*Amazon Last Mile Routing Research Challenge:*

THE PROBLEM

Despite tremendous advances in route optimization over the last decades, there remains an important gap between theoretical route planning and real-life route execution: in real-life operations, the quality of a route is not exclusively defined by its theoretical length, duration, or cost. Experienced delivery drivers have tacit knowledge about the complex operational environment in which they serve customers on a daily basis. To allow for safer, more efficient, and sustainable last-mile delivery, it is critical to leveraging this tacit information for improved route planning.

THE OBJECTIVE

The primary goal of the Last Mile Routing Research Challenge is to encourage participants to develop innovative approaches leveraging artificial intelligence, machine learning, deep learning, computer vision, and other non-conventional methods to produce solutions to the route sequencing problem that outperform traditional, optimization-driven operations research methods in terms of solution quality and computational cost.

## **Downloading the Data**

The AWS command line interface (CLI) ([Amazon Web Services 2022](https://pubsonline.informs.org/doi/10.1287/trsc.2022.1173#B1)) is the recommended tool to access the research challenge data set. After installing the AWS CLI, the following command can be executed to download the data to a local directory specified by the user: aws s3 sync–no-sign-request s3://amazon-last-mile-challenges/almrrc2021/{local directory}

*Miscellaneous -*

* The contributed methodology divides the problem into two parts: on the zone level, it aims to infer the sequence of zones after which, within each zone, the sequence of stops is determined. It stems from the idea that first a routing problem is abstractified into zones, for which a sequence is established by solving a Traveling Salesman Problem TSP
* The quality score is based both on the level of time window adherence and the amount of backtracking in the observed sequence. Backtracking occurs when a delivery vehicle delivers packages within some neighborhood or geographical area, leaves the neighborhood or geographical area, then returns later during the route. Backtracking is inefficient and should be limited when possible.
* the most successful contributions used optimization algorithms for the TSP enriched by constraints or rules extracted from historical data.
* The approach in this paper learns precedence constraints from a single best-fitting historical tour and enters them into a *precedence-constrained TSP* solver based on local search techniques.
* A portion of the driver tours include return visits to a stop because of an undelivered package during an ini- tial visit. The routes in the final scoring phase of the challenge are limited to those with no such undelivered packages (so no return visits to stops). There are 1,107 such routes that also receive the high-quality rating. This High+Delivered collection of instances is the test set for our discussion.
* In cases in which a stop in the training data is missing a zone ID, we assign to it the ID of the closest stop (in Euclidean distance) associated with the same depot.

**Scoring**

The scoring metric used in the competition is complex, combining two ways to measure the difference between two tours, one involving only the positions of the stops in the tours and the other also taking into account the travel time between stops. The two measures are multi- plied and scaled to produce a single numerical score, the lower the better, with a perfect match of the driver route receiving 0.0. Details of the score computation can be found at the challenge GitHub site (https://github. com/MIT-CAVE/rc-cli/tree/main/scoring). The over- all score for a competing team is the average of their tour scores taken over the full hidden data set.

To indicate the range of outcomes, random tours on the High+Delivered instances received a score of 0.91545, averaged over 10 trials. Much better are the optimal ATSP tours, attaining a score of 0.07030. Our target is to find routing constraints that drive this score toward 0.0.