A Brief History of NP-Completeness, 1954–2012

David S. Johnson

2010 Mathematics Subject Classification: 68-03, 68Q17, 68Q25, 68W25, 90C05, 90C22
Keywords and Phrases: NP-completeness, polynomial time, approximation algorithms, bin packing, unique games conjecture

The year 2012 marks the 40th anniversary of the publication of the influential paper "Reducibility among combinatorial problems" by Richard Karp [37]. This paper was the first to demonstrate the wide applicability of the concept now known as NP-completeness, which had been introduced the previous year by Stephen Cook and Leonid Levin, independently. 2012 also marks the 100th anniversary of the birth of Alan Turing, whose invention of what is now known as the "Turing machine" underlay that concept. In this chapter, I shall briefly sketch the history and pre-history of NP-completeness (with pictures), and provide a brief personal survey of the developments in the theory over the last 40 years and their impact (or lack thereof) on the practice and theory of optimization. I assume the reader is familiar with the basic concepts of NP-completeness, P, and NP, although I hope the story will still be interesting to those with only a fuzzy recollection of the definitions.

The New Prehistory

When the Garey & Johnson book *Computers and Intractability: A Guide to the Theory of NP-Completeness* [23] was written in the late 1970s, the sources of the theory were traced back only to 1965. In particular, we cited papers by Cobham [13] and Edmonds [18], which were the first to identify the class of problems solvable in polynomial time as relevant to the concept of efficient solvability and worthy of study. We also cited a second paper of Edmonds [17], which in a sense introduced what was later to be called the class NP, by proposing the notion of a problem having a "good characterization."

It turns out, however, that a pair of eminent mathematicians had touched on the issues involved in NP-completeness over a decade earlier, in handwritten private letters that took years to come to light. The first to be rediscovered (and the second to be written) was a letter from Kurt Gödel to John von Neumann, both then at the Institute for Advanced Study in Princeton, New Jersey. Gödel is perhaps most famous for his 1931 "Incompleteness Theorems" about mathematical logic. His letter, written in German and dated 20 March 1956, was not publicized until 1989, when Juris Hartmanis published a translation and commentary [27].

In this letter, Gödel considered first a problem of finding proofs in a given proof system: Given a first order formula F and an integer n, is there is a proof of F having length no more than n? Let A be a Turing machine that solves this problem, and, following Gödel, let $\psi_A(F, n)$ denote the number of steps that A takes when applied to the instance consisting of formula F and bound n. Now let $\phi_A(n)$ be the worst-case value of $\psi_A(F, n)$ over all formulas F of length n. Note that a Turing machine A performing exhaustive search would have a value for $\phi_A(n)$ that was no worse than exponential in n. Gödel pointed out how wonderful it would be if there were an A with $\phi_A(n) = O(n)$ or even $O(n^2)$, observing that such a speedup had already been observed for the problem of computing the quadratic residue symbol. Finally, he asked "how strongly in general" one could improve over exhaustive search for combinatorial problems, in particular mentioning the problem of primality testing (a problem whose worst-case complexity remained open for almost 50 more years, until it was shown to be polynomial-time solvable by Agrawal, Kayal, and Saxena in 2002 [3]).

Note that Gödel did not make the generalization from $O(n)$ and $O(n^2)$ to polynomial time. He was more interested in algorithms that might plausibly be practical. He was also not measuring running time in terms of the modern concept of "input length". For that he would have had to explicitly specify that n was written in unary notation. (If n were written in standard binary notation, then exhaustive search for his problem might have been *doubly exponential* in the input size.) On the other hand, he does seem to have assumed binary, or at least decimal, input size when he discussed primality testing. Moreover, he used the idea of worst-case running time analysis for algorithms and problems, something that was not all that common at the time, and which dominates algorithmic research today. And he does seem to have an idea of the class of problems solvable by exhaustive search, which can be viewed as a generalization of NP, and his final question hints at the question of P versus NP. At any rate, Gödel's letter, once discovered, was immediately recognized as an important precursor to the theory of NP-completeness. When an annual prize for outstanding journal papers in theoretical computer science was established in 1992, it was only natural to name it the Gödel Prize. More recently, the letter has even lent its name to a well-written and popular blog on algorithms and computational complexity (*Gödel's Lost Letter and P = NP*, `http://rjlipton.wordpress.com`).

The other famous mathematician whose letters foreshadowed the theory of NP-completeness was John Nash, Nobel Prize winner for Economics and subject of both the book and the movie *A Beautiful Mind*. In 1955, Nash sent several handwritten letters about encryption to the United States National Security Agency, which were not declassified and made publicly available until 2012 [1]. In them, he observes that for typical key-based encryption processes,

Figure 1: Stephen Cook, Richard Karp, and Leonid Levin, photographed in the 1980s

if the plain texts and encrypted versions of some small number of messages are given, then the key is determined. This is not technically correct, since in addition there must be sufficient entropy in the plain texts, but Nash's arguments apply as well to the problem of finding *some* key consistent with the encryptions. His central observation was that even if the key is determined, it still may not be easy to find.

