Published in ACM TRANSACTIONS ON ALGORITHMS 1:1, 160-176 (2005)

The NP-Completeness Column

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

Abstract. This is the 24th 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]." This edition of the column describes the history and purpose of the column and the status of the open problems from [G&J] and previous columns.

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: NP-completeness, open problems, primality testing, perfect graphs, coding theory, lattice bases

1. A BELATED REVIVAL

With this article, I resume a long-dormant column on NP-completeness whose first 23 editions appeared in *J. Algorithms* from 1981 through 1992. When the column first appeared, just two and a half years after the publication of [G&J], its main purpose was to provide timely additions and updates to the list of NP-complete and open problems at the end of that book. As the column evolved, however, it tended to devote more of its effort to providing brief reports and tutorials on new theoretical developments related to NP-completeness, covering such topics as Levin's concept of "random NP" [Col 11, 1984], the complexity of "uniqueness" [Col 15, 1985], zero-knowledge proofs [Col 21, 1988], and the PCP theorem [Col 23, 1992]. The revived column will contain material of both types, and in addition will provide pointers to other relevant sources of information as they appear, including books, tutorial articles, and websites.

The revival of the column was inspired in part by the creation of the new *ACM Transactions on Algorithms* as a successor to *J. Algorithms*. In addition, I hope that the research I do in preparing the column will help me make progress on a planned 2nd edition of [G&J]. Much has happened in the 13 years since the last

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

Table I. The current status of the open problems from [G&J] and previous columns.

Problem Name	Source	Status	Covered in
GRAPH ISOMORPHISM	[G&J]	Open	–
SUBGRAPH HOMEOMORPHISM (FOR A FIXED GRAPH H)	[G&J]	P	[Col 19, 1987]
GRAPH GENUS	[G&J]	NPC	[Col 21, 1988]
CHORDAL GRAPH COMPLETION	[G&J]	NPC	[Col 1, 1981]
CHROMATIC INDEX	[G&J]	NPC	[Col 1, 1981]
PARTIAL ORDER DIMENSION	[G&J]	NPC	[Col 1, 1981]
PRECEDENCE CONSTRAINED 3-PROCESSOR SCHEDULING	[G&J]	Open	–
LINEAR PROGRAMMING	[G&J]	P	[Col 1, 1981]
TOTAL UNIMODULARITY	[G&J]	P	[Col 1, 1981]
SPANNING TREE PARITY PROBLEM	[G&J]	P	[Col 1, 1981]
COMPOSITE NUMBER	[G&J]	P	This Column
MINIMUM LENGTH TRIANGULATION	[G&J]	Open	–
IMPERFECT GRAPH	[Col 1, 1981]	P	This Column
GRAPH THICKNESS	[Col 2, 1982]	NPC	[Col 5, 1982]
EVEN COVER (MINIMUM WEIGHT CODEWORD)	[Col 3, 1982]	NPC	This Column
"UNRESTRICTED" TWO-LAYER CHANNEL ROUTING	[Col 5, 1982]	Open	–
GRACEFUL GRAPH	[Col 6, 1983]	Open	–
ANDREEV'S PROBLEM	[Col 17, 1986]	Open	–
SHORTEST VECTOR IN A LATTICE	[Col 18, 1986]	"NPC"	This Column

column appeared (and the 26 years since the first edition of [G&J]). As with the very first column [Col 1, 1981], this edition of the column surveys developments with respect to the open problem list in [G&J], this time augmenting the coverage to include the open problems highlighted in previous columns. Table I summarizes the current status of all these problems. Eight of the twelve open problems from [G&J] and one of the seven open problems from the columns had been resolved by 1992, and their resolutions were covered in previous columns. Since then one of the four open problems from [G&J] and two of the open problems from the columns have been resolved, and one of the column problems has been partially resolved (in a sense to be explained later). Section 2 will cover the resolved and partially resolved problems, while Section 3 will discuss the problems that remain open. The next column will likely cover hardness-of-approximation results and the complexity conjectures on which they rely. Suggestions of topics and results to be covered by future columns are welcome.

While readers await the next column, they might wish to investigate some of the many other sources that now provide information about developments in the field.

SIGACT News has been running a Computational Complexity column moderated by Lane Hemaspaandra since 1993, available in `.pdf` format from the ACM Digital Library (`portal.acm.org/dl.cfm`). Although the column covers a wide range of topics, many are directly relevant to NP-completeness. In particular, the 46th edition, which appears in the March 2005 issue and was written by guest columnist Scott Aaronson, is a delightful survey of the wide variety of proposals for using physical processes to obtain exponential speedups over classical Turing machines and thus solve NP-complete problems efficiently. It covers a wide variety of suggestions, from soap bubbles to various exploitations of quantum mechanics, and explains why each is unlikely to work.

A second relevant column has been appearing regularly in the *Bulletin of the EATCS* since 1987. Originally entitled "The Structural Complexity Column" and moderated by Juris Hartmanis, it morphed in 1997 into "The Complexity Column," moderated first by Eric Allender, then by Lance Fortnow, and currently by Jacobo Torán. An index and downloadable `.pdf` versions of recent editions are available from `http://theorie.informatik.uni-ulm.de/Personen/toran/beatcs`. The October 2004 edition presents an interesting survey by Jörg Flum and Martin Grohe on the relatively new concept of "fixed parameter tractability" and its associated complexity classes (see also Downey and Fellows [1999]).

There are also several regularly updated websites/blogs that may be of interest. Lance Fortnow has been writing a "Computational Complexity" weblog since August 2002, with daily updates. This is where many of us first hear about major results, and where we could even find technical reviews of the early episodes of the CBS television series *Numb3rs*, in the second episode of which the question of P versus NP played a central role. A webpage maintained by Scott Aaronson, "The Complexity Zoo" (`http://www.complexityzoo.com`) provides notation and definitions for hundreds of complexity classes, both common and obscure, along with some of the facts known about them. Pierluigi Crescenzi and Viggo Kann maintain an "NP Optimization Problem" website (`http://ww.nada.kth.se/~viggo/problemlist/compendium.html`), which collects hardness-of-approximation results, updated at least through March 2000. Gerhard Woeginger maintains "The P-versus-NP" page (`http://www.win.tue.nl/~gwoegi/P-versus-NP.htm`) with many interesting links plus a list of supposed proofs that P = NP (and P $\neq$ NP), all but one of which appeared since the brief survey of such claims in [Col 20, 1987]. Indeed, 17 of the 19 claims in the list have occurred since the year 2000, when the Clay Institute announced prizes of $1,000,000 for resolving the P versus NP question and six other famous problems in mathematics (see `http://www.claymath.org/millennium`).

Finally, readers who have not seen the earlier editions of this column (or have forgotten them) can now obtain them online. I recently compiled `.pdf` versions of all 23 columns from the original `troff` source files, and have posted them at `http//www.research.att.com/~dsj/columns`. These are inexact replicas, with slightly different pagination and some subpar equation formatting due to changes in the underlying typesetting software. In addition, the figures in Columns 5 and 16 had to be recreated because the software that originally generated them was no longer functional. (Oh, the joys of software evolution!) Definitive electronic versions of the columns can be found at *Science Direct* (`http://www.sciencedirect.com`),

where the *J. Algorithms* webpage makes scanned `.pdf` copies of all the columns available for a fee.

