d on his independent 1971 that defined the P vs. NP By the time Levin's paper he west, P vs. NP had already d itself as computing's rtant question.

pagliazzo, in a classic 1995 described five worlds with grees of possibilities for the oblem:

thmica: P = NP or something equivalent," such as fast ic algorithms for NP.

stica: NP problems are hard st case but easy on average.

and: We can easily create problems, but not hard NP where we know the solution.

e worst of all possible worlds, an neither solve hard prob- erage nor do we get any ap- ptographic advantage from lty of these problems.

ypt: Cryptographic one-way exist, but we do not have cryptography.

mania: Public-key cryptog- ssible—that is, two parties nge secret messages over nels.

orlds are purposely not for- ned but rather suggest the possibilities given our of the P vs. NP problem. al belief, though not univer- we live in Cryptomania.

iazzo draws upon a "you it all" from P vs. NP theo- an either solve hard NP or have cryptography, but have both (you can have Perhaps, though, we are to a de facto *Optiland*. Ad- machine learning and op- in both software and are allowing us to make on problems long thought r impossible—from voice n to protein folding—and e most part, our crypto- otocols remain secure.

tion called "What if P=NP?" 009 survey,[13] I wrote, "Learn- es easy by using the princi- am's razor—we simply find st program consistent with Near-perfect vision recogni- uage comprehension and translation, and all other learning tasks become trivial. We will also have much better predictions of weather and earthquakes and other natural phenomenon."

Today, you can use face-scanning to unlock your smartphone, talk to the device to ask it a question and often get a reasonable answer, or have your question translated into a different language. Your phone receives alerts about weather and other climatic events, with far better predictions than we would have thought possible just a dozen years ago. Meanwhile, cryptography has gone mostly unscathed beyond brute-force-like attacks on small key lengths. Now let's look at how recent advances in computing, optimization, and learning are leading us to Optiland.

Solving Hard Problems

In 2016, Bill Cook (no relation to Steve) and his colleagues decided to tackle the following challenge:[9] How do you visit every pub in the U.K. in the shortest distance possible? They made a list of 24,727 pubs and created the ultimate pub crawl, a walking trip that spanned 45,495,239 meters—approximately 28,269 miles—a bit longer than walking around the earth.

Cook had cheated a bit, eliminating some pubs to keep the size reasonable. After some press coverage in the U.K.,[7] many complained about missing their favorite watering holes. Cook and company went back to work, building up the list to 49,687 pubs. The new tour length would be 63,739,687 meters, or about 39,606 miles (see Figure). One needs just a 40% longer walk to reach more than twice as many pubs. The pub crawl is just a traveling salesman problem, one of the most famous of the NP-complete problems. The number of possible tours through all the 49,687 pubs is roughly three followed by 211,761 zeros. Of course, Cook's computers don't search the whole set of tours but use a variety of optimization techniques. Even more impressive, the tour comes with a proof of optimality based on linear program duality.

Taking on a larger task, Cook and company aimed to find the shortest tour through more than two million stars where distances could be computed. Their tour of 28,884,456 parsecs is within a mere 683 parsecs of optimal.

Beyond Traveling Salesman, we have seen major advances in solving satisfiability and mixed-integer programming—a variation of linear programming where some, but not necessarily all, of the variables are required to be integers. Using highly refined heuristics, fast processors, specialized hardware, and distributed cloud computing, one can often solve problems that arise in practice with tens of thousands of variables and hundreds of thousands or even millions of constraints.

Faced with an NP problem to solve, one can often formulate the problem as a satisfiability or mixed-integer programming question and throw it at one of the top solvers. These tools have been used successfully in verification and automated testing of circuits and code, computational biology, system security, product and packaging design, financial trading, and even to solve some difficult mathematical problems.

Data Science and Machine Learning

Any reader of *Communications* and most everyone else cannot dismiss the transformative effects of machine

Shortest route through 49,687 U.K. pubs. Used by permission. (http://www.math.uwaterloo.ca/tsp/uk).