If the key is a binary string of length r, exhaustive search will work (as it did for Gödel), but takes time exponential in r. For weak cryptosystems, such as substitution ciphers, there are faster techniques, taking time $O(r^2)$ or $O(r^3)$, but Nash conjectured that "for almost all sufficiently complex types of enciphering," running time exponential in the key length is unavoidable.

This conjecture would imply that P $\neq$ NP, since the decryption problem he mentions is polynomial-time equivalent to a problem in NP: Given the data on plain and encrypted texts and a prefix x of a key, is there a key consistent with the encryptions which has x as a prefix? It is a stronger conjecture, however, since it would also rule out the possibility that all problems in NP can, for instance, be solved in time $n^{O(\log n)}$, which, although non-polynomial, is also not what one typically means by "exponential." Nash is also making a subsidiary claim that is in essence about the NP-hardness of a whole collection of decryption problems. This latter claim appears to be false. Nash proposed an encryption scheme of the type he specified, but the NSA observed in private notes that it provided only limited security, and since the publication of the letters modern researchers have found it easy to break [2]. Also, like Gödel, Nash did not make the leap from low-order polynomial time to polynomial time in general. He did however, correctly foresee the mathematical difficulty of the P versus NP problem. He admitted that he could not prove his conjecture, nor did he expect it to be proved, even if it were true.

Cook, Karp, and Levin

The theory of NP-completeness is typically traced back to Steve Cook's 1971 paper "The complexity of theorem-proving procedures" [14], which provided the first published NP-completeness results. However, Leonid Levin, then a student in Moscow, proved much the same results at roughly the same time, although his results were not published until 1973. Over the years, the contemporaneous and independent nature of Levin's accomplishment have come to take precedence over publication dates, and what used to be called "Cook's Theorem" is now generally referred to as the "Cook-Levin Theorem." Let me say a bit about these two parallel developments.

When Cook wrote his paper, he was an Associate Professor in the Computer Science Department of the University of Toronto, where he is now a University Professor. Earlier, he had received his PhD from Harvard in 1966, and spent four years as an Assistant Professor in the Mathematics Department of University of California, Berkeley, which foolishly denied him tenure. Cook's paper appeared in the proceedings of the 1971 ACM Symposium on Theory of Computing (STOC), and there are apocryphal stories that it almost was not accepted. This seems unlikely, although it wouldn't be the first time a major breakthrough was not recognized when it occurred. The paper's significance was certainly recognized as soon as it appeared. Not only did the paper prove that SATISFIABILITY is NP-complete (in modern terminology), but it also proved the same for 3SAT, and hinted at the broader applicability of the concept by showing that the same also holds for SUBGRAPH ISOMORPHISM (more specifically, the special case now known as the CLIQUE problem). I was a grad student at MIT at the time, and Albert Meyer and Mike Fischer included these results in their Fall 1971 Algorithms course. Others had also been busy, as became clear at the March 1972 conference on "Complexity of Computer Computations" at the IBM T.J. Watson Research Center in Yorktown Heights, NY, where Richard Karp presented his famous paper.

Karp was also a Harvard PhD recipient (1959), and after an 11-year stint at the same IBM Research Center that housed the conference, had moved to a professorship at UC Berkeley in 1968, where he remains today, after a brief sojourn to the University of Washington in Seattle. Karp's paper showed that 19 additional problems were NP-complete, including such now-famous characters as VERTEX COVER, CHROMATIC NUMBER, the directed and undirected HAMILTONIAN CIRCUIT problems, SUBSET SUM, and the KNAPSACK problem. Most of the proofs were due to Karp himself, but a few were attributed to Gene Lawler, Bob Tarjan, and "the Algorithms Seminar at Cornell." The paper appears to be the first to use the notations P and NP, although its term for "NP-complete" was "polynomial complete," a locution used in several early papers before the modern terminology took hold. The paper also introduced the distinction between a *polynomial transformation*, where an instance of the first problem is transformed into one of the second that has the same yes-no answer, and a *polynomial reduction*, in which the first problem is solved using one or more calls to a subroutine that solves the second. Cook had stated his results in terms of the latter notion, but his proofs had essentially relied only on the first.

This was the first conference that I had attended, and I was suitably awed by all the famous participants whom I was meeting for the first time - including John Hopcroft, Michael Rabin, Bob Tarjan, Jeff Ullman, and Richard Karp himself. I even got to sit across the table from Dick at one lunch. I took the opportunity to mention to him that I had already proved one polynomial completeness result myself, that for BIN PACKING, the problem that was to be the topic of my thesis. Albert Meyer had proposed I work on it just a month earlier, saying "This is perfect for you, Johnson. You don't need to know anything – you just have to be clever." Albert had learned about the problem from a preprint of a 1972 STOC paper by Garey, Graham, and Ullman [21]. In the problem, one is given a sequence of numbers $a_1, a_2, \ldots, a_n \in (0,1]$ and a target k, and asked if the numbers be partitioned into k sets, each summing to no more than 1. Dick showed polite interest, but, as the words came out of my mouth, I was embarrassed to realize how trivial my proof was compared to the ones in his paper (SUBSET SUM is the special case of BIN PACKING where $k = 2$ and the $\sum_{i=1}^{n} a_i = 2$.)

