SIGACT News Complexity Theory Column 74

Lane A. Hemaspaandra
Dept. of Computer Science, University of Rochester
Rochester, NY 14627, USA

Introduction to Complexity Theory Column 74

First, warmest congratulations to Scott Aaronson on winning the NSF's Alan T. Waterman Award! The award is a huge honor and a wonderful recognition of all the tremendous work Scott has done as both a researcher and an expositor. (Wait. What is that overhead? Not bird, nor plane, nor even frog. It's just... a 500-foot-tall robotic marmoset from Venus flying determinedly towards MIT! Scott, it may already be time to reconsider the answers you gave to Bill's poll questions in this issue's column—while you still can.)

Those with very long memories will remember the September and December 1996 articles in this column on the future of computational complexity theory, featuring comments by Allender, Feigenbaum, Goldreich, Goldsmith, Papadimitriou, Pitassi, Razborov, Rudich, Sipser, and Wigderson. Now, you'd sort of think they would have then buckled down and resolved P versus NP within a couple of years. But no, they did not. (And please don't believe any rumor that one of them said, "Of *course* I have resolved P versus NP. Years ago. But why kill the goose that lays the golden eggs?" Totally untrue!) So in 2001 Bill Gasarch realized that it was time to look again toward the future, and he designed and conducted a terrific poll on the future of P versus NP, which ran as the June 2006 complexity theory column. And, as you surely know all too well dear reader, the oh-so-tricksy P versus NP question since then has continued to evade resolution (well, to evade any public resolution, give or take such flawed resolutions as those that are cataloged at `http://www.win.tue.nl/~gwoegi/P-versus-NP.htm`, the fascinating web site where Gerhard Woeginger provides links to more than eighty papers claiming to resolve P versus NP). So Bill Gasarch again has been so very energetic and generous as to design and preside over another poll, asking about P versus NP and other central issues.

My deep thanks to Bill for his tremendous efforts on this poll, to all the people who participated in his 2002 and 2012 polls, and to the two 1996 articles' participants. I hope that people in the distant future will look at these four articles to help get a sense of people's thoughts back in the dark ages when P versus NP had not yet been resolved.

As Bill and I were exchanging email about the article and the poll results, he commented that he could now say to his class with authority things such as: Most theoreticians think P $\neq$ NP.

And I thought about this a bit. Yes, most theoreticians do think P $\neq$ NP. But earlier in my career, all of the following statements were certainly true:

- Most theoreticians think NL $\neq$ coNL.
- Most theoreticians think that IP doesn't have a snowball's chance of containing PSPACE, and who would be so silly as to even suggest that it might?
- Most theoreticians think that neither P^{PP} nor PH contains the other.
- Most theoreticians think P $\neq$ BPP.

And yet thanks to Immerman & Szelepcsényi, Shamir, and Toda, all those issues (except the last one) have definitively been resolved, and in the opposite direction of what most people believed was the case. Briefly put, the field's intuitions have sometimes been wrong before—and could be wrong about P versus NP.

Is there a take-away item from the field's previous flawed intuitions? I think there is. The fact that the theoretical community has come up with some intuition-shattering surprises quietly whispers to each of us that more surprises are still to come, and that we should not just try to prove the things we suspect correct, but should try to disprove them. That isn't at all new thought (see for example the words of Anil Nerode in Dick Lipton's poll-answer below), but is the natural way to approach difficult open issues. Decades ago, when I was in graduate school, we were told (by John Hopcroft, I believe) that even if one works basically all week trying to prove P $\neq$ NP, one should set aside Friday afternoon for trying to prove P = NP. So clearly the take-away item is, with warm thanks and credit to Professor Hopcroft and to many of the comments in the answers people gave to Bill's poll questions: Let's dare to think crazy thoughts and explore crazy directions, trying to develop crazy new techniques, on Friday afternoons (and perhaps beyond)!

Finally, please stay tuned to future issues of the column, in which you'll find articles by: Amir Yehudayoff; Lane A. Hemaspaandra & Ryan Williams; Rafael Pass; and Dorit Aharonov.

Guest Column: The Second P =? NP Poll[1]

William I. Gasarch[2]

1 The Poll

In 2001 I (innocently!) asked Lane if I could do a SIGACT News article on a poll of what computer scientists (mostly theorists) thought about P vs. NP. It was to be an objective record of subjective opinions. I asked (by telegraph in those days) over 100 theorists, of which exactly 100 responded, which made taking percentages very easy. That poll appeared in the SIGACT News Complexity Column in 2002 and is also available on Wikipedia, and has exceeded my wildest dreams for usefulness.

Ten years have passed, so it's time to do it again. Hence I emailed out (and posted on my blog) the following questions:

1. Do you think P = NP or not? You may give other answers as well.
2. When do you think it will be resolved?
3. What kinds of techniques do you think will be used?
4. Will the problem still be relevant given advances in algorithms and in SAT Solvers?
5. Feel free to comment on anything else: Graph Isomorphism, Factoring, Derandomization, Quantum computers, and/or your own favorite problem.
6. Do I have your permission to print your response? I will do this for some people—how many depends on how many answer the poll.
7. What is your highest degree in and where is it from? This information will be used for statistics only.

I purposely did not use surveymonkey or a similar device since I want people to have the freedom to say things like *I hope* P vs. NP *is never resolved!* or *highest degree: 105 when I was really sick.*

The number of respondents was 152. I was hoping for 200 so that the percentages would again be very easy. Oh well. In Section 2, I summarize the results and compare them to the results from 2002. In Section 3, I give some excerpts from some of the responses. For space reasons I could not include everyone. Also these are edited so you might think: *Gee, I didn't write that, but it captures what I thought.*

[2] Dept. of Computer Science, University of Maryland, College Park, MD, 20742, USA. gasarch@cs.umd.edu.

2 Summary of Results

2.1 Does P = NP?

The following is a chart of what my 2002 poll said and what the 2012 poll says. DK stands for Don't Know, DC stands for Don't Care. Ind stands for Independent. I assume they mean Independent of ZFC.

	P ≠ NP	P = NP	Ind	DC	DK	DK and DC	other
2002	61(61%)	9(9%)	4(4%)	1(1%)	22(22%)	0(0%)	3(3%)
2012	126 (83%)	12 (9%)	5 (3%)	5 (3%)	1(0.6%)	1 (0.6%)	1 (0.6%)

The *other* responses in 2002 were that the problem is *not* independent. The *other* response in 2012 was that the responder didn't want P = NP to be true and also didn't want it resolved. If he gets his wish then I may do this poll every 10 years for the rest of my life.

There is a definite trend—more people think P ≠ NP now than they did in 2002. How strongly held are these opinions? Although I did not ask people what the strength of their opinion was in either poll (1) in 2002, 7 out of the 61 P ≠ NP votes (11%) said they had some doubts that P ≠ NP, (2) in 2011, 16 out of the 125 (again 11%) said they had some doubts that P ≠ NP. Hence, of the people that think P ≠ NP, the level of confidence is about the same.

There were 28 people who answered the poll both in 2002 and 2012. Of these, 24 thought P ≠ NP both then and now. Since 28 is a perfect number, this has to be correct.

I have been asked *what do the bigshots in the field think?* To make that question rigorous I looked at all of the respondents who have won either a Turing Award, Fields Medal, Nevanlinna Prize, Godel Prize, Kanellakis award, Knuth Prize, or are in the National Academy of Science or Engineering. I also looked into other awards but the ones above subsumed them. Of these 21 people, 17 (81%) think P ≠ NP (though 2 hold that opinion weakly), 2 (9%) think P = NP, and 2 (9%), perhaps the wisest of them all, said they didn't know. These opinions are very similar to that of all the participants.

2.2 When Will P vs. NP Be Resolved?

The following is a chart of what my 2002 poll said and what the 2012 poll says with regard to *when* P vs. NP will be resolved. The years are 2000+ unless it is written in full. So 02–09 means 2002–2009 but 2200–3000 means 2200–3000.

	02–09	10–19	20–29	30–39	40–49	50–59	60–69	70–79
2002	5(5%)	12(12%)	13(13%)	10(10%)	5(5%)	12 (12%)	4(4%)	0(0%)
2012	0(0%)	2(.01%)	17(11%)	18(12%)	5(3%)	10 (6.5%)	10 (6.5%)	9(6%)

	80–89	90–99	100–109	110–119	150–159	2200–3000	4000–4100
2002	1(1%)	0(0%)	0(0%)	0(0%)	0(0%)	5(5%)	0(0%)
2012	4(3%)	5(3%)	2(1.2%)	5(3%)	2(1.2%)	3(2%)	3(2%)

	Long Time	Never	Don't Know	Sooner than 2100	Later than 2100
2002	0(0%)	5(5%)	21(21%)	62(62%)	17 (17%)
2012	22(14%)	5(3%)	8(5%)	81(53%)	63 (41%)

In 2002 62% thought that P vs. NP would be resolved by 2100; however, in 2012 only 53% felt that way. Also note that in 2002 17% thought it would be resolved later than 2100; however, in 2012 63% felt that way. Hence people are more pessimistic now then they were then about when P vs. NP will be resolved.

