

## ALGORITHMS

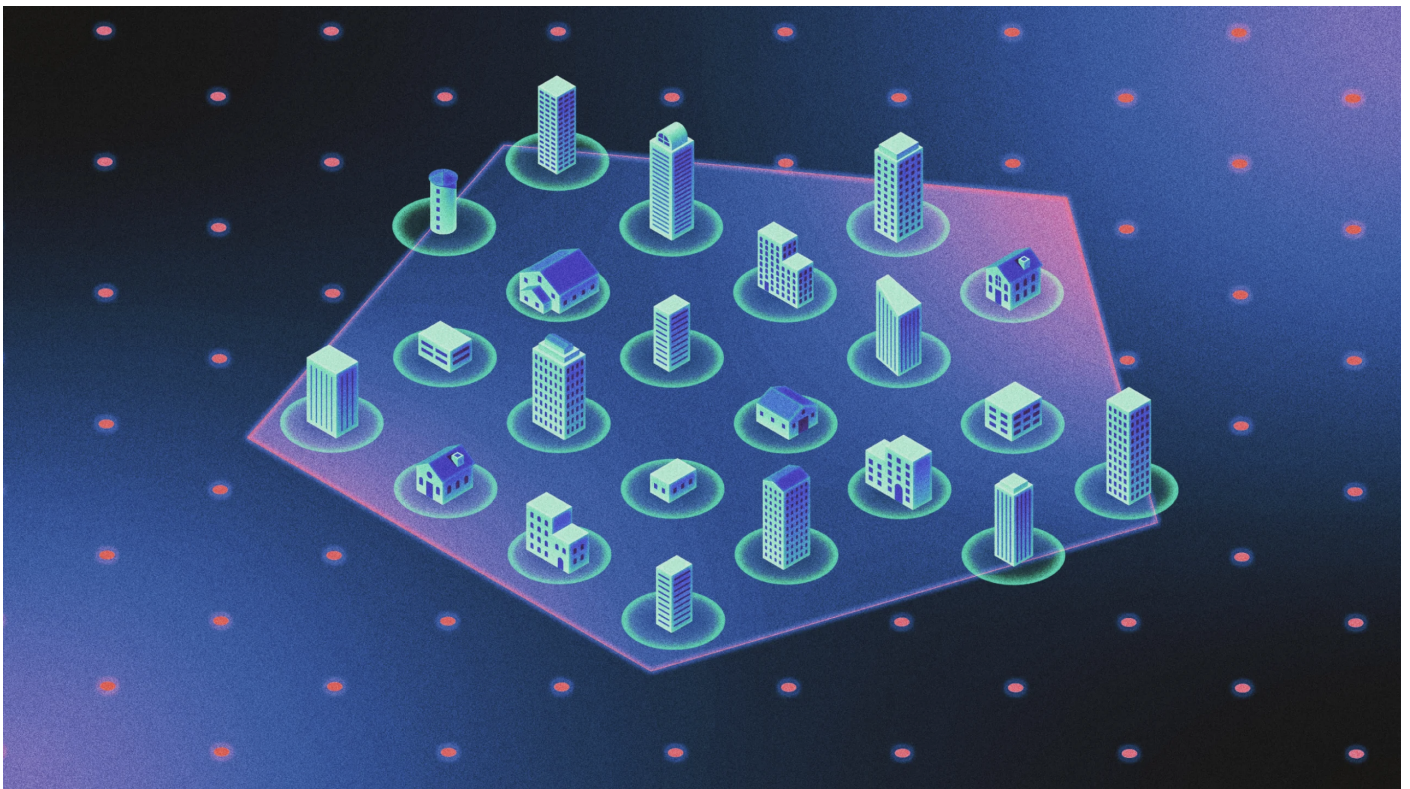
### Researchers Approach New Speed Limit for Seminal Problem

By LAKSHMI CHANDRASEKARAN

January 29, 2024

*Integer linear programming can help find the answer to a variety of real-world problems. Now researchers have found a much faster way to do it.*

2 |



Kristina Armitage/Quanta Magazine

**T**he traveling salesperson problem is one of the oldest known computational questions. It asks for the ideal route through a certain list of cities, minimizing mileage. Despite seeming simple, the problem is notoriously difficult. While you can use brute force to check all the possible routes until you find the shortest path, such a strategy becomes untenable after just a handful of cities. Instead, you can apply a rigorous mathematical model called linear programming, which roughly approximates the problem as a set of equations and methodically checks the possible combinations to find the best solution.

But sometimes, you need to optimize for problems involving whole-number amounts. What good is a factory optimization plan that manufactures 500.7 couches? For this, researchers often turn to a variant of linear programming called integer linear programming (ILP). It's popular in applications that involve discrete decisions, including production planning, airline crew scheduling and vehicle routing. "Basically, ILP is the bread and butter of operations research both in theory and practice," said Santosh Vempala, a computer scientist at the Georgia Institute of Technology.