In addition to many other interesting papers, the conference included a lively panel discussion, a transcript of which is contained in the proceedings [45]. It covered issues raised by many of the preceding talks, but the discussion kept coming back to the P versus NP question. The most remembered (and prescient) comment from the panel was by John Hopcroft. He observed that, although a consensus seemed to be forming that the two classes were not equal, for all we currently knew, every problem in NP could be solved in linear time. He concluded that it would be "reasonably safe" to conjecture that, within the next five years, no one would prove that any of the polynomial complete problems even required more than quadratic time. It is now 40 years and counting, and we still have yet to see any such proofs.

Meanwhile, in a much different world, Leonid Levin was thinking about the same issues, but not getting nearly the same publicity. In the Soviet Union at the time, many researchers were considering questions related to the P versus NP question. In particular, there was the notion of the class of problems that could only be solved by *perebor*, the Russian name for algorithms that were essentially based on exhaustive search [52]. Levin was a PhD student at the University of Moscow. In 1971, he completed a thesis on Kolmogorov complexity, but although it was approved by Kolmogorov (his advisor) and by his thesis committee, the authorities refused to grant the degree for political reasons. (Levin admits to having been a bit intractable himself when it came to toeing the Soviet line [51, 151–152].) Levin continued to work on other things, however, in particular *perebor*, coming up with his version of NP-completeness that same year, and talking about it at various seminars in Moscow and Leningrad [52]. He also wrote up his results, submitting them for publication in June 1972 [52], although the paper did not appear until the second half of 1973. Its title, translated into English, was "Universal sequential search problems" [42] ("Sequential search" was a mistranslation of *perebor*).

The 2-page paper was brief and telegraphic, a trait shared by many of Levin's subsequent papers (e.g., see [55, 43]), omitting proofs entirely. A corrected translation appears as an appendix in [52]. In his paper, Levin deals with the generalization of NP to search problems: Relations $A(x,y)$ on strings, such that for all pairs (x,y) such that $A(x,y)$ holds, the length of y is polynomially bounded in the length of x, and such that for all pairs (x,y), one can determine in polynomial time whether $A(x,y)$ holds. Here x stands for an instance of the problem, and y a corresponding "solution." The search problem for A is, given x, find a y such that $A(x,y)$ holds. The corresponding problem in NP is, given x, does there exist a y such that $A(x,y)$ holds. Levin mentions this version, calling it a "quasi-search" problem, but concentrates on the search problem version. He describes what we would now view as the standard notion of a polynomial reduction from one search problem A to another one, and calls a problem a "universal search problem" if there exist polynomial reductions to it from all the search problems in the above class. He then goes on to list six search problems that he can prove are universal search problems. These include the search versions of SATISFIABILITY, SET COVER, and SUBGRAPH ISOMORPHISM, along with others that were not on Karp's list, such as the following tiling problem: Given a square grid whose boundary cells each contain an integer in the range from 1 to 100, together with rules constraining the contents of interior cells, given the contents of the four neighboring cells (to the left, right, top, and bottom), find a legal tiling that agrees with the given assignment to the boundary cells.

Those who heard Levin speak about these results were immediately impressed. Trakhtenbrot [52] quotes Barzdin, who heard Levin speak in Novosibirsk in April, 1972, as saying "Just now Levin told me about his new results; it is a turning point in the topic of *perebor*!" Note that this is clear evidence that the work of Cook and Karp had not yet received wide attention in Russia. However, neither did the work of Levin. In 1973, when Russian theoreticians finally did take up NP-completeness, it was mainly through the Cook and Karp papers [25]. Levin's impact appears not to have spread much beyond those who had heard him speak in person.

In 1978, Levin emigrated to the US, where I first met him while visiting MIT. There he finally received an official PhD in 1979, after which he took up a position at Boston University, where he is now a Full Professor. He has made many additional contributions to complexity theory, including

- A theory of average case completeness [43], using which he shows that a variant of his above-mentioned tiling problem, under a natural notion of a uniform distribution for it, cannot be solved in polynomial expected time unless every other combination of a problem in NP with a reasonably constrained probability distribution can be so solved.

- A proof that the one-way functions needed for cryptography exist if and only if pseudorandom number generators exist that cannot in polynomial time be distinguished from true random number generators [28].

- A proof that a 1965 precursor of the ellipsoid algorithm, in which simplices play the role of ellipses, also runs in polynomial time [55] (thus there *is* a simplex algorithm that runs in polynomial time ...).

