A Fast Way to Solve Hard Problems

A new algorithm to solve linear programming problems is so fast that experts in the field are taken aback

A young mathematician at AT&T Bell Laboratories has found a new and extremely successful way to solve some of the most common problems put on a computer. The problems are linear programming problems and the method, devised by Narendra Karmarkar, is so quick and so versatile that even experts in the field are astonished. "It's almost a shocking result," says Michael Garey of AT&T Bell Laboratories.

Linear programming problems frequently arise in business and also are common in such fields as engineering, agriculture, and the social sciences. Karmarkar's method not only appears to be substantially faster than the current method but it also can be applied to problems that are too large or complex for the current method to handle.

Until now, linear programming problems have been solved with the simplex method, which was devised about 40 years ago by George Dantzig of Stanford University. The method, which is quite fast to begin with, has been honed down to its essentials by generations of computer scientists and is sold commercially in the form of a highly efficient assembly language code.

But Karmarkar's method promises to beat the simplex. Even now, when his program is written in the higher level, and therefore slower, language of Fortran, Karmarkar's method is faster. He tested his method against what is said to be the fastest simplex program available—an IBM assembly language program. When he tried to solve problems consisting of 5000 variables, the size of a typical linear programming problem, the IBM program took 50 times longer to find the solution. And the larger linear programming problems become, the greater the advantage that Karmarkar's method has over the simplex.

Linear programming problems are algebraic monstrosities. They typically consist of thousands or tens of thousands of linear inequalities. The objective is to choose values for the variables so as to maximize or minimize a linear function and, at the same time, satisfy the inequality constraints.

An often-used example of a linear programming problem is the "diet problem." A person has certain dietary requirements—so much vitamin A, so much vitamin C, so much fiber, so much protein, and so on. He wants to select his foods so that all his nutrient requirements are met and the food costs as little as possible. The constraints are the equations saying that the amount of each nutrient must at least equal the minimum daily requirement. The variables are the amounts of each food that must be eaten. And the function that must be minimized is the one that says the total cost of the diet should be as low as possible.

Geometrically, linear programming problems look like multidimensional polygons, called polytopes, in high-dimensional spaces. The solution that is sought is a vertex of the polytope. The problem is to find that vertex. The simplex method does it by hopping from vertex to adjacent vertex, searching for the best one. The number of vertices can easily exceed 10^{100} and if the computer had to check each one, the algorithm would take so long that it would never find the solution. In practice, however, the simplex algorithm finds the best vertex without searching through them all, although the method can still be foiled by occasional problems which it cannot solve in a reasonable length of time.

> For problems with 5000 variables, the IBM program took 50 times longer.

The simplex method is an example of what computer scientists call an exponential time algorithm, meaning that the length of time it takes to solve a problem increases, at least for some examples, as an exponential function of the size of the problem.

Although mathematicians were basically satisfied with the simplex algorithm, they have been tantalized by the possibility that a much better method might be found. A few years ago, their hopes were raised when a young Russian investigator published a report of a new method, called the ellipsoid algorithm, for linear programming problems (*Science*, 2 November 1979, p. 545). This was a so-called polynomial time algorithm—the holy grail of computer science. Such methods run in a length of time that is only a polynomial function of the number of variables in the problem. The difference between an exponential time algorithm and a polynomial time one can be substantial. A polynomial time algorithm requiring x^3 steps can take only 1/5 of a second of computer time when x is 100. An exponential time algorithm requiring 3^x steps might require billions of centuries of computer time when x is 100.

The Russian algorithm for linear programming problems proved to be disappointing, however. Its running time was a function of such a large degree polynomial that it was far slower than the simplex method for the sorts of problems that occur in everyday life—although it might beat the simplex for very large problems of the sort that usually do not occur in practice.

What Karmarkar has done is to find a different sort of polynomial time algorithm for linear programming problems—one that fulfills the potential of this class of methods. Instead of having the computer search from vertex to vertex of the multidimensional polytope to find the best one, he "plunges into the interior," says Ronald Graham, director of mathematical and statistical research at AT&T Bell Laboratories. He closes in on the sought after vertex by creating a sequence of spheres inside the polytope that converge to it. Kamarkar's method does its work in a strange mathematical space. In this space, the polytope is transformed to a regular geometric figure that is the multidimensional analog of an equilateral triangle. This warping of the original space, says Graham, is "a little like the hyperbolic geometry of relativity theory. It is like looking at the problem through a lens that distorts, especially at the edge of the figure."

Karmarkar can prove that his method has to be fast. His analysis guarantees that for a polytope in n dimensions, at every step he will move at least $1/n$th of the way to the solution. And in practice, he does much better. In addition, he says, his algorithm should have new applications. For example, Karmarkar says, it should be possible to use parallel processors to speed up the solutions to linear programming problems to such an extent that they can be obtained in "real time"—almost instantaneously. Thus it could be used in pattern recognition problems and in computer programs that pilot airplanes, which are instances where linear programming problems were simply untouchable.