2.3 What Techniques Will be Used?

The number of people who ventured a guess as to what techniques will be used is 77. The most intelligent and least fun comment on this was *Won't know until it's solved.* The least informative but most fun comments were *I'm not telling* and *The aliens will tell us.*

Classifying the techniques talked about is difficult. Nevertheless, here is a summary;

1. *New Techniques and Hard Math*: 65 either said this directly or indirectly.
2. *Logic:* 14 think that the techniques will be from Logic. Within that there was a great variety of logic: Computability theory, model theory, proof theory.
3. *Combinatorics:* 8 people think that the techniques will be from combinatorics.
4. *Deep Math but not Logic or Combinatorics:* 11 people. Geometric Complexity theory was mentioned several times.
5. *Known Techniques:* 5 people think the techniques are already known today.
6. *An Algorithm:* 6 people, all of whom think P = NP, think that there will be an algorithm. These people also thought it would be solved sooner rather than later.
7. *Misc:* The following were mentioned once or twice each: Miracle, Nonconstructive, Computer assisted (2 said that), Information complexity, show Factoring is not in P. completely new Meta Argument, Different Model of Comp, and Machine Learning inspired.

Scott Aaronson summed up a common view by saying roughly: *the resolution will take many yellow books*[3] *including yellow books that haven't been written yet.*

2.4 Relevancy

The number of people who gave an opinion on the relevance of P vs. NP given that we have SAT solvers was 123. Of those 15 said NO, 106 said YES, and 3 said that it depends on the answer. If we discount the latter people then we get that, of the 120 that answered the question, 86% think that P vs. NP is relevent; however, we will add a caveat to that later.

Of the 15 No's (a) 5 gave no reason, (b) one each pointed to faster machines, quantum computers, SAT solvers, polynomial time being the wrong notion because it's asymptotic, and quadratic time already not being feasible, (c) 2 said it's not practical, and (d) 2 just said that interest would fade. One said that even though it's not relevant it helped us ask the right questions.

Of the 106 YES's (a) 57 gave no reason, (b) 11 said that it was only of academic interest, (c) 11 said that SAT solvers aren't that good (one noting that they are not good for factoring), (d) 6 said

[3]Springer-Verlag has two series of books *Undergraduate Texts in Mathematics* and *Graduate Texts in Mathematics.* Hence the term *Yellow Books* indicates hard mathematics. At least it will after this Poll appears.

that studying P vs. NP and SAT Solvers go hand-in-hand (one claims that people began writing SAT solvers because of the P vs. NP problem), (e) 4 said that P vs. NP will be relevant to crypto. The other answers were one-each and hard to classify; however, here is one that I personally liked (it's not mine): *If* P $\neq$ NP *then we get Awesome Algorithms! If* P = NP *then we get Awesome Cryptography!*

As noted above 86% of the people who answered this question think that P vs. NP is relevant. But of the YES votes, 11 thought that P vs. NP is only of academic interest. If we count them as No's then we get 26 No's, 97 YES's, and hence 78% think that P vs. NP is relevant.

2.5 Other Problems

1. *Graph Isomorphism:* Of the 21 people who commented on it, 14 think GI $\in$ P, 6 think GI $\notin$ P, 2 thought GI will take $n^{O(\log n)}$ time which I count as not in P.

2. *Factoring:* Of the 21 people who commented on it, 8 think factoring is in P, 13 think factoring is not in P.

3. *Derandomization* Of the 13 people who commented on derandomization all 13 think that P = BPP and/or other randomized classes can be derandomized.

4. *Quantum* The number of people who gave an opinion on quantum computing is 16.

   Eight had clearly negative opinions: (a) 5 said quantum computing is of no practical value, (b) 1 said that the computations will be quadratically more than we think, (c) 1 said that aside from simulating quantum systems, their practicality is far off, and (d) 1 said quantum computing is a hoax.

   Two had clearly positive opinions: (a) 1 said P $\neq$ BQP and GI $\in$ BQP, and (b) 1 said quantum computing will be important.

   Six had opinions that I cannot construe as positive or negative: (a) 2 said NP is not in AP, (b) 1 said BPP $\neq$ BQP, (c) 1 said BQP does not contain NP, (d) 1 said that NP and BQP are incomparable, and (e) 1 said quantum computers will factor integers up to 10^4 by 2100.

3 What Various People Said

Here are some excerpts from some of the answers I received.

1. **Scott Aaronson** I believe P $\neq$ NP on basically the same grounds that I think I won't be devoured tomorrow by a 500-foot-tall robotic marmoset from Venus, despite my lack of proof in both cases.

   In his recent book *The Beginning of Infinity,* David Deutsch argues that we can't even make *probabilistic* predictions about some future event, to whatever that extent that event depends on new knowledge being created. I agree with him here: a proof of P $\neq$ NP, like other major mathematical advances, would depend almost entirely on new knowledge, and for that reason my uncertainty applies not only to the number of years but to the log of that number: decades? centuries? millennia? I have no idea. Maybe your question should be rephrased:

will humans manage to prove P $\neq$ NP *before they either kill themselves out or are transcended by superintelligent cyborgs? And if the latter, will the cyborgs be able to prove* P $\neq$ NP*?*

Obviously I don't know—but if we look at the techniques used in (say) Ryan Williams' recent result, and then remember that that proof only separates NEXP from ACC_0, we can get a weak hint about the scale of the techniques that would be needed to separate P from NP. Right now, Mulmuley's GCT is the only approach out there that even *tries* to grapple with the biggest barrier we know, beyond even relativization, natural proofs, and algebraization: the barrier that many nontrivial problems (including matching and linear programming) are in P! That's not to say Mulmuley's specific program will succeed: indeed, I suspect that the right chain of reasoning might diverge from Mulmuley's at an earlier rather than later point. But I fear Mulmuley is basically right about the quantity of yellow books that will need to be brought to bear (including yellow books that haven't been written yet).

GI, Factoring, Derandomization: Probably GI $\in$ P. Factoring is not in P, but with nothing like the strength of conviction with which I think P $\neq$ NP. I think P = BPP (with essentially the same strength of conviction as P $\neq$ NP), and likewise PromiseP = PromiseBPP, L = RL, etc.

Quantum computing: I think BPP $\neq$ BQP (though not with the same strength of conviction as P $\neq$ NP), and also think quantum computers will ultimately be found to be compatible with the laws of physics (they're certainly compatible with the *known* laws).

2. **Moez A. Abdel-Gawad** P $\neq$ NP. Realistically, I think it will never be resolved. It is losing/lost practical relevance, and we already gave it ("wasted on it?") the best of our brains.

   Techniques: Are miracles techniques?

   Relevance: No. P vs. NP is losing practical relevance (except as an example of some challenging problem).

   Other: Nothing else, unless philosophy or religion are allowed. My favorite problem is type checking (inside language compilers and interpreters). Decidability issues limit the expressiveness of types, but I think that won't change much even in the very unlikely event that P = NP is proven to be true. More so if P $\neq$ NP.

3. **Yiorgos Adamopoulos** I do not *want* it to be equal. I do not *want* it to be resolved. It is the holy grail and as long as it exists, people find interesting results. But if I were to estimate a solution, I would give it a time span equal to Fermat's Theorem. I think Operations Research people will find the solution to this. There's too much money involved.

   Right now, my favorite problem is Liu's conjecture: $2m/(m+1)$ on preemptive versus non-preemptive multitasking.

4. **Manindra Agrawal** P $\neq$ NP will be resolved by the year 2030 using Combinatorics and algebra. No fancy technique. GI, Factoring, Derandomization, are all in P. Likely that P = BQP.

5. **Eric Allender** P $\neq$ NP will be resolved within 25 years, though this estimate is completely meaningless, of course.

Techniques: If I knew, then I wouldn't tell you.

GI*, Factoring, Derandomization:* It is likely that we'll know about GI long before we know P $\neq$ NP. Would not be shocking if factoring was in P. I think that there are problems in E that require exponential-size circuits, and hence I also believe that BPP = P.

Quantum computers: I suspect that research on quantum computers will lead to insights about physics that will, in turn, lead to the discovery that physically-realizable quantum computers are no more powerful than classical computers.

6. **Daniel Apon** P $\neq$ NP. I think we're far enough away from resolving P vs. NP that it's effectively unpredictable when we'll get there. So—An arbitrarily large amount of time. Say, 2000 years. I don't believe P vs. NP is resolvable with current techniques (even clever, complex mixtures of current techniques). We will need to develop new mathematical machinery to get traction with the problem.