Cook and Karp also have made significant contributions to complexity theory since their original breakthroughs. Karp's many contributions are well known in the mathematical programming community and too extensive to list here. Cook's main work has been in the study of proof complexity, but he is responsible for introducing at least one additional complexity class, one that provides an interesting sidelight on NP-completeness.

This is the class SC, the set of decision problems that can be solved by algorithms that run in polynomial time and require only polylogarithmic space, that is, use $O(\log^k n)$ space for some fixed k. Here "SC" stands for "Steve's Class," the name having been suggested by Nick Pippenger in recognition of Steve's surprising 1979 result that deterministic context-free languages are in this class [15], but also in retaliation for Steve's having introduced the terminology "NC" ("Nick's Class") for the set of decision problems that can be solved in polylogarithmic time using only a polynomial number of parallel processors [26]. The significance of these two classes is that, although it is easy to see that each is contained in P, one might expect them both to be *proper* subclasses of P. That is, there are likely to be problems in P that cannot be solved in polynomial time if restricted to polylog space, and ones that cannot be solved in polylog time if restricted to a polynomial number of processors. By analogy with NP-completeness, one can identify candidates for such problems by identifying ones that are "complete for P" under appropriate reductions. One famous example, complete for P in both senses, is LINEAR PROGRAMMING [16].

Both Cook and Karp have won multiple prizes. Cook won the 1982 ACM Turing Award (the top prize in computer science) and the 1999 CRM-Fields Institute Prize (the top Canadian award for research achievements in the mathematical sciences). Karp won the Lanchester Prize in 1977, the Fulkerson Prize in discrete mathematics in 1979, the ACM Turing Award in 1985, the ORSA-TIMS von Neumann Theory Prize in 1990, and many others. Levin is long overdue for his own big award, although I expect this will come soon. And, of course, the biggest prize related to NP-completeness is still unawarded: The question of whether P equals NP is one of the six remaining open problems for the resolution of which the Clay Mathematics Institute is offering a $1,000,000 Millenium Prize.

Garey, Johnson, and *Computers and Intractability*

My own most influential connection to the theory of NP-completeness is undoubtedly the book *Computers and Intractability: A Guide to the Theory of NP-completeness*, which I wrote with Mike Garey and which was published in

Figure 2: Michael Garey and David Johnson in 1977

1979. At the time, we optimistically promised the publishers that we would sell 5,000 copies, but it has now sold over 50,000, picking up some 40,000 citations along the way, according to Google Scholar.

My early involvement with the theory, beyond the lunchtime conversation mentioned above, mainly concerned one of the methods for coping with NP-completeness: Designing and analyzing approximation algorithms. While at MIT I wrote a PhD thesis on approximation algorithms for the bin packing problem [32] and a paper exploring how the same approach could be extended to other problems, such as graph coloring, set covering, and maximum satisfiability [33].

On the strength of this research, I was recruited to come to work at Bell Labs by Ron Graham and Mike Garey, whose initial paper on bin packing had introduced me to the topic. After receiving my PhD in June 1973, I moved to New Jersey and began my Bell Labs/AT&T career. One of my first collaborations with Mike was in producing a response to a letter Don Knuth had written in October to many of the experts in the field. The letter sought a better name than "polynomial complete" for the class of problems that Cook and Karp had identified. Knuth asked for a vote on three terms he was proposing ("Herculean," "formidable," and "arduous"). We did not particularly like any of Knuth's alternatives, and proposed "NP-complete" as a write-in candidate. We were not the only ones, and when Knuth announced the results of his poll in January 1974 [41], he gave up on his original proposals, and declared "NP-complete" the winner, with "NP-hard" chosen to designate problems that were at least as hard as all the problems in NP, although possibly not in NP themselves. See Knuth's article or [23] for an amusing summary of some of the other proposals he received.

Mike and I also began an active research collaboration, covering both bin packing and scheduling algorithms and the proof of new NP-completeness results. When Karp wrote a journal article [38] derived from his original proceedings paper, his expanded list, now of 25 problems, included some of our new results. This set the stage for our book [23], with its much longer list, although the actual genesis of the book was more happenstance. In April 1976, Mike and I attended a conference at Carnegie-Mellon University on "New Directions and Recent Results in Algorithms and Complexity," where I gave a talk on the various types of approximation guarantees we had seen so far. Afterwards, at a coffee break, an editor for the Prentice-Hall publishing company came up to me and suggested that Mike and I write a book on approximation algorithms. In thinking about that proposal, we realized that what was needed, before any book on approximation algorithms, was a book on NP-completeness, and by the time we left the conference we were well on our way to deciding to write that book ourselves.

