

Instances for Two-Tier Multi-Modal City Logistics

Publication date: 11.12.24

License: Creative Commons Attribution 4.0 International ([Read more](#))

DOI: 10.5281/zenodo.14361024

Creator: Dr. Julia Lange ([Google Scholar Profile](#), [0000-0001-8317-5751](#))

Affiliation: School of Business Studies and Economics, University of Kaiserslautern-Landau, Kaiserslautern, Germany

Contributors:

Teodor Gabriel Crainic ([0000-0002-4424-0984](#)) and *Walter Rei* ([0000-0001-6626-8251](#))

School of Management, Université du Québec à Montréal and CIRRELT - Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Montréal, Canada

Timo Gschwind ([0000-0002-7715-4994](#))

School of Business Studies and Economics, University of Kaiserslautern-Landau, Kaiserslautern, Germany

Motivation and Scientific Background

These instances provide data for **Planning Problems in Two-Tier Multi-Modal City Logistics**, where a set of individual commodity flows, called *demands*, must be shipped in a metropolitan area.

The considered **urban transportation system** is

- *two-tiered* - consisting of two layers with dedicated vehicle fleets and handover facilities (*satellites*) to connect them, and
- *multi-modal* - involving different modes of transportation like truck, cargo bike and public transport services.

The **outer tier** (also: first tier) connects

- *distribution centers* at the cities outskirts or industrial areas, which act as access points for demands coming from and moving towards the external zone,
- with *satellite locations*, where demands are (de)consolidated.
- A fleet of *urban vehicles* like small trucks, trams or tram compartments is operated.

The **inner tier** (also: second tier) connects

- satellites with *customer locations*.
- A fleet of *city freighters*, which are environmentally friendly vehicles like electric vehicles or cargo bikes, is operated.

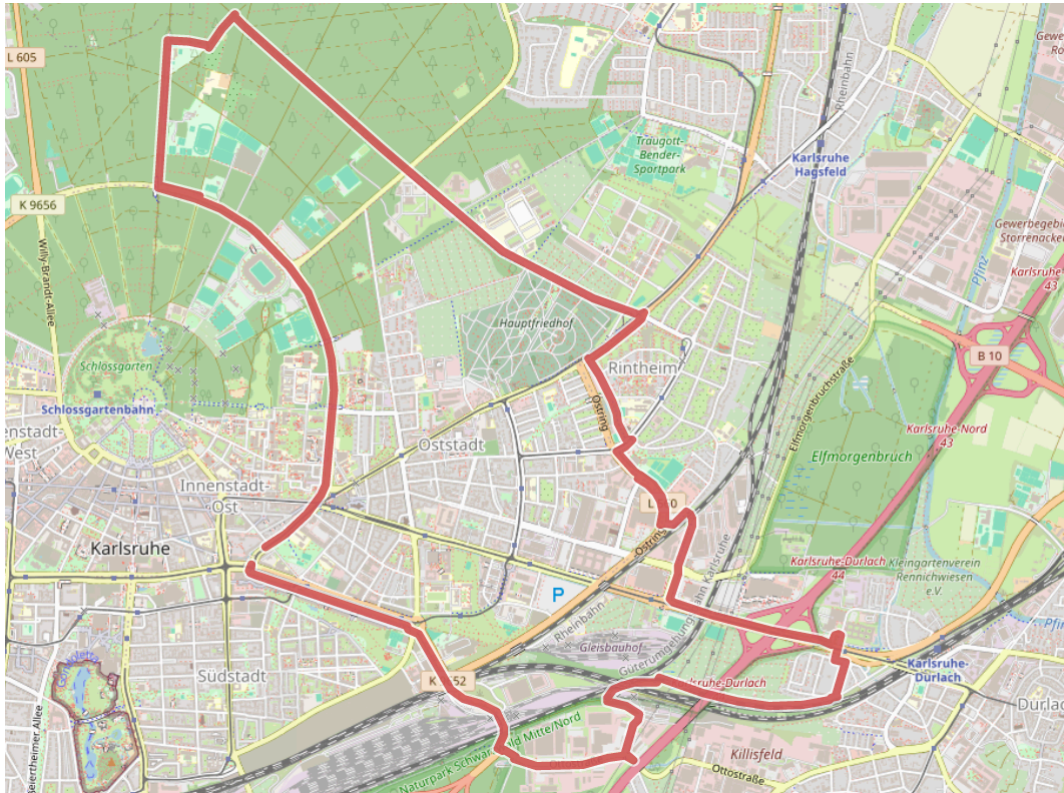
Three types of **demands** are considered:

- e2c - external to customer, inbound, delivery required,
- c2e - customer to external, outbound, pickup required,
- c2c - customer to customer, inner-city, pickup and delivery required.

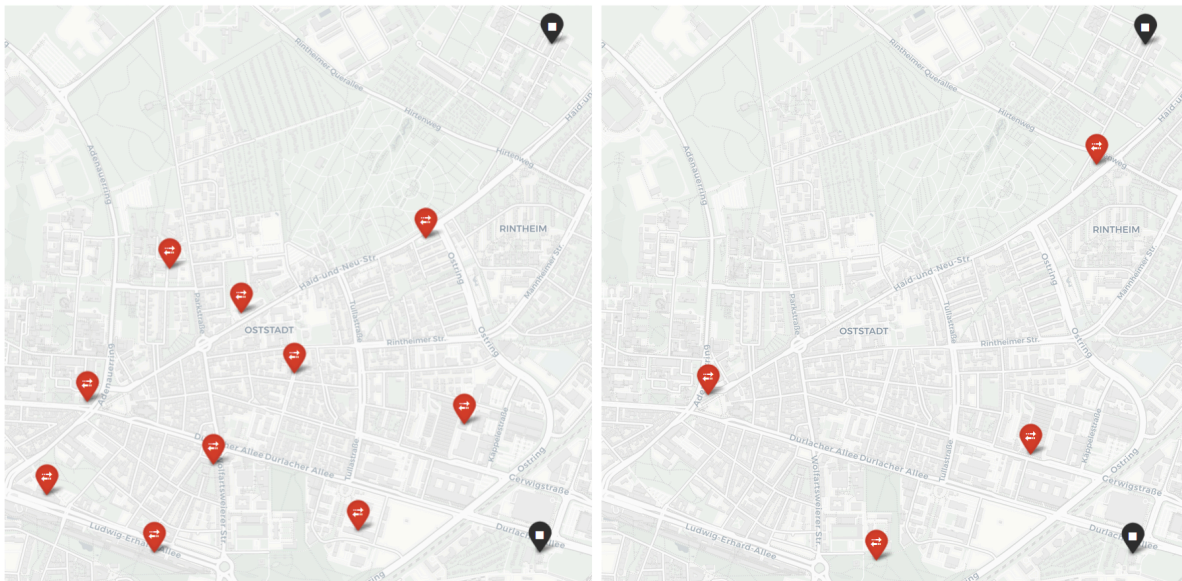
Related research areas, where these instances are applicable: City logistics, (Service) Network Design, Vehicle Routing

Geographical Background

The instances are based on the district 'Oststadt' of Karlsruhe, a medium-size German city. The neighborhood has 21,000 inhabitants. It is well-connected to 2 highways (A5 and A8) and features a public tram system.



The instances involve two distribution centers (black, squares), ten road-based and four rail-based satellites (red, arrows). Road-based satellites are formed by parking lots of supermarkets, gas stations or public institutions. Rail-based satellites are tram stations. All locations are chosen by the authors with regard to an appropriate surrounding.



The customer locations are randomly taken from the address list of the district.

Map Source: <https://maps.openrouteservice.org/>

Description of the Instance Set

General Information

Number of instances: 46,656

Instance IDs: 1000 to 47,655

Characteristics: The instances feature different parameter values, some of which are based on distributions. Whenever this is the case, 3 samples are drawn that are identified as 'replicates' with a certain ID. Every parameter value (sample) is combined with every other parameter value (sample).