   Relevance: Absolutely. We'll continue building bigger computers and faster algorithms in a crypto arms race far past the proof that P $\neq$ NP.

   Other: SAT will be shown to require strongly exponential time. We will eventually develop techniques to show tight time complexities for problems of "intermediate" complexity (superpolynomial and subexponential). Some relatively simple-to-state, natural problem will be shown to have an OPTIMAL, "strange" time complexity, like $\log(\log(n))\log(n)(n^{\log(n)})(2^{\log^*(n)})$; others will follow.

7. **Boaz Barak** I am almost certain that P $\neq$ NP. I tend to agree with Scott Aaronson that, given all the evidence for it, in other fields such as physics P $\neq$ NP would have already been declared a *law of nature*. I don't see it being resolved in the next decade, but beyond that I can't say.

   Techniques: As a cryptographer, I hope the proof will show something stronger such as subexponential sized circuits can't solve random instances of satisfiability, hence giving us average-case hardness and one-way functions as well. Indeed, some known lower bounds such as those for AC_0, showed such stronger statements of exponential average-case hardness.

   Related to that, I have a hope/hunch that the way the natural proof paradigm will be breached will not be by violating "largeness" (i.e., proving hardness for a specific function) but by violating "constructiveness" (i.e., coming up with proof techniques that are not captured by efficient algorithms). It makes intuitive sense that we will need non-constructive arguments to prove computational hardness.

   Of course the problem will still be relevant. People come up every day with NP problems they want to solve and the SAT solvers can't handle. In particular for many cryptosystems, the corresponding SAT instances take exponential time for current solvers. This will not change as SAT almost certainly requires exponential time to solve.

   It's easier for me to imagine that the proof of P $\neq$ NP will show that random instances of SAT are hard, than to imagine that it will show that factoring or some other structured problem is hard. So, one of my favorite research questions is whether we can base *public* key cryptography on more *unstructured* problems.

8. **Dave Barrington** P $\neq$ NP. This will be shown by 2040 using some techniques from mainstream mathematics.

   Relevant: Yes, it will still confirm our intuition that the growth rate for NP-complete problems is superpolynomial.

   Derandomization: I think L = RL will be solved by 2020 (in that direction).

9. **Paul Beame** More than being 10 years since your last poll, March 2011 is 25 years since a Science Digest article (Dr Crypton column) with photos of Johan Håstad and David Barrington about their hot new results and about Razborov's monotone circuit lower bound. The article quotes Mike Sipser and Ron Graham on the expectation that the problem would be resolved by 2000. (Ron Graham says that *A proof in the next three years would not surprise me.*) But it concludes that Barrington's result might *convince mathematicians not to be so cocky about their convictions in the complex field of complexity theory.*

   P $\neq$ NP. I do not think that the question is independent of set theory but even if it were, the right choice of consistent extension would be to add P $\neq$ NP. The question is 40 years old (at least as far as the larger field of research has been aware of it), so I would bet even money on it being resolved in the next 40 years. Our experience so far has been that algebra (non-solvable groups, finite fields) has allowed surprising algorithms. However, I do not think that *yellow books not yet written*, i.e., new deep results in traditional mathematics like that required for GCT, will be necessary or especially useful. It won't be some simple trick that everyone has missed for years but rather the combination of many small advances over the years that makes the final statement accessible. One question that the experts might have had a different answer for 25 years ago is whether it will also prove that NP is not contained in BPP or P/poly. I suspect that it might not.

   Relevant: Yes. I've spent a lot of time with SAT solvers and their behavior is quite brittle. Much of their success on big practical problems has to do with the fact that the hard cores of those problems are quite small. Nonetheless, many large practical problems will be solvable using SAT solvers.

   I think that a quasipolynomial-time approximation algorithm beating the conjectured bounds will be found for Unique Games within the next decade. The problem is roughly 10 years old so that seems roughly like the right timeline to discuss, though this might be sooner.

10. **Adam Bender** P $\neq$ NP will be resolved in 2025 using techniques not invented yet.

    Relevant: People will talk like it is, but in practice, no.

11. **Marc Bezem** P $\neq$ NP. In relation to SAT, I think Boolean formulas are very compact representations having a huge information content. I do not see why their satisfiability should be P. It is now open for 40 years. That's not long for a truly difficult problem. Give it 100 years! (More precisely, 60 more years.) The problem is so fundamental that it will always be relevant.

12. **Nader Bshouty** P = NP will be resolved before I die using new techniques in Algebraic Function Fields and Algebraic Geometry over finite fields. The time complexity though will be n^C for a very large C, like TOW(10).

13. **Sam Buss** I think P $\neq$ NP. However, our evidence for this conjecture is not as convincing as we might like. On the other hand, I am agnostic about whether the polynomial hierarchy is proper; for instance, it would not be so shocking if it turned that $\Sigma_2^p = \Pi_2^p$. I am also agnostic about whether these classes even equal PSPACE. It will be resolved by 2020. I of course am not so certain about this, but have made this predication in print before. So for the sake of consistency, I stand by this prediction.

    Techniques: I have no idea. The current idea that is "in the air" is that improvements in algorithms for SAT and other hard problems will lead to improved separation results. This seems like a reasonable approach. The conventional wisdom is that P $\neq$ NP is hard to prove because proving lower bounds is hard in principle. But this could be circumvented by showing, for instance, that the polynomial time hierarchy collapses and equals exponential time; from this the ordinary time hierarchy would give P $\neq$ NP. That is to say, it may not be necessary to give fundamentally new proof methods for lower bound; instead we might discover unexpected new algorithms, and this might be good enough to resolve P vs. NP.

    Proof complexity is another possibility. For a snapshot of the current state of the art, you can refer to a recent survey of mine, *Towards* NP $-$ P *via Proof Complexity and Search (to appear).*

14. **Stas Busygin** In 2002 my answer was that P = NP and this would be shown in 2010. I retract my answer from 2002 and now think P $\neq$ NP. The 100 years mentioned by Ketan Mulmuley seems a reasonable lower bound for when it will be solved. Algebraic geometry is a good bet, though I lack the expertise to argue whether GCT program is on the right track. It's easier to say which techniques will NOT be used: all of those from alleged P $\neq$ NP "proofs" that appear every month or so, including phase transition/statistical physics arguments from the infamous manuscript publicized in August 2010.

    Relevant: Of course the problem will be relevant! I would rather consider the present advances in SAT solvers irrelevant! Who uses them in machine learning/AI applications?

    Quantum computing may be the key to new efficient algorithms. Not because of we are going to have a quantum computer soon (D-Wave is still more a hype than something real), but because we can discover special cases of quantum systems that can be simulated efficiently due to a special structure and help us solve seemingly intractable problems. Holographic algorithms is a good start in this direction as we know they correspond to modeling certain quantum systems of fermions.

15. **Stephen Cook** P $\neq$ NP will not be resolved in the next 20 years and will need new techniques.

    Relevant: Certainly. In particular it will be relevant to cryptography.

16. **Bruno Courcelle** P $\neq$ NP is true and provable in ZFC. It is completely ridiculous, as somebody proposed, to put P $\neq$ NP as a new axiom.

    Techniques: Combinatorial arguments. Descriptive complexity has failed.

    Relevant: Yes, for Theory.

17. **Ernie Croot** P $\neq$ NP, but at the same time I don't think that NP-complete problems should require exponential time to solve—I think they should require time intermediate between

polynomial and exponential. Obviously if any one NP-complete problem could be solved in time $2^{n^{o(1)}}$ then all of them will be sub-exponential time. It will be solved sooner than people think. I would guess it will be resolved within the next 20 years.

Although methods from representation theory show some promise (or so I am told) in possibly addressing the P vs. NP problem, I think some of the more recent developments in additive combinatorics might also prove useful here. Particularly, I think the work of Harald Helfgott and others on growth and generation in $SL_n(F_p)$ is the sort of theory that might be applicable.

To my knowledge nobody in CS has looked at applying ideas from this part of additive combinatorics to computational complexity, though several people have applied ideas related to Gowers U^k norms to problems in machine learning and such.

It will be highly relevant. Cryptology is one place it will remain relevant.

GI*, Factoring:* GI $\in$ P—we just do not have the right sort of techniques yet. (I personally would look into properties of harmonic functions on graphs.) Factoring can be solved in subexponential time, perhaps $2^{(\log N)^{o(1)}}$ operations. I would probably start by trying to improve upon the Number Field Sieve, which works by extending the Quadratic Sieve to number fields. Is there an analogue of "Number Field" for quaternions (i.e., "division algebra extensions" of $Q(i, j, k)$, where Q is the rational numbers)? Can one further improve the NFS using quaternion arithmetic?