One of my tasks was to collect NP-completeness results for our planned list, which in those days before personal computers meant writing the details by hand onto file cards, stored in plastic box. At that time, it was still possible to aim for complete coverage, and our eventual list of some 300 problems covered most of what had been published by the time we finished our first draft in mid-1978, including many results we came up with ourselves when we identified interesting gaps in the literature, and for which we provided the unhelpful citation "[Garey and Johnson, unpublished]." We did keep notes on the proofs, however (in that same plastic box), and most can still be reconstructed ... After detailed discussions about what we wanted to say, I wrote first drafts of the chapters, with Mike then clarifying and improving the writing. (A quick comparison of the writing in [23] with that in this memoir will probably lead most readers to wish Mike were still doing that.)

We did resort to computers for the actual typesetting of the book, although I had to traipse up to the 5th floor UNIX room to do the typing, and put up with the invigorating smell of the chemicals in the primitive phototypesetter there. Because we were providing camera-ready copy, we had the final say on how everything looked, although our publisher did provide thorough and useful copy-editing comments, including teaching us once and for all the difference between "that" and "which." There was only one last-minute glitch, fortunately caught before the book was finalized – the cover was supposed to depict the graph product of a triangle and a path of length two, and the initial artist's rendering of this was missing several edges.

Over the years, the book has remained unchanged, although later printings include a 2-page "Update" at the end, which lists corrigenda and reports on the status of the twelve open problems listed in Appendix A13 of the book. As of today only two remain unresolved: GRAPH ISOMORPHISM and PRECEDENCE CONSTRAINED 3-PROCESSOR SCHEDULING. Of the remaining ten, five are now known to be polynomial-time solvable and five are NP-complete. For details, see [35, 46]. A second edition is perpetually planned but never started, although I have resumed my NP-completeness column, now appearing on a sporadic basis in *ACM Transactions on Algorithms*, as groundwork for such an undertaking.

We never did write that book on approximation algorithms, and indeed no such book seems to have appeared until Dorit Hochbaum's *Approximation Algorithms for NP-Hard Problems* [29] appeared in 1997. This was an edited collection, to which Mike, Ed Coffman, and I contributed a chapter. The first textbook on approximation algorithms was Vijay Vazirani's *Approximation Algorithms* [53], which did not appear until 2001. Although Mike and I never got around to writing a second book, there is a second "Garey and Johnson" book of a sort. In 1990, our wives, Jenene Garey and Dorothy Wilson, respectively a Professor of Nutrition at NYU and a school teacher, coauthored *The Whole Kid's Cookbook*, copies of which were sold to raise funds for the Summit Child Care Center, a local institution where Dorothy had worked.

The Last Forty Years: Hardness of Approximation

It would be impossible, in the limited space left to me, to give a thorough history of the developments in the theory of NP-completeness since the 1970s, so in this section I shall restrict myself to just one thread: applying the theory to approximation algorithms.

An approximation algorithm does not necessarily return an optimal solution, but settles for some feasible solution which one hopes will be near-optimal. A standard way to evaluate an approximation algorithm A is in terms of the "worst-case guarantee" it provides. Let us suppose for simplicity that the problem X for which A is designed is a minimization problem. Then A provides a worst-case guarantee equal to the maximum, over all instances I of the problem, of $A(I)/OPT(I)$, where $A(I)$ is the value of the solution that algorithm yields for instance I, and $OPT(I)$ is the optimal solution value. For example, Christofides' algorithm for the Traveling Salesman Problem (TSP) has a worst-case guarantee of $3/2$ if we restrict attention to instances satisfying the triangle inequality [12].

We are of course most interested in approximation algorithms for NP-hard problems that run in polynomial time. Unfortunately, it turns out that sometimes designing such an approximation algorithm can be just as hard as finding an optimal solution. The first paper to make this observation appeared in 1974, written by Sahni and Gonzalez [49]. They showed, for example, that if one does *not* assume the triangle inequality, then for any constant k, the existence of a polynomial-time approximation algorithm for the TSP with worst-case guarantee k or better would imply P = NP. The proof involves a "gap" construction, by transforming instances of HAMILTON CIRCUIT to TSP instances whose optimal tours have length n if the Hamilton Circuit exists, and otherwise have length greater than kn (for example by letting the distance between u and v be 1 if $\{u, v\}$ is an edge in the original graph, and kn otherwise).

By the time our NP-completeness book appeared, there were a few more results of this type. Of particular interest were results ruling out "approximation schemes." A *polynomial-time approximation scheme* (PTAS) for a problem is a collection of polynomial-time algorithms A_ϵ, where A_ϵ has a worst-case guarantee of $1+\epsilon$ or better. In 1975, Sahni [48] showed that the Knapsack Problem has such a scheme. His algorithms, and many like them, were seriously impractical, having running times exponential in $1/\epsilon$, although for any fixed ϵ they do run in polynomial time. Nevertheless, over the years much effort has been devoted to finding such schemes for a wide variety of problems.

