and History

This section contains basic definitions and gives an impressionistic view of developments in approximation algorithms up to the mid 1980s. The rest of the article will cover more recent developments.

An *NP-minimization problem* is specified by a polynomial-time computable bivariate *cost function* c and a constant k. For any input x, the goal is to find a solution $y, |y| \leq k\, |x|^k$ such that $c(x, y)$ is minimized. For example, in the TSP, x is a set of points and their mutual distances, a solution y is a tour and $c(x, y)$ is its length. (An NP-maximization problem is specified analogously.)

*Supported by NSF CAREER award NSF CCR-9502747, an Alfred Sloan Fellowship, and a Packard Foundation Fellowship. Email: `arora@cs.princeton.edu`
This survey was a plenary lecture at *ACM Symposium on Theory of Computing*, 1998. The latest version appears at `www.cs.princeton.edu/~arora/`

An approximation algorithm is said to achieve an *approximation ratio* $\alpha \geq 1$ for the above problem if for each input x it computes a solution y of cost at most $\alpha \cdot OPT$, where OPT is the cost of the optimum. If such an algorithm exists, we say that the problem is *approximable* upto a factor α. If we can show —assuming P $\neq$ NP or some similar conjecture— that such an algorithm does not exist, then we say that the problem is *inapproximable* upto factor α. We allow α to depend upon $|x|$; for example it could be $\log|x|$. A *fully polynomial time approximation scheme* or FPTAS is an algorithm that, given input x and error bound $\epsilon > 0$, computes a solution of cost at most $(1+\epsilon)OPT$ in time that is polynomial in $|x|$ and $1/\epsilon$. A *polynomial time approximation scheme* or PTAS is an algorithm that, for any fixed $\epsilon > 0$, can achieve an approximation ratio $1+\epsilon$ in time that is polynomial in the input size. The running time could be exponential (or worse) in $1/\epsilon$ (for example, $|x|^{1/\epsilon}$).

Some approximation algorithms were discovered before NP-completeness, most notably Graham's algorithm for scheduling to minimize makespan [69], which achieves an approximation ratio 2. After the discovery of NP-completeness, Garey, Graham, and Ullman [57] and then Johnson [79] formalized the notion of an approximation algorithm. Johnson also described simple approximation algorithms for MAX-CUT, MAX-SAT, and SET-COVER (Lovász [104] gave a similar algorithm for SET-COVER). Sahni and Gozalez [118] showed that achieving any constant approximation ratio for the TSP is NP-hard. Notable approximation algorithms were discovered, including Sahni's PTAS [117] for KNAPSACK, Ibarra and Kim's improvement of this algorithm to an FPTAS [78], and Christofides's algorithm for metric TSP with approximation ratio 1.5 [37]. Garey and Johnson identified the notion of *strong NP-completeness* [59] and showed that most NP-hard problems of interest do not have FPTAS's if P $\neq$ NP. Some preliminary work was also done on classifying problems according to approximability [14, 15, 16].

The early 1980s saw further success in design of algorithms, including Fernandez de la Vega and Lueker's PTAS [53] and Karmarkar and Karp's FPTAS [84] for BIN-PACKING[1], PTAS's for some geometric packing and covering problems (see the chapter by Hochbaum in [75] for a survey), and for various problems on planar graphs [102, 22]. (Planar graphs are easier to treat because they have small separators.) Wigderson showed how to color 3-colorable graphs with $O(\sqrt{n})$ colors [124]. Hochbaum and Shmoys [76] designed a PTAS for MIN-MAKESPAN. In another work [77] they proved a *tight* result for the k-CENTER problem: they showed how to achieve an approximation ratio 2 and proved that achieving a ratio $2-\epsilon$ is NP-hard.

In the late 1980s two important developments occurred. Leighton and Rao [98] used a powerful method based upon linear programming duality to design algorithms that achieve approximation ratio $O(\log n)$ or $O(\log^2 n)$ for a variety of graph separation and layout problems. The influence of this work can be seen in many approximation algorithms today. Around the same time, Papadimitriou and Yannakakis [111] sought to lay the study of approximability on a sound footing and made a start by defining a class of optimization problems they called MAX-SNP. Using a certain *approximability preserving* reduction they defined *completeness* for MAX-SNP and showed that MAX-3SAT is MAX-SNP-complete (see Section 4.2). This showed that an inapproximability result for MAX-3SAT would generalize to a large class of problems, which motivated the discovery of the PCP Theorem a few years later.

Papadimitriou and Yannakakis also noted that the classical style of reduction (Cook-Levin-Karp [41, 99, 85]) relies on representing a computational history by a combinatorial problem. A computational history is a very *non-robust* object, since even changing a bit in it can affect its correctness. This nonrobustness lay at the root of the difficulty in proving inapproximability results.

Luckily, more robust representations of computational histories were around the corner, thanks to work on *interactive proofs* and *program checking*.

3 Probabilistically Checkable Proofs

The concept of PCPs evolved out of *interactive proofs*, which were defined in the mid 1980s by Goldwasser, Micali, and Rackoff [67] and Babai [17] as a probabilistic extension of the nondeterminism used in NP. Interactive proofs found many applications in cryptography and complexity theory (see Goldreich's article [64]), one of which involved an early version of probabilistically checkable proof systems (Fortnow, Rompel, and Sipser [54]). In 1990, Lund, Fortnow, Karloff and Nisan [106] and Shamir [120] showed IP=PSPACE, thus giving a new probabilistic definition of PSPACE in terms of interactive proofs. They introduced a revolutionary algebraic way of looking at boolean formulae. In restrospect, this algebraization can also be seen as a "robust" representation of computation (cf. Section 2). The inspiration to use polynomials came from works on *pro-*

[1]The approximation ratio of these BIN-PACKING algorithms approaches 1 only if the value of the optimum approaches ∞.