18. **B. Donat** P vs. NP is Independent of PA and ZF and this will be shown before 2021 using model theory. The beauty of P vs. NP resides in its proof of independence.

19. **Underwood Dudley** Because I'm unencumbered with any knowledge about the problem I can follow my intuition and confidently say that of course P $\neq$ NP.

    My training was in number theory, a field far older than that containing P and NP, and I can therefore answer with the perspective that distance gives. There are many problems in number theory that are not resolved. The Goldbach conjecture has been around since 1742; in the decades from 1920 to 1970 progress was made but since then it has come to a standstill and it may be that no one is working on it any more. Twenty years ago several people were more or less sure that they were on the brink of showing that there were infinitely many pairs of amicable numbers. Well, they weren't. There's a little hope that the twin primes conjecture maybe could be verified if just a little lightning would strike, but at the moment the skies seem clear with no thunderclouds on the horizon.

    Let's face it, not all questions are going to be answered, especially since we can ask a potential infinity of them. The abc conjecture, which has Fermat's theorem as a trivial corollary, is not going to be settled any century soon. Schinzel's Hypothesis H, which immediately implies the truth of the twin primes conjecture, is likewise too hard. In fact, I think that everyone has forgotten about it.

    So, although it goes counter to the can-do American spirit, I think I won't live to see the P vs. NP question settled, nor will many of the readers of this survey. The finite lifetime of the human race may not provide enough time. I'd be glad to be proved wrong, of course.

20. **Ronald Fagin** I have much better intuition for NP than I do for P, because of the connection of NP with logic (via Fagin's Theorem). I believe that P $\neq$ NP and NP $\neq$ co-NP. I remember

asking a famous theoretician once whether he worked on the P vs. NP problem, and he told me, *No. First I need an idea.* I do feel like I have an idea, namely using logical games on graphs. In fact, it can be shown that NP $\neq$ co-NP if and only if a certain player has a winning strategy in a certain class of games on graphs (based on mathematical logic). Because, at least to some extent, our brains are capable of finding winning strategies in games, I think that the logical game-theoretic approach is as good a candidate as any for resolving the P vs. NP problem.

As I mentioned in the 2002 poll, I have proven (at least twice) that NP $\neq$ co-NP, and I've also proven (also at least twice) that NP = co-NP. All but one of these proofs made use of these logical games. One of my proofs that NP does not equal co-NP (from the late 1990's) survived for about 3 days and fooled some very smart people into believing it. It turned out that the bug in my proof was (believe it or not) a misuse of Fagin's Theorem.

21. **Lance Fortnow** P $\neq$ NP. It will be resolved in an unpredictably long time. If I knew what kinds of techniques would be used I wouldn't tell. SAT solvers have made the problem more relevant since the ability to solve small problems makes us thirst to solve longer ones.

    GI*, Factoring, Derandomization:* GI $\in$ P, factoring is not in P, Derandomization: Easy.

    Quantum Computers: Of little practical use.

    Favorite Problem: SAT, which is hard.

22. **Harvey Friedman** P $\neq$ NP. It will be solved in 2030 using combinatorics.

    Relevant: The solution will be followed by sharpened forms that are very relevant, particularly to security.

23. **William Gasarch** P $\neq$ NP, however, I am not dogmatic on this. When I first saw the Graph Minor Theorem used to get Vertex Cover for fixed k into $O(n^3)$ times I thought that a different very-hard-math-thing-that-I-don't-understand might be able to get SAT $\in$ P. This hasn't happened yet but it could. Also, I am more convinced that separating the two is hard then I am convinced that they are different. Litmus test: If someone told me that the problem had been solved, but not which direction, I would guess P = NP. It will be solved between 200 and 400 years from now. When Jon Katz saw my answer he said: *If it's not solved within 200 years it's not going to be solved.*

    Techniques: I hope it uses Ramsey Theory and Logic so I might understand the proof. If it comes out of Geometric Complexity Theory I will not understand the proof. In any case it will be new techniques.

    We will show that P $\neq$ NP by showing that factoring is not in P. SAT might not be a good candidate for separation. This kind of thing has happened before (once): The proof that $AC_0 \neq NC_1$ was done by showing PARITY $\notin AC_0$. Note that PARITY is not complete for NC_1 under AC_0 reductions. The word problem for S_5 is, but was not useful for separation. Factoring may be a better candidate for separation since you can generate instances that seem hard, where for SAT this seems hard to do.

    GI*, Factoring, Derandomization:* We could show P $\neq$ NP in 400 years but still not know the status of GI. OR we could find GI $\in$ P tomorrow. As noted above, Factoring is NOT in P.

L = RL will be shown before I do this poll again. P = BPP but this won't be proven for a while.

Quantum Computers: They will never be practical; however, just as the Prob Method is now a STANDARD thing to know even if you are not working on probability, Quantum methods will be a standard thing to know even if you don't work on quantum computing.

Other: Within 10 years all supermarkets will have self-checkouts that work nicely and that you are expected to use—except in New Jersey which will outlaw them to create more jobs (as they do now for self-service gas).

24. **Betsy George** I think P $\neq$ NP, but a little part of me secretly hopes P = NP. If it was shown that P = NP then this would likely get a lot of media attention and hopefully attract more people into computational complexity or at the very least just elevate the general population's knowledge of the topic. (Hopefully the press would be good, and not just laughing that so many people in the field had made the wrong assumptions for years.)

    When: 2050–2075?????

    Techniques: No clue, but I *hope* it's something really elegant.

    Relevant: Yes. The more you understand the theory behind something the better you can understand the applications.

25. **Oded Goldreich** P $\neq$ NP. *When:* Only a guess: In 200–400 years.

    Techniques: No clue. That's the point... I think we can have no clue at this.

    Relevant: Of course. I don't see the relevance of advances in SAT Solvers to the question at stake.

    GI*, Factoring, Derandomization:* GI could be in P. Factoring is probably hard. Derandomization should be possible, I expect significant progress in our lifetime.

    Quantum computers: Their conjectured power is based on a philosophically flawed *inference* by which what our current theory of QM does not rule out is actually doable in real life. I don't believe that QC is significantly more powerful than TM.

26. **Judy Goldsmith** I think that the question (P versus NP) becomes less important, as we develop better and better heuristic solvers for constraint problems, SAT, and higher-class problems such as QBF, which is complete for PSPACE. It may never get resolved. Relevance: I think that it drives the development of heuristic solvers for complete problems for NP, PSPACE, EXP, and beyond.

27. **William Gowers** I think that P $\neq$ NP. My reason, which may not be very good, is that there are analytic sets that are not Borel. Although I don't believe that the proof of the latter fact can be converted into a proof that P does not equal NP, I feel that it shows that P has no reason to equal NP. A counterargument, however, is that the set of bipartite graphs on $N \times N$ containing a complete matching is not Borel, so the analogy between the two situations is definitely not perfect.

    I think that someone has to come along with a stunning new idea, and those are rather unpredictable so the best one can do is give a wild guess about the half life of the problem.

I would go for 30 years for this half life. (Just to be clear, I'm saying that there is roughly a 50-50 chance that the big idea will have been had in the next 30 years, and at any time in the future if nobody is closing in on that big idea then there will still be roughly a 50-50 chance of it happening in the next 30 years after that. Except that if the problem is still open in 2300, it seems reasonable to think that the half life will by then be bigger, so maybe I should say that the half life will remain at 30 years for the next century or so.)

This is pure prejudice, but I'm inclined to think that a direct attack on proving circuit complexity lower bounds is still the best way to go, despite all the known difficulties. Of course, there will be new techniques, but I don't expect some magic bullet from algebraic geometry to do the job.

Relevant: It depends what you mean by *relevant.* It will undoubtedly be very, very interesting, and any successful proof methods could well have a big impact, but whether a solution would have a big effect on how we go about doing mathematics or writing computer programs is much less clear. If P is shown not to equal NP, then it will just confirm what almost everyone uses as a working hypothesis anyway, and if P is shown to equal NP, a lot will depend on the nature of the proof. For instance, a brilliant trick that made it easy to write programs to solve any NP problem that came along (without the need to convert it into some other problem such as SAT) would, I imagine, have a dramatic effect on computing. But a theoretical proof that 3-SAT can be solved in time n^{100} by means of some bizarrely opaque algorithm might have no impact at all. I also think that by the time the problem is resolved there may well have been a great deal of progress in automating human thought processes—such as how we solve maths problems—so that if P were shown to equal NP, then all it would add would be a general way of producing proofs that we do not understand for problems that we are not interested in.

The apparent hardness of factoring is often given as evidence that quantum computers are more powerful than classical computers. But one could take a slightly different attitude and argue that the fact that factoring can be solved by a quantum computer shows that factoring isn't quite as hard a problem as it looked. Could a quantum computer be simulated classically? Or rather, could a quantum computation be achieved by classical means? I would say that it is unlikely, but not unimaginably unlikely.