Given how impractical PTASs tend to be, one could perhaps view this ever-popular pastime of designing them as providing "negative-negative" results, rather than positive ones. One can rule out the existence of such a scheme (assuming P $\neq$ NP) by proving that there exists an ϵ such that no polynomial-time approximation can have a worst-case guarantee of $1+\epsilon$ or better unless P = NP. This is trivially true for BIN PACKING, since if an algorithm could guarantee a ratio less than $3/2$, then one could use it to solve the SUBSET SUM problem. The existence of a PTAS for a problem thus merely shows that there is no ϵ such that one can prove a $1+\epsilon$ inapproximability result.

There is one particular type of PTAS, however, that can perhaps be viewed more positively. Shortly after Sahni's KNAPSACK PTAS appeared, Ibarra and Kim [31] significantly improved on it, designing what we now call a *fully* polynomial-time approximation scheme (FPTAS): An algorithm A that takes as input both an instance I and an $\epsilon > 0$, returns a solution that is no worse than $(1+\epsilon)OPT(I)$, and runs in time bounded by a polynomial not just in the size of I, but also in $1/\epsilon$.

Unfortunately, it was quickly realized that FPTASs were much less common than ordinary PTASs. In particular, the TSP with the triangle inequality could not have an FPTAS unless P $\neq$ NP, something that could not then be ruled out for ordinary PTASs. This was because it was "NP-hard in the strong sense," which means it was NP-hard even if we restrict all numbers in the input (in this case the inter-city distances) to integers that are bounded by some fixed polynomial in the input length, rather than the exponentially large values normally allowed by binary notation. It is an easy result [22] that no optimization problem that is strongly NP-hard can have an FPTAS unless P = NP (in which case none is needed).

On the other end of the scale (problems for which no algorithms with a bounded performance guarantee could exist, or at least were known), there were fewer results, although the best performance guarantee then available for the SET COVER problem was $H(n) = \sum_{i=1}^{\infty} 1/i \sim \ln n$ [33, 44], and no algorithms for CLIQUE were known with guarantees better than $O(n/polylog(n))$ [33]. Whether this was best possible (assuming P $\neq$ NP) was unknown, and the field remained in this state of ignorance for more than a decade. Indeed, although there was the occasional interesting problem-specific result, approximation algorithms remained only a minor thread of algorithms research until 1991, when a seemingly unrelated result in NP-completeness theory suddenly gave them an explosive new life.

This result was the discovery of a new characterization of NP, in terms of "probabilistically checkable proofs" (PCPs). A PCP is a proof whose validity can be estimated by looking at only a few, randomly chosen, bits. If the proof is valid, then any choice of those bits will support this fact. If it is defective, than a random choice of the bits to be examined will, with probability 1/2 or greater, confirm that the proof is not valid. This basic concept developed out of a series of papers, starting with the study of interactive proofs involving multiple provers and one verifier. (These papers include one with Leonid Levin as a co-author [10].)

If $f(n)$ and $g(n)$ are two functions from the natural numbers to themselves, let PCP(f, g) denote that class of all problems that have PCPs using $O(f(n))$ random bits and looking at $O(g(n))$ bits of the proof. In late 1991, Feige, Goldwasser, Lovász, Safra, and Szegedy [20] showed that NP $\subseteq$ PCP$(\log n \log\log n, \log n \log\log n)$ and that, surprisingly, this highly-technical result implied that CLIQUE could not be approximated to any constant factor unless NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$. This is a weaker conclusion than P = NP, but not much more believable, and in any case, the implication was strengthened to P = NP in early 1992, when Arora and Safra [7] showed that NP = PCP$(\log n, \log n)$. Shortly thereafter, Arora, Lund, Motwani, Sudan, and Szegedy [5] improved this to NP = PCP$(\log n, 1)$, which had even stronger consequences for approximation. In particular, it implied that many famous problems could not have PTASs, including MAX 2-SAT, VERTEX COVER, and the triangle-inequality TSP. There is not room here to give the details of the proofs of these results or all the references, but the key idea was to produce a gap construction for the problem in question, based on the relation between the random bits used by the verifier in a PCP for 3SAT, and the proof bits at the addresses determined by those random bits. For a contemporaneous survey, providing details and references, see [34].

In the twenty years since these breakthrough results, there has been an explosion of inapproximability results exploiting variants and strengthenings of the original PCP results, and based on a variety of strengthenings of the hypothesis that P $\neq$ NP. For surveys, see for instance [36, 54]. Today we know that CLIQUE cannot be approximated to a factor $n^{1-\epsilon}$ for any constant $\epsilon > 0$ unless P = NP [56]. We also know that the Greedy algorithm for SET COVER, mentioned above, cannot be bettered (except in lower-order terms) unless NP $\subseteq$ DTIME$[n^{O(\log\log n)}]$ [19].

