# **Millemiglia** : A Novel Instance Generator for Simulating Middle-Mile Logistics Networks

M. Petris, A.Lotfi, T. Cuvelier, B. De Backer, C. Archetti

Google Research ESSEC BUSINESS SCHOOL

3 July 2024

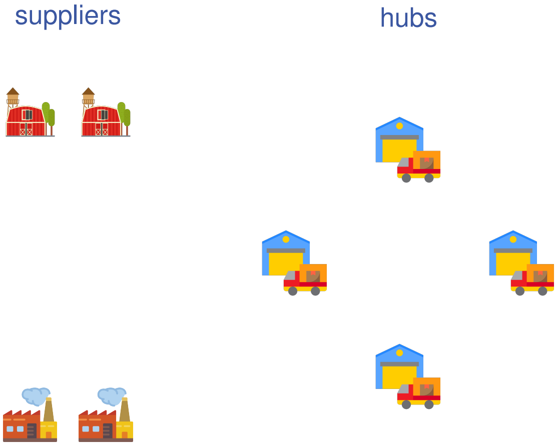
# Introduction

## Context: supply chain

## Context: supply chain

suppliers

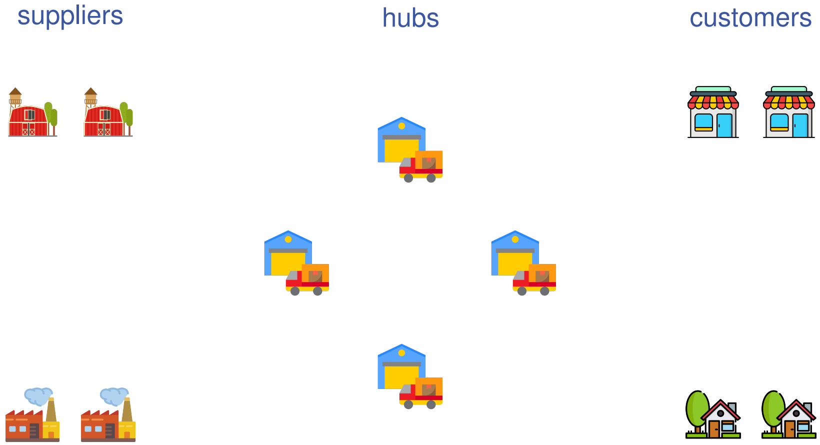
# Context: supply chain



suppliers

hubs

# Context: supply chain

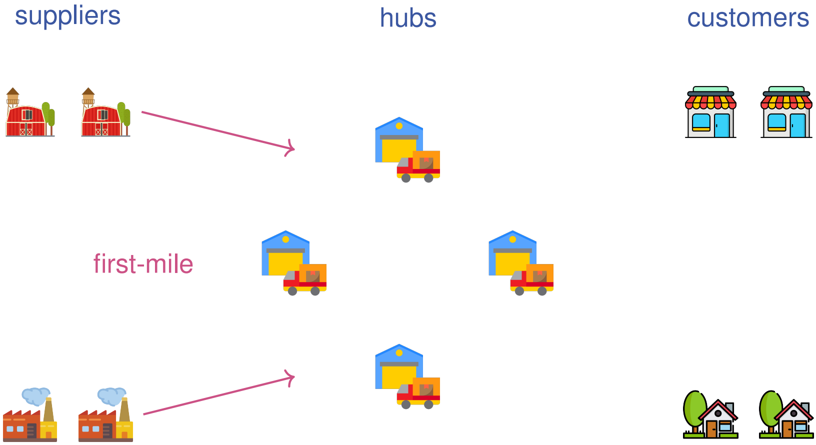


suppliers

hubs

customers

# Context: supply chain



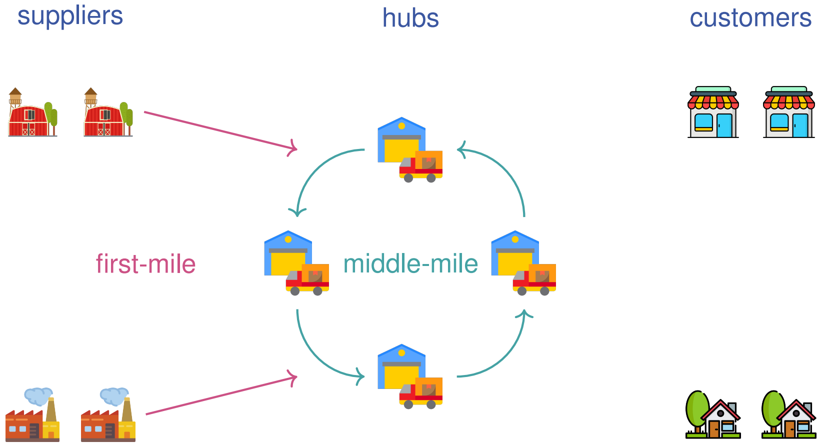
suppliers

hubs

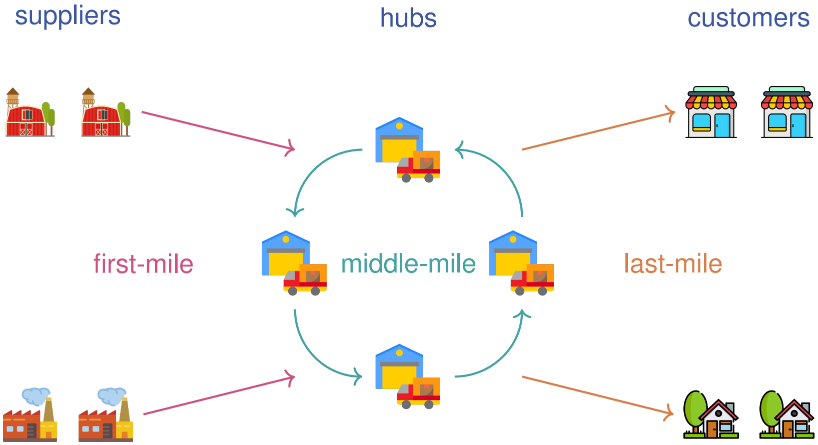
customers

first-mile

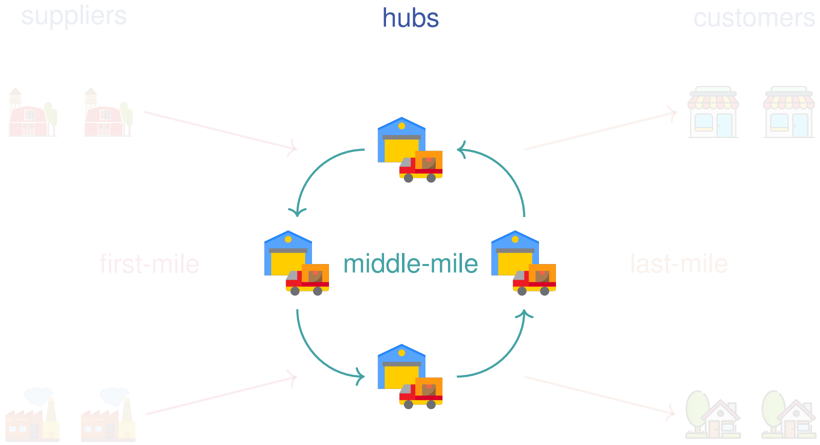
# Context: supply chain



# Context: supply chain



# Context: supply chain



# The Middle-Mile Logistic Problem

Given a **logistic network** composed of

- **hubs**
- **lines** and **rotations**
- **capacitated vehicles**

# The Middle-Mile Logistic Problem

Given a **logistic network** composed of

- **hubs**
- **lines** and **rotations**
- **capacitated vehicles**

and a set of **shipments** (O-D pairs, quantity) disclosed within a **given time horizon**,

# The Middle-Mile Logistic Problem

Given a **logistic network** composed of

- **hubs**
- **lines** and **rotations**
- **capacitated vehicles**

and a set of **shipments** (O-D pairs, quantity) disclosed within a **given time horizon**, **deliver the shipments** from their origin to their destinations **at minimum cost.**