Instance Data

General Information

Time Horizon (in minutes)

All time-related parameters (of demands, services and tours) are set in a way so that demands can be shipped within this time horizon. But note: capacity restrictions may still be violated and instances may get infeasible.

Locations

Distribution Center (DC)

- location ID
- address (street + house number)
- coordinates (longitude, latitude)

The same DC locations are used in all instances.

Satellite (Sat)

- location ID
- address (street + house number)
- coordinates (longitude, latitude)
- type (road or rail)
- region ID
- maximum allowed waiting time (in minutes)
- maximum allowed number of urban vehicles
- maximum allowed number of city freighters
- maximum size of the parking area (in square meters)

The considered district 'Oststadt' is clustered in regions. There is always at least one satellite per region.

Customer Locations (Cust)

- location ID
- address (street + house number)
- coordinates (longitude, latitude)
- maximum allowed waiting time (in minutes)

Demands

- origin (location ID)
- destination (location ID)

- direction (indicates type with 1: e2c, -1: c2e, 0: c2c)
- volume portion of city freighter required
- volume portion of urban vehicle required
- handover time (in minutes)
- pickup/delivery time window(s) (from minute to minute)
- availability time (in minutes)

Demand data is generated based on a set of *demand sizes* [XS, S, M, L, XL].

- Each demand size is chosen from the following distribution:
(XS: 0.1, S: 0.2, M: 0.4, L: 0.2, XL: 0.1).
- Handover times are size-dependent between 5 and 10 minutes.
- Volume portions are size- and vehicle-dependent between 0.0005 and 0.125.

Availability times do only exist for e2c demands. They are randomly determined according to the following rule:

- With a probability of 50%, the availability time is 0.
- With a probability of 50%, the availability time is chosen from a triangular distribution with lower bound and mode equal to 0 and upper bound equal to $1/2 \cdot \text{time horizon}$.

Transportation Network

Distance Matrix (Dima)

- travel time (in minutes, of corresponding vehicle)

outer-tier/urban vehicle = truck or tram, inner-tier vehicle/city freighter = cargo bike

Services (Outer Tier, Urban Vehicles)

- stops list (route)
- start time window
- fixed cost

Characteristics:

- Services have at most 4 stops and can never visit a satellite more than once.
- Every service visits at least one DC at the beginning and/or at the end of the route.
- For each satellite, a feasibility-assuring service connecting it to its nearest DC with maximum start time flexibility is involved.
- It is assured that every satellite is present sufficiently often in the set of services.
- Start time window widths can be:
time horizon/15, time horizon/10, time horizon/5, time horizon/2 and time horizon (= maximum flexibility, always available).
- Fixed cost = total travel time * flexibility multiplier,
where flexibility multiplier is 2 for start time window width equal to time horizon/2 and 3 for start time window width equal to time horizon.

Tours (Inner/Second Tier)

- stops list (route)
- fixed cost

Characteristics:

- Tours start and end at a satellite, might be cyclic or acyclic and can never visit a customer more than once.
- For each customer location, a feasibility-assuring tour starting and ending at its nearest satellite and visiting only this customer location is involved.
- For each c2c demand, a feasibility-assuring tour visiting the two required customer locations and starting and ending at their nearest satellites is involved.
- Except for the feasibility-assuring tours, the sequences of customer locations visited are randomly generated and time windows of demands are not considered.
- Fixed cost = total travel time

Instance Parameters and Dependencies

The **instance parameters** varied are:

- Number of customer locations (= number of demands)
- Satellite type
- Satellite density
- Time horizon
- Demands type distribution (e2c portion, c2e portion, c2c portion)
- Time window widths distribution (short, medium, long)
- Customer waiting duration policy distribution (no, short, long)

For some parameters, abbreviations are used to indicate the parameter values in the instance file names. The abbreviations are described below and their usage is explained in the next section.

