Team develops mathematical solver for analog computers

December 12 2018, by Deanna Csomo Mccool

Zoltán Toroczkai, professor in the Department of Physics and concurrent professor in the Department of Computer Science and Engineering at the University of Notre Dame. Credit: Matt Cashore/University of Notre Dame

Your computer performs most tasks well. For word processing, certain computations, graphic arts and web surfing, the digital box on your desk is the best tool for the job. But the way your computer works, with its style of mathematics that relies on the binary code system of "on" and

"off" 1s and 0s, isn't ideal for solving every problem.

That's why researchers such as Zoltán Toroczkai, professor in the Department of Physics and concurrent professor in the Department of Computer Science and Engineering at the University of Notre Dame, are interested in reviving analog computing at a time when digital computing has reached its maximum potential.

Toroczkai and collaborators have been working toward developing a novel mathematical approach that will help advance computation beyond the digital framework. His most recent paper, published in *Nature Communications*, describes a new mathematical, analog "solver" that can potentially find the best solution to NP-hard problems.

NP-hardness is a theory of computational complexity, with problems that are famous for their difficulty. When the number of variables is large, problems associated with scheduling, protein folding, bioinformatics, medical imaging and many other areas are nearly unsolvable with known methods. After testing their new method on a variety of NP-hard problems, the researchers concluded their solver has the potential to lead to better, and possibly faster, solutions than can be computed digitally.

Analog computers were used to predict tides from the early to mid-20th century, guide weapons on battleships and launch NASA's first rockets into space. They first used gears and vacuum tubes, and later, transistors, that could be configured to solve problems with a range of variables. They perform mathematical functions directly. For instance, to add 5 and 9, analog computers add voltages that correspond to those numbers, and then instantly obtain the correct answer. However, analog computers were cumbersome and prone to "noise"—disturbances in the signals—and were difficult to re-configure to solve different problems, so they fell out of favor.

Digital computers emerged after transistors and integrated circuits were reliably mass produced, and for many tasks they are accurate and sufficiently flexible. Computer algorithms, in the form of software, are sets of instructions that tell the computer hardware how to perform. Because the process is restricted to the use of 0s and 1s, this also makes their programming simpler, and allowed digital computing to dominate for nearly 70 years.

However, their restrictions may prevent digital computers from solving NP-hard problems with many variables. One such problem is the "Traveling Salesman" problem, in which a salesperson must start in one city and return to that city at the end of a trip, but in between, must travel to all the different cities on a list. What's the most efficient route among all the points? The problem becomes exponentially more challenging with the addition of more cities. The difficulty with such optimization problems, Toroczkai noted, is "while you can always come up with some answer, you cannot determine if it's optimal. Determining that there isn't a better solution is just as hard as the problem itself."

A challenge for analog computing rests with the design of continuous algorithms. Unlike digital computing, which has a long history in algorithm development, algorithms for analog computers lack a similar knowledge base and thus are very difficult to design. Toroczkai's approach is different from the types of algorithms for digital computers, in all aspects.

The next step is to design and build devices based on this approach, a process that will be tackled within Notre Dame's College of Engineering. The analog computers would be built for specific tasks, and not for everyday computing needs. This work is part of a larger-scale, multi-institutional effort, called Extremely Energy Efficient Collective Electronics (EXCEL), led by Notre Dame's Suman Datta, Freimann Chair of Engineering and professor of electrical engineering, in

collaboration with Sharon Hu, professor of computer science and engineering.

"There are mostly engineering problems that need to be solved at this point, such as spurious capacities and better noise control, but it's going to get there," Toroczkai said. "Ideally I would like to see that you have this box on your desk that is your scheduler. And it is going to do much better of a job than your regular computer."

More information: Botond Molnár et al. A continuous-time MaxSAT solver with high analog performance, *Nature Communications* (2018). DOI: 10.1038/s41467-018-07327-2

Provided by University of Notre Dame

Citation: Team develops mathematical solver for analog computers (2018, December 12) retrieved 19 April 2024 from https://techxplore.com/news/2018-12-team-mathematical-solver-analog.html

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.