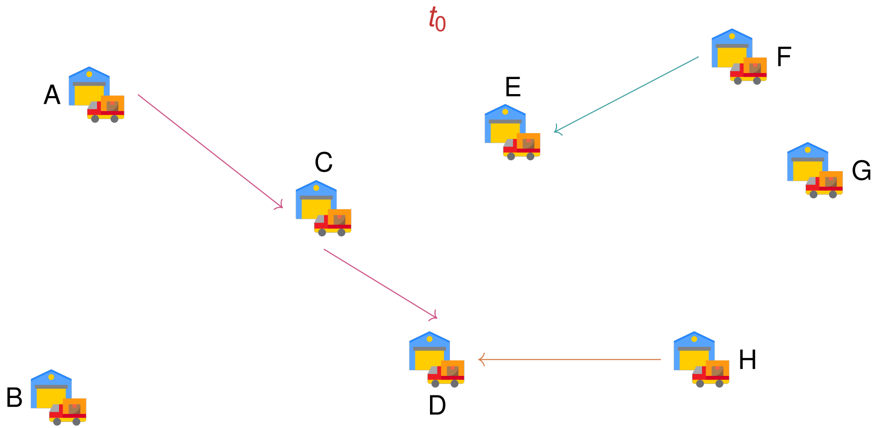
# The Middle-Mile Logistic Problem

Given a **logistic network** composed of

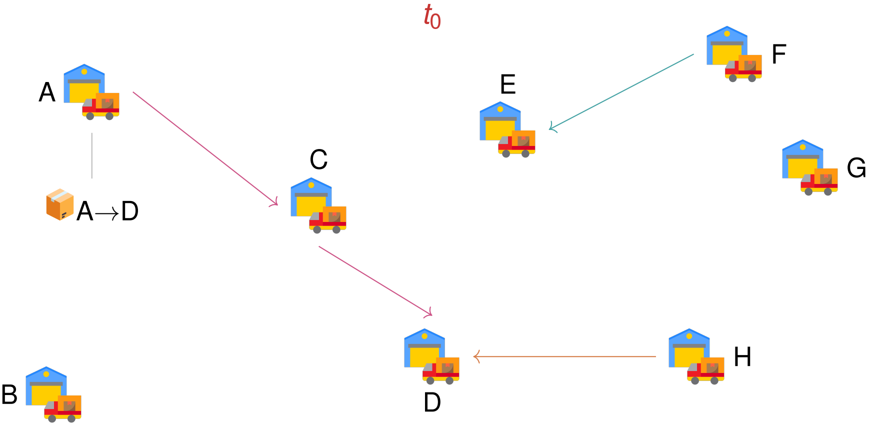
- **hubs**
- **lines** and **rotations**
- **capacitated vehicles**

and a set of **shipments** (O-D pairs, quantity) disclosed within a **given time horizon**, **deliver the shipments** from their origin to their destinations **at minimum cost.**

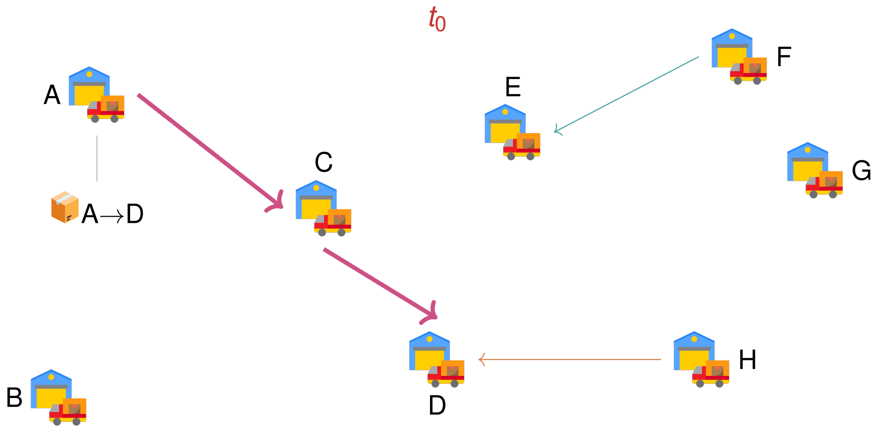
# The Middle-Mile Logistic Problem: an example