Number of Customer Locations and Number of Demands

- 10, 20, 30
- Number of demands equals number of customer locations for each instance.

Customer locations are sampled 3 times per number of locations. Instances in files '...10-1...' involve the same 10 customer locations (customer sample 1).

Demands may require pickup/delivery at the same customer location, only when c2c demands are involved.

Satellite Type (st)

- road, also: stRO
- rail, also stRA

The satellite type defines the urban vehicle type used: road-based satellites require small trucks, rail-based satellites require trams.

Satellite Density (sd)

- as given, also: sdAG
10 for road-based satellites, 4 for rail-based satellites, deterministic
- halved, also: sdHA
for road-based satellites only, 5 satellites, equally distributed over 2 regions, selection sampled 3 times
- minimal, also: sdMI
for road-based satellites only, 2 satellites, one per region, selection sampled 3 times

Time Horizon

- 120 minutes, 240 minutes

Demand Type Distribution (dp)

- (1, 0, 0), also: dpOIN
only e2c (inbound) demands, classical setting, deterministic
- (0.8, 0.2, 0), also: dpIO
80% e2c (inbound) and 20% (outbound) c2e demands, demand type to customer location assignment sampled 3 times
- (0.65, 0.2, 0.15), also: dpRE
65% e2c, 20% c2e and 15% c2c demands (regular city logistics scenario), demand type to customer location assignment sampled 3 times

Time Window Widths Distribution (tw)

- (1, 0, 0), also: twOSH
all time windows 15 minutes in width, deterministic
- (0,1, 0), also: twOME
all time windows 30 minutes in width, deterministic
- (0, 0, 1), also: twOLO
all time window widths = time horizon, represents no active time windows, deterministic
- (1/3, 1/3, 1/3), also: twEQ
33% short, 33% medium and 33% long time window widths, time window width to demand assignment sampled 3 times

Customer Waiting Policy Distribution (cwt)

- (1, 0, 0), also: cwtOSH
no waiting allowed at any customer location, deterministic
- (0,1, 0), also: cwtOME
maximum waiting duration at all customer locations = 10 minutes, deterministic
- (0, 0, 1), also: cwtOLO
maximum waiting duration at all customer locations = time horizon, represents no active waiting duration restrictions, deterministic
- (1/3, 1/3, 1/3), also: cwtEQ
33% no, 33% short and 33% long waiting allowed, maximum waiting duration to customer location assignment sampled 3 times

Technical Details

Files and Directories

The folder **2TCL_ka_instances.zip** contains 46,656 instance files of type JSON.

The following files provide additional information on the instance set

- *CLInstance_Characteristics_Abbrev.txt* -
shows translation of abbreviations of instance generation parameter values used in file names
- *CLInstances_Characteristics_Overview_1000-47655.xlsx* and
CLInstances_Characteristics_Overview_1000-47655.json -
hold summary of instance parameter values for all instances

Instance Files

format: JSON (.json)

File Name

```
1 ka_east_{number of customers}-{customer replicate}_\  
2 {sat. type}_{sat. density}-{sat. density replicate}_\  
3 ttBI_{time horizon}_\  
4 {customer waiting policy distr.}-{customer waiting policy replicate}_\  
5 {number of demands}_{demand type distr.}_\  
6 {time window widths distr.}-{time window widths replicate}_\  
7 {dsDE}-{demand size replicate}_\  
8 {instance ID}.json
```

Abbreviations are used to indicate instance parameter values. See description above and overview in CLInstance_Characteristics_Abbrev.txt