2. CLOSED AND ALMOST CLOSED PROBLEMS

In this section, I discuss the four open problems that have been resolved/partially resolved since the last column appeared. Two turned out to be polynomial-time solvable, and I'll start with these. As we shall see, although the results represent major theoretical breakthroughs, the running times for the discovered algorithms represent yet another strong challenge to the notion that "polynomial-time solvable" is a synonym for "efficiently solvable in practice."

COMPOSITE NUMBER [G&J]

INSTANCE: Positive integer N.

QUESTION: Are there positive integers $p, q > 1$ such that $N = p \cdot q$?

Comment: The search for an efficient algorithm for this problem goes back at least to 1801, when Gauss discussed it in his *Disquisitiones Arithmeticae*. When the NP-Completeness Column went on hiatus in 1992, it was known that the problem was unlikely to be NP-complete. The complementary problem PRIMES (is N a prime?) was known also to be in NP [Pratt 1975], and no problem in NP $\cap$ co-NP can be NP-complete unless the polynomial hierarchy PH collapses to NP. Also, there was an algorithm of Adleman et al. [1983] that tests for primality in time $n^{O(\log \log n)}$, where $n = \lceil \log N \rceil$ is the length of the binary representation of N (the input size). Such a running time would apply to all other NP-complete problems if primality testing were NP-complete. (Given that COMPOSITE NUMBER is in P if and only if PRIMES is in P, I shall for simplicity refer to the problem under consideration as "primality testing" in what follows.)

Still more was known about primality testing in 1992. In particular, it had recently been shown to be in ZPP, the set of all problems solvable in polynomial time by *Las Vegas* algorithms: randomized algorithms that for each instance either report the correct answer or say "I don't know," the latter occurring with probability less than 1/2. This followed from a result of Adleman and Huang [1992], who adapted the elliptic curve approach of Goldwasser and Kilian [1986] to show that primes could be recognized by a polynomial-time *Monte Carlo* algorithm: a randomized algorithm that says "no" if the input is composite and says "yes" at least half the time when the input is prime. The Las Vegas algorithm was obtained by running this algorithm in parallel with one of the previously-discovered polynomial-time Monte Carlo algorithms for recognizing composite numbers [Rabin 1976; 1980; Solovay and Strassen 1978]. Unfortunately, although the Monte Carlo algorithms for composite numbers run in time $O(n^2 \log n \log \log n)$, the running time for the Adleman-Huang algorithm is $O(n^{30})$ or worse (M.-D. Huang, personal communication, 2005). This means that the Las Vegas algorithm was far from practical, although its mere existence led many to conjecture that primality testing is in P.

Ten years later, in a preprint that was released on the Internet on August 6, 2002, and covered in *The New York Times* just two days later, Manindra Agrawal, Neeraj Kayal, and Nitin Saxena (AKS) verified this conjecture. The preprint [Agrawal et al. 2002] showed that primality testing is indeed in P, essentially by proving

that if a number is composite, one can find a certificate of this fact by examining only a polynomial number of candidates. Moreover, the running time of their algorithm was much faster than that of the randomized ZPP algorithm of Adleman and Huang [1992]. It was "only" $\tilde{O}(n^{12})$, where "$\tilde{O}(f(n))$" is a shorthand for "$O(f(n)\log^k(f(n)))$ for some constant k." For a high-level introduction to their algorithm and the primality problem in general, see Aaronson [2003].

There was one drawback to the initial AKS result, however. The algorithm they described worked for all sufficiently large N, but they couldn't say precisely how large *large* was. Thus for some B the general approach that uses the algorithm of Adleman et al. [1983] for $N < B$ and the algorithm of Agrawal et al. [2002] for $N \geq B$ would correctly test for primality and run in time $\tilde{O}(n^{12})$, but they couldn't say what value of B would suffice. Their proof relied on a number-theoretic result of Fouvry [1980], whose proof was in a sense nonconstructive. In particular, the computation of B appears to depend on the Extended Riemann Hypothesis (ERH). If the ERH is true (as most number theorists believe), then a value for B can effectively be computed (although if the ERH is true, there are better primality testing algorithms, such as the algorithm of Miller [1976], which runs in time $\tilde{O}(n^4)$ if implemented with the same fast subroutines as AKS (C. Pomerance, personal communication, 2005)). On the other hand, if the ERH is false, the value of B seems to depend on the size of the smallest counterexample to the ERH, which can't be known unless the ERH has actually been disproved.

Fortunately, a way around this nonconstructivity was found by the time AKS wrote the journal version of their paper [Agrawal et al. 2004]. In this version, the authors constructively design an $\tilde{O}(n^{10.5})$ algorithm using a suggestion from Hendrik Lenstra and reduce the time for the nonconstructive Fouvry-based algorithm to $\tilde{O}(n^{7.5})$. I don't have the space to go into the technical details, but the $\tilde{O}(n^{10.5})$ now relies only on elementary number theory and is easy to follow. For alternative coverages of the details of the algorithms with a bit more background, see Granville [2004] or the forthcoming 2nd Edition of Crandall and Pomerance [2001].

The progress did not stop here, however. Two further developments are also covered in the last two references. First, Lenstra and Pomerance [2005] produced an $\tilde{O}(n^6)$ constructive primality tester, which appears to be within polylog factors of the best possible for the type of approach introduced by AKS. Second, building on work of Berrizbeitia [2002], Bernstein [2004] and Mihăilescu and Avanzi [2003] independently developed randomized Las Vegas algorithms that run in $O(n^{4+o(1)})$ time. The time is a bit worse than $\tilde{O}(n^4)$, possibly something like $O(n^4(\log n)^{\log\log\log n})$ for the Bernstein algorithm (C. Pomerance, personal communication, 2005), but represents a major improvement over the Adleman-Huang approach mentioned at the beginning of this discussion, and might even be practical [Granville 2004]. An excellent website covering questions related to primality, including the state of the art for practical primality testing, is `http://primes.utm.edu`. Currently the best general-purpose algorithms are not AKS-based or known to obey polynomial worst-case time bounds, but instead build upon the older elliptic curve approach of Goldwasser and Kilian [1986].

Note, however, that even if practical, these primality testing algorithms do not address what is currently the most important problem about composite numbers:

how to factor them. When one of the above algorithms asserts that a number is composite rather than prime, it produces a certificate of compositeness, but not the actual factors. This is fortunate, since today much of electronic commerce uses the RSA encryption scheme [Rivest et al. 1978], and the security of that scheme depends on the assumption that factoring is hard. At present, it does appear to be. The running time bound for the current best general algorithm for factoring integers is still exponential, although the exponential is better than many we see. By a widely-believed heuristic argument the time bound is $O(2^{cn^{1/3}(\log n)^{2/3}})$ [Pomerance 1996], where $c \sim 1.902$ [Coppersmith 1993].

