whaling practiced until some 18 years ago. If whalers consistently took the larger, older individuals, he suggests, the groups may have "lost their social knowledge and may be less successful." **–ELIZABETH PENNISI**

QUANTUM COMPUTING

Souped-Up Software Gets a Virtual Test

Amazing things, quantum computers. On paper, they can outpace conventional computers a billionfold, bringing new worlds of computation within human reach. The only hitch is that no one has built one that does that yet. That raises a practical problem for designers of quantum software: How do you test a potential "killer app" for a machine that doesn't exist?

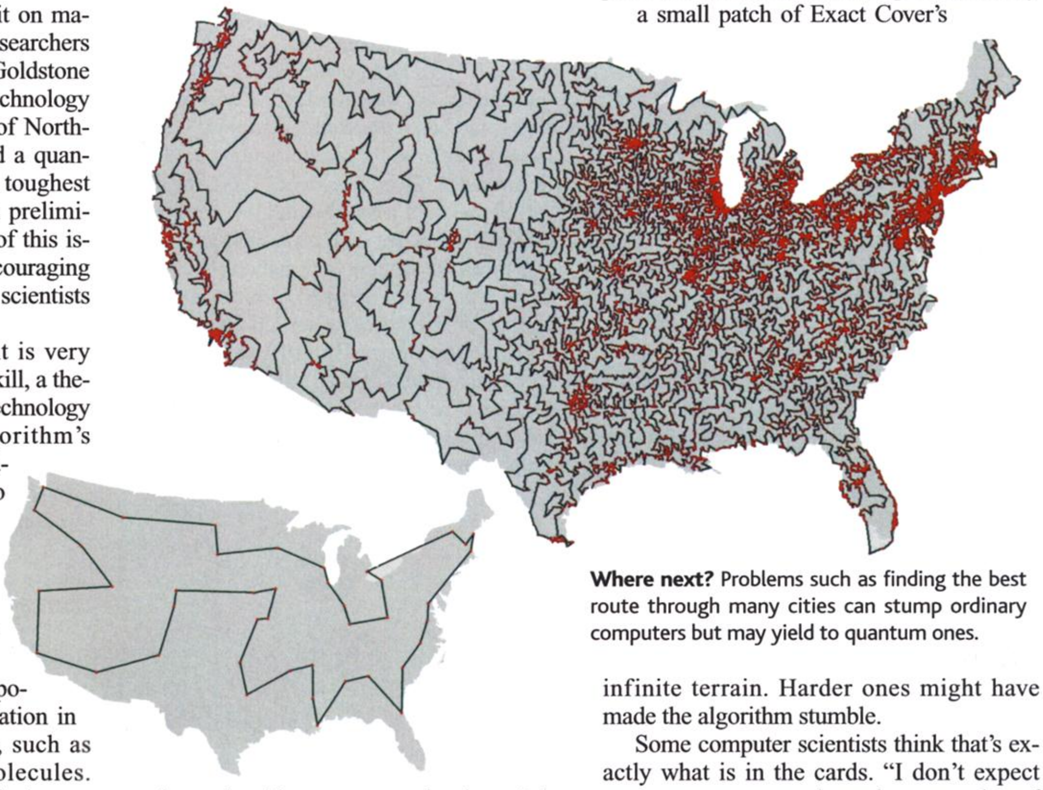
If you have time, you can run it on machines that do exist. That's how researchers led by Edward Farhi and Jeffrey Goldstone of the Massachusetts Institute of Technology in Cambridge and Sam Gutmann of Northeastern University in Boston pitted a quantum algorithm against one of the toughest problems in computer science. In preliminary tests, described on page 472 of this issue, the algorithm racked up an encouraging virtual track record that left some scientists hankering for more.

"If it is truly powerful, then it is very broadly applicable," says John Preskill, a theorist at the California Institute of Technology in Pasadena. Although the algorithm's prospects remain "highly speculative," Preskill says, "the incentive to press forward with the daunting task of building large-scale quantum computers will be greatly strengthened if quantum computers are really as powerful as the work of Farhi *et al.* suggests."

The dream machines get their potential power from storing information in objects that obey quantum laws, such as electrons, atomic nuclei, or molecules. Whereas each bit stored in a classical computer can take on only one of two values—0 or 1—the "qubits" in a quantum computer can also exist in a strange state called superposition, in which, in a sense, they possess every possible value at once. That gives quantum computers an amazing knack for parallel processing, raising hopes that they might conquer problems that ordinary classical computers can't handle.

The Mount Everest of computer science is a class of problems known as NP-complete. Algorithms designed to solve NP-complete problems mushroom exponentially into impossibly long calculations as the size of the input increases. One famous example is to find the most efficient route for a traveling salesman who must visit every city on his map once and only once. As the number of cities increases, the problem quickly becomes so complex that, in general, conventional computer algorithms can't solve it for more than a few thousand cities. (The current world record is 3038.)

To tame that exponential monster, computer scientists are hunting for a problem-solving algorithm whose run-time grows more slowly, with some power of the size of the input. One such "polynomial time" algorithm is all they need, because mathematicians have proved that any algorithm that solves one NP-complete problem in polynomial time will crack every other NP-complete problem, too. Last year the Clay Mathematics Institute in Cambridge, Massachusetts, offered a $1 million bounty to anyone who either writes such an algorithm or proves that it can't be done (*Science*, 26 May 2000, p. 1328).



Where next? Problems such as finding the best route through many cities can stump ordinary computers but may yield to quantum ones.

SOURCES: (TOP TO BOTTOM) APPLEGATE, BIXBY, CHVATAL, AND COOK (1998); GEORGE DANTZIG, RAY FULKERSON, AND SELMER JOHNSON (1954)

An NP-complete problem, Farhi and his colleagues decided, was just the thing for road-testing a virtual quantum computer. A year earlier, they had devised a way to program a quantum computer to solve an NP-complete problem called Exact Cover. Exact Cover is like Twenty Questions played with bits: Given a series of rules describing a string of ones and zeroes, the player must decide whether the string exists. The quantum algorithm "isn't clever," Farhi says, but it always gets a solution sooner or later. "The question is how long is long enough."

To find out, the scientists programmed a cluster of workstations to simulate a quantum computer running the algorithm, by running in sequence the operations that a quantum machine would perform simultaneously. Then they fed it various combinations of rules and waited for it to crank out the answers. Although the problems weren't difficult (a nonquantum desktop PC could have solved each one in a fraction of a second, Farhi says), the simulation took days to find each solution. The quantum run-time, it turned out, grew in proportion to the length of the bit string, squared. That put the algorithm solidly within polynomial time—the realm of practical solvability.

Time to alert the Clay Institute? Unfortunately not, Farhi says. Even if quantum algorithms qualify for the prize, a few promising results are a far cry from a mathematical proof, he points out. Besides, the simple problems in the simulation represented only a small patch of Exact Cover's infinite terrain. Harder ones might have made the algorithm stumble.

Some computer scientists think that's exactly what is in the cards. "I don't expect any quantum approach to give a speedup of NP-complete problems in polynomial time," says Charles Bennett, a quantum-computing researcher at IBM's Thomas J. Watson Research Center in Yorktown Heights, New York. To do that, he thinks, an algorithm would have to target some still-unknown Achilles' heel in the problems themselves—a feat he considers unlikely.

Preskill, however, is guardedly optimistic about the algorithm. Although the evidence is still "far from conclusive," he says, "I think it is a promising idea that ought to be pursued aggressively." Farhi says that's just what he has in mind. **–MARK K. ANDERSON**

Mark K. Anderson is a writer in Northampton, Massachusetts.