Data Hierarchy and Structure

```
1 - locations:  
2   - dc:  
3     - location ID:  
4       - address: string  
5       - coordinates: [int, int] # [longitutde, lattitude]  
6   - sat:  
7     - location ID:  
8       - address: string  
9       - coordinates: [int, int] # [longitutde, lattitude]  
10      - type: string # road or rail  
11      - region: int  
12      - maxWaitingTime: int  
13      - maxNumCityFreighter: int  
14      - maxNumUrbanVehicle: int  
15      - sizeOfParkingArea: int  
16   - cust:  
17     - location ID:  
18       - address: string  
19       - coordinates: [int, int] # [longitutde, lattitude]  
20       - maxWaitingTime: int  
21 - dima: [nested list of floats] # list index = location ID - 1  
22                                     # 9999.0 = link does not exist  
23 - timehorizon: int  
24 - demands:  
25   [  
26     [int, int, int, float, float, int, [[int, int], [int, int]], float]  
27     # origin, destination, direction, \  
28     # vol. portion city freighter, vol. portion urban vehicle, \  
29     # handover time, \  
30     # [pickup time window, delivery time window], availability time  
31   ]  
32 - tours:  
33   [  
34     [[location ID, location ID, ...], float, 1]  
35     # [stops list], fixed cost, dummy for variable cost  
36   ]  
37 - services:  
38   [  
39     [[location ID, location ID, ...], [int, int], float, 1]
```

```

40         # [stops list], [start time window], fixed cost, dummy for variable cost
41     ]
42     - characteristics:
43         - Number of Customers: int
44         - Customer Replicate: int,          # 1, 2, 3
45         - Satellite Type: string,          # road or rail
46         - Satellite Density: string,        # as given, halved, minimal
47         - Satellite Density Replicate: int, # 1, 2, 3
48         - Tour Vehicle: "bike",
49         - Instance Density: int,           # 120, 240 (equal to time horizon)
50         - Time Horizon: int,
51         - Number of Demands: int,
52         - Demand Type Distribution: tuple string,
53         - Demand Size Distribution: "default",
54         - Demand Size Replicate: int,      # 1, 2, 3
55         - TW Widths Distribution: tuple string,
56         - TW Widths Distr. Replicate: int, # 1, 2, 3
57         - Customer Waiting Policy Distribution: tuple string,
58         - Customer Waiting Policy Replicate: int, # 1, 2, 3
59         - Number of Tours: int,
60         - Number of Services: int,
61         - Instance ID: int

```

The object is nicely processable as a Python dictionary.

Values taken from distributions are sampled 3 times. In case of deterministic values, the replicate number is set to 1.

Instance Characteristics Overview

format: JSON (.json)

Data Hierarchy and Structure

```

1  - Id: {row index: int}
2  - Name: {row index: "ka_east"}
3  - NumCustomers: {row index: int}
4  - CustomerReplicate: {row index: int}
5  - NumDemands: {row index: int}
6  - DemandTypeDistribution: {row index: tuple string}
7  - DemandSizes: {row index: "default"}
8  - DemandSizeReplicate: {row index: int}
9  - Sattype: {row index: string}
10 - SatDensity: {row index: string}
11 - SatDensityReplicate: {row index: int}
12 - NumSatellites: {row index: int}
13 - TourType: {row index: "bike"}
14 - NumTours: {row index: int}
15 - NumServices: {row index: int}
16 - InstanceDensity: {row index: int}
17 - TimeHorizon: {row index: int}
18 - TWwidthsDistr: {row index: tuple string}
19 - TWwidthsDistrReplicate: {row index: int}
20 - CustWaitPolicyDistr: {row index: tuple string}
21 - CustWaitPolicyDistrReplicate: {row index: int}
22 - FileName: {row index: string}
23 - GenerationDateTime: {row index: date time string}

```

This object is nicely processable as a pandas data frame using `pandas.read_json()`.

General Remarks

Note

- Addresses are given in German format. Street Name + House Number
- Coordinate are given as [Longitude, Latitude].
- Numbers are given in English notation. Decimals indicated by '.', example: 24.78 (a number between 24 and 25)
- The location IDs always start with 1 and 2 (the DCs), continue with the satellites (3 to x) and finish with the customer locations (x+1 to x+n).
- **Feasibility:** All time restrictions are set in a way so that feasibility is best possibly assured. However, it may appear that instances are infeasible due to capacity restrictions (too many demands at the same time in the same vehicle or at the same satellite).