gram checking [34] (see also [101, 23, 35]). Babai, Fortnow, and Lund [19] used similar methods to give a new probabilistic definition of NEXPTIME, the exponential analogue of NP. To extend this result to NP, Babai, Fortnow, Levin, and Szegedy [20] and Feige, Goldwasser, Lovász, Safra, and Szegedy [49] studied variants of what we now call probabilistically checkable proof systems (Babai et al. called their systems *holographic* proofs). Feige et al. also proved the first inapproximability result to come out of the PCP area. They showed that if any polynomial-time algorithm can achieve a constant approximation ratio for the MAX-CLIQUE problem, then each NP problem is solvable in $n^{O(\log\log n)}$ time. This important result drew everybody's attention to the (as yet unnamed) area of probabilistically checkable proofs. A year later, Arora and Safra [12] formalized and named the class PCP and used it to give a new probabilistic definition of NP. (The works of Babai et al. and Feige et al. were precursors of this new definition.) They also showed that approximating MAX-CLIQUE is NP-hard. Soon, Arora, Lund, Motwani, Sudan, and Szegedy [10] proved the PCP Theorem (see below) and showed that MAX-SNP-hard problems do not have a PTAS if P $\neq$ NP. Many sources attribute the PCP theorem jointly to [12, 10]. For brief surveys of all the above developments see [18, 64, 80]. In the years since the discovery of the PCP Theorem, other variants of PCP have been studied and used in inapproximability results.

Now we define the class PCP. A $(r(n), q(n))$-*restricted verifier* for a language L, where r, q are integer-valued functions, is a probabilistic turing machine M that, given an input of size n, checks membership proofs for the input in the following way. The proof is an array of bits to which the verifier has random-access (that is, it can *query* individual bits of the proof).

- The verifier reads the input, and uses $O(r(n))$ random bits to compute a sequence of $O(q(n))$ addresses in the proof.

- The verifier queries the bits at those addresses, and depending upon what they were, outputs "accept" or "reject".

- If the input x is in L then

$$\text{there exists proof } \Pi \text{ s.t. } \Pr[M^{\Pi}\text{accepts}] = 1, \tag{1}$$

  If $x \notin L$ then

$$\text{for every proof } \Pi,\ \Pr[M^{\Pi}\text{accepts}] \leq 1/2 \tag{2}$$

  (In both cases the probability is over the choice of the verifier's random string.)

PCP$(r(n), q(n))$ is the complexity class consisting of every language with an $(r(n), q(n))$-restricted verifier. Since NP is the class of languages whose membership proofs can be checked by a deterministic polynomial-time verifier, NP $= \cup_{c\geq 0}$PCP$(0, n^c)$. The PCP Theorem gives an alternative definition:

$$\mathrm{NP} = \mathrm{PCP}(\log n, 1) \tag{3}$$

Other PCP-like classes have been defined by using variants of the definition above, and shown to equal NP(when the parameters are chosen appropriately). We mention some variants and the best results known for them.

1. The probability 1 in condition (1) may be allowed to be $c < 1$. Such a verifier is said to have *imperfect completeness* c.

2. The probability 1/2 in condition (2) may be allowed to be $s < c$. Such a verifier is said to have *soundness* s. Using standard results on random walks on expanders, it can be shown from the PCP theorem that every NP language has verifiers with perfect completeness that use $O(k)$ query bits for soundness 2^{-k} (here $k \leq O(\log n)$).

3. The number of query bits, which was $O(q(n))$ above, may be specified more precisely together with the leading constant. The constant is important for many inapproximability results. Building upon past results on PCPs and using fourier analysis, Håstad [74] recently proved that for each $\epsilon > 0$, every NP language has a verifier with completeness $1 - \epsilon$, soundness 1/2 and only 3 query bits. He uses this to show the inapproximability of MAX-3SAT upto a factor $8/7 - \epsilon$.

4. The *free bit* parameter may be used instead of query bits [50, 29]. This parameter is defined as follows. Suppose the query bit parameter is q. After the verifier has picked its random string, and picked a sequence of q addresses, there are 2^q possible sequences of bits that could be contained in those addresses. If the verifier accepts for only t of those sequences, then we say that the free bit parameter is $\log t$ (note that this number need not be an integer).

5. *Amortized free bits* may be used [29]. This parameter is defined as $\lim_{s\to 0} f_s/\log(1/s)$, where

f_s is the number of free bits needed by the verifier to make soundness $< s$. Bellare and Sudan show (modifying a reduction from [49]) that if every NP language has a verifier that uses $O(\log n)$ random bits and F amortized free bits then MAX-CLIQUE is inapproximable upto a factor $n^{1/(1+F+\delta)}$ for each $\delta > 0$. Håstad has shown that for each $\epsilon > 0$, every NP language has a verifier that uses $O(\log n)$ random bits and ϵ amortized free bits [73]. This implies that MAX-CLIQUE is inapproximable upto a factor $n^{1-\epsilon}$.

6. The proof may contain not bits but letters from a larger alphabet Σ. The verifier's soundness may then depend upon Σ. In a *p prover 1-round interactive proof system*, the proof consists of p arrays of letters from Σ. The verifier is only allowed to query 1 letter from each array. Since each letter of Σ is represented by $\lceil \log |\Sigma| \rceil$ bits, the number of *bits* queried may be viewed as $p \cdot \lceil \log |\Sigma| \rceil$. Constructions of such proof systems for NP appeared in [30, 96, 52, 27, 50, 113]. Lund and Yannakakis [108] used these proof systems to prove inapproximability results for SETCOVER and many subgraph maximization problems. The best construction of such proof systems is due to Raz and Safra [114]. They show that for each $k \leq \sqrt{\log n}$, every NP language has a verifier that uses $O(\log n)$ random bits, has $\log |\Sigma| = O(k)$ and soundness 2^{-k}. The parameter p is $O(1)$.