28. **Fred Green** P $\neq$ NP will be proven but it will not be in my lifetime. Maybe 100 years from now. Very deep mathematics that we don't know yet will be used. No one knows if GCT is the right direction, but it wouldn't surprise me if something of that scope turned out to be useful. (If it turns out differently, I may actually be around to be surprised.) It will still be relevant, but will be of more academic interest than it is now.

    We will find that P = BPP (possibly in the next 10 years), P $\neq$ BQP (possibly in the next 120 years), and that factoring is hard classically. One of Uwe Schöning's descendants will prove that Graph Isomorphism is in BQP. Quantum computation will be found to be moderately useful, and the first 1KQb quantum computer will become operational on April 1, 2094. A correct proof of the Riemann Hypothesis will be announced on the following day by an eccentric young Russian mathematician.

29. **Yuri Gurevich** It seems that the P vs. NP problem is becoming the most famous mathematical problem, yes mathematical. In a vast majority of applications, it is not the worst-case (or

even average-case) complexity that matters but the typical case. Is the typical case that much different from the worst and average cases? Yes, it is. For one thing, the asymptotic behavior may be irrelevant because the range of input sizes is finite. By now, I have no idea which way the problem will be solved or when. But note that a negative solution does not necessarily mean that NP-complete problems are necessarily intractable in practice; even though the solver has to spend a lot of time on hard instances, the challenger may have to spend much more time to produce hard instances. (We addressed this issue in *The challenger-solver game*, Bull. of EATCS, Oct. 1989.) Similarly, a positive solution does not necessarily mean that NP-complete problems are easy; the polynomial bound may be high or even unknown.

30. **Lane A. Hemaspaandra:** P $\neq$ NP. I want to say *with my luck it will be resolved the day after I die* except if I say that Bill Gasarch will probably hire an assassin to speed up the resolution of this problem.

    In recent years, especially in AI circles, I've found many people who feel that worst-case hardness results (e.g., NP-completeness results) for a problem say virtually nothing about whether there can be polynomial-time heuristic algorithms for the problem that are right wildly often. My coauthors and I in response try to mention that work in structural complexity theory from the (gasp!) late 1900s made it clear that at the extreme that isn't so: If any polynomial-time algorithm's symmetric difference with any NP-hard set has a polynomially bounded census function then P = NP, and if any Polynomial Time algorithm's symmetric difference with any NP-hard set has an $n^{\log^{O(1)}(n)}$-bounded census function then EXP = NEXP and NP falls into quasipolynomial time. So sufficiently good typical-case algorithms would have sweeping worst-case implications.

31. **Danny Hermelin** Don't know, could go both ways, but probably not equal. If they are equal, however, I think this will have devastating implications on TCS. That is, our model of efficient and non-efficient computation must be completely wrong.

    Techniques: Maybe some completely new meta argument, coupled with other techniques. I used to think the model theory approach seemed really promising (much before the latest wrong P $\neq$ NP proof).

    Relevant: Yes. The P vs. NP question is really about how good our model for efficient algorithms is. Not equal means the model is good, equal means it's probably worthless. Already graph minor theory gives some indication why the model is not perfect.

32. **Jeremy Hurwitz** P $\neq$ NP. Specifically, I believe that the Exponential Time Hypothesis is true (SAT requires 2^{cn} time for some constant $c > 0$). I would put the over/under (or when it will be solved) at around 70 years from now. This may be partly unfounded optimism, since it would imply a 50% chance of a proof being discovered before I die.

    Techniques: Here's one approach: Fix a graph property such as CLIQUE. Given a Polynomial Time algorithm A, we will construct a random graph G (possibly Erdős-Renyi, but most likely constructed according to a more complicated procedure that depends on A). We then "only" need to show that Prob[A errs on G] > 0 for all A.

    This approach is motivated by recent work on the planted clique problem in random graphs. Specifically, Ben Rossman has shown that any monotone circuit which solves k-CLIQUE with

high probability over random graphs with edge probability p must either have size $\Omega(n^{k/4})$ or fail to solve the problem with high probability.

Having said that, I doubt the final technique has a name yet. In much the same way that hardness-of-approximation exploded when Khot proposed the Unique Games Conjecture, I think someone will discover a key technique/observation/lemma which will suddenly solve many lower-bound problems.

Relevant: Definitely! Studying heuristics (when and why they work) goes hand-in-hand with studying P vs. NP.

GI, *Factoring:* I think that GI $\in$ P (or at least is tractable in practice), while Factoring will remain a secure basis for crypto.

33. **Neil Immerman** P $\neq$ NP will be resolved somewhere between 2017 and 2034, using some combination of logic, algebra, and combinatorics.

    Relevant: Yes, definitely. The quality and usefulness of SAT solvers will only increase the importance of our understanding what is easy and what is hard.

    I think that the most fundamental question is how to automatically parallelize computations. As we learn to understand the hardware/parallel time trade-off, the other open questions in complexity including P vs. NP will be revealed as well.

34. **Russell Impagliazzo** P $\neq$ NP. *When:* There's no way to tell, but I don't think we're close.

    Techniques: If I knew that, why would I be wasting time writing this email?

    Relevance: Yes. Like ILP, SAT solvers are not NP oracles. They solve some large instances of NP-problems quickly. Others, they simply fail to solve in any reasonable amount of time. For example, I had my graduate complexity theory class reduce Sudoku puzzles to SAT and use Zchaff to solve them. Success (in terms of getting the algorithm to terminate before the due date) depended heavily on the reduction used and no one could solve 36 by 36 Sudoku puzzles with this method.

    To my knowledge, SAT solvers have never been used in cryptanalysis, which shows limitations. (People have tried reducing factoring, but this is a loser because the resulting formulas are really huge even for easy to factor sizes. Better would be to reduce your favorite block cipher—I don't know if anyone has tried this.)

    Relevance: I don't know why *advances in algorithms* would make the P vs. NP problem less significant.

    You should also keep in mind that many interesting problems are in PH rather than NP, and so far quantified SAT-solving hasn't had the same success.

35. **Gil Kalai** P $\neq$ NP will be resolved by 2030. The problem does not interact so well with other areas of mathematics. A clever combinatorial proof based on familiar considerations from computational complexity seems always a (small) possibility. I find some of the current proposed directions interesting, but I cannot regard any as very promising.

    Relevant: Yes, fully relevant. It is interesting also to understand theoretically why SAT solvers do so well.

GI, *Factoring, Derandomization:* I don't have a guess about GI $\in$ P. There are too many conflicting pieces of evidence. Factoring I would expect not to be in P. Obviously, I even more strongly believe that BPP $\neq$ BQP. Practically, derandomization of randomized algorithms is, of course, possible. Theoretically as practice suggests we can expect that derandomization is possible, and I find the theoretical support impressive. The most daring derandomization conjectures (that can be deduced from extremely strong hardness assumptions) are less convincing.

I am not sure that the issue is entirely understood even on the level of posing the problem. For example, if you can prove to me using the probabilistic method that a certain randomized algorithm can be replaced by a random algorithm in a certain (very large) class of deterministic algorithms, I will be confused if this is a genuine derandomization (while technically speaking it is).

Quantum computers: I think that it is not plausible that computationally superior QC can be built. I have an even stronger belief that Quantum Mechanics can accommodate infeasibility of computationally-superior QC. Infeasibility of scalable QC may imply that certain computations in quantum field theory are not scalable as well and lose their relevance for large systems. (I am not aware, and will be eager to learn about other physical consequences of infeasibility of superior-QC.)

I expect that quantum error correction is the key to the problem, and that superior QCs are excluded by the following *Principle of No QEC*: In every implementation of quantum error correcting code with one encoded qubit, the probability of not getting the intended qubit is at least some > 0, independently of the number of qubits used for encoding.

36. **Richard Karp** I believe intuitively P $\neq$ NP but it is only an intuition. Relevance: Absolutely.

37. **Jon Katz** P $\neq$ NP. Also NP $\not\subseteq$ BPP. But I am open to the possibility that NP $\subseteq$ P/poly. *When:* Perhaps never. As a lower bound, not in the next 10 years. Entirely new techniques/approaches need to be developed.

    Techniques: Unclear. It could go either way here: it could require deep connections with hard mathematics, or it could involve an extremely clever proof that uses no more than undergraduate math techniques.

    Relevant: Finding that P = NP would be relevant irrespective of any heuristics or approximation algorithms. Proving that P $\neq$ NP, which is anyway the prevailing opinion, will not be itself change anything. But the hope is that it might lead to other advances (e.g., provably hard functions for crypto).

    P vs. NP gets a lot of attention because of its importance outside complexity theory. But there are other questions that may be *easier* to resolve, yet no less important (at least within the field). My favorite here is P $\neq$ PSPACE (surely this should be easy to show!), but one that appears within reach in the next 10 years is an unconditional separation of NEXP from BPP.