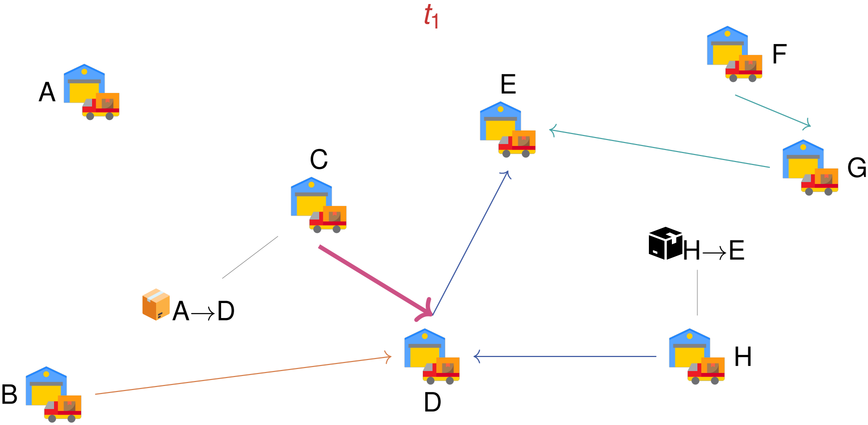
# The Middle-Mile Logistic Problem: an example



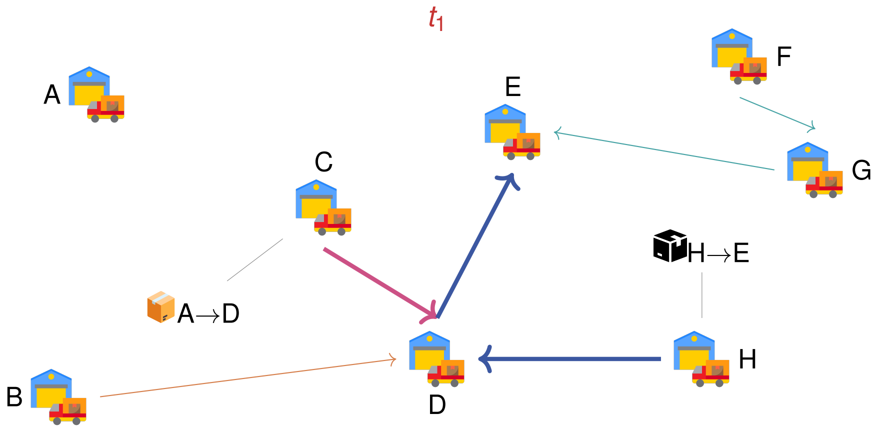
# The Middle-Mile Logistic Problem: an example



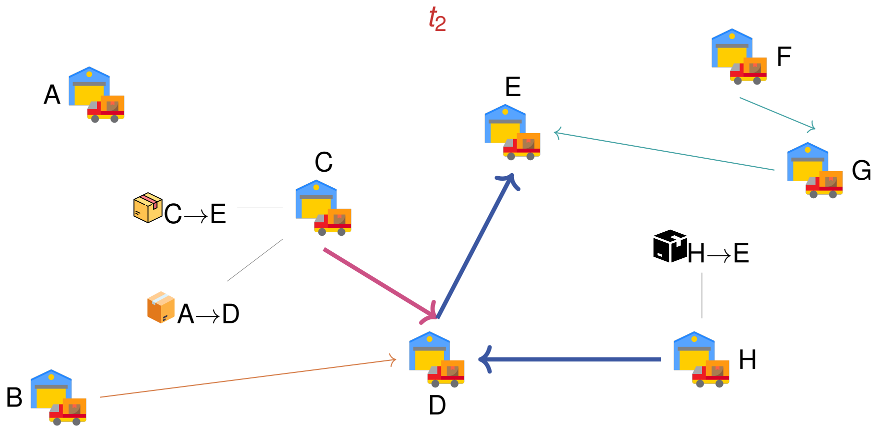
# The Middle-Mile Logistic Problem: an example