We mention here a few of the ideas at the core of all the above-mentioned results. First, note that it suffices to design verifiers for 3SAT since 3SAT is NP-complete and a verifier for any other language can transform the input to a 3SAT instance as a first step. The verifier for 3SAT is designed by "lifting" the question of whether or not the input 3CNF formula is satisfiable to an algebraic domain: the formula has a satisfying assignment iff a related polynomial exists with certain specified properties. The verifier expects the membership proof to contain this polynomial presented *by value*. The verifier uses certain algorithms that, given a function specified by value, can check that it satisfies the desired properties. Two of those algorithms are Sum Check [106] and Low Degree Test (invented in [19] and improved in [20, 49, 116, 12, 10, 114, 13]). An important technique introduced in [12] and used in all subsequent papers is *verifier composition*, which composes two verifiers to give a new verifier some of whose parameters are lower than those in either verifier. Verifier composition relies on the notion of a *probabilistically checkable encoding*, a notion to which Arora and Safra were led by results in [20]. (Later, in the proof of the PCP Theorem [10], *Hadamard codes* were used to implement such encodings.) Another result that plays a crucial role in recent works on PCP is Raz's *parallel repetition theorem* [113]. Finally, the work of Håstad [73, 74] uses encodings based upon the so-called *Long Code* [26].

A striking feature of the PCP area is that each advance has built upon previous papers, often using them in a "black-box" fashion. Consequently, a proof of Håstad's MAX-CLIQUE result from first principles would fill well over 100 pages!

3.1 Connection to Inapproximability

The connection between PCPs and inapproximability was first established by the result of Feige et al. [49] on MAX-CLIQUE. (Condon [38] had discovered a connection somewhat earlier; but she worked with a different notion of proof verification and a less natural optimization problem.) For a while, this connection was viewed as "just" coincidental, and this viewpoint began to change only after the PCP theorem was shown to be equivalent to the inapproximability of MAX-SNP [11, 10]. The connection finally became undeniable after Lund and Yannakis [108, 107] used PCP constructions to prove the inapproximability of SETCOVER, CHROMATIC-NUMBER, and many MAX-π-SUBGRAPH problems. Since then many other inapproximability results have been discovered as described in [9, 43].

As pointed out in [9], just as 3SAT is the "canonical" problem in the theory of NP-completeness, MAX-3SAT is the "canonical" problem in the theory of inapproximability. Once we prove the inapproximability of MAX-3SAT, we can prove most other inapproximability results, though not always in the strongest possible form.

The inapproximability of MAX-3SAT is intimately related to the PCP theorem: It is easy to see that $\mathrm{NP} \subseteq \mathrm{PCP}(\log n, 1)$ iff there is a reduction τ from SAT to MAX-3SAT that ensures the following for some fixed $\epsilon > 0$:

$$I \in \mathrm{SAT} \quad \Rightarrow \quad \text{MAX-3SAT}(\tau(I)) = 1$$
$$I \notin \mathrm{SAT} \quad \Rightarrow \quad \text{MAX-3SAT}(\tau(I)) \leq \frac{1}{1+\epsilon}.$$

Here MAX-3SAT$(\tau(I))$ is the maximum fraction of clauses in formula $\tau(I)$ that are simultaneously satisfiable.

Let's check that the "if" part holds: if the above reduction exists then $\mathrm{NP} \subseteq \mathrm{PCP}(\log n, 1)$. Given any NP language L and an input, the verifier reduces it to SAT, and then to MAX-3SAT using the above reduction. It expects the membership proof to contain a satisfying assignment to the MAX-3SAT instance. To check this membership proof probabilistically, it picks a clause uniformly at random, reads the values of the variables in it (notice, this requires reading only 3 bits), and accepts iff these values satisfy the clause. Clearly, if the original input is in L, there is a proof which the verifier accepts with probability 1. Otherwise every proof is rejected with probability at least $1 - 1/(1 + \epsilon)$. Of course, repeating the verification $O(1/\epsilon)$ times makes the rejection probability $\geq 1/2$.

The "only if" part (NP $\subseteq$ PCP$(\log n, 1)$ implies the existence of the above-mentioned reduction) is only a little more difficult. It involves replacing a verifier's actions by an equivalent 3CNF formula (see [9]).

3.1.1 Other inapproximability results

In the past few years, two types of research has been done on inapproximability. One tries to improve the inapproximability results that are already known. Usually this involves improving the parameters of some known verifier. MAX-3SAT, MAX-CLIQUE, SET-COVER, and CHROMATIC NUMBER are the main problems for which such improved results have been obtained. The first paper in this area was Bellare, Goldwasser, Lund, and Russell [27]; others are Feige and Kilian [50, 51]; Bellare and Sudan [29]; Bellare, Goldreich, and Sudan [26]; Fürer [56], Håstad [73, 74]; Feige [47]; Raz and Safra [114]; and Arora and Sudan [13]. Thanks to this work, we now know of *tight* results for the four problems mentioned above: in other words, the inapproximability results for these problems almost match the performance of the best algorithms. The tight approximation ratios for the various problems are: $8/7 - \epsilon$ for MAX-3SAT [74], $n^{1-\epsilon}$ for MAX-CLIQUE [73], $n^{1-\epsilon}$ for CHROMATIC NUMBER [51] and $(1 - \epsilon) \ln n$ for SET-COVER [47].

Other papers have tried to prove inapproximability results for problems that were not already known to be inapproximable. In the two years after the discovery of the PCP Theorem, many new results were discovered by Lund and Yannakakis [108, 107], Arora, Babai, Stern, and Sweedyk [6], Bellare [24], Bellare and Rogaway [28], Zuckerman [128], etc.. Since then, there has been essentially no progress except for a few MAX-SNP-hardness results.

Crescenzi and Kann [43] maintain a compendium that lists the current approximation status of important optimization problems.

3.2 Other applications of PCP Techniques

The PCP Theorem and related techniques have found many other theoretical applications in complexity theory and cryptography, including new definitions of PSPACE [40] and PH [92], probabilistically checkable codes, zero-knowledge proofs, checkable VLSI computations, etc. See [3] for a survey. One interesting application first observed by Babai et al. [20] (and given prominence in the *New York Times* [94] article on the PCP Theorem) is that the PCP Theorem implies that formal mathematical proofs can be rewritten —with at most a polynomial blowup in size— such that a probabilistic checker can verify their correctness by examining only a constant number of bits in them. (The constant does not depend upon the length of the proof.) To see that this follows from the PCP Theorem, just notice that for most known axiomatic systems, the following language is in NP (and consequently has a $(\log n, 1)$-restricted verifier):

$$\{< T, 1^n >: T \text{ is a theorem with a proof of size} \leq n\} .$$

4 Classifying NP-hard problems by approximability

Soon after the discovery of NP-completeness, it was recognized that optimization versions of NP-complete problems are not necessarily polynomial-time equivalent in terms of approximability. As mentioned in Section 2, KNAPSACK was found to have an FPTAS whereas TSP was shown to be inapproximable upto any constant factor.