38. **Samir Khuller** P $\neq$ NP. If P = NP then I think we would have found a poly time algorithm for one of the NP-complete problems. However, proving that no poly time algorithm exists, is much harder.

39. **Joe Kilian** P $\neq$ NP. I choose the optimally pessimistic position: The problem will not be resolved and will not be shown independent of set theory.

    Techniques: Not really applicable, given my pessimism. But I do believe that some progress will be made separating NP from the lower classes. I think that this will be by alternative descriptions of these complexity classes in terms of more information theoretic games (though with some bounds on communication, random bits, or something like that). Then it will be shown that some problems in NP are not so transformable, at least given the bounds on resources. Don't expect this to work beyond fairly weak classes.

    Relevance: The P vs. NP problem relates to so many problems that heuristic improvements to SAT solvers will be irrelevant. For example, I predict that SAT solvers will never be the best way to factor. What would kill the problem is a convincing demonstration that quantum computers can, in practice, solve problems that the best known non-quantum techniques cannot begin to handle (say, factoring 10000-bit numbers). At this point, P will be considered much less interesting as a class, and the QP versus NP (or QNP) question will become the important problem.

    GI*, Factoring, Derandomization:* It is conceivable that someone will come up with a natural candidate Polynomial-time algorithm for GI (by *natural* I mean not using clever scheduling techniques to eventually run all possible polynomial time algorithms), but that it will take another century to prove that it always works. Some significant progress will be made on Factoring probably not reducing it to polynomial time. P = BPP will be proven.

    Quantum computers: I'm a pessimist—I suspect that we will never get in practice the exponential speedups available in theory; we will eventually learn that there are physical limits to how many "independent" configurations quantum computers can exploit just as there are limits to how small we can make computer components. These limits will be generous, but we will not be able to, for example, factor n-bit numbers by constructing a superposition of more than 2^n states.

40. **Clyde Kruskal** I do not really know much more about P vs. NP now than I did ten years ago for the first poll, so as much as possible I am not changing my answers.

    P $\neq$ NP. But I am a little less convinced of this now than for the last poll.

    When: 2036. For the last poll I basically added the amount of time researchers had been working on the problem to the current date. If I did that this time I would get something like 2056, but then I would definitely be wrong at least once, so I will stick with 2036.

    Techniques: The solution will almost surely use techniques that I will not understand.

    Relevance: Definitely relevant theoretically. Not clear if it will have any practical relevance even if P = NP.

    Last time I said: If P = NP then I think NP-complete problems will have very high degree polynomial times. Otherwise, it does not seem reasonable that we do not yet have a polynomial time solution. To amend that, I suppose the degree of the polynomial could be small but the multiplicative constants very large.

41. **Oliver Kullmann** P $\neq$ NP, however ETH (the conjecture that SAT requires time $2^{\Omega(n)}$) is clearly false. In fact, I think that SAT is just barely not in P. P vs. NP will be solved in

20–30 years using beautiful techniques. I think it is the most important question, which the solution will *reveal*.

Techniques: Highly infinitary. Like the set of real numbers contains in a nutshell the universe of sets.

Relevance: The time of SAT is still to come, we just scratched the surface. The P vs. NP problem stands for a deeper understanding of complexity, while *polynomial time* is a straw man.

Quantum blabla is a hoax.

Mathematics has to face its ghosts, disorder, chaos—however all current approaches are reactionary, and just want to continue as before. No more disavowals via approximations, statistics, a little bit of order, but the REAL!

42. **Greg Kuperberg** P $\neq$ NP. No idea when it will be resolved.

    Techniques: All I know about it is the obstruction theorems: The proof won't relativize and won't algebrize, and probably won't be a natural proof. I also read somewhere that if they are different, then that fact probably isn't independent of the axioms. On the other hand, the problem is not really my business.

    Relevant: Of course.

    I suppose that P = BPP and that neither NP nor BQP contain each other. Why not believe all of this? I suppose also that there are instances of Abelian groups that cannot be analyzed in P or BPP, even though all Abelian groups can be analyzed in BQP. I am not convinced that graph isomorphism is in BQP. On the other hand, I am not convinced that graph isomorphism is not in P because there is no theory of hard instances.

43. **David Lewis** P $\neq$ NP will be proven on March 13, 2027.

    Techniques: Combinatorics, computability theory. Breakthrough will come from a new Turing-equivalent machine model with a different perspective on time in computation. Before P vs. NP is solved, this will enable meaningful polynomial lower bounds on NP-complete problems to be achieved. The real embarrassment about P vs. NP is not that we can't solve it, but that we can't even show SAT is unsolvable in quadratic time!

    GI*, Factoring:* GI $\in$ P, factoring is in P.

    Quantum Computers: I won't live to see one that is useful except for simulating quantum systems.

44. **Richard Lipton** P = NP will be solved on Dec 12, 2012. Ken Regan and I feel *The End is Near*. Ken guessed 2030–2040. 2030 is only 19 years away. Now compare that to the *Natural Proofs* paper of Alexander Razborov and Stephen Rudich was almost 19 years ago. It seems to follow that any civilization advanced enough to contact us, such as the Vegans in Carl Sagan's *Contact*, or the Vogones in *The Hitchhiker novels* would already know the answer. We differ on whether they would tell us the answer, with Dick saying yes, just before sending us off to coincide with the Mayan Quinquemillennium.

    The approach based on upper bounds, as used by Ryan Williams, seems to me to be the best chance.

Relevant: Yes.

Factoring: This is in time $n^{(\log \log n)/100}$. Actually, lets go all the way and guess that it's in polynomial time. As to when it will done, I think there are two times: when it is solved and when it is made public. The first is now. The second is 2020.

Mind you, all of these projected answers should be treated as *quantum noise*—we weren't even calling them predictions. To quote Anil Nerode's remarks from Bill's original poll:

Being attached to a speculation is not a good guide to research planning. One should always try both directions of every problem. Prejudice has caused famous mathematicians to fail to solve problems whose solution was opposite of their expectations, even though they had developed all the methods required.

45. **Ray Miller** P $\neq$ NP. As I remember the discussions by some of the experts that considered this when I was at IBM research and at theory conferences, this was the prevailing view, and may still be.

    When: I cannot say, but I believe it will be a long time before it is solved. Possibly it will take 50 or more years from now.

    Techniques: I have no idea, but think new proof techniques will be required.

    Relevance: Much has been done with new algorithms for approximate solutions for the problems that appear to be in the NP realm, thus somewhat circumventing the need to know whether P = NP or not, so it probably won't be that important for a practical prospective, however this may change if computer speeds really increase dramatically—say by quantum computing.

46. **Hunter Monroe** P $\neq$ NP, NP $\neq$ co-NP, and PH does not collapse. It will be resolved on June 27, 2012 (one year from today).

    Techniques: Basic computability theory will be used to show that the complement of the bounded halting problem has infinitely often superpolynomial speedup (a stronger statement than P $\neq$ NP). The proofs will fit on one page and be intelligible to undergraduates. A clever diagonalization argument will make an end run around the BGS oracle constraint. It will also be shown that there is no fastest algorithm for most familiar computational problems in particular integer and matrix multiplication (as a machine independent statement, these have hard to minimize circuits). It will be shown that the reason these problems in Boolean form (BOOLEAN CONVOLUTION and BOOLEAN MM) have gaps between their monotone and nonmonotone circuit complexity (these gaps are known facts) is a closely related property, but that NP-complete problems such as HAMILTONIAN CIRCUIT do not have a gap, and that the latter has easy to minimize circuits (essentially enumerating all possible Hamiltonian circuits).

    Relevance: Absolutely. Mathematical research is not a trivial endeavor. Its nontriviality reflects undecidability pushing back at you.

    The hardness of inverting multiplication (factoring) will be proven using the undecidability of arithmetic with multiplication and the decidability of arithmetic without multiplication (the Presburger arithmetic). Graph Isomorphism will have some form of speedup. Randomized algorithms will be found to have no special advantage against NP-complete problems such as

HAMILTON CIRCUIT, but are useful against precisely problems with an undecidable flavor, because randomization skirts the law of the excluded middle (something can be true or false with a probability).

47. **Dana Moshkovitz** P $\neq$ NP. I think that the tools we need for the proof haven't been discovered yet. If I had to, I would bet on a resolution in a few hundreds of years from now.

    Techniques: Everything we already have and much more.

    Relevant: Yes. If anything, we keep discovering more and more reasons for why the question is relevant—we keep hitting more and more barriers for algorithms that boil down to P vs. NP.

48. **Ketan Mulmuley** P $\neq$ NP. The technique will be Geometric complexity theory (GCT). GCT shows that deep upper bound problems at the frontiers of algebraic geometry and representation theory are hidden underneath the P vs. NP and related problems. Hence, though going via algebraic geometry is not formally necessary, doing so would amount to reinventing the wheels of this field that have been developed over centuries.