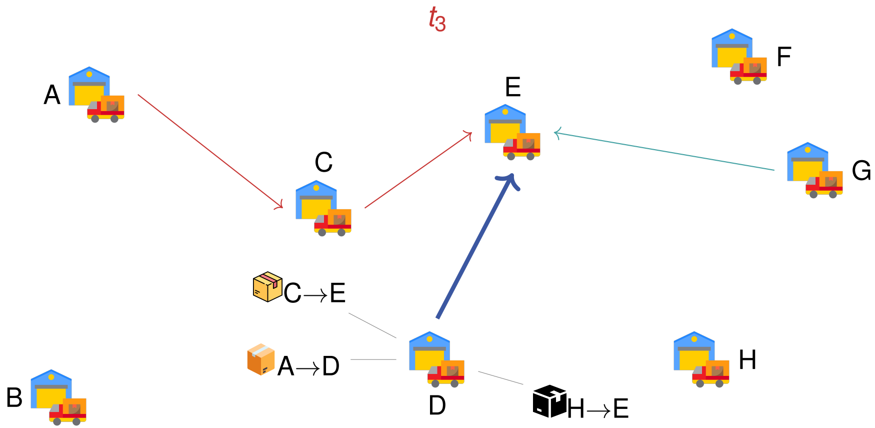
# The Middle-Mile Logistic Problem: an example



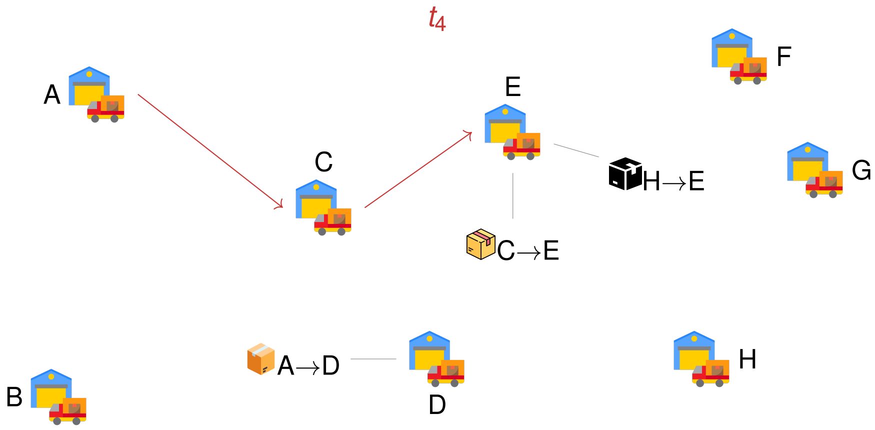
# The Middle-Mile Logistic Problem: an example



# The Middle-Mile Logistic Problem: an example



# The Middle-Mile Logistic Problem: an example



# The Middle-Mile Logistic Problem: space-time graph

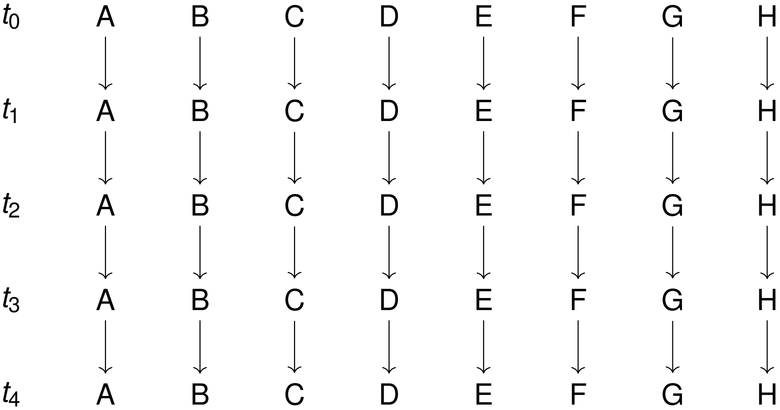
The Middle-Mile Logistic Problem can be modelled on a **acyclic space-time graph** where

- **vertices** are pairs **(hub,time)**
- **waiting arcs**
- **travelling arcs** (lines - rotations)