We are therefore led naturally to the following research program. (i) Define an *approximation preserving reduction* appropriately, such that if two problems are interreducible then they have the same approximability. (ii) Show that the interreducibility relation divides NP-hard optimization problems into a small number of equivalence classes and give "complete" problems for each class.

Of course, the above program is too ambitious in general because the number of equivalence classes may not be small and may even be infinite. (For example a classical result of Ladner [95] says that there are infinitely many equivalence classes in NP for the usual polynomial-time reducibility.) Nevertheless, we could conceivably declare such a program a success if all interesting problems could be shown to fall into a small number of classes. (For example, Ladner's result notwithstanding, in practice most decision problems are found to be either NP-complete or in P, and that is why we consider NP-completeness a successful theory.)

Moreover, the task in (i) —defining an approximation-preserving reduction— is also harder than it looks. After several competing definitions the consensus choices these days are the *A-reduction* [45] and the *AP-reduction* [44]. An A-reduction is defined such that if a problem Π is A-reducible to problem Γ and Γ is approximable upto a factor α, then we can conclude that Π is approximable upto a factor $O(\alpha)$. Thus A-reductions are useful for studying coarse structure of classes of problems, when approximation ratios can be specified with big-Oh notation. An AP-reduction is defined such that if problem Π is AP-reducible to problem Γ and Γ is approximable upto a factor $1+\alpha$, then we can conclude that Π is approximable upto a factor $1+O(\alpha)$. Thus AP-reductions are suited for studying the fine structure of problem classes for which approximation ratios are close to 1.

Whatever the definition of an approximation-preserving reduction, coming up with reductions between concrete problems is definitely not easy. Even seemingly related problems such as MAX-CLIQUE and CHROMATIC NUMBER (both of which were recently proven to be inapproximable upto a factor $n^{1-\epsilon}$ [73, 51]) are not known to be interreducible. Of course, once a class of interreducible problems has been identified, we still have to find the "right" approximation ratio to associate with it. Namely, we have to find an α such that $\Theta(\alpha)$ is achievable by some polynomial-time algorithm and achieving a ratio $o(\alpha)$ is impossible if $\mathrm{P} \neq \mathrm{NP}$.

Having outlined some of the difficulties, I now trace the progress made on the above program.

4.1 Empirical Classification of Known Inapproximability Results

In a survey article, Arora and Lund [9] briefly describe how to prove most known inapproximability results. They list at least two dozen important problems for which inapproximability was proved using PCP-based techniques. They make the empirical observation that these problems divide into four broad classes, based upon the approximation ratio that is provably hard to achieve for them. Class I contains all problems for which achieving an approximation ratio $1+\epsilon$ is NP-hard for some fixed $\epsilon > 0$. Classes II, III, and IV contain problems for which the corresponding ratios are $\Theta(\log n)$, $2^{\log^{1-\epsilon} n}$ for every fixed $\epsilon > 0$, and n^{ϵ} for some fixed $\epsilon > 0$. (Of course, each class contains every higher numbered class.) Inapproximability results within a class seem to use similar techniques, and in fact the authors identify a canonical problem in each class which can be used to prove the inapproximability results for all other problems in that class. The inapproximability of the canonical problems can be proved using MAX-3SAT.

The authors pose an open question whether or not these empirically observed classes can be derived from some deeper theory. (Only Class I seems to have an explanation, using MAX-SNP.)

4.2 MAX-SNP

Papadimitriou and Yannakakis [111] were interested in determining which problems have a PTAS and which don't. They defined a class of optimization problems, MAX-SNP, as well as a notion of completeness for this class. Roughly speaking, a MAX-SNP-complete problem is one that behaves just like MAX-3SAT in terms of approximability: MAX-3SAT has a PTAS iff every MAX-SNP-complete problem does. (This made MAX-3SAT a plausible candidate problem for an inapproximability result, and in particular motivated the discovery of the PCP theorem.)

MAX-SNP contains constraint-satisfaction problems, where the constraints are local. The goal is to satisfy as many constraints as possible. The concept of "local" constraints is formalized using logic: constraints are local iff they are definable using a quantifier-free propositional formula. The inspiration to use this definition came from Fagin's characterization of NP using 2nd order logic [46]

A maximization problem is in MAX-SNP if there is a sequence of relation symbols $G_1, \ldots, G_m$, a relation symbol S, and a quantifier-free formula $\phi(G_1, \ldots, G_m, S, x_1, \ldots, x_k)$ (where each x_i is a variable) such that the following are true: (i) there is a polynomial-time algorithm that, given any instance I of the problem produces a set $\mathcal{U}$ and a sequence of relations $G_1^{\mathcal{U}}, \ldots, G_m^{\mathcal{U}}$ on $\mathcal{U}$, where each $G_i^{\mathcal{U}}$ has the same arity ("arity" refers to the number of arguments) as the relation symbol G_i. (ii) The value of the optimum solution on instance I, denoted $\mathrm{OPT}(I)$ equals

$$\max_{S^{\mathcal{U}}} |\{\vec{x} \in \mathcal{U}^k : \phi(G_1^{\mathcal{U}}, \ldots, G_m^{\mathcal{U}}, S^{\mathcal{U}}, \vec{x}) = \text{ TRUE}\}|,$$

where $\vec{x} = (x_1, \ldots, x_k)$, $S^{\mathcal{U}}$ is a relation on $\mathcal{U}$ with the same arity as S, and $\mathcal{U}^k$ is the set of k-tuples of $\mathcal{U}$.