The method also should enable researchers to solve enormous linear programming problems, which have as many as a million inequalities and were long considered impossible to solve. Such problems arise in the design of traffic routing in the telephone system.

In addition, Karmarkar anticipates that his method should solve nonlinear programming problems, which resemble linear programming ones except that the functions are nonlinear and instead of having a polytope with edges, you have a multidimensional figure with curved faces. Such problems arise in models of the economy, for example.

Although the computer science and operations research communities are taken aback by Karmarkar's result, they are not entirely surprised that a practical polynomial time algorithm for linear programming problems has been found. What surprises them is that it is so efficient. As Lászlo Lovász, the researcher from the University of Budapest who helped valiate the ellipsoid algorithm and make it more efficient, says, "I would have expected that someone would have first found a polynomial time method that was comparable to the simplex and that, after a time and with refinements, it might be better." The incredibly efficient algorithm that Karmarkar found seems "spectacular", according to Lovász.

Karmarkar's algorithm is still so new, however, that its implications are yet to be fully established. He got his theoretical results last year and started implementing the method and testing it against the simplex this summer. But one thing is certain, says Graham, "People will be experimenting a lot with this method."

—GINA KOLATA

Decision Near on *Galileo* Asteroid Flyby

Less than a year ago, the scientists in charge of the upcoming *Galileo* mission to Jupiter realized that the spacecraft would have a chance to explore a major asteroid on the way; the National Aeronautics and Space Administration (NASA) is now very near a decision on whether or not to seize that opportunity. For once the issue is not money. It is management: can the *Galileo* operations team at the Jet Propulsion Laboratory handle the extra work load without jeopardizing the mission as a whole?

The object in question is 29 Amphitrite, namesake of the legendary wife of Neptune and, at 200 kilometers diameter, one of the 30 largest objects in the main asteroid belt between Mars and Jupiter. The planetary science community is understandably enthusiastic about having a look: asteroids are thought to be relics of the formation of the solar system and thus have a very high scientific priority. Unfortunately, the first dedicated United States mission to an asteroid is not scheduled until the mid-1990's.

Aside from its relatively large size, Amphitrite is intriguing because of its composition. It is classified as an S-type asteroid, which means that the sunlight reflected from its bright, slightly pinkish surface contains absorption bands of the minerals pyroxene and olivine, as well as the signature of nickel-iron metal.

Since the asteroid belt is thought to be the source of many of the meteorites that fall to Earth, the obvious thing to do with such spectra is to compare them with the spectra of samples in the museum collections. It turns out, however, that the S-type asteroids are ambiguous. On the one hand they match the most common class of meteorites, the chondrites, which are pyroxene-olivine rocks containing millimeter-sized flecks of nickel-iron metal. If Amphitrite is a giant chondrite, then it is a mass of primitive stuff that condensed right out of the original solar nebula, and it has probably remained unaltered for 4.6 billion years.

On the other hand, S-type asteroids also match the stony-iron meteorites, which are solid chunks of metal laced with rock. The stony-irons appear to be relics of chondritic asteroids that somehow melted, allowing the molten nickel-iron to sink through the lighter, rocky magma. Later, after these asteroids had cooled and solidified, they were presumably ruptured by collisions with other asteroids, their crusts stripped away, and their metallic cores laid bare.

So perhaps Amphitrite is an exposed core. If so, a once-over by *Galileo* might help explain where the heat came from to melt the asteroid and why the S-types melted when the others stayed cool.

The *Galileo* encounter with Amphitrite would come on 6 December 1986, nearly 6 months after the May 1986 launch. From a closest-approach distance of possibly 10,000 kilometers, the spacecraft's cameras would image the asteroid with approximately the same resolution that the Voyager cameras achieved among the moons of Saturn. Unlike Voyager, however, *Galileo* also carries a Near-Infrared Mapping Spectrometer that could make a surface composition map. Ultraviolet, thermal, and polarization measurements would also be possible.

The encounter's major cost to the mission would be a 3-month delay in *Galileo's* arrival at Jupiter in 1989, which translates to an additional outlay of some $10 million on top of a total mission cost of nearly $1 billion. "It's not a serious concern," says Geoffery Briggs, head of NASA's plantary science division. "The added outlays come at the end of the mission [in the early 1990's], and *Galileo* is a good candidate for an extended mission anyway."

The real concern, he says, is that the *Galileo* team at the Jet Propulsion Laboratory is already stretched thin, with most of the current activity focused on completing the spacecraft itself. "Do we have the resources—and the management attention—to do the asteroid?" asks Briggs. "We don't want to close our eyes to the value of the science, but we don't want to jeopardize the mission by having people overlook something stupid."

On the other hand, Briggs points out that the JPL team is working hard to pin down just what Amphitrite will involve. He hopes to take a recommendation to NASA administrator James M. Beggs within a few weeks. "I hope by then, with the facts in front of us, that it will be a straightforward decision."—M. MITCHELL WALDROP