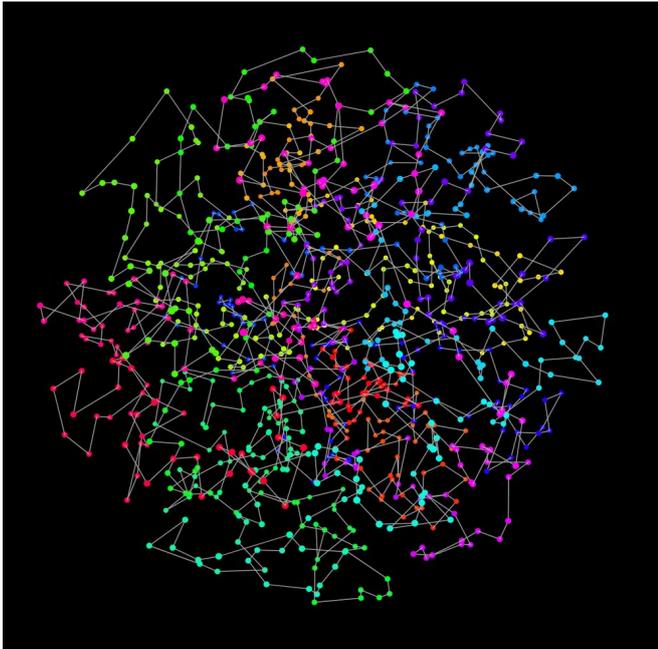# A new approach to planning the best route breaks a barrier that's stood for nearly half a century

12 April 2021, by Nathan Klein

This collection of dots and lines is the shortest traveling salesperson problem tour that passes through 1,000 points. Credit: [William Cook et al.](link), [CC BY-ND](link)

Computers are good at answering questions. What's the shortest route from my house to Area 51? Is 8,675,309 a prime number? How many teaspoons in a tablespoon? For questions like these, they've got you covered.

There are certain innocent-sounding questions, though, that computer scientists believe computers will never be able to answer—at least not within our lifetimes. These problems are the subject of the [P versus NP question](link), which asks whether problems whose solutions can be checked quickly can also be solved quickly. P versus NP is such a fundamental question that either designing a fast algorithm for one of these hard problems or

proving you can't would net you a [cool million dollars](link) in prize money.

My favorite hard problem is the [traveling salesperson problem](link). Given a collection of cities, it asks: What is the most efficient route that visits all of them and returns to the starting [city](link)? To come up with practical answers in the real world, computer scientists use approximation algorithms, methods that don't solve these problems exactly but get close enough to be helpful. Until now, the best of these algorithms, developed in 1976, guaranteed that its answers would be no worse than 50% off from the best answer.

I work on approximation algorithms as a [computer scientist](link). My collaborators Anna Karlin and Shayan Oveis Gharan and I have found a way to beat that 50% mark, though just barely. We were able to prove that a specific approximation algorithm puts a crack in this long-standing barrier, a finding that opens the way for more substantial improvements.

This is important for more than just planning routes. Any of these hard problems can be encoded in the traveling salesperson problem, and vice versa: Solve one and you've solved them all. You might say that these hard problems are all the same computational gremlin wearing different hats.

The best route is hard to find

The problem is usually stated as "find the shortest route." However, the most efficient solution can be based on a variety of quantities in the real world, such as time and cost, as well as distance.

To get a sense of why this problem is difficult, imagine the following situation: Someone gives you a list of 100 cities and the cost of plane, train and bus tickets between each pair of them. Do you think

you could figure out the cheapest itinerary that visits them all?

Consider the sheer number of possible routes. If you have 100 cities you want to visit, the number of possible orders in which to visit them is 100 factorial, meaning 100 × 99 × 98 x … × 1. This is larger than the number of atoms in the universe.

Going with good enough

Unfortunately, the fact that these problems are difficult does not stop them from coming up in the real world. Besides finding routes for traveling salespeople (or, these days, delivery trucks), the traveling salesperson problem has applications in many areas, from mapping genomes to designing circuit boards.

To solve real-world instances of this problem, practitioners do what humans have always done: Get solutions that might not be optimal but are good enough. It's OK if a salesperson takes a route that's a few miles longer than it has to be. No one cares too much if a circuit board takes a fraction of a second longer to manufacture or an Uber takes a few minutes longer to carry its passengers home.

Computer scientists have embraced "good enough" and for the past 50 years or so have been working on so-called approximation algorithms. These are procedures that run quickly and produce solutions that might not be optimal but are provably close to the best possible solution.