To understand this definition in more familiar terms, note that the sequence of relation symbols $G_1, \ldots, G_m, S$, as well as their arities, are fixed for the problem. Thus when the universe $\mathcal{U}$ has size n, the sequence of relations $G_1^{\mathcal{U}}, \ldots, G_m^{\mathcal{U}}$ implicitly defines an "input" of size $O(n^c)$ where c is the largest arity of a relation in G. Solving the optimization problem involves finding a relation $S^{\mathcal{U}} \subseteq \mathcal{U}^k$ that maximizes the number of k-tuples satisfying ϕ. Since $|\mathcal{U}|^k = n^k$, this relation $S^{\mathcal{U}}$ can be viewed as a "feasible solution," which can be specified using n^k bits.

As an example, let us reformulate MAX-CUT, the problem of partitioning the vertex set of an undirected graph into two parts such that the number of edges crossing the partition is maximized. Let the universe $\mathcal{U}$ be the vertex set of the graph, and let G consist of E, a binary relation whose interpretation is "adjacency." Let S be a unary relation (interpreted as one side of the cut), and $\phi(E, S, (u, v)) = (u < v) \wedge E(u, v) \wedge (S(u) \neq S(v))$. Clearly, the optimum value of MAX-CUT on the graph is

$$\max_{S \subseteq \mathcal{U}} \left| \left\{ (u, v) \in \mathcal{U}^2 : \phi(E, S, (u, v)) = \text{ TRUE} \right\} \right|.$$

Papadimitriou and Yannakakis defined an approximation-preserving reduction called an *L-reduction*, and showed that every MAX-SNP problem L-reduces to MAX-3SAT, MAX-CUT, Metric TSP, Vertex Cover, etc. Thus these problems are *MAX-SNP-hard*. MAX-CUT and MAX-3SAT are also in MAX-SNP, so they are *MAX-SNP-complete*.

Khanna, Motwani, Sudan and Vazirani [89] have shown an interesting structural fact about MAX-SNP: every NP-optimization problem that is approximable within a constant factor is AP-reducible to MAX-3SAT. This fact is a consequence of the PCP Theorem.

4.3 Constraint Satisfaction Problems

A *constraint satisfaction problem* consists of c boolean functions $f_1, \ldots, f_c$ of arity at most k, where c, k are some constants. An instance of the problem consists of n boolean variables and m functions, each of which is drawn from the c specified functions. For each of the m functions, a sequence of k inputs is specified to which the function is to be applied. Note that each 0/1 assignment to the inputs results in every function being set to 0/1. The goal is to find an assignment to the inputs that makes all functions 1. Note that 3SAT is a special case of this problem ($k = 3$ and $c = 16$; the 16 allowable functions are x, $\overline{x}$, $x \vee y$, $x \vee \overline{y}$, etc.). MAX-CSP problems are defined similarly, except the goal is to maximize the number of functions set to 1. MAX-CUT, MAX-2SAT, and MAX-3SAT lie in MAX-CSP.

A classical result of Schaefer [119] shows that each constraint satisfaction problem is either in P or NP-complete. (In other words, the infinitely many levels in NP exhibited by Ladner's theorem [95] are absent among constraint satisfaction problems.) Schaefer also gives simple properties of the constraint functions which determine whether the problem is NP-complete.

Thus MAX-CSP problems present a good testing ground for the hypothesis that approximability properties divide all interesting optimization problems into a small number of classes. (A simple greedy algorithm achieves a constant approximation ratio for every MAX-CSP problem, so this class does not contain the rich diversity of approximation ratios found in the real world.) Creignou [42] and Khanna, Sudan, and Williamson [91] show that, indeed, there are only two types of MAX-CSP problems: those that can be optimally solved in polynomial time, and those that are MAX-SNP-hard. (Which case holds depends upon certain syntactic properties of the constraint functions.)

More recently Khanna, Sudan, and Trevisan [90] have studied *minimization* problems connected with constraint satisfaction. These classes contain familiar problems such as s-t min-cut, vertex cover, hitting set with bounded size sets, integer programs with two variables per inequality, deleting the minimum number of edges to make a graph bipartite, and the nearest codeword problem. Again, the classes are found to divide into a constant number of equivalence classes.

The above research touches a good number of problems, and seems to provide clues to what a more comprehensive theory might look like.

5 Good Upperbounds: New Algorithms

Approximation algorithms is a big area and some of its high points of the past few years are surveyed in chapters of [75]. See, for example, Goemans and Williamson's primal-dual technique for network design [63]; Linial, London, and Rabinovich's discovery of geometric embeddings of graphs [103] and their use in algorithms (see Shmoy's chapter [121]); and the use of linear programming techniques in scheduling problems (see Hall's chapter [71]).

Below I focus only on some algorithms whose discovery appears to have been stimulated by the issues raised by inapproximability results. Sometimes, inapproximability results can direct algorithm designers to the right problem and the right result.

5.1 MAX-CUT, MAX-2SAT, MAX-3SAT

Soon after the work on MAX-SNP and PCP showed that PTAS's were unlikely for problems such MAX-CUT and MAX-2SAT, Goemans and Williamson discovered an algorithm that achieves approximation ratio 1.13 for MAX-CUT and MAX-2SAT. The best algorithms before that point could only achieve approximation ratios 2 and 1.33 respectively.

Their chief tool, semidefinite programming [70], has since been applied to other constraint satisfaction problems. Karloff and Zwick [83] have used it to achieve an approximation ratio 8/7, for MAX-3SAT. This is an example of a *tight* result, since Håstad had already shown that achieving an approximation ratio $8/7 - \epsilon$ is hard. (Thus Karloff and Zwick knew the "correct" approximation ratio to shoot for, which must have helped.) Semidefinite programming has also been used in better algorithms for coloring 3-colorable graphs [82] and for the MIN-BANDWIDTH problem [33] (a stronger version of the latter result was independently obtained by Feige using geometric embeddings [48]).