49. **Ryan O'Donnell** P $\neq$ NP. It will be resolved in about 50 years. No idea what kind of techniques. It's an important problem in pure theory/mathematics, independent of SAT Solvers.

    I guess I believe all the *"standard" conjectures*: P = BPP, NP $\neq$ P, BQP $\neq$ P, the ETH (exponential time hypothesis, which states that SAT requires $2^{\Omega(n)}$ time). I believe the Unique Games Conjecture. I might guess that Graph Isomorphism is in P. About Factoring I have no idea.

50. **Jim Owings** From a philosophical point of view, there is no way that P = NP could be true. If it is true, then that would be an accident of nature. However, accidents do happen. My vote is yes, it is true, P = NP.

    The techniques would have to be unheard of at the present time. I have no way of supporting my position, but I am not just trying to be contrarian. There are things in mathematics that we know are true (such as the Collatz problem), but which we have no hope of proving. This is not a good analogy, though, because the Collatz conjecture is intuitively true, whereas P = NP is intuitively false.

51. **Andy Parrish** P $\neq$ NP. The key breakthrough will be found in 2036, due to increased focus for the 100th anniversary of Turing's paper "On Computable Numbers," but it will take a few years to close all gaps. The proof will be inspired by state-of-the-art machine learning techniques.

    Relevant: Yes—we always take for granted the problems that are "easy" with current technology, and strive to solve "hard" problems.

52. **Chris Pollett** As in the last poll, I think P $\neq$ NP. It will hopefully be proven before 2020.

    In the last poll, I thought that a sequence of results that stronger and stronger logical systems could not prove P = NP would eventually lead to an outright result that P $\neq$ NP. I still like this approach, but I could also imagine something where a combination of diagonalization together with everything under the sun a la Williams might yield a proof.

    I am still guessing P = BPP.

53. **Jim Purtilo** Does P = NP or not? Dibs on "I don't know and I don't care." I saw that on a T-shirt once. It will be resolved at Closing time after whatever is the hotel bar at which FOCS is next held. (It will of course then be unresolved again the next morning.) The techniques will be Turbo-encabulation.

    Will the problem still be relevant given advances in algorithms and in SAT *Solvers?* It's relevant now?

    Feel free to comment on anything else: Graph Isomorphism, Factoring, Derandomization, Quantum computers, and/or your own favorite problem.

    I think the GOP is screwed and has no decent bench to challenge Obama in 2012.

54. **Ken Regan** P $\neq$ NP. It will be solved in 2030–2040. Algebraic Geometry including deeper aspects of cohomology theory than have been used before. See the 2005 survey paper *Betti Number Bounds, Applications and Algorithms* by Saugata Basu, Richard Pollack, and Marie-Francoise Roy in Current Trends in Combinatorial and Computational Geometry: Papers from the Special Program at MSRI. (It's also at CiteSeer.) See also Joel Friedman's work *Cohomology in Grothendiek Topologies and Lower Bounds in Boolean Complexity* at `http://arxiv.org/abs/cs/0512008` and its sequel `http://arxiv.org/abs/cs/0604024`.

    Relevant: Yes. I have not seen a treatment of the idea that SAT instances resulting from natural reductions to SAT tend to be harder than instances of the same size that show up in test suites for these solvers.

    GI*, Factoring, Derandomization:* GI $\in$ P. I am sympathetic to the view that Factoring is in P, or at least in BPP. I am not sure I believe the strong hypothesis of $2^{\Omega(n)}$ size lower bounds on non-uniform circuits for problems in exponential time, on which major de-randomization results are based. Something tells me a $2^{\Omega(n/\log n)}$ upper bound may be possible.

    Quantum Computing: Generally I agree with Gil Kalai's standpoint in his paper *How Quantum Computers Fail* (paper is here: `http://arxiv.org/abs/1106.0485`, slides on the topic are here: `http://gilkalai.files.wordpress.com/2011/06/iqi.pdf`) that quantum noise might not follow the assumptions needed for the Fault Tolerance Theorem.

55. **Jonathan M. Rosenberg** P $\neq$ NP. I suspect this is like the problem of *intermediate growth* in combinatorial group theory. Originally no examples were known; now there are methods of generating lots of them. I think it will be solved in 10 years. The reason why it's taking so long is that the tools have to be developed from scratch.

    Relevant: Not so much. It may not make much difference for many practical problems.

56. **Michael Sipser** As far as I know, not much has changed mathematically relevant to P vs. NP since 2002 so I have nothing to add to my previous response to your questionnaire. Use my prior response; however, update it so that I predict it will be solved 25 years from now, not from 2002. Here is the old response updated.

    As you may know, when I was a graduate student in the mid 1970s I predicted that it would be solved by the century's end. I also bet Len Adleman an ounce of gold that I would be right. Now that I've paid off, I'm more reluctant to make a prediction once again. But I'll go out on a limb and give it another 25 years, so by around 2037. And I'll stick with my

earlier prediction that the resolution will be a proof that P $\neq$ NP. The technique would be combinatorial, but that isn't saying much. No more bets, however.

57. **Peter Shor** P $\neq$ NP. It will be resolved 30 years from now (at the ever-receding research horizon). I might as well add my 2 cents. Whenever anybody has some possible but very unlikely future technology, like fusion, the horizon is always 30 years away. Fusion energy has been 30 years away for the past 70 years. So I'll say the same thing: P vs. NP will be resolved 30 years from now.

    Techniques: Deep Math.

    Relevant: It will definitely be relevant. Advances in algorithms can only take you so far in the face of NP-complete problems. There will be significant advances solving in NP-complete problems more efficiently, but they will still be seen as hard.

    NP-complete problems won't be possible on quantum computers, either.

58. **John Sidles** The P vs. NP question will be proved undecidable by 2014, by a graduate student. The proof will involve reflecting upon Turing machines in P that verify their own non-verifiability.

    Relevant: Narrowly no; broadly yes.

    This and similar proofs will help build a better-integrated understanding of simulation, sampling, and verifying; this unification will be a central theme of the 21st century STEM enterprise.

    (Scott's question) Given that P $\neq$ NP is undecidable in ZF/C, is it nonetheless true? (Maimonides' answer) *Though the Messiah may tarry, yet do we await him each day* and for this reason we will regard P $\neq$ NP as true ... secure in the knowledge that no formal disproof will ever be given, and the faith that no practical disproof will ever be given.

59. **Steven Skiena** P $\neq$ NP. A reasonable estimator for rare events is to double the interval to date without the event. If we take the question as arising seriously about 1972, this means it will be solved in 2052.

    Techniques: Nothing I will find myself understanding. Perhaps it will be solved by a non-computer scientist.

    Relevant: P versus NP seems to have become of perhaps less practical interest but increased theoretical interest over the past ten years. Complexity theoretic issues seem to impact bigger areas of math/science (e.g., quantum physics and game theory), but the practical stakes for negative anticipated result seem small.

    Relatively little of these topics (Graph Isomorphism, Factoring, Derandomization, Quantum Computers) will make their way into future undergraduate algorithms courses, similar to how the basic theory taught in Calculus courses does not advance these days. Almost all of the theory that I teach in my undergrad algorithms course was known 15 to 20 years ago. Of course, there will continue be advances at the graduate/research level.

60. **Dan Spielman** P $\neq$ NP will be proven in 2050.

    Relevant: No, it will not be relevant. But, its irrelevance will be due to the ubiquity of quantum computers. The question we will want answered by then is whether NP is in BQP.

Graph isomorphism will be shown to be in time $n^{O(\log n)}$.

61. **Y. C. Tay** An alien who knows the answer may say: "Not, but that's not the right way to formulate the problem."

    When: After a reformulation of computational complexity.

    Techniques: Non-set-theoretic techniques.

62. **Denis Therien** P = NP will be proven on June 4, 2029 (my 75th birthday) by finding an algorithm for an NP-complete graph question.

    Relevance: No.

63. **Moshe Vardi** Several years ago Ron Fagin collected "bets" on the outcome of the P versus NP question. I believe that I am one of the very few people who placed nontrivial odds in FAVOR of P = NP. When asked to justify my bet, I answered that it is essentially a "protest vote." I do not really have any deep intuition in favor of P = NP. I do not, however, believe that the evidence in favor of P $\neq$ NP is as strong as it is widely believed to be. The main argument in favor of P $\neq$ NP is the total lack of fundamental progress in the area of exhaustive search. This is, in my opinion, a very weak argument. The space of algorithms is very large and we are only at the beginning of its exploration. Witness the non-constructive tractability proofs in the area of graph minors and tractability proofs in the area of group theory that are based on the very deep classification of finite simple groups. The resolution of Fermat's Last Theorem also shows that very simple questions may be settled only by very deep theories. Over the two decades we have seen several major lines of attack on the P versus NP question. I myself was involved on one of them, that of finite-model theory. All these lines of attack yielded beautiful theories, but there is little reason to believe that they led us any closer to resolving the problem.