The above statement applies only to algorithms for classical computers. Shor [1997] showed that a quantum computer, if it could be built, would be able to factor integers in polynomial expected time. This at least suggests that factoring is not NP-hard, since to date no one has seen how to use quantum computers to solve NP-hard problems. Another reason why factoring is not likely to be NP-hard is that it can be solved in polynomial time with an oracle to a problem in NP ∩ co-NP, in particular the problem, given N and m, of whether N has a nontrivial factor smaller than m. Here the unique prime factorization of N provides a polynomial-time checkable certificate for both yes and no answers. The complexity of factoring thus remains a major open problem and a suitable replacement for COMPOSITE NUMBER in any future list.

IMPERFECT GRAPH [Col 1, 1981]

INSTANCE: Graph $G = (V, E)$.

QUESTION: Is G *not* a perfect graph, that is, is there a subset $V' \subseteq V$ such that the subgraph of G induced by V' has a chromatic number which is larger than its maximum clique size?

Comment: Perfect graphs are of wide interest in graph theory and combinatorial optimization. Entire books have been written about them (e.g., Golumbic [1980]), and many NP-hard problems can be solved in polynomial time when restricted to them. For example, the weighted versions of CHROMATIC NUMBER and CLIQUE are solvable in polynomial time for perfect graphs using the ellipsoid method [Grötschel et al. 1981a]. The algorithms for these problems are applicable to all graphs and either report that the graph is not perfect or return the correct answer (in which case the graph may or may not be perfect). Nevertheless, it would be useful to tell in advance whether a graph is perfect and hence such algorithms are guaranteed to work.

That the problem of recognizing graphs that are *not* perfect is in NP was first proved by Grötschel et al. [1981b]. A simpler proof is now possible, since membership in NP follows immediately from the famous "strong perfect graph conjecture," which recently became a theorem thanks to Maria Chudnowsky, Neil Robertson, Paul Seymour, and Robin Thomas [Chudnovsky et al. 2005c]. This theorem says that a graph is perfect if and only if neither it nor its complement contains an "odd hole," that is, an induced subgraph that is an odd cycle of length 5 or more.

Hsu [1987] showed that perfect *planar* graphs could be recognized in polynomial time. We now know that the general problem is also in P. The proof exploits the strong perfect graph theorem, which reduces the problem to that of testing for

the existence of an odd hole. It also requires major new ideas and a somewhat different set of authors: Chudnovsky, Gérard Cornuéjols, Xinming Liu, Seymour, and Kristina Vušković [Chudnovsky et al. 2005a].

This result is a major breakthrough, but the algorithm itself has drawbacks. First, it runs in time $O(|V|^9)$. Second, it isn't guaranteed to find an odd hole in G when one exists. When odd holes exist in both G and its complement, there is no guarantee as to which type of hole the algorithm will find. Thus the problem of determining whether a given graph G contains an odd hole remains open.

In contrast, the problem of telling whether G contains an even hole *can* be solved in polynomial time. This was first shown by Conforti et al. [2002], whose "cleaning" technique was exploited in the perfect graph recognition algorithm of Chudnovsky et al. [2005a]. The running time for the original even hole detection algorithm (not quantified by Conforti et al. [2002]) was estimated to be $O(|V|^{40})$ by Chudnovsky et al. [2005b], who found a somewhat simpler algorithm and reduced the time bound to $O(|V|^{31})$. As to further improvements, Chudnovsky et al. [2005b] sketch ideas that get the time down to $O(|V|^{15})$, but that is the best known so far. In addition, the complexity of the problem of finding the *smallest* even hole when one exists remains open, and telling whether an even hole containing a specified vertex exists is NP-complete, as is the analogous problem for odd holes [Bienstock 1991]. It is thus perhaps fortunate that there currently appear to be no practical applications for finding even (or odd) holes. The next resolved problem also has to do with parity issues and does have practical import.

EVEN COVER/MINIMUM WEIGHT CODEWORD [Col 3, 1982]

INSTANCE: Collection C of subsets of a given finite set X, positive integer K.

QUESTION: Is there a nonempty subcollection $C' \subset C$ with $|C'| \leq K$, such that each element of X is in an even number (possibly zero) of sets from C'?

Comment: This problem is solvable in polynomial time if no $c \in C$ has size greater than 2 [JáJá and Venkatesan 1981] or if $K \geq |C|$, in which case it reduces to solving a system of linear equations over GF(2). It is a combinatorial restatement of the classic coding theory problem that asks whether the minimum weight of a non-zero codeword in a binary linear code is K or less. Such a code is defined by a binary $m \times n$ "parity check" matrix H. The codewords are all those binary vectors x such that $Hx^t = \mathbf{0}$ (the all-zero vector), and the weight of a codeword is its "Hamming weight," the number of nonzeros that it contains. (The rows of H correspond to the elements in X and the columns correspond to the sets in C.)

The minimum weight codeword problem is closely related to the nearest codeword problem, called DECODING OF LINEAR CODES in [G&J] and proved NP-complete by Berlekamp et al. [1978]. In this latter problem, the instance is augmented by an m-bit binary vector s, and we ask if there is a binary vector x with Hamming weight K or less such that $Hx^t = s$. If we want to obtain the maximum likelihood decoding of a received codeword y, we must solve the minimization version of this problem for $s = Hy^t$. (In practice, we often settle for the easier task of bounded-error decoding.) Guruswami and Vardy [2005] recently showed that the arbitrary-finite-field version of the maximum likelihood decoding problem remains NP-complete even when restricted to the important special case of Reed-Solomon codes. Such

codes have many real-world applications, including the encoding scheme used on compact discs. The NP-completeness proof unfortunately requires that the field size q be exponential in n, whereas in applications we typically have $q = n + 1$.

Returning to the original domain of binary codes, Berlekamp et al. [1978] also showed that it is NP-complete to tell if there is a codeword with Hamming weight *exactly* w, a much closer variant on the minimum weight codeword problem. The original problem, however, remained open until 1997, when Alexander Vardy proved that it too is NP-complete [Vardy 1997]. The proof is decidedly nontrivial. Starting from the DECODING OF LINEAR CODES problem, Vardy first presents a transformation to FINITE-FIELD SUBSET SUM: Given an integer $m > 0$, distinct elements $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta \in \mathrm{GF}(2^m)$, and a positive integer K, is there a subset of K or fewer elements of $\{\alpha_i : 1 \leq i \leq n\}$ which sums to β in $\mathrm{GF}(2^m)$? This problem is then transformed to a version of MINIMUM WEIGHT CODEWORD for codes over $\mathrm{GF}(2^m)$, and this is reduced to the original problem by a complicated argument. See [Vardy 1997] for the full details and additional background on the problem.

There have also been a variety of results about the hardness of *approximating* the minimum weight codeword. Dumer et al. [2003] proved that the minimum weight codeword is hard to approximate within any constant factor, assuming NP $\not\subseteq$ RP, a slightly stronger hypothesis than NP $\not\subseteq$ P, but one that is still widely believed. They proved still stronger non-approximability bounds under the stronger but still plausible hypothesis that NP $\not\subseteq$ RQP, where RQP stands for "random quasi-polynomial time," the set of all languages recognizable by Monte Carlo algorithms in time $O(2^{\log^k n})$ for some k. Under this assumption, the minimum weight codeword is hard to approximate to within $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$. Both results hold for linear codes over any finite field $GF(q)$, $q \geq 2$, and for randomized algorithms that are only guaranteed to meet the bound with probability exceeding $1/2$.