5.2 Geometric Network Design Problems

The status of geometric problems such as Euclidean TSP or Steiner tree was a major open question in the post-PCP years. The more general case where the points lie in a metric space was known to be MAX-SNP-hard [32]. I tried unsuccessfully to prove inapproximability in the geometric case, and then became convinced that no such result was provable. This led me to try to design a PTAS, at which I succeeded [4]. Insights gained from my failed attempt to prove inapproximability were invaluable. (Note, however, that Mitchell independently arrived at a similar PTAS a few months later, while working from a different angle [109].)

For similar reasons, I suspect that graph bisection and layout problems may also have good approximation algorithms; I do not currently see how to apply PCP techniques to prove an inapproximability result for them.

5.3 Problems on Dense Graphs and Hypergraphs

Recent work shows that dense instances of problems such as MAX-CUT, MAX-ACYCLIC SUBGRAPH, MAX-k-SAT, MIN-BISECTION etc., have PTAS's ([8]; some of these results were independently obtained by Fernandez de la Vega [dlV94]). (A *dense* graph is one in which each vertex has degree $\Theta(n)$; denseness for formulae is defined analogously.) We were led to this PTAS when we were investigating whether MAX-SNP problems are inapproximable on dense instances, and we could not come up with an inapproximability result.

The work on dense graphs has recently been extended in [7, 55, 66]. The last two papers make the PTAS's extremely efficient.

6 Future Directions

Simplifying the proofs of the results on PCP's is an extremely important task. Current proofs fill several dozen pages each. As has happened in many other areas in the past, the process of simplifying the proofs will very likely lead to new insights.

Very few new inapproximability results (as opposed to improvements of known inapproximability results) have been proved in the past couple of years, and the time is ripe for the next big result. This result (either an algorithm or an inapproximability result) could come from the field of graph separation and layout problems, which have been intensively studied [121]. The MIN-BISECTION problem (divide a graph into two equal halves so as to minimize the number of edges crossing between them) is particularly intriguing. We currently know neither an algorithm that achieves an approximation ratio better than around $n^{1/2}$, nor an inapproximability result of *any* sort.

Recently, Ajtai [1] and Ajtai and Dwork [2] have shown deep connections between (conjectured) inapproximability and cryptography by constructing cryptographic primitives whose security is based upon the inapproximability of certain lattice problems (such as SHORTEST VECTOR and NEAREST VECTOR). A strong inapproximability result for these problems (which would probably have to be based upon an assumption other than P $\neq$ NP) would therefore have exciting implications.

acknowledgements

I thank Madhu Sudan, Oded Goldreich, Sanjeev Khanna, David Shmoys, Luca Trevisan, and David Williamson for helpful advice and discussions while I was writing this survey.

References

[1] M. Ajtai. Generating hard instances of lattice problems. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1996.

[2] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1997

[3] S. Arora. *Probabilistic Checking of Proofs and Hardness of Approximation Problems.* PhD thesis, U.C. Berkeley, 1994. Available from `http://www.cs.princeton.edu/~arora` .

[4] S. Arora. Polynomial-time approximation schemes for Euclidean TSP and other geometric

problems. *Proceedings of 37th IEEE Symp. on Foundations of Computer Science*, pp 2-12, 1996.

[5] S. Arora. Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. *Proceedings of 38th IEEE Symp. on Foundations of Computer Science*, 1997.

[6] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317-331, 1997.

[7] S. Arora, A. Frieze, and H. Kaplan. New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems. *Proceedings of the Thirty Seventh Annual Symposium on the Foundations of Computer Science,* IEEE, 1996, pages 21–30.

[8] S. Arora, D. Karger, and M. Karpinski. Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$-Hard Problems. *Proceedings of the Twenty Seventh Annual Symposium on the Theory of Computing,* ACM, 1995.

[9] S. Arora and C. Lund. Hardness of approximations. In [75].

[10] 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.

[11] S. Arora, R. Motwani, S. Safra, M. Sudan, and M. Szegedy. PCP and approximation problems. *Unpublished note*, 1992.

[12] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. To appear *Journal of the ACM.* Preliminary version in *Proceedings of the Thirty Third Annual Symposium on the Foundations of Computer Science,* IEEE, 1992.

[13] S. Arora and M. Sudan. Improved low degree testing and its applications. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1997

[14] G. Ausiello, A. D'Atri, and M. Protasi. On the structure of combinatorial problems and structure preserving reductions. In *Proc. 4th Intl. Coll. on Automata, Languages and Programming*, 1977.

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

[16] G. Ausiello, A. Marchetti-Spaccamela, and M. Protasi. Toward a unified approach for the classification of NP-complete optimization problems. *Theoretical Computer Science*, 12:83–96, 1980.

[17] L. Babai. Trading group theory for randomness. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 421–429, 1985.

[18] L. Babai. Transparent proofs and limits to approximations. In *Proceedings of the First European Congress of Mathematicians*. Birkhauser, 1994.

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

[20] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 21–31, 1991.

[21] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254-276, 1988.

[22] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *JACM* 41(1):153–180, 1994. Prelim. version in *Proceedings of the Twenty Third Annual Symposium on the Foundations of Computer Science*, IEEE, 1982.

[23] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. *Proceedings of the Seventh Annual Symposium on Theoretical Aspects of Computer Science,* Lecture Notes in Computer Science Vol. 415, Springer Verlag, 1990.

[24] M. Bellare. Interactive Proofs and approximation: Reductions from two provers in one round. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems.* IEEE Computer Press, 1993. Preliminary version: IBM Research Report RC 17969 (May 1992).

[25] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi and M. Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory* 42(6):1781-1795, November 1996.

[26] M. Bellare, O. Goldreich, and M. Sudan. Free bits and non-approximability– towards tight results. In *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995. Full version available from ECCC.

[27] 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.

[28] M. Bellare and P. Rogaway. The complexity of approximating a nonlinear program. *Journal of Mathematical Programming B*, 69(3):429-441, 1995. Also in *Complexity of Numerical Optimization*, Ed. P. M. Pardalos, World Scientific, 1993.

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

[30] M. Ben-or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi prover interactive proofs: How to remove intractability assumptions. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 113–121, 1988.

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

[32] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.

