PHYS.ORG

# Fundamental algorithm gets first improvement in 10 years

September 27 2010, By Larry Hardesty

Graphic: Christine Daniloff

The maximum-flow problem, or max flow, is one of the most basic problems in computer science: First solved during preparations for the Berlin airlift, it's a component of many logistical problems and a staple of introductory courses on algorithms. For decades it was a prominent research subject, with new algorithms that solved it more and more efficiently coming out once or twice a year. But as the problem became better understood, the pace of innovation slowed. Now, however, MIT researchers, together with colleagues at Yale and the University of Southern California, have demonstrated the first improvement of the max-flow algorithm in 10 years.

The max-flow problem is, roughly speaking, to calculate the maximum amount of “stuff” that can move from one end of a network to another, given the capacity limitations of the network’s links. The stuff could be data packets traveling over the Internet or boxes of goods traveling over the highways; the links’ limitations could be the bandwidth of Internet connections or the average traffic speeds on congested roads.

More technically, the problem has to do with what mathematicians call graphs. A graph is a collection of vertices and edges, which are generally depicted as circles and the lines connecting them. The standard diagram of a communications network is a graph, as is, say, a family tree. In the max-flow problem, one of the vertices in the graph — one of the circles — is designated the source, where all the stuff comes from; another is designated the drain, where all the stuff is headed. Each of the edges — the lines connecting the circles — has an associated capacity, or how much stuff can pass over it.

## Hidden flows

Such graphs model real-world transportation and communication networks in a fairly straightforward way. But their applications are actually much broader, explains Jonathan Kelner, an assistant professor of applied mathematics at MIT, who helped lead the new work. “A very, very large number of optimization problems, if you were to look at the fastest algorithm right now for solving them, they use max flow,” Kelner says. Outside of network analysis, a short list of applications that use max flow might include airline scheduling, circuit analysis, task distribution in supercomputers, digital image processing, and DNA sequence alignment.

Traditionally, Kelner explains, algorithms for calculating max flow would consider one path through the graph at a time. If it had unused capacity, the algorithm would simply send more stuff over it and see what happened. Improvements in the algorithms’ efficiency came from cleverer and cleverer ways of selecting the order in which the paths were explored.

## Graphs to grids

But Kelner, CSAIL grad student Aleksander Madry, math undergrad Paul Christiano, and Professors Daniel Spielman and Shanghua Teng of, respectively, Yale and USC, have taken a fundamentally new approach to the problem. They represent the graph as a matrix, which is math-speak for a big grid of numbers. Each node in the graph is assigned one row and one column of the matrix; the number where a row and a column intersect represents the amount of stuff that may be transferred between two nodes.

In the branch of mathematics known as linear algebra, a row of a matrix can also be interpreted as a mathematical equation, and the tools of linear algebra enable the simultaneous solution of all the equations embodied by all of a matrix's rows. By repeatedly modifying the numbers in the matrix and re-solving the equations, the researchers effectively evaluate the whole graph at once. This approach turns out to be more efficient than trying out edges one by one.

If N is the number of nodes in a graph, and L is the number of links between them, then the execution of the fastest previous max-flow algorithm was proportional to $(N + L)^{(3/2)}$. The execution of the new algorithm is proportional to $(N + L)^{(4/3)}$. The researchers haven't in fact written a program that implements their algorithm, and in practice, the performance of an algorithm can depend on factors like how efficiently it's coded and how well it manages memory. But in theory, for a network like the Internet, which has about 100 billion nodes, the new algorithm could solve the max-flow problem 100 times faster than its predecessor.

The immediate practicality of the algorithm, however, is not what impresses John Hopcroft, the IBM Professor of Engineering and Applied Mathematics at Cornell and a recipient of the Turing Prize, the highest award in computer science. “My guess is that this particular framework is going to be applicable to a wide range of other problems,” Hopcroft says. “It’s a fundamentally new technique. When there’s a breakthrough of that nature, usually, then, a subdiscipline forms, and in four or five years, a number of results come out.”

*This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: Fundamental algorithm gets first improvement in 10 years (2010, September 27) retrieved 27 April 2024 from https://phys.org/news/2010-09-fundamental-algorithm-years.html