# The Middle-Mile Logistic Problem: space-time graph

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $t_0$ | A | B | C | D | E | F | G | H |
| $t_1$ | A | B | C | D | E | F | G | H |
| $t_2$ | A | B | C | D | E | F | G | H |
| $t_3$ | A | B | C | D | E | F | G | H |
| $t_4$ | A | B | C | D | E | F | G | H |

# The Middle-Mile Logistic Problem: space-time graph



$t_0$ A B C D E F G H

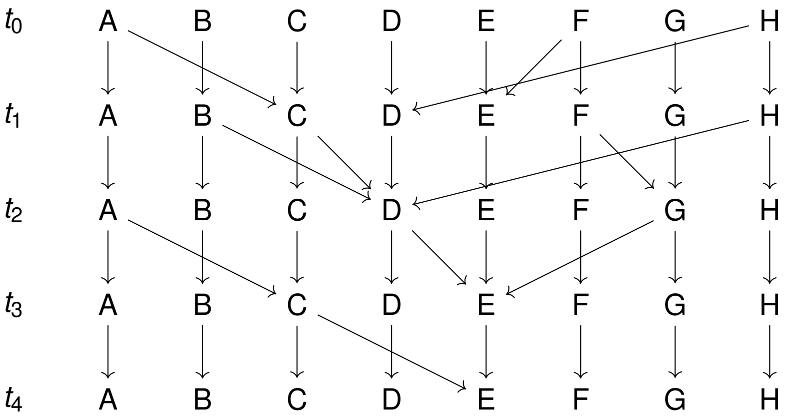
$t_1$ A B C D E F G H

$t_2$ A B C D E F G H

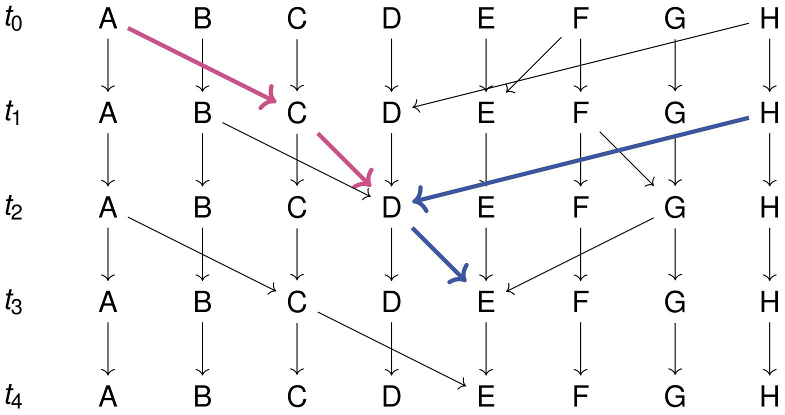
$t_3$ A B C D E F G H

$t_4$ A B C D E F G H

# The Middle-Mile Logistic Problem: space-time graph



# The Middle-Mile Logistic Problem: space-time graph



# Instance generator : Goals & Procedures

# Project goals

Creating diverse and realistic MM networks datasets for :

- **Benchmarking Optimization Approaches**
- **Training Machine Learning Models**

## Motivation

Middle-mile logistic

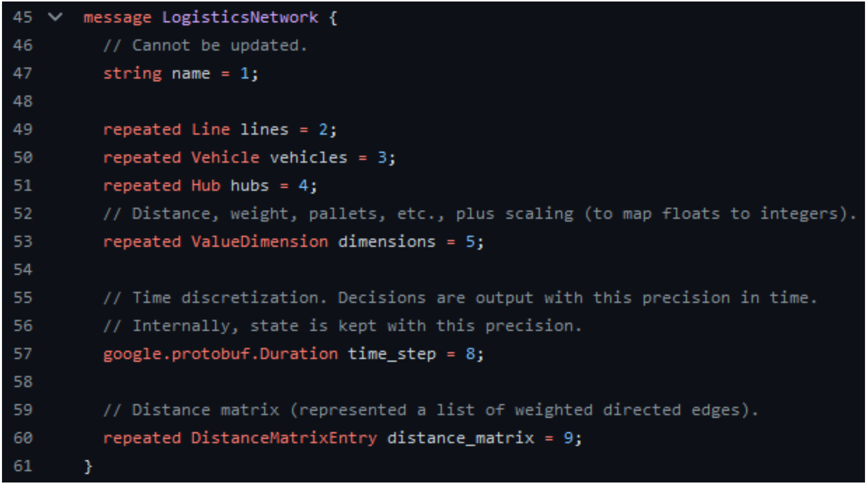
- aims to **consolidate loads** to generate **cost savings**
- **literature is scarce** w.r.t. last-mile
- **Lack of existing libraries** for instance generation.