Somewhat stronger results hold for the nearest codeword problem, where the above inapproximability bounds hold under the weaker assumptions that NP $\not\subseteq$ P and NP $\not\subseteq$ QP, respectively [Arora et al. 1997]. We shouldn't expect *weaker* results for this problem than for MINIMUM WEIGHT CODEWORD, since there is an approximation-preserving polynomial-time Turing reduction from that problem to this one [Goldreich et al. 1999]. The current best attainable polynomial-time guarantees for both problems are ϵn for any fixed ϵ and $\epsilon n / \log n$ if randomization is allowed [Berman and Karpinski 2002].

The above results imply only that no universal polynomial-time algorithms can work for all codes. They thus leave open the possibility that efficient decoding algorithms might still exist for each individual code. One way of interpreting such a claim would be to say that there is a fixed polynomial p such that each code H has a Boolean decoding circuit C_H with size bounded by $p(n)$ where n is the codeword length. Note that we do not say anything about how easy it might be to construct C_H given H, just that the circuit exists. Hence this can be called "decoding with (unlimited) preprocessing." Unfortunately, it does not appear to be a likely possibility. As shown by Bruck and Naor [1990], the existence of such a set of size-bounded circuits for optimal decoding would imply that PH collapses to Σ_2^p. The same consequence follows even if all we want the circuits to do is approximate the distance to the nearest codeword to within a factor less than $5/3$

[Feige and Micciancio 2004]. This holds for linear codes over arbitrary finite fields; the inapproximability bound increases to 3 for binary codes [Regev 2004a].

The next problem is a close cousin to MINIMUM WEIGHT CODEWORD.

SHORTEST VECTOR IN A LATTICE [Col 18, 1986]

INSTANCE: Collection of vectors $v_1, \ldots, v_n$, each a member of $\mathbf{Q}^n$, integer $B > 0$.

QUESTION: Is there a nonzero vector $a = (a_1, \ldots, a_n) \in \mathbf{Z}^n$ such that if $x = \sum_{i=1}^n a_i v_i$, then the Euclidean length $||x|| = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$ is B or less?

Comment: In what follows, we shall use the standard abbreviation "SVP" for this problem. The "lattice" to which the problem name refers is the set of all integer combinations of the vectors v_i. The approximation algorithm story for this problem, although similar to that for MINIMUM WEIGHT CODEWORD, differs in significant ways. For the earlier problem, even the most trivial approximation algorithm will provide a factor-of-n guarantee, and although that seems bad, here things can be much worse. Fortunately, even what seem like horrible approximation algorithms have proved useful. The now-classical LLL basis reduction algorithm of Lenstra et al. [1982] could only guarantee a factor of $2^{(n-1)/2}$ and yet it and its variants have been key ingredients in polynomial-time algorithms for factoring polynomials [Kannan et al. 1988; Schönhage 1984], breaking of certain public-key cryptosystems [Brickell 1985], solving integer programs in fixed dimensions [Lenstra 1983] and solving the "simultaneous Diophantine approximation" problem [Lagarias 1985].

As of 1992, however, we didn't even know if one needed to settle for approximation algorithms. We did know the following: The problem is solvable in polynomial time if $B = 0$, as it then becomes just the problem of solving homogeneous linear equations over $\mathbf{Z}$. It becomes NP-hard if one replaces the Euclidean norm by the ℓ_∞ norm, i.e., $\max\{|x_i| : 1 \leq i \leq n\}$ [van Emde Boas 1981]. Also NP-complete is the related closest vector problem (CVP), in which one is in addition given a target vector y, and asked if an a exists such that $||\sum_{i=1}^n a_i v_i - y|| \leq B$, where the norm $||\cdot||$ is any ℓ_p metric for $p \geq 1$ [van Emde Boas 1981].

In 1998, Miklós Ajtai showed that SVP is NP-hard under randomized reductions [Ajtai 1998]. This is not quite as strong as NP-hardness. Once again the intractability of the problem depends on the hypothesis that NP $\not\subseteq$ RP rather than NP $\not\subseteq$ P. As we have seen, this is widely viewed as strong evidence, but the stronger evidence provided by a standard NP-completeness proof would be even better, and determining whether such a proof exists remains a significant open problem. For tutorials covering the technical details of Ajtai's proof and subsequent results, see Cai [1999] and Kumar and Sivakumar [2001].

SVP also appears to be hard to approximate, again based on hypotheses that are plausible but a bit weaker than NP $\not\subseteq$ P. See Table II, which summarizes the currently best results about the approximability of this problem, results which I'll discuss (and clarify) in the following paragraphs.

The first nonapproximability result was somewhat weak. Ajtai [1998] showed that assuming NP $\not\subseteq$ RP there is a constant k such that no randomized polynomial-time algorithm could be guaranteed to get within a factor of $1 + 1/2^{n^k}$, a factor

Table II. Complexity results about approximation guarantees for SVP

Guarantee	Result	Reference
$O(1)$	Ptime only if NP $\subseteq$ RP	[Khot 2004]
$2^{(\log n)^{1/2-\epsilon}}$	Ptime only if NP $\subseteq$ RQP	[Khot 2004]
$O\left(\sqrt{n/\log n}\right)$	Not NP-hard unless PH = Σ_p^2	[Goldreich and Goldwasser 2000]
$O\left(\sqrt{n}\right)$	Not NP-hard unless NP = co-NP	[Aharonov and Regev 2004]
$O\left(n\sqrt{\log n}\right)$	Worst case hardness implies average case hardness	[Micciancio and Regev 2004]
$2^{O\left(\frac{n \log\log n}{\log n}\right)}$	BPP	[Ajtai et al. 2001]
$2^{O\left(\frac{n(\log\log n)^2}{\log n}\right)}$	P	[Schnorr 1987]

that unfortunately approaches 1 as n, the dimension of the lattice, approaches ∞. Significant improvements quickly followed, however. Micciancio [2001] showed that, assuming NP $\not\subseteq$ RP, no constant ratio less than $\sqrt{2}$ can be guaranteed by a randomized polynomial-time algorithm, and more recently Khot [2004] has shown that *no* constant ratio guarantee is possible under this assumption. Under the stronger assumption that NP $\not\subseteq$ RQP, no randomized polynomial-time algorithm can guarantee a ratio of $2^{\log^{1/2-\epsilon} n}$ for any constant $\epsilon > 0$ [Khot 2004]. Note, however, that this is not quite as strong as the analogous result we saw above for MINIMUM WEIGHT CODEWORD, where the exponent is $\log^{1-\epsilon} n$.

Moreover, the gap between what has been proved difficult and what is achievable remains exponential. The $O(2^{(n-1)/2})$ performance guarantee of the original LLL algorithm has been bettered, but to date the best polynomial-time guarantee is for an algorithm of Schnorr [1987] whose guarantee is exponential in $O(n(\log\log n)^2/\log n)$ [Ajtai 2003]. The bound improves when randomization is allowed, but only to exponential in $O(n \log\log n/\log n)$ [Ajtai et al. 2001].