The long-reigning champ of approximation

One of the first and most famous approximation algorithms is for the traveling salesperson problem and is known as the Christofides-Serdyukov algorithm. It was designed in the 1970s by Nicos Christofides and, independently, by a Soviet mathematician named Anatoliy Serdyukov whose work was not widely known until recently.
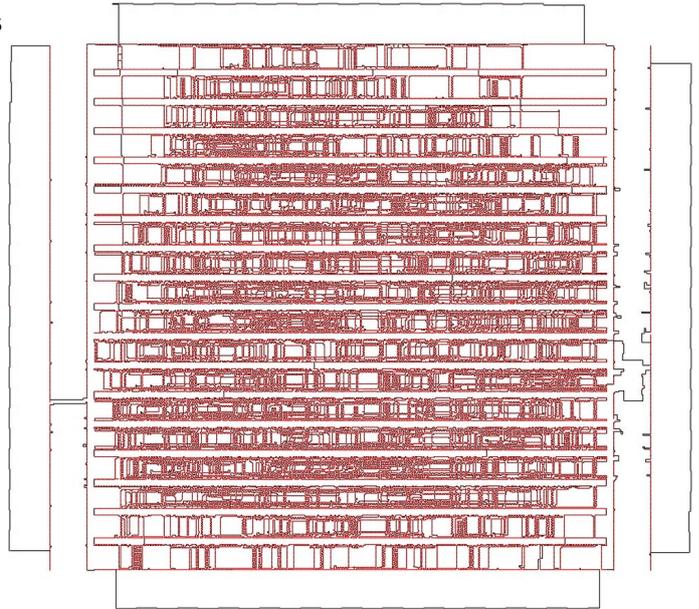
This TSP tour is an efficient layout for circuits on a circuit board. Credit: William Cook et al., CC BY-ND

The Christofides-Serdyukov algorithm is quite simple, at least as algorithms go. You can think of a traveling salesperson problem as a network in which each city is a node and each path between pairs of cities is an edge. Each edge is assigned a cost, for example the traveling time between the two cities. The algorithm first selects the cheapest set of edges that connect all the cities.

This, it turns out, is easy to do: You just keep adding the cheapest edge that connects a new city. However, this not a solution. After connecting all the cities, some might have an odd number of edges coming out of them, which doesn't make sense: Every time you enter a city with an edge, there should be a complementary edge you use to leave it. So the algorithm then adds the cheapest collection of edges that makes every city have an even number of edges and then uses this to produce a tour of the cities.

This algorithm runs quickly and always produces a solution that's at most 50% longer than the optimal one. So, if it produces a tour of 150 miles, it means that the best tour is no shorter than 100 miles.

Of course, there's no way to know exactly how close to optimal an approximation algorithm gets for a particular example without actually knowing the optimal solution—and once you know the optimal solution there's no need for the approximation algorithm! But it's possible to prove something about the worst-case scenario. For example, the Christofides-Serdyukov algorithm guarantees that it produces a tour that is at most 1.5 times the length of the shortest collection of edges connecting all the cities—and, therefore, at most 1.5 times the length of the optimal tour.

A really small improvement that's a big deal

Since the discovery of this algorithm in 1976, computer scientists had been unable to improve upon it at all. However, last summer my collaborators and I proved that a particular algorithm will, on average, produce a tour that is less than 49.99999% away from the optimal solution. I'm too ashamed to write out the the true number of 9s (there are a lot), but this nevertheless breaks the longstanding barrier of 50%.

The algorithm we analyzed is very similar to Christofides-Serdyukov. The only difference is that in the first step it picks a random collection of edges that connects all the cities and, on average, looks like a traveling salesperson problem tour. We use this randomness to show that we don't always get stuck where the previous algorithm did.

While our progress is small, we hope that other researchers will be inspired to take another look at this problem and make further progress. Often in our field, thresholds like 50% stand for a long time, and after the first blow they fall more quickly. One of our big hopes is that the understanding we gained about the traveling salesperson problem while proving this result will help spur progress.

Getting closer to perfect

There is another reason to be optimistic that we will see more progress within the next few years: We think the algorithm we analyzed, which was devised in 2010, may be much better than we were able to prove. Unlike Christofides' algorithm, which can be shown to have a hard limit of 50%, we suspect this

algorithm may be as good as 33%.

Indeed, experimental results that compare the approximation algorithm to known optimal solutions suggest that the algorithm is quite good in practice, often returning a tour within a few percent of optimal.

The current theoretical limit is around 1%—meaning that (unless P=NP) there is no algorithm that will be able to get within 1% of optimal. The question that theoreticians hope to answer is: How close can we get?

This article is republished from The Conversation under a Creative Commons license. Read the original article.

Provided by The Conversation