## Library overview

- Based on the work : **Middle-Mile Logistics Through the Lens of Goal-Conditioned Reinforcement Learning** [Eberhard et al., 2023]
- Coded in **C++** with **protocol buffers**
- **Open source**
  `https://github.com/or-tools/millemiglia`
- Can generate realistic instances with diverse size and structure.

## Procedure:

# Procedure: Generate logistic network



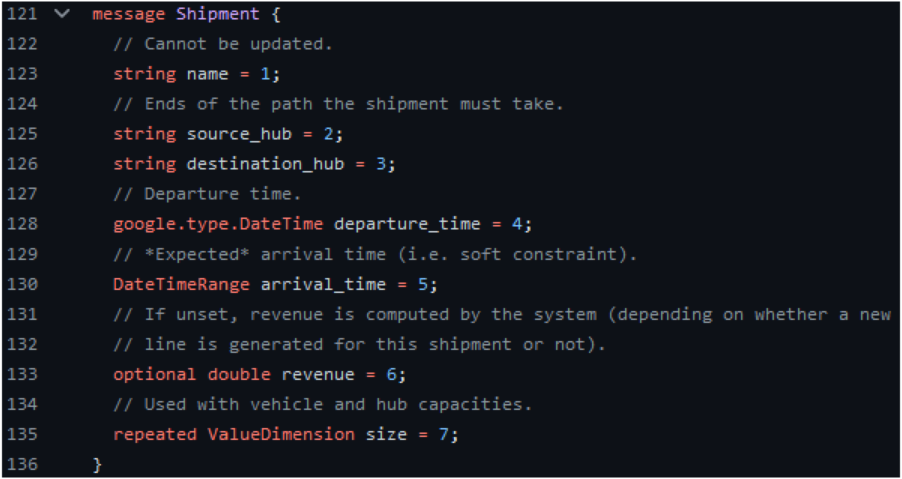
```
message LogisticsNetwork {
  // Cannot be updated.
  string name = 1;

  repeated Line lines = 2;
  repeated Vehicle vehicles = 3;
  repeated Hub hubs = 4;
  // Distance, weight, pallets, etc., plus scaling (to map floats to integers).
  repeated ValueDimension dimensions = 5;

  // Time discretization. Decisions are output with this precision in time.
  // Internally, state is kept with this precision.
  google.protobuf.Duration time_step = 8;

  // Distance matrix (represented a list of weighted directed edges).
  repeated DistanceMatrixEntry distance_matrix = 9;
}
```

# Procedure: Generate shipments



```
message Shipment {
  // Cannot be updated.
  string name = 1;
  // Ends of the path the shipment must take.
  string source_hub = 2;
  string destination_hub = 3;
  // Departure time.
  google.type.DateTime departure_time = 4;
  // *Expected* arrival time (i.e. soft constraint).
  DateTimeRange arrival_time = 5;
  // If unset, revenue is computed by the system (depending on whether a new
  // line is generated for this shipment or not).
  optional double revenue = 6;
  // Used with vehicle and hub capacities.
  repeated ValueDimension size = 7;
}
```

# Generate logistic network: space graph

Input:

- **Number of hubs**
- **Graph density**

# Generate logistic network: space graph

Input:

- **Number of hubs**
- **Graph density**

Output:

- **Graph** with a few high degree nodes (hubs)

# Generate logistic network: space graph

Input:

- **Number of hubs**
- **Graph density**

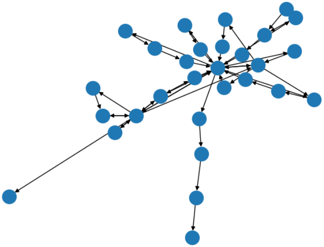
Output:

- **Graph** with a few high degree nodes (hubs)

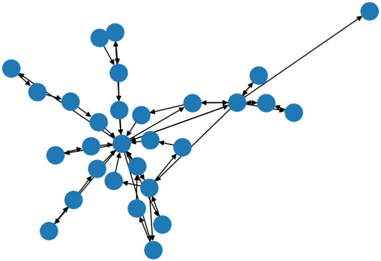
Procedure:

- Modified **Albert-Barabási** random graph model [Albert and Barabási, 2000]

# Space graph



Real graph



Generated graph

# Generate logistic network: lines

Input:

- **Graph**
- **Time horizon**
- **Number of lines** and **maximal length of a line**
- **Maximal number of rotations per line**

# Generate logistic network: lines

Input:

- **Graph**
- **Time horizon**
- **Number of lines** and **maximal length of a line**
- **Maximal number of rotations per line**

Output:

- **Lines-rotations**

# Generate logistic network: lines

Procedure:

1. **Lines** are **paths** in the space graph
    - **Arcs** are sampled $\propto$ **degree**

# Generate logistic network: lines

Procedure:

1. **Lines** are **paths** in the space graph
    - **Arcs** are sampled $\propto$ **degree**
2. **Rotations** assign **time information** to the lines
    - **Times** are sampled with **uniform distribution**

# Generate logistic network: vehicles

Input:

- **Line-rotations**
- **Maximal vehicle capacity**

# Generate logistic network: vehicles

Input:

- **Line-rotations**
- **Maximal vehicle capacity**

Output:

- **One vehicle for each line-rotation**

# Generate logistic network: vehicles

Input:

- **Line-rotations**
- **Maximal vehicle capacity**

Output:

- **One vehicle for each line-rotation**

Procedure:

- **Capacity** is sampled **uniform distribution**
- **Cost** $\propto$ **capacity**

## Generate shipments

Input:

- **Shipment number**
- **Logistic network**
- **Maximal path length**
- **Minimal** and **maximal size**

# Generate shipments

Input:

- **Shipment number**
- **Logistic network**
- **Maximal path length**
- **Minimal** and **maximal size**

Output:

- **Set of shipments**

# Generate shipments

Input:

- **Shipment number**
- **Logistic network**
- **Maximal path length**
- **Minimal** and **maximal size**

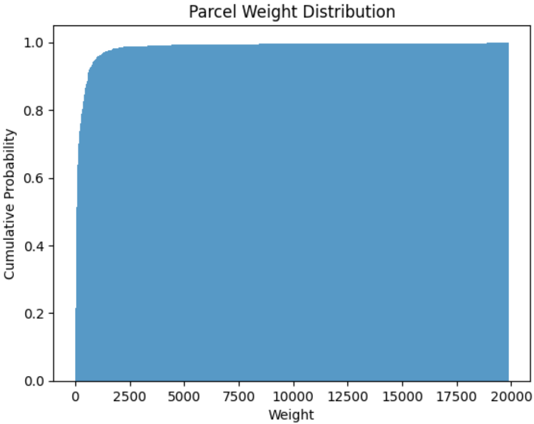
Output:

- **Set of shipments**

Procedure:

- (**Source**, **departure time**) and (**destination**, **arrival time**) are found by sampling a **path** in the space-time network
- **Sizes** are sampled with **Lomax distribution**

# Shipment sizes distribution



## Future Work

- Incorporate more complex features in the instance generator.
- Gather feedback from the research community to refine and expand Millemiglia's capabilities.

Thank you!

# BIBLIOGRAPHY I

Albert, R. and Barabási, A.-L. (2000).
Topology of evolving networks: local events and universality.
*Physical review letters*, 85(24):5234.

Eberhard, O., Cuvelier, T., Valko, M., and Backer, B. D. (2023).
Middle-mile logistics through the lens of goal-conditioned reinforcement learning.
*NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning*.