What is the likelihood that we can close the gap? It appears that our only hope may be to find better algorithms, since there are serious technical limitations on our ability to prove stronger hardness-of-approximation results. Aharonov and Regev [2004], improving on results of Lagarias et al. [1990], Banaszczyk [1993], and Goldreich and Goldwasser [2000], have shown the following: If there were deterministic or randomized polynomial transformations that proved SVP is NP-hard to approximate to within a factor of $c\sqrt{n}$ for sufficiently small $c > 0$, then NP = co-NP.

More specifically, here is what they proved. The standard technique for proving that a given minimization problem A is hard to approximate within a factor of r is to transform instances x of some NP-hard problem B to ordered pairs (y, t) where y is an instance of A and t is a potential solution value, such that $OPT(y) \leq t$ if the answer for x in B is yes, and $OPT(y) > rt$ if the answer for x in B is no. Thus a polynomial-time approximation algorithm with a worst-case ratio guarantee of r or less for A could be used to solve B in polynomial time. Similarly, if there is a polynomial-time nondeterministic Turing machine (NDTM) that on any pair (y, t) where y is an instance of A always accepts when $OPT(y) > rt$ and never accepts

when $OPT(y) \leq t$ (no matter what it did when $t < OPT(y) \leq rt$), then B is in co-NP. In essence, Aharonov and Regev [2004] simply show that there is a c such that such an NDTM exists for SVP when $r = c\sqrt{n}$.

In addition, the problem of approximating SVP to within the even smaller factor of $r = c\sqrt{n/\log n}$ cannot be NP-hard under polynomial transformations unless NP $\subseteq$ co-AM [Goldreich and Goldwasser 2000]. This has almost as dire consequences since, by Boppana et al. [1987], it would imply that the polynomial hierarchy would collapse to Σ_p^2. To prove it, Goldwasser and Goldreich exhibit a bounded-round interactive proof system that will always convince the (polynomial-time) verifier when $OPT(y) > rt$ and convinces with probability less than $1/2$ when $OPT(y) \leq t$. By a result of Cai and Nerurkar [2000], both this and the previous result extend to rule out proofs of NP-hardness that use polynomial-time Turing reductions, assuming they show how to solve B using a subroutine for A that always gives the right answer when $OPT(y) \leq t$ or $OPT(y) > rt$.

It would be nice if these results did not hold, since there would be some wonderful consequences if we could prove that SVP were hard to approximate to within slightly larger factors. In particular, as first shown by Ajtai [1996], a variant on the MINIMUM WEIGHT CODEWORD problem would be hard *on average*, not just in the worst case. Here are some of the details. To make statements about "hardness on average" we must not only define the problem, but also specify the probability distribution over instances. For Ajtai's result (and subsequent improvements on it), the problem is one of finding a "small" solution to a set of modular equations. An instance consists of an $n \times m$ matrix A over $\mathbf{Z}_q$ for some n, m, and q, together with a rational number b, and the goal is to find a nonzero m-dimensional vector x over $\mathbf{Z}_q$ such that $Ax^t = \mathbf{0}$ and the Euclidean length of x is no more than b. Note that if we measured x by the number of nonzeros it contains, then we would have the MINIMUM WEIGHT CODEWORD problem over $GF(q)$. As to distributions, for each set of values n, m, q we take the uniform distribution over the set $X_{n,m,q}$ of all matrices A with those parameters, and fix $b = \sqrt{m}q^{n/m}$. By Minkowski's theorem, the desired vector will always exist, but the proof is nonconstructive and so finding one can still be hard.

What Ajtai proved was that there exist constants $\alpha, \beta, k \geq 1$ such that the following holds: If there were a randomized polynomial-time algorithm $\mathcal{A}$ that for each distribution $X_{n,\lceil \alpha n \log n \rceil, n^\beta}$, $n \geq 1$, finds the desired vector with probability $1/2$ (over both the instance distribution and its own random bits), then there would be a second randomized polynomial-time algorithm $\mathcal{B}$ that for any instance of SVP finds an estimate z for the length of the shortest vector v such that $||v|| \leq z \leq n^k||v||$. Conversely, if one cannot approximate SVP to within a factor of n^k for *all* instances in polynomial time, then one cannot obtain the abovementioned average-case behavior for the modular equation problem in polynomial time either.

Ajtai did not derive an explicit value for k in his paper, but it has been estimated to be about 10. A subsequent sequence of papers progressively improved the exponent, both by tightening some of the mathematical arguments and by deriving improved schemes for generating the collection of random calls to Algorithm $\mathcal{A}$ used by algorithm $\mathcal{B}$ [Cai and Nerurkar 1997; Micciancio 2004; Micciancio and Regev 2004]. Now all we need in order to get the desired average-case hardness is that for

some sufficiently large C there is no polynomial-time algorithm with a $Cn\sqrt{\log n}$ worst-case guarantee. Note that by the abovementioned results of Goldreich and Goldwasser [2000] and Aharonov and Regev [2004] this is still more than a factor of $\sqrt{n}$ worse than the largest guarantee that is likely to be NP-hard to obtain. It is however much closer than it was originally, and perhaps further improvements in the average-to-worst-case connection will yield something we *can* prove NP-hard.

One might also wonder whether the average-case hardness of this particular problem would be all that useful. As Ajtai [1996] shows, however, it would imply the existence of a one-way function that is hard to invert on average (a randomized modular variant on SUBSET SUM). This is a key cryptographic primitive, and at least in principle can be used to construct digital signatures. (For recent developments along this line, see Micciancio and Vadhan [2003].) And although this particular function is not a *trapdoor* one-way function and hence not usable for public-key encryption, Ajtai and Dwork [1997] showed that such a trapdoor function (and resulting public-key cryptosystem) could be derived, again with average-case hardness guaranteed, if a variant on SVP is hard in the worst-case. This variant is unfortunately a bit more restricted than the ones we have been considering. It is the *unique* shortest vector problem $u(n^k)$-SVP, in which we are given a lattice and asked to find the shortest vector v, but we only need to provide the correct answer in situations where any other vector u with $||u|| \leq n^k||v||$ is a constant multiple of v. Ajtai and Dwork [1997] show that if this problem is hard for $k = 8$ then their cryptosystem is secure. Modified cryptosystems with the exponent k as small as 1.5 have since been found [Regev 2004b], but note that if $u(n^k)$-SVP is hard, then so must be the more general problem of approximating the shortest vector to within a factor of n^k, which we have already seen is unlikely to be proved NP-hard for $k \geq 1/2$. Indeed Cai [1998] has shown that $u(n^k)$-SVP cannot be proved NP-hard by standard means even for $k = 1/4$ unless the polynomial hierarchy collapses. Thus it remains to be seen how or whether the worst-case/average-case connection might be exploited to prove security for such cryptosystems assuming P $\neq$ NP. (This is probably just an academic question anyway, given that the system encodes each bit of the message separately using many arithmetic operations and random bits and hence is not likely to be practical.)

Results similar to those listed in Table II have been proved for several variants of SVP in addition to $u(n^k)$-SVP. Consider the previously mentioned closest vector problem (CVP). This problem is provably at least as hard to approximate as SVP [Goldreich et al. 1999]. Indeed, stronger nonapproximability results have been proved for it. Dinur et al. [2003] have shown that, assuming NP $\not\subseteq$ P, CVP cannot be approximated to within a factor of $n^{c/\log\log n}$ for some constant $c > 0$, a stronger conclusion under a weaker hypothesis than that shown in Table II for SVP. (This bound was claimed by Dumer et al. [2003] to carry over to the nearest codeword problem, where it would have been much closer to the best upper bounds known, but the claim was based on incomplete information and has since been retracted (D. Micciancio, personal communication, 2005).) Achievable guarantees are similar to those for SVP (e.g., see Babai [1986]), and similar results have been obtained about non-NP-hardness [Aharonov and Regev 2004] and the value of preprocessing [Regev 2004a].