[33] A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum bandwidth and other problems These proceedings.

[34] M. Blum and S. Kannan. Designing programs that check their work. In *Proc. 21st ACM Symp. on Theory of Computing*, pages 86–97, 1989.

[35] M. Blum, M. Luby, and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS* 47(3):549–595, 1993. *Prelim. Version in Proceedings of the Twenty Second Annual Symposium on the Theory of Computing*, ACM, 1990.

[36] R. B. Boppana and M. M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, June 1992.

[37] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In J.F. Traub, editor, *Symposium on new directions and recent results in algorithms and complexity*, page 441. Academic Press, NY, 1976.

[38] A. Condon. The complexity of the max-word problem and the power of one-way interactive proof systems. *Computational Complexity*, 3:292–305, 1993.

[39] A. Condon, J. Feigenbaum, C. Lund and P. Shor. Probabilistically Checkable Debate Systems and Approximation Algorithms for PSPACE-Hard Functions. *Proceedings of the Twenty Fifth Annual Symposium on the Theory of Computing*, ACM, 1993.

[40] A. Condon, J. Feigenbaum, C. Lund and P. Shor. Random debaters and the hardness of approximating stochastic functions. *SIAM Journal on Computing*, 26(2):369-400, April 1997.

[41] S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pages 151–158, 1971.

[42] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *JCSS* 51(3):511–522, 1995.

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

[44] P. Crescenzi, V. Kann, R. Silvestri, and L. Trevisan. Structure in aproximation classes. *Proc. 1st Combinatorics and Computing conference*, pages 539–548. LNCS 959, Springer Verlag, 1995.

[45] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93:241–262, 1991.

[46] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. Karp, editor, *Complexity of Computer Computations*, pages 43–73. AMS, 1974.

[47] U. Feige. A threshold of $\ln n$ for approximating set cover. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing*, ACM, 1996, pages 314–318.

[48] U. Feige. Approximating the bandwidth via volume respecting embeddings. *These proceedings.*

[49] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques *Journal of the ACM*, 43(2):268–292, 1996. *Preliminary version: Approximating clique is almost NP-complete, Proceedings of the Thirty Second Annual Symposium on the Foundations of Computer Science, IEEE, 1991.*

[50] 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.

[51] U. Feige and J. Kilian. Zero Knowledge and the Chromatic Number (Preliminary Version). *Proceedings of the Eleventh Annual Conference on Complexity Theory*, IEEE, 1996.

[52] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems. In *Proc. 24th ACM Symp. on Theory of Computing*, pages 733–741, 1992.

[dlV94] W. Fernandez de la Vega. MAXCUT has a randomized approximation scheme in dense graphs. *Random Structures & Algorithms*, vol. 8, 1996. Prelim. manuscript, October 1994.

[53] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1+\epsilon$ in linear time. *Combinatorica*:1(4), 349–355, 1981.

[54] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *Proceedings of the 3rd Conference on Structure in Complexity Theory*, pages 156–161, 1988.

[55] A. Frieze and R. Kannan. The Regularity Lemma and Approximation Schemes for Dense Problems. *Proceedings of the Thirty Seventh Annual Symposium on the Foundations of Computer Science*, IEEE, 1996, pages 12–20.

[56] M. Fürer. Improved hardness results for approximating the chromatic number. In *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995.

[57] M. R. Garey, R. L. Graham, and J. D. Ullman. Worst case analysis of memory allocation algorithms. In *Proceedings of the Fourth Annual Symposium on the Theory of Computing*, ACM, 1972, pages 143–150.

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

[59] M. Garey and D. Johnson. "Strong" NP-completeness results: motivation, examples and implications. *Journal of the ACM*, 25:499-508, 1978.

[60] M. R. Garey and D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness.* W. H. Freeman, 1979.

[61] N. Garg, V.V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)-cut theorems and their applications. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 698 –707, 1993.

[62] M. Goemans and D. Williamson. A 0.878 approximation algorithm for MAX-2SAT and MAX-CUT. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 422–431, 1994.

[63] M. Goemans and D. Williamson. The primal-dual method for approximation algorithms and its applications to network design problems. In [75].

[64] O. Goldreich. Probabilistic proof systems. Technical Report RS-94-28, Basic Research in Computer Science, Center of the Danish National Research Foundation, September 1994. *Proceedings of the International Congress of Mathematicians*, Birkhauser Verlag 1994.

[65] O. Goldreich. A Taxonomy of Proof Systems. In *Complexity Theory Retrospective II*, L.A. Hemaspaandra and A. Selman (eds.), Springer-Verlag, New York, 1997.

[66] O. Goldreich, S. Goldwasser and D. Ron. Property Testing and its Connection to Learning and Approximation. *Proceedings of the Thirty Seventh Annual Symposium on the Foundations of Computer Science,* IEEE, 1996, pages 339–348.

[67] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proofs. *SIAM J. on Computing*, 18:186–208, 1989.

[68] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symp. on Theory of Computing*, pages 218–229, 1987.

[69] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.

[70] M. Grötschel, L. Lovász, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer Verlag, Berlin. 1988.

[71] L. Hall. Approximation algorithms for scheduling. In [75].

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

[73] J. Håstad. Clique is Hard to Approximate within $n^{1-\epsilon}$. *Proceedings of the Thirty Seventh Annual Symposium on the Foundations of Computer Science,* IEEE, 1996, pages 627–636.

[74] J. Håstad. Some optimal inapproximability results. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1997, pages 1–10.

[75] D. Hochbaum, ed. Approximation Algorithms for NP-hard problems. PWS Publishing, Boston, 1996.

[76] D. Hochbaum and D. Shmoys. Using dual approximation algorithms for scheduling problems: practical and theoretical results. *Journal of the ACM* 34(1):144–162, 1987.

[77] D. Hochbaum and D. Shmoys. A best-possible heuristic for the k-center problem. *Math. of Op. Res.* 10(2):180–184, 1985.

[78] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subsets problems. *JACM*, 22(4):463–468, 1975.

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

[80] D. S. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 13:502–524, 1992.