    When: I think it is impossible to give an intelligent answer to this question.

    Techniques: I think it is impossible to give an intelligent answer to this question.

    Relevance: While the P vs. NP quandary is a central problem in computer science, we must remember that a resolution of the problem may have limited practical impact. Recall that the difference is between theory and practice is that *in theory, they are not that different, but in practice, they are quite different.* This seems to apply to the theory and practice of SAT and similar problems. An important role of theory is to shed light on practice, and there we have large gaps. We need, I believe, a richer and broader complexity theory, a theory that would explain both the difficulty and the easiness of problems like SAT.

64. **Avi Wigderson** My response from 10 years ago did not change, so I did not offer a new one. Here is his response from 10 years ago:

    I think this project is a bit premature. I think we know too little of what is relevant to even guess answers to your questions, certainly if "we" is replaced by "I".

    The only thing I can definitely say, is that it is one of the most important and interesting questions every asked by humans, and more people and resources should participate in filling up the holes that would allow better guesses of answers to your question.

65. **Ryan Williams** I lean towards P $\neq$ NP, but I would not bet anything significant on it. I think it is premature to conjecture which way the question will be resolved. We cannot rule out linear time SAT algorithms. Common sense says that the universe is simply not nice enough that P should equal NP. But I don't know how to justify that formally.

    Do we believe P $\neq$ NP because the universe ain't wired that way, or is it because we are too ignorant to find these great algorithms? (I do believe that either P = NP or P $\neq$ NP can be proved within existing mathematical foundations.) Perhaps someone would counter that *because* we are too ignorant to presently resolve P vs. NP this in itself is reason for believing P $\neq$ NP. That's about as anthropocentric as an argument can get. Conjectures should be based on knowledge, not ignorance.

    When: Within my lifetime.

    Techniques: What I am about to say is nonsense (in case that doesn't become obvious), but I hope it will be productive nonsense. The proof of either P = NP or P $\neq$ NP will use a number of new ideas that strengthen connections between the analysis of algorithms and proofs of lower bounds. Note this prediction includes the possibility of using geometric complexity theory, but does not exclude completely different approaches. If P $\neq$ NP, the first proof will be a proof by contradiction. Perhaps a proof of P = NP will rise from the ashes of a strong failed attempt to prove the opposite. Strassen's algorithm for matrix multiplication was apparently discovered by trying to prove lower bounds on multiplying 2 by 2 matrices.

    Relevance: Yes.

    Derandomization NEXP $\neq$ BPP ought to be proved within the next ten years. Unlike P vs. NP, I would bet on LOGSPACE $\neq$ NP being proved soon. The problem seems significantly more tractable to resolve. I also predict that someone out there reading the responses to this poll will ignore all of us, and go discover something wonderful that we never even considered.

66. **Joshua Zelinsky** P $\neq$ NP. Relevant: Yes. Some people seem very confident that P = BPP. They confuse me.

    Relevant: Yes.

67. **Doron Zeilberger**

    (a) *Do you think* P = NP *or not? You may give other answers as well.* P $\neq$ NP.

    (b) *When do you think it will be resolved?* Before 2030.

    (c) *What kinds of techniques do you think will be used?* Circuit-lower bounds heavily assisted by large-scale computer computation, like the Kepler conjecture and the Four Color Theorem.

    (d) *Will the problem still be relevant given advances in algorithms and in* SAT *Solvers?* It was never relevant, it is a nice but irrelevant math challenge. Its *practical* interest is based on the flawed assumption that *polynomial* is fast. If the degree is a googolplex, it is not that fast. Also on the *asymptotic fallacy*, and the *asymptotic snobbery* of mainstream complexity theory. Life is finite, and often *asymptotics* may take zillion of big bangs to kick in.

(e) *Feel free to comment on anything else: Graph Isomorphism, Factoring, Derandomization, Quantum computers:* Like most major questions in the history of mathematics, P vs. NP is a stupid question, whose answer is obviously yes, but for which is may be either very difficult, or impossible to prove, because it is unnatural. It is reminiscent of irrationality questions in number theory. It is easy enough to prove that $\sqrt{2}$ is irrational. It is trivial to prove that e is irrational, and not too hard to prove that π is irrational. But no one has any clue how to prove (rigorously) that $e + \pi$ is irrational. The probability that it is rational is $\aleph_0/aleph$ which is a very small 0, but the notion of *fraction of integer* does not live well with the definition of $e + \pi$.

(f) Shannon proved that most Boolean Functions have super-polynomial circuits, there is no reason that the optimal circuit for, say, CLIQUE, would be polynomial size, CLIQUE being pretty *random*. So P $\neq$ NP is obviously true (as true as the fact that $e + \pi$ is irrational). Proving either (and tons of other obviously true stupid questions that mix apples and oranges, like the Goldbach conjecture) is extremely difficult and the questions are artificial.

(g) (Bill's advice:) See Zeilberger's opinion number 76 titled *Why* P *Does Not Equal* NP *and Why Humans Will Never Prove It by Themselves*, `http://www.math.rutgers.edu/~zeilberg/Opinion76.html`.

68. **David Zuckerman** P $\neq$ NP. When: Of course I have no idea, but given the lack of progress within the last 40 years, I will guess about 100 years.

    Relevant: Yes, of course.

    Derandomization: I believe strongly that BPL = L and BPP = P.

    Factoring: I would guess Factoring is not in P; on the other hand, conditioned on someone resolving the complexity of Factoring in my lifetime, I'd guess that it's in P, because that would be easier to prove.

    Graph Isomorphism: If forced to guess, I'd probably say it's in P.

69. **Anonymous** P $\neq$ NP: It will never be resolved, but if it is it might come from advances in circuit complexity; but more likely it would be something we haven't thought of yet.

    Relevant: Very much so.

    Perhaps twenty years ago there was the danger that because the P vs. NP problem was so seemingly intractable that there was no place for complexity to go. Fortunately that has not at all occurred and there are many very interesting problems within reach.

70. **Anonymous** P = NP will be shown between 50 and 100 years from now. To show that there exists an algorithm whose running time is $n^{O(1)}$ because there are only finitely many obstacles to nonexistence. However, the proof will be non-constructive, and perhaps another century will go by before the exponent is shown to be at most $10^{10^{10}}$. (Or perhaps it will be shown to be uncomputable in some sense.)

    A similar situation arises with respect to graph classes. We know from the Robertson–Seymour Theorem that linear algorithms exist to solve all questions of the form *does this graph belong to minor-closed class X*, yet in most cases we have no way to discover the algorithm in our lifetimes.

When you ask *will it still be relevant* do you mean at the time the question is answered? or next year? SAT solvers do some amazing things but they fail disastrously on many problems. The world has lots and lots of cases where we can solve special cases of *unsolvable* problems in a reasonable time, hence unsolvable problems have never scared me; but don't ask me to find a tough leaper tour with a SAT solver. The way most people seem to think the P versus NP is relevant is I think really irrelevant; yet I definitely believe the question is highly relevant, for mathematical elegance rather than for application in practice.

If the outcome to P versus NP turns out as I predict, Mike Paterson said long long ago that we will know we asked the wrong question. But I'm not sure that we did. There will just have to be another way to dichotomize between what we can do in practical polynomial time and what we can do polynomially only *in principle.*

71. **Anonymous** I'd like to hope P = NP since it would be more satisfying to me and I like to think that cleverness could do something. I think it will be resolved out of the blue. I don't know what techniques it will use, but I think they might be from outside of CS/Math in some way.

    Relevance: Yes

    Other problems: It annoys me that the jump from 2-SAT to 3-SAT is so big but that from 3-SAT to 4-SAT isn't.

72. **Anonymous** It is possible that P = NP. We have not yet exhausted possibilities for new paradigms of algorithms. But I do not rule out P $\neq$ NP. If in fact P = NP (or R = NP) then solution may arrive in the coming decade. If P $\neq$ NP then proof may require many many years.

    The reason for thinking that if, in fact, P = NP an algorithm may be found sooner than a proof in case P $\neq$ NP, is that computer scientists have already found many powerful forms of algorithms, whereas methods for proving lower bound results are very weak. In either case new breakthrough techniques will be required.

    Relevance: The problem is of fundamental importance. An efficient algorithms will change the world. A proof that P $\neq$ NP, especially if average high complexity will be established for some subsets of NP problems, will provide a provable basis for modern cryptography.

    Factorization: is a promising, fruitful problem. Especially attempting to improve existing (non-polynomial) algorithms to enable practical factorization of larger integers.