Two other variants for which similar results have been proved, together with sample references, are the "shortest n independent vectors" problem (Given a lattice basis A and a real number b, are there n linearly independent vectors in the lattice all of which have Euclidean length b or less?) [Ajtai 1996; Micciancio and Regev 2004] and the "covering radius" problem (Given a lattice with a rational basis in $\mathbf{R}^n$, what is the smallest r such that every point in $\mathbf{R}^n$ is within distance r of some lattice point?) [Micciancio 2004; Guruswami et al. 2005]. The latter reference also covers the analogous problem for codes.

3. STILL OPEN AFTER ALL THESE YEARS

As shown in Table I, three of the "Open Problems of the Month" from past columns remain open. As far as I can tell there has been no progress on any of them. In two cases this is perhaps no great loss. ANDREEV'S PROBLEM is a highly technical problem historically related to monotone circuit complexity and "UNRESTRICTED" TWO-LAYER CHANNEL ROUTING concerns a hypothetical VLSI design technology that time has long since passed by. The third open problem left from past columns, GRACEFUL GRAPH, does retain a certain cachet in the graph theory community and so might be worth revisiting. Readers interested in learning about this problem, or foolhardy enough to want to find out about the other two, are referred to the relevant columns listed in the table. I conclude this edition of the column by highlighting the three remaining open problems from [G&J].

GRAPH ISOMORPHISM [G&J]

INSTANCE: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

QUESTION: Are G_1 and G_2 isomorphic, i.e., is there a one-to-one onto function $f : V_1 \rightarrow V_2$ such that $\{u, v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$?

Comment: This problem remains open, but there is much evidence suggesting that it is not NP-complete. Although the best general algorithm currently known for it has running time $2^{O(\sqrt{n \log n})}$ [Babai and Luks 1983], we do not even know whether GRAPH ISOMORPHISM is logspace-hard for P. The strongest current hardness results are those of Toran [2004], which shows that GRAPH ISOMORPHISM is hard (under AC^0 transformations) for Nondeterministic Logspace (NL) and several other classes inside P. Furthermore, the problem of *counting* the number of isomorphisms between two graphs is polynomial-time equivalent to the problem of telling whether even one exists [Mathon 1979], whereas the counting versions of most NP-hard problems are #P-hard. Perhaps most significantly, if GRAPH ISOMORPHISM *were* NP-complete, then the polynomial hierarchy PH would collapse to Σ_2^p, an unlikely hypothesis that we have seen several times in this column. The collapse would occur because there is a bounded-round interactive proof for GRAPH NON-ISOMORPHISM [Goldreich et al. 1991], which places GRAPH ISOMORPHISM in the class co-AM. Thus if it were NP-complete we would have NP $\subseteq$ co-AM, and the already-mentioned result of Boppana et al. [1987] would apply.

Thus GRAPH ISOMORPHISM is often mentioned as a candidate for a problem whose complexity lies between that of P and the NP-complete problems. As such, it has engendered a new complexity class of its own: "GI" or the class of all problems that are polynomial-time equivalent to GRAPH ISOMORPHISM. Members include

the problem of determining whether two polytopes, given by their vertex-facet incidence matrices, are combinatorially isomorphic [Kaibel and Schwartz 2003] and the question of whether there is a homeomorphism between two 2-complexes [Shawe-Taylor and Pisanski 1994]. Much of the research on GRAPH ISOMORPHISM since [G&J] has concentrated on populating this class and on classifying special cases of GRAPH ISOMORPHISM as polynomial-time solvable or themselves "GI-complete." There is not space here to cover all such results, but an extensive early summary can be found in [Col 1, 1981].

PRECEDENCE CONSTRAINED 3-PROCESSOR SCHEDULING [G&J]

INSTANCE: Set T of unit length tasks, partial order $\prec$ on T, and a deadline $D \in Z^+$.

QUESTION: Can T be scheduled on 3 processors so as to satisfy the precedence constraints and meet the overall deadline D, i.e., is there a schedule $\sigma : T \rightarrow \{0, 1, \ldots, D-1\}$ such that $t \prec t'$ implies $\sigma(t) < \sigma(t')$ and such that for each integer i, $0 \leq i \leq D-1$, there are at most 3 tasks $t \in T$ for which $\sigma(t) = i$?

Comment: This problem remains open, even when "3" is replaced by any fixed $K > 3$. It remains a fundamental problem in scheduling theory, but there appears to have been no progress whatsoever on it since 1981. Thus readers interested in polynomial-time solvable subcases and related NP-hardness results can still get the up-to-date story in [Col 1, 1981].

MINIMUM LENGTH TRIANGULATION [G&J]

INSTANCE: Collection $C = \{(a_i, b_i) : 1 \leq i \leq n\}$ of pairs of integers giving the coordinates of n points in the plane, and a positive integer B.

QUESTION: Is there a triangulation of the set of points represented by C that has total "discrete-Euclidean" length B or less? Here a triangulation is a collection L of line segments, each joining two points in C and no two intersecting except possibly at their endpoints, such that L partitions the interior of the convex hull into triangular regions. The discrete-Euclidean length of a line segment joining (a_i, b_i) and (a_j, b_j) is given by $\left\lceil \left((a_i - a_j)^2 + (b_i - b_j)^2\right)^{1/2} \right\rceil$, and the total length of a triangulation is the sum of the lengths of its constituent line segments.

Comment: The discrete Euclidean metric is used in the problem statement to insure that the problem is in fact in NP. This problem remains open whether one uses the true or discrete Euclidean metric, and even if one considers the variant in which one allows the collection C to be augmented by a user-supplied set of "Steiner" points before the triangulation is constructed.

There is some question, however, as to how fundamental the problem really is. For applications involving triangulations, such as mesh generation, it is not the *length* of the triangulation that is important, but how well-behaved the triangles themselves are. The original, non-Steiner problem is polynomial-time solvable for several objective functions related to this goal. If you want to maximize the minimum internal angle over all triangles, the easy-to-construct Delaunay triangulation suffices, and in fact can be shown to provide a triangulation that is locally optimal with respect to "equiangularity" [Sibson 1978]. If you want to minimize the max-

imum angle, there is a polynomial-time algorithm for this as well [Edelsbrunner et al. 1992]. You can also in polynomial time find triangulations that minimize the maximum edge length, that maximize the minimum triangle height, minimize the maximum aspect ratio, and optimize a variety of other objective functions [Edelsbrunner and Tan 1993; Bern et al. 1993; D'Azevedo and Simpson 1989].

This said, there still does remain significant interest in the original problem, if only because of its status as an anointed open problem. On the assumption that it is NP-hard, a variety of researchers have worked on polynomial-time approximation algorithms for it, and we now have algorithms with bounded-ratio guarantees for both the Steiner and non-Steiner versions of the problem [Eppstein 1994; Levcopoulos and Krznaric 1998]. See [Bern and Eppstein 1997] for a survey.