Other hypotheses under which hardness of approximation results have been proved include NP $\nsubseteq$ DTIME$[n^{O(\log\log\log n)}]$, NP $\nsubseteq \cup_{k>0}$ DTIME$[n^{\log^k n}]$, NP $\nsubseteq \cap_{\epsilon>0}$ DTIME$[2^{n^\epsilon}]$, and NP $\nsubseteq$ BPP, the latter a class of problems solvable by randomized algorithms in polynomial time. Currently, the most popular hypothesis, however, is the "Unique Games Conjecture" (UGC) of Subhash Khot [39]. Suppose we are given a prime q, a small $\epsilon > 0$, and a list of equations of the form $x_j - x_k = c_h \pmod q$ in variables x_i and constants c_h. The conjecture says that it is NP-hard to distinguish between the case where at least a fraction $1 - \epsilon$ of the equations can be simultaneously satisfied and the case when no more than a fraction ϵ of the equations can – a very large gap. As with the PCP results, this conjecture initially came from a problem involving multiple prover systems, and it was in this context that it obtained its name.

The reason this rather specialized hypothesis has garnered attention is that it implies that for many important problems, our currently best approximation algorithms cannot be improved upon unless $P = NP$. For instance, no polynomial-time approximation algorithm for VERTEX COVER can guarantee better than the factor of 2 already guaranteed by several simple approximation algorithms [9]. Similarly, the Goemans-Williamson algorithm [24] for MAX CUT, which exploits semidefinite programming and randomized rounding and has a worst-case guarantee of $(2/\pi)/(\min_{0<\theta\leq\pi}((1-\cos(\theta))/\theta)) \sim .878$, cannot be improved upon by any polynomial-time algorithm [40]. More generally, for any Constraint Satisfaction Problem (CSP) where the goal is to find an assignment to the variables that satisfies a maximum number of the constraints, it can be shown that a standard algorithm, based on semidefinite programming and rounding, achieves the best possible worst-case approximation ratio of any polynomial-time algorithm, assuming P $\neq$ NP and the UGC [47], and even though for many such problems we do not at this point know what that ratio is.

Whether the UGC is true is, of course, an open question, and researchers tend to be more skeptical of this than of P $\neq$ NP. Moreover, its impact seems restricted to problems where approximation algorithms with finite worst-case ratios exist, while the other conjectures mentioned above have led to many nonconstant lower bounds, such as the roughly $\ln n$ lower bound for SET COVER. This has had the interesting side effect of making algorithms with non-constant worst-case ratios more respectable – if one cannot do better than $\Omega(\log n)$, then maybe $O(\log^2 n)$ isn't so bad? Indeed, a recently well-received paper had the breakthrough result that the LABEL COVER problem had a polynomial-time approximation algorithm with an $O(n^{1/3})$ worst-case ratio, beating the previous best of $O(n^{1/2})$ [11].

Let me conclude by addressing the obvious question. All this definitely makes for interesting theory, but what does it mean for practitioners? I believe that the years have taught us to take the warnings of NP-completeness seriously. If an optimization problem is NP-hard, it is rare that we find algorithms that, even when restricted to "real-world" instances, always seem to find optimal solutions, and do so in empirical polynomial time. Even that great success of optimization, the CONCORDE code for optimally solving the TSP [4], appears to have super-polynomial running time, even when restricted to simple instances consisting of points uniformly distributed in the unit square, where its median running time seems to grow exponentially in $\sqrt{n}$ [30].

Thus, the classical justification for turning to approximation algorithms remains valid. How that is refined by our hardness-of-approximation results is less clear. Many approximation algorithms, such as the greedy algorithm for SET COVER, seem to come far closer to optimal than their worst-case bounds would imply, and just because a problem is theoretically hard to approximate in the worst case does not mean that we cannot devise heuristics that find relatively good solutions in practice. And frankly, once exact optimization runs out of gas, what other choice do we have but to look for them?

References

[1] http://www.nsa.gov/public_info/_files/nash_letters/nash_letters1.pdf.

[2] http://www.gwern.net/docs/1955-nash.

[3] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Ann. Math.*, 160:781–793, 2004. Journal version of a 2002 preprint.

[4] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, editors. *The Traveling Salesman Problem.* Princeton University Press, Princeton, NJ, 2006.

[5] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. In *Proc. 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 14–23, Los Alamitos, CA, 1992. IEEE Computer Society. Journal version, see [6].

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation algorithms. *J. ACM*, 45(3):501–555, 1998.

[7] S. Arora and S. Safra. Probabilistically checkable proofs; a new characterization of NP. In *Proc. 33rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 2–13, Los Alamitos, CA, 1992. IEEE Computer Society. Journal version, see [8].

[8] S. Arora and S. Safra. Probabilistically checkable proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[9] P. Austrin, S. Khot, and M. Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011.

[10] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd Ann. ACM Symp. on Theory of Computing*, pages 21–31, New York, 1991. Association for Computing Machinery.

[11] M. Charikar, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61:190–206, 2011.