[81] V. Kann. *On the approximability of NP-complete optimization problems.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1992.

[82] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Proceedings of the Thirty Fifth Annual Symposium on the Foundations of Computer Science,* IEEE, 1994, pages 2–13.

[83] H. Karloff and U. Zwick. A 7/8-Approximation Algorithm for MAX 3SAT? *Proceedings of the Thirty Eigth Annual Symposium on the Foundations of Computer Science,* IEEE, 1997 .

[84] N. Karmarkar and R.M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proc.* 23^{rd} *IEEE Symposium on Foundations of Computer Science*, pp 312–320, 1982.

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

[86] R. M. Karp. Probabilistic analysis of partitioning algorithms for the TSP in the plane. *Math. Oper. Res.* **2** (1977), 209-224.

[87] D. Karger, R. Motwani, and G.D.S. Ramkumar. On approximating the longest path in a graph. In *Proceedings of Workshop on Algorithms and Data Structures*, pages 421–430. LNCS (Springer-Verlag), v. 709, 1993.

[88] 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.

[89] S. Khanna, R. Motwani, M. Sudan, and U. Vazirani. On syntactic versus computational views of approximability. *Proceedings of the Thirty Fifth Annual Symposium on the Foundations of Computer Science,* IEEE, 1994, pages 819–836.

[90] S. Khanna, M. Sudan, L. Trevisan. Constraint satisfaction: the approximability of minimization problems. *Proceedings of the Twelfth Annual Conference on Complexity Theory*, IEEE, 1997,

pages 282–296.

[91] S. Khanna, M. Sudan, and D. Williamson. A complete classification of the approximability of maximization problems derived from Boolean constraint satisfaction. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1997, pages 11-20.

[92] M. Kiwi, C. Lund, A. Russell, D. Spielman, and R. Sundaram Alternation in interaction. *Proceedings of the Ninth Annual Conference on Structure in Complexity Theory*, IEEE, 1994, pages 294–303.

[93] P. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. In *Proc. of the 6th Conference on Structure in Complexity Theory*, pages 353–366, 1991.

[94] G. Kolata. New shortcut found for long math proofs. *New York Times*, April 7, 1992.

[95] R. Ladner. On the Structure of Polynomial Time Reductibility. *JACM*, 22(1):155–171, 1975.

[96] D. Lapidot and A. Shamir. Fully parallelized multi prover protocols for NEXPTIME. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 13–18, 1991.

[97] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys. The traveling salesman problem. John Wiley, 1985

[98] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 422–431, 1988.

[99] L. Levin. Universal'nyĭe perebornyĭe zadachi (universal search problems : in Russian). *Problemy Peredachi Informatsii*, 9(3):265–266, 1973.

[100] L. Levin. Average case complete problems. *SIAM Jour. of Computing*, 1986, Vol. 15, pp. 285–286. Extended abstract appeared in *Proc. 16th ACM Symp. on Theory of Computing*, 1984.

[101] R. Lipton. New directions in testing. In *Distributed Computing and Cryptography*, J. Feigenbaum and M. Merritt, editors. Dimacs Series in Discrete Mathematics and Theoretical Computer Science, 2. AMS 1991.

[102] R. Lipton and R. Tarjan. Applications of a planar separator theorem. *SIAM J. Computing*, 9(3):615–627, 1980.

[103] N. Linial, E. London, and Y. Rabinovich. The Geometry of Graphs and Some of its Algorithmic Applications. *Combinatorica*, 1995.

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

[105] C. Lund. *The Power of Interaction.* MIT Press, Cambridge, Mass., 1992.

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

[107] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proceedings of International Colloquium on Automata, Languages and Programming, ICALP*, pages 40–51, 1993.

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

[109] J. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: Part II- A simple PTAS for geometric k-MST, TSP, and related problems. Preliminary manuscript, April 30, 1996. *To appear in SIAM J. Computing.*

[110] A. Panconesi and D. Ranjan. Quantifiers and approximation. In *Proc. of the 22nd ACM Symp. on the Theory of Computing*, pages 446–456, 1990.

[111] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991. *Prelim. version in Proc. ACM STOC 1988.*

[112] C. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993.

[113] R. Raz. A parallel repetition theorem. *Proceedings of the Twenty Seventh Annual Symposium on the Theory of Computing,* ACM, 1995.

[114] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. *Proceedings of the Twenty Eighth Annual Symposium on the Theory of Computing,* ACM, 1997.

[115] R. Rubinfeld. *A Mathematical Theory of Self-Checking, Self-Testing and Self-Correcting Programs.* Ph.D. thesis, U.C. Berkeley, 1990.

[116] R. Rubinfeld and M. Sudan. Testing polynomial functions efficiently and over rational domains. In *Proc. 3rd Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 23–32, 1992.

[117] S. Sahni. Approximate algorithms for the 0/1 knapsack problem. *Journal of the ACM*, 22(1):115–124, 1975.

[118] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23:555–565, 1976.

[119] T. J. Schaefer. The Complexity of Satisfiability Problems. In *Proc. 10th ACM STOC*, pp 216–226, 1978.

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

[121] D. Shmoys. Cut problems and their application to divide-and-conquer. In [75].

[122] H. U. Simon. On approximate solutions for combinatorial optimization problems. *SIAM J. Algebraic Discrete Methods*, 3:294–310, 1990.

[123] M. Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, U.C. Berkeley, 1992.

[124] A. Wigderson. Improving the Performance Guarantee for Approximate Graph Coloring. *JACM* 30(4):729–735, 1983.

[125] M. Yannakakis. The effect of a connectivity requirement on the complexity of of maximum subgraph problems. *Journal of the ACM*, 26:618–630, 1979.

[126] M. Yannakakis. Edge deletion problems. *SIAM Journal of Computing*, 10:77–89, 1981.

[127] M. Yannakakis. On the approximation of maximum satisfiability. In *Proceedings of 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1–9, 1992.

[128] D. Zuckerman. NP-complete problems have a version that's hard to approximate. In *8th Structure in Complexity Theory Conf.*, pages 305–312, 1993.