Two-sided mobility platforms.

Ride-pooling in pandemic
+ Supply-demand interactions

Rafał Kucharski r.m.kucharski@tudelft.nl

Intro

myself

Rafał Kucharski

- now: PostDoc @ TU Delft working in Critical MaaS team of prof. Oded Cats
- before:
  - PhD in Dynamic Traffic Assignment: *Modelling Rerouting Phenomena* (with prof. Guido Gentile, Rome)
  - R&D software developer (PTV SISTeMA)
  - transport modeller (models for Kraków, Warsaw and more)
  - data