

Q&A: Enhancing last-mile logistics with machine learning

April 17 2024, by Lauren Hinkel



Credit: CC0 Public Domain

Across the country, hundreds of thousands of drivers deliver packages and parcels to customers and companies each day, with many click-to-door times averaging only a few days. Coordinating a supply chain feat of this magnitude in a predictable and timely way is a longstanding problem of operations research, where researchers have been working to optimize the last leg of delivery routes.

This is because the last phase of the process is often the costliest due to inefficiencies like long distances between stops due to increased e-commerce demand, weather delays, traffic, lack of parking availability, customer delivery preferences, or partially full trucks—inefficiencies that became more exaggerated and evident during the pandemic.

With newer technology and more individualized and nuanced data, researchers are able to develop models with better routing options but, at the same time, need to balance the computational cost of running them. Matthias Winkenbach, MIT principal research scientist, director of research for the MIT Center for Transportation and Logistics (CTL), and a researcher with the MIT-IBM Watson AI Lab, discusses how artificial intelligence could provide better and more computationally efficient solutions to a combinatorial optimization problem like this one.

What is the vehicle routing problem, and how do traditional operations research (OR) methods address it?

The vehicle routing problem is faced by pretty much every logistics and delivery company like USPS, Amazon, UPS, FedEx, and DHL every single day. Simply speaking, it's finding an efficient route that connects a set of customers that need to be either delivered to or something that needs to be picked up from them. It's deciding which customers each of those vehicles—that you see out there on the road—should visit on a given day and in which sequence.

Usually, the objective here is to find routes that lead to the shortest, the fastest, or the cheapest route. But very often they are also driven by constraints that are specific to a customer. For instance, if you have a customer who has a delivery time window specified, or a customer on the 15th floor in the high-rise building versus the ground floor. This

makes these customers more difficult to integrate into an efficient delivery route.

To solve the vehicle routing problem, we obviously can't do our modeling without proper demand information and, ideally, customer-related characteristics. For instance, we need to know the size or weight of the packages ordered by a given customer or how many units of a certain product need to be shipped to a certain location.

All of this determines the time that you would need to service that particular stop. For realistic problems, you also want to know where the driver can park the vehicle safely. Traditionally, a route planner had to come up with good estimates for these parameters, so very often, you find models and planning tools that are making blanket assumptions because there weren't stop-specific data available.

Machine learning can be very interesting for this because nowadays most of the drivers have smartphones or GPS trackers, so there is a ton of information as to how long it takes to deliver a package. You can now, at scale, in a somewhat automated way, extract that information and calibrate every single stop to be modeled in a realistic way.

Using a traditional OR approach means you write up an optimization model, where you start by defining the objective function. In most cases, that's some sort of cost function. Then, there are a bunch of other equations that define the inner workings of a routing problem. For instance, you must tell the model that if the vehicle visits a customer, it also needs to leave the customer again. In academic terms, that's usually called flow conservation.

Similarly, you need to make sure that every customer is visited exactly once on a given route. These and many other real-world constraints together define what constitutes a viable route. It may seem obvious to

us, but this needs to be encoded explicitly.

Once an optimization problem is formulated, there are algorithms out there that help us find the best possible solution; we refer to them as solvers. Over time, they find solutions that comply with all the constraints. Then, it tries to find routes that are better and better, so cheaper and cheaper ones, until you either say, "OK, this is good enough for me," or until it can mathematically prove that it found the optimal solution.

The average delivery vehicle in a U.S. city makes about 120 stops. It can take a while to solve that explicitly, so that's usually not what companies do because it's just too computationally expensive. Therefore, they use so-called heuristics, which are algorithms that are very efficient in finding reasonably good solutions but typically cannot quantify how far away these solutions are from the theoretical optimum.

You're currently applying machine learning to the vehicle routing problem. How are you employing it to leverage and possibly outperform traditional OR methods?

That's what we're currently working on with folks from the MIT-IBM Watson AI Lab. Here, the general idea is that you train a model on a large set of existing routing solutions that you either observed in a company's real-world operations or that you generated using one of these efficient heuristics.