[12] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In *Symposium on New Directions and Recent Results in Algorithms and Complexity, J.F. Traub, (ed.)*, page 441. Academic Press, NY, 1976.

[13] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Proc. 1964 International Congress for Logic Methodology and Philosophy of Science*, pages 24–30, Amsterdam, 1964. North Holland.

[14] S. Cook. The complexity of theorem proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.

[15] S. A. Cook. Deterministic CFL's are accepted simultaneously in polynomial time and log squared space. In *Proc. 11th Ann. ACM Symp. on Theory of Computing*, pages 338–345, New York, 1979. Association for Computing Machinery.

[16] D. P. Dobkin, R. J. Lipton, and S. P. Reiss. Linear programming is log space hard for P. *Inf. Proc. Lett.*, 8(2):96–97, 1979.

[17] J. Edmonds. Minimum partition of a matroid into independent subsets. *J. Res. Nat. Bur. Standards Sect. B*, 69:67–72, 1965.

[18] J. Edmonds. Paths, trees, and flowers. *Canad. J. Math*, 17:449–467, 1965.

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

[20] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. 32nd Ann. IEEE Symp. on Foundations of Computer Science*, pages 2–12, Los Alamitos, CA, 1991. IEEE Computer Society.

[21] M. R. Garey, R. L. Graham, and J. D. Ullman. Worst-case analysis of memory allocation algorithms. In *Proc. 4th Ann. ACM Symp. on Theory of Computing*, pages 143–150, New York, 1972. Association for Computing Machinery.

[22] M. R. Garey and D. S. Johnson. Strong NP-completeness results: Motivation, examples, and implications. *J. ACM*, 25(3):499–508, 1978.

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, New York, 1979.

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

[25] M. Goldberg, V. Lifschitz, and B. Trakhtenbrot. *A Colloquium on Large Scale Finite Mathematics in the U.S.S.R.* Delphi Associates, Falls Church, VA, 1984. This is the transcript of a discussion which I attended and of which I have a preliminary typescript. Various websites list it as a book with an ISBN number and the same number of pages as my typescript, and Google displays a picture of what appears to be a hardcover version, but no one seems to be offering it for sale.

[26] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo, editors. *Limits to Parallel Computation: P-Completeness Theory.* Oxford University Press, New York, 1995.

[27] J. Hartmanis. The structural complexity column: Gödel, von Neumann and the P=?NP problem. *Bull. European Assoc. for Theoretical Comput. Sci.*, 38:101–107, 1989.

[28] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandum generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[29] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, 1997.

[30] H. H. Hoos and T. Stützle, 2009. Private Communication.

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

[32] D. S. Johnson. *Near-Optimal Bin Packing Algorithms.* PhD thesis, Massachusetts Institute of Technology, 1973.

[33] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comp. Syst. Sci.*, 9:256–278, 1974.

[34] D. S. Johnson. The NP-completeness column: An ongoing guide – the tale of the second prover. *J. Algorithms*, 13:502–524, 1992.

[35] D. S. Johnson. The NP-completeness column. *ACM Trans. Algorithms*, 1(1):160–176, 2005.

[36] D. S. Johnson. The NP-completeness column: The many limits on approximation. *ACM Trans. Algorithms*, 2(3):473–489, 2006.

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

[38] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

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

[40] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

[41] D. E. Knuth. A terminological proposal. *SIGACT News*, 6(1):12–18, 1974.

[42] L. A. Levin. Universal sequential search problems. *Problemy Peredachi Informatskii*, 9(3):115–116, 1973.

[43] L. A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

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

[45] R. E. Miller and J. W. Thatcher, editors. *Complexity of Computer Computations.* Plenum Press, New York, 1972.

[46] W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. *J. ACM*, 55(2):Article A11, 2008.

[47] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–154, New York, 2008. Association for Computing Machinery.

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

[49] S. Sahni and T. Gonzalez. P-complete problems and approximate solutions. In *Proc. 15th Ann. IEEE Symp. on Foundations of Computer Science*, pages 28–32, Los Alamitos, CA, 1974. IEEE Computer Society. A journal article expanding on the inapproximability results of this paper appears as [50].

[50] S. Sahni and T. Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, 1976.

[51] D. Shasha and C. Lazere. *Out of their Minds.* Copernicus, New York, 1995.

[52] B. A. Trakhtenbrot. A survey of Russian approaches to *perebor* (brute-force search) algorithms. *Ann. History of Computing*, 6:384–400, 1984.

[53] V. V. Vazirani. *Approximation Algorithms.* Springer-Verlag, Berlin, 2001.

[54] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, New York, 2011.

[55] B. Yamnitsky and L. A. Levin. An old linear programming algorithm runs in polynomial time. In *Proc. 23rd Ann. IEEE Symp. on Foundations of Computer Science*, pages 327–328, Los Alamitos, CA, 1982. IEEE Computer Society.

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

David S. Johnson
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932-0971
dsj@research.att.com