REFERENCES

Aaronson, S. 2003. The prime facts: From Euclid to AKS. (Manuscript, available from `http://www.scottaaronson.com/writings/prime.pdf`.)

Adleman, L. M., and Huang, M.-D. 1992. *Primality Testing and Abelian Varieties over Finite Fields*. Lecture Notes in Mathematics, vol. 1512. Springer-Verlag, New York.

Adleman, L. M., Pomerance, C., and Rumely, R. S. 1983. On distinguishing prime numbers from composite numbers. *Ann. Math. 117*, 173–206.

Agrawal, M., Kayal, N., and Saxena, N. 2002. PRIMES is in P. (Manuscript, available from `http://www.cse.iitk.ac.in/primality.pdf`.)

Agrawal, M., Kayal, N., and Saxena, N. 2004. PRIMES is in P. *Ann. Math. 160*, 781–793.

Aharonov, D., and Regev, O. 2004. Lattice problems in NP ∩ coNP. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computing*. IEEE Computer Society, Los Alamitos, Calif., 362–371.

Ajtai, M. 1996. Generating hard instances of lattice problems. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. ACM, New York, 99–108. (Full version available from `http://eccc.uni-trier.de/eccc` as ECCC technical report TR96-007).

Ajtai, M. 1998. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*. ACM, New York, 10–19.

Ajtai, M. 2003. The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*. ACM, New York, 396–406.

Ajtai, M., and Dwork, C. 1997. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. ACM, New York, 284–293. (Full version available from `http://eccc.uni-trier.de/eccc` as ECCC technical report TR96-065).

Ajtai, M., Kumar, R., and Sivakumar, D. 2001. A seive algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*. ACM, New York, 601–610.

Arora, S., Babai, L., Stern, J., and Sweedyk, E. Z. 1997. The hardness of approximating 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, Los Alamitos, Calif., 1993, 724–733.)

Babai, L. 1986. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica 6*, 1–13.

Babai, L., and Luks, E. 1983. Canonical labelling of graphs. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*. ACM, New York, 171–183.

Banaszczyk, W. 1993. New bounds in some transference theorems in the geometry of numbers. *Math. Ann. 296*, 625–635.

Berlekamp, E. R., McEliece, R. J., and van Tilborg, H. C. A. 1978. On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory IT-24*, 384–386.

Berman, P., and Karpinski, M. 2002. Approximating minimum satisfiability of linear equations. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, Pa., 74–83.

Bern, M., Edelsbrunner, H., Eppstein, D., Mitchell, S., and Tan, T. S. 1993. Edge insertion for optimal triangulations. *Disc. Comput. Geom. 10*, 47–65.

Bern, M., and Eppstein, D. 1997. Approximation algorithms for geometric problems. In *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, Ed. PWS Publishing Company, Boston, Mass., 296–345.

Bernstein, D. J. 2004. Proving primality in essentially quartic random time. (Manuscript, available from `http://cr.yp.to/primetests/quartic-20041203.pdf`.)

Berrizbeitia, P. 2002. Sharpening PRIMES is in P for a large family of numbers. (Manuscript, available from `arxiv.org/abs/math.NT/0211334`.)

Bienstock, D. 1991. On the complexity of testing for odd holes and induced odd paths. *Disc. Math. 90*, 85–92. (Corrigendum in *Disc. Math. 102* (1992), 109.)

Boppana, R., Hastad, J., and Zachos, S. 1987. Does co-NP have short interactive proofs? *Inf. Process. Lett. 25*, 27–32.

Brickell, E. F. 1985. Breaking iterated knapsacks. In *Advances in Cryptology: Proceedings of the of CRYPTO 84*. Lecture Notes in Computer Science, vol. 196. Springer-Verlag, Berlin, Germany, 342–358.

Bruck, J., and Naor, M. 1990. The hardness of decoding linear codes with preprocessing. *IEEE Trans. Inf. Theory 36*, 381–385.

Cai, J.-Y. 1998. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. *Theor. Comput. Sci. 207*, 105–116.

Cai, J.-Y. 1999. Some recent progress on the complexity of lattice problems. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*. IEEE Computer Society, Los Alamitos, Calif., 158–179.

Cai, J.-Y., and Nerurkar, A. 1997. An improved worst-case to average-case connection for lattice problems. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computing*. IEEE Computer Society, Los Alamitos, Calif., 468–477.

Cai, J.-Y., and Nerurkar, A. 2000. A note on the non-NP-hardness of approximate lattice problems under general Cook reductions. *Inf. Process. Lett. 76*, 61–66.

Chudnovsky, M., Cornuéjols, G., Liu, X., Seymour, P., and Vušković, K. 2005a. Recognizing Berge graphs. *Combinatorica 25*, 143–186.

Chudnovsky, M., Kawarabayashi, K., and Seymour, P. 2005b. Detecting even holes. *J. Graph Theory 48*, 85–111.

Chudnovsky, M., Robertson, N., Seymour, P., and Thomas, R. 2005c. The strong perfect graph theorem. *Ann. of Math.*, to appear.

Conforti, M., Cornuéjols, G., Kapoor, A., and Vušković, K. 2002. Even hole-free graphs, part II: Recognition algorithm. *J. Graph Theory 40*, 238–266.

Coppersmith, D. 1993. Modifications to the number field sieve. *J. Graph Theory 6*, 169–180.

Crandall, R., and Pomerance, C. B. 2001. *Prime Numbers: A Computational Perspective*. Springer-Verlag, New York.

D'Azevedo, E. F., and Simpson, R. B. 1989. On optimal interpolation triangle incidences. *SIAM J. Sci. Statist. Comput. 10*, 1063–1075.

Dinur, I., Kindler, G., Raz, R., and Safra, S. 2003. Approximating CVP to within almost polynomial factors is NP-hard. *Combinatorica 23*, 205–243. (Preliminary version in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computing*, IEEE, Los Alamitos, Calif., 1998, 475–484.)

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer-Verlag, Heidelberg.

Dumer, I., Micciancio, D., and Sudan, M. 2003. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inf. Theory 49*, 22–37. (Preliminary version in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computing*, IEEE, Los Alamitos, Calif., 1999, 475–484.)

Edelsbrunner, H., and Tan, T. S. 1993. A quadratic time algorithm for the minmax length triangulation. *SIAM J. Comput. 22*, 527–551.

Edelsbrunner, H., Tan, T. S., and Waupotitsch, R. 1992. An $o(n^2 \log n)$ time algorithm for the minmax angle triangulation. *SIAM J. Sci. Statist. Comput. 13*, 994–1008. (Preliminary version in *Proceedings of the 6th ACM Sump. on Computational Geometry*, ACM, New York, 1990, 44–52.)

Eppstein, D. 1994. Approximating the minimum weight Steiner triangulation. *Disc. Comput. Geom. 11*, 163–191. (Preliminary version in *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, Penn., 1992, 48–57.)

Feige, U., and Micciancio, D. 2004. The inapproximability of lattice and coding problems with preprocessing. *J. Comput. Syst. Sci. 69,* 1, 45–67. (Preliminary version in *Proceedings of the 17th Annual IEEE Conference on Computational Complexity*, IEEE, Los Alamitos, Calif., 2002, 44–52.)