In most [machine-learning](#) models, you no longer have an explicit objective function. Instead, you need to make the model understand what kind of problem it's actually looking at and what a good solution to the problem looks like. For instance, similar to training a [large language](#)

[model](#) on words in a given language, you need to train a route learning model on the concept of the various delivery stops and their demand characteristics.

Like understanding the inherent grammar of natural language, your model needs to understand how to connect these delivery stops in a way that results in a good solution—in our case, a cheap or fast solution. If you then throw a completely new set of customer demands at it, it will still be able to connect the dots quite literally in a way that you would also do if you were trying to find a good route to connect these customers.

For this, we're using model architectures that most people know from the language processing space. It seems a little bit counterintuitive because what does language processing have to do with routing? But actually, the properties of these models, especially transformer models, are good at finding structure in language—connecting words in a way that they form sentences. For instance, in a language, you have a certain vocabulary, and that's fixed.

It's a discrete set of possible words that you can use, and the challenge is to combine them in a meaningful way. In routing, it's similar. In Cambridge, there are like 40,000 addresses that you can visit. Usually, it's a subset of these addresses that need to be visited, and the challenge is: How do we combine this subset—these "words"—in a sequence that makes sense?

That's kind of the novelty of our approach—leveraging that structure that has proven to be extremely effective in the language space and bringing it into combinatorial optimization. Routing is just a great test bed for us because it's the most fundamental problem in the logistics industry.

Of course, there are already very good routing algorithms out there that emerged from decades of operations research. What we are trying to do in this project is show that with a completely different, purely machine learning-based methodological approach, we are able to predict routes that are pretty much as good as, or better than, the routes that you would get from running a state-of-the-art route optimization heuristic.

What advantages does a method like yours have over other state-of-the-art OR techniques?

Right now, the best methods are still very hungry in terms of computational resources that are required to train these models, but you can front-load some of this effort. Then, the trained model is relatively efficient in producing a new solution as it becomes required.

Another aspect to consider is that the operational environment of a route, especially in cities, is constantly changing. The available road infrastructure, traffic rules, and speed limits might be altered, the ideal parking lot may be occupied by something else, or a construction site might block a road. With a pure OR-based approach, you might actually be in trouble because you would have to basically resolve the entire problem instantly once new information about the problem becomes available.

Since the operational environment is dynamically changing, you would have to do this over and over again. If you have a well-trained model that has seen similar issues before, it could potentially suggest the next-best route to take almost instantaneously. It's more of a tool that would help companies adjust to increasingly unpredictable changes in the environment.

Moreover, optimization algorithms are often manually crafted to solve

the specific problem of a given company. The quality of the solutions obtained from such explicit algorithms is bounded by the level of detail and sophistication that went into the design of the algorithm. A learning-based model, on the other hand, continuously learns a routing policy from data.

Once you have defined the model structure, a well-designed route learning model will distill potential improvements to your routing policy from the vast amount of routes it is being trained on. Simply put, a learning-based routing tool will continue to find improvements to your routes without you having to invest in explicitly designing these improvements into the algorithm.

Lastly, optimization-based methods are typically limited to optimizing for a very clearly defined objective function, which often seeks to minimize cost or maximize profits. In reality, the objectives that companies and drivers face are much more complex than that, and often, they are also somewhat contradictory. For instance, a company wants to find efficient routes, but it also wants to have a low emissions footprint.

The driver also wants to be safe and have a convenient way of serving these customers. On top of all of that, companies also care about consistency. A well-designed route learning model can eventually capture these high-dimensional objectives by itself, and that is something that you would never be able to achieve in the same way with a traditional optimization approach.

So, this is the kind of machine learning application that can actually have a tangible real-world impact in industry, on society, and on the environment. The logistics industry has problems that are much more complex than this.

For instance, if you want to optimize an entire supply chain—let's say,

the flow of a product from the manufacturer in China through the network of different ports around the world, through the distribution network of a big retailer in North America to your store where you actually buy it—there are so many decisions involved in that, which obviously makes it a much harder task than optimizing a single vehicle route.

Our hope is that with this initial work, we can lay the foundation for research and also private-sector development efforts to build tools that will eventually enable better end-to-end supply chain optimization.

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: Q&A: Enhancing last-mile logistics with machine learning (2024, April 17) retrieved 2 May 2024 from <https://techxplore.com/news/2024-04-qa-mile-logistics-machine.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.