Since they first formulated ILP over 60 years ago, researchers have discovered various algorithms that solve ILP problems, but they've all been relatively slow in terms of the number of steps required. The best version they could come up with — a kind of speed limit — comes from the trivial case where the problem's variables (such as whether a salesman visits a city or not) can only assume binary values (zero or 1). In this case, ILP has a runtime that scales exponentially with the number of variables, also called the dimension. (If there's only one variable, it takes just two steps to test every possible combination and solve the problem; two variables mean four steps, three mean eight steps, and so on.)

Unfortunately, once the variables take a value beyond just zero and 1, the algorithm's runtime grows much longer. Researchers have long wondered if they could get closer to the trivial ideal. Progress has been slow, with the record set in the 1980s and only incremental improvements made since.

But recent work by Victor Reis, currently at the Institute for Advanced Study, and Thomas Rothvoss, at the University of Washington, has made the biggest runtime leap in decades. By combining geometric tools to limit the possible solutions, they created a new, faster algorithm for solving ILP in almost the same time as the trivial binary case. The result received a best-paper award at the 2023 Foundations of Computer Science conference.

"This new algorithm is extremely exciting," said Noah Stephens-Davidowitz, a mathematician and computer scientist at Cornell University. "It represents the first [major] improvement to ILP solvers in nearly 40 years."

ILP works by transforming a given problem into a set of linear equations that must satisfy some inequalities. The specific equations are based on the details of the original problem. But while these details may differ, the basic makeup of ILP problems remains the same, giving researchers a single way to attack a multitude of problems.



Victor Reis helped create a new algorithm that solves integer linear programming problems in record time.

—

Maryna Viazovska



That's not to say it's easy work. It wasn't until 1983 that the mathematician Hendrik Lenstra proved that the general problem was even solvable, providing the first algorithm that could do it. Lenstra thought about ILP geometrically. First, he turned the inequalities at the heart of ILP into a convex shape, such as any regular polygon. This shape represents the constraints of the individual problem you're solving, whether it's couch production or airline scheduling, so the shape's interior corresponds to all possible values that could solve the inequalities, and thus the problem. Lenstra called this shape the convex body. The problem's dimension influences the dimension of this shape: With two variables it takes the form of a flat polygon; in three dimensions it is a Platonic solid, and so on.

Lenstra next imagined all the integers as an infinite set of grid points, known in mathematics as a lattice. A two-dimensional version looks like a sea of dots, and in three dimensions it looks like the points where steel beams in a building connect. The dimension of the lattice also depends on the dimension of a given problem.

To solve a given ILP problem, Lenstra showed that you just look for where the possible solutions meet the set of integers: at the intersection of the convex body and the lattice. And he came up with an algorithm that could search this space exhaustively — but to be effective, it sometimes had to break the problem up into pieces of smaller dimensions, adding many steps to the runtime.







Thomas Rothvoss and Reis combined previous work with a geometric approach to achieve an ILP algorithm that runs almost as quickly as the theoretical ideal.

—  
Dennis Wise/University of Washington

In the following years, several researchers explored how to make this algorithm run faster. In 1988, Ravi Kannan and László Lovász introduced a concept called the covering radius, borrowed from the study of error-correcting codes, to help the convex body and lattice intersect more efficiently. Roughly, the covering radius makes sure that the convex body always contains at least one integer point, no matter where you place it on the lattice. As a result, the size of the covering radius also determines how efficiently you can solve the ILP problem.

So it all came down to determining the size of the ideal covering radius. Unfortunately, this proved to be a hard problem on its own, and the best Kannan and Lovász could do was narrow down a possible value by searching for upper and lower bounds. They showed that the upper bound — the maximum size of the covering radius — scaled up linearly with the dimension. This was pretty fast, but not enough to significantly speed up the ILP runtime. Over the next 30 years, other researchers could only do slightly better.

What ultimately helped Reis and Rothvoss break through was an unrelated mathematical result that focused purely on lattices. In 2016, Oded Regev and Stephens-Davidowitz showed, in effect, how many lattice points could fit within a specific shape. Reis and Rothvoss applied this to other shapes, which allowed them to better estimate the number of lattice points contained in an ILP covering radius, lowering the upper bound. “The latest breakthrough came with the realization that you can actually do other kinds of shapes,” Regev said.

This new tightened upper bound was a vast improvement, allowing Reis and Rothvoss to achieve a dramatic speedup of the overall ILP algorithm. Their work brings the runtime to  $(\log n)^{O(n)}$ , where  $n$  is the number of variables and  $O(n)$  means it scales linearly with  $n$ . (This expression is considered “almost” the same as the run time of the binary problem.)

“It’s a triumph at the intersection of math, computer science and geometry,” said Daniel Dadush of the national research institute CWI in the Netherlands, who helped pioneer the algorithm Reis and Rothvoss used to measure the ILP runtime.

For now, the new algorithm hasn’t actually been used to solve any logistical problems, since it would take too much work updating today’s programs to make use of it. But for Rothvoss, that’s beside the point. “It’s about the theoretical understanding of a problem that has fundamental applications,” he said.

As to whether the computational efficiency of ILP could be further improved, researchers are still hopeful that they’ll keep approaching the ideal runtime — but not anytime soon. “That would require a fundamentally new idea,” Vempala said.