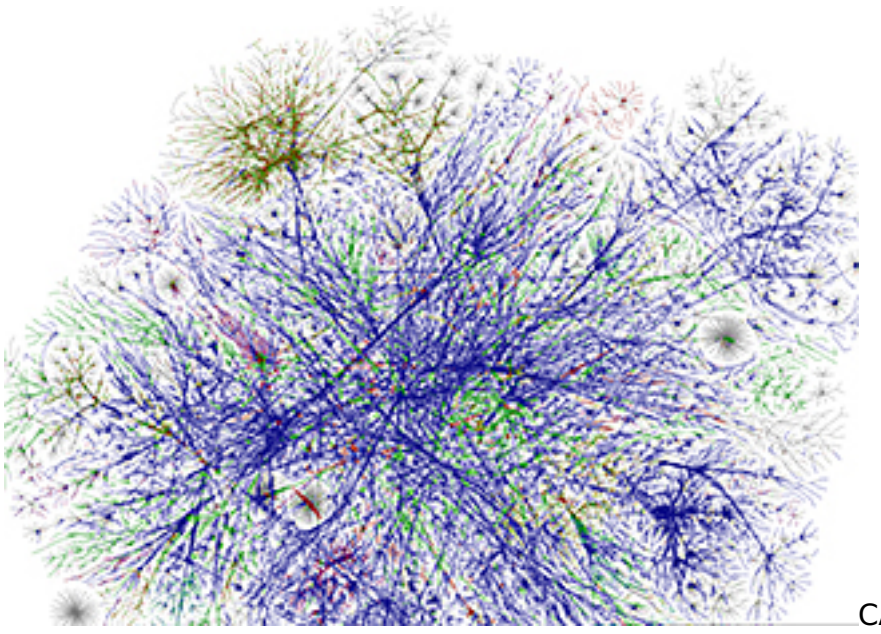
New Algorithm Dramatically Streamlines Max Flow Solutions

Helen Knight, MIT



CAMBRIDGE, MA -- Finding the most efficient way to transport items across a network like the U.S. highway system or the Internet is a problem that has taxed mathematicians and computer scientists for decades.

To tackle the problem, researchers have traditionally used a maximum-flow algorithm, also known as “max flow,” in which a network is represented as a graph with a series of nodes, known as vertices, and connecting lines between them, called edges.

Given that each edge has a maximum capacity — just like the roads or the fiber-optic cables used to transmit information around the Internet — such algorithms attempt to find the most efficient way to send goods from one node in the graph to another, without exceeding these constraints.

But as the size of networks like the Internet has grown exponentially, it is often prohibitively time-consuming to solve these problems using traditional computing techniques, according to Jonathan Kelner, an associate professor of applied mathematics at MIT and a member of MIT’s Computer Science and Artificial Intelligence Laboratory (CSAIL).

So in a paper to be presented at the ACM-SIAM Symposium on Discrete Algorithms in Portland, Ore., this week, Kelner and his colleague Lorenzo Orecchia, an applied mathematics instructor, alongside graduate students Yin Tat Lee and Aaron Sidford, will describe a new theoretical algorithm that can dramatically reduce the number of operations needed to solve the max-flow problem, making it possible to tackle even huge networks like the Internet or the human genome.

“There has recently been an explosion in the sizes of graphs being studied,” Kelner says. “For example, if you wanted to route traffic on the Internet, study all the connections on Facebook, or analyze genomic data, you could easily end up with graphs with millions, billions or even trillions of edges.”

Previous max-flow algorithms have come at the problem one edge, or path, at a time, Kelner says. So for example, when sending items from node A to node B, the algorithms would transmit some of the goods down one path, until they reached its maximum capacity, and then begin sending some down the next path.

“Many previous algorithms,” Kelner says, “would find a path from point A to point B, send some flow along it, and then say, ‘Given what I’ve already done, can I find another path along which I can send more?’ When one needs to send flow simultaneously along many different paths, this leads to an intrinsic limitation on the speed of the algorithm.”

But in 2011 Kelner, CSAIL graduate student Aleksander Madry, mathematics undergraduate Paul Christiano, and colleagues at Yale University and the University of Southern California developed a technique to analyze all of the paths simultaneously.

The researchers viewed the graph as a collection of electrical resistors, and then imagined connecting a battery to node A and a ground to node B, and allowing the current to flow through the network. “Electrical current doesn’t pick just one path, it will send a little bit of current over every resistor on the network,” Kelner says. “So it probes the whole graph globally, studying many paths at the same time.”

This allowed the new algorithm to solve the max-flow problem substantially faster than previous attempts.

Now the MIT team has developed a technique to reduce the running time even further, making it possible to analyze even gigantic networks, Kelner says.

Unlike previous algorithms, which have viewed all the paths within a graph as equals, the new technique identifies those routes that create a bottleneck within the network. The team’s algorithm divides each graph into clusters of well-connected nodes, and the paths between them that create bottlenecks, Kelner says.

“Our algorithm figures out which parts of the graph can easily route what they need to, and which parts are the bottlenecks. This allows you to focus on the problem areas and the high-level structure, instead of spending a lot of time making unimportant decisions, which means you can use your time a lot more efficiently,” he says.

The result is an almost linear algorithm, Kelner says, meaning the amount of time it takes to solve a problem is very close to being directly proportional to the number of nodes on the network. So if the number of nodes on the graph is multiplied by 10, the amount of time would be multiplied by something very close to 10, as opposed to being multiplied by 100 or 1,000, he says. “This means that it scales essentially as well as you could hope for with the size of the input,” he says.

Source URL (retrieved on *07/07/2015 - 3:57pm*):
http://www.scientificcomputing.com/news/2014/01/new-algorithm-dramatically-streamlines-max-flow-solutions