Fouvry, E. 1980. Théorème de Brun-Titchmarsh; application au théorème de Fermat. *J. Number Theory 12*, 128–138.

Goldreich, O., and Goldwasser, S. 2000. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci. 60*, 540–563. (Preliminary version in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1998, 1–9.)

Goldreich, O., Micali, S., and Widgerson, A. 1991. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM 38*, 691–729. (Preliminary version in *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computing*, IEEE, Los Alamitos, Calif., 1986, 174–187.)

Goldreich, O., Micciancio, D., Safra, S., and Seifert, J. P. 1999. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett. 71*, 55–61.

Goldwasser, S., and Kilian, J. 1986. Almost all primes can be quickly certified. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*. ACM, New York, 316–329. Journal version is [Goldwasser and Kilian 1999].

Goldwasser, S., and Kilian, J. 1999. Primality testing using elliptic curves. *J. ACM 46,* 4, 450–472.

Golumbic, M. C. 1980. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York.

Granville, A. 2004. It is easy to determine whether a given integer is prime. *Bull. AMS 42*, 3–38.

Grötschel, M., Lovász, L., and Schrijver, A. 1981a. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica 1*, 169–197.

Grötschel, M., Lovász, L., and Schrijver, A. 1981b. Polynomial algorithms for perfect graphs. *Ann. Disc. Math. 21*, 322–356.

Guruswami, V., Micciancio, D., and Regev, O. 2005. The complexity of the covering radius problem. *Computational Complexity 14*, 90–120. (Preliminary version in *Proceedings of the 19th IEEE Conference on Computational Complexity*, IEEE, Los Alamitos, Calif., 2004, 161–173.)

Guruswami, V., and Vardy, A. 2005. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. *IEEE Trans. Inf. Theory 51*, to appear.

Hsu, W.-L. 1987. Recognizing planar perfect graphs. *J. ACM 34*, 255–288.

JáJá, J., and Venkatesan, S. 1981. On the complexity of a parity problem related to coding theory. Report CS-81-5, Department of Computer Science, Pennsylvania State University.

Kaibel, V., and Schwartz, A. 2003. On the complexity of polytope isomorphism problems. *Graphs and Combinatorics 19*, 215–230.

Kannan, R., Lenstra, A. K., and Lovász, L. 1988. Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. *Math. Comp. 50*, 235–250.

KHOT, S. 2004. Hardness of approximating the shortest vector problem in lattices. In *Proceedings of the 45th IEEE Symposium on Foundations of Computing*. IEEE Computer Society, Los Alamitos, Calif., 126–135.

KUMAR, R., AND SIVAKUMAR, D. 2001. Complexity theory column 33 (Guest Column): Complexity of SVP – A reader's digest. *SIGACT News 32,* 3, 40–52.

LAGARIAS, J. C. 1985. The computational complexity of simultaneous Diophantine approximation problems. *SIAM J. Comput. 14*, 196–209.

LAGARIAS, J. C., LENSTRA, JR., H. W., AND SCHNORR, C.-P. 1990. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica 10*, 333–348.

LENSTRA, A. K., LENSTRA, H. W., AND LOVÁSZ, L. 1982. Factoring polynomials with rational coefficients. *Math. Ann. 261*, 515–534.

LENSTRA, H. W., AND POMERANCE, C. 2005. Primality testing with Gaussian periods. To appear.

LENSTRA, JR., H. W. 1983. Integer programming with a fixed number of variables. *Math. Oper. Res. 8*, 538–548.

LEVCOPOULOS, C., AND KRZNARIC, D. 1998. Quasi-greedy triangulations approximating the minimum weight triangulation. *J. Algorithms 27*, 303–338. (Preliminary version in *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, Penn., 1996, 392–401.)

MATHON, R. 1979. A note on the graph isomorphism counting problem. *Inf. Process. Lett. 8*, 131–132.

MICCIANCIO, D. 2001. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput. 30*, 2008–2035.

MICCIANCIO, D. 2004. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM J. Comput. 34*, 118–169. (Preliminary version in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2002, 609–618.)

MICCIANCIO, D., AND REGEV, O. 2004. Worst-case to average-case reductions based on Gaussian measures. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computing*. IEEE Computer Society, Los Alamitos, Calif., 372–381.

MICCIANCIO, D., AND VADHAN, S. 2003. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *Advances in Cryptology - CRYPTO 2003, Proceedings of the 23rd Annual International Cryptology Conference*. Lecture Notes in Computer Science, vol. 2729. Springer-Verlag, Berlin, Germany, 282–298.

MIHĂILESCU, P., AND AVANZI, R. M. 2003. Efficient "quasi"-deterministic primality test improving AKS draft. (Manuscript, available from the first author at `preda@uni-math.gwdg.de`.)

MILLER, G. L. 1976. Riemann's hypothesis and test for primality. *J. Comput. Syst. Sci. 13*, 300–317.

POMERANCE, C. 1996. A tale of two sieves. *AMS Notices 43*, 1473–1485.

PRATT, V. 1975. Every prime has a succinct certificate. *SIAM J. Comput. 4*, 214–220.

RABIN, M. 1976. Probabilistic algorithms. In *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, Ed. Academic Press, New York, 21–39.

RABIN, M. 1980. Probabilistic algorithm for testing primality. *J. Number Theory 12*, 128–138.

REGEV, O. 2004a. Improved inapproximability of lattice and coding problems with preprocessing. *IEEE Trans. Inf. Theory 50*, 2031–2037. (Preliminary version in *Proceedings of the 18th IEEE Conference on Computational Complexity*, IEEE, Los Alamitos, Calif., 2003, 315–322.)

REGEV, O. 2004b. New lattice-based cryptographic constructions. *J. ACM 51*, 899–942. (Preliminary version in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2003, 407–416.)

RIVEST, R., SHAMIR, A., AND ADLEMAN, L. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 120–126.

SCHNORR, C.-P. 1987. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci. 53*, 201–224.

Schönhage, A. 1984. Factorization of univariate integer polynomials by Diophantine approximation and an improved basis reduction algorithm. In *Automata, Languages, and Programming*. Lecture Notes in Computer Science, vol. 172. Springer-Verlag, Berlin, Germany, 436–447.

Shawe-Taylor, J., and Pisanski, T. 1994. Homeomorphism of 2-complexes is graph isomorphism complete. *SIAM J. Comput. 23*, 120–132.

Shor, P. W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput. 26*, 1484–1509.

Sibson, R. 1978. Locally equiangular triangulations. *Comput. J. 21*, 243–245.

Solovay, R. M., and Strassen, V. 1978. A fast Monte-Carlo test for primality. *SIAM J. Comput. 6*, 84–85.

Toran, J. 2004. On the hardness of graph isomorphism. *SIAM J. Comput. 33*, 1093–1108.

van Emde Boas, P. 1981. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Report 81-04, Department of Mathematics, Univ. Amsterdam, Amsterdam, The Netherlands.

Vardy, A. 1997. The intractability of computing the minimum distance of a code. *IEEE Trans. Inf. Theory 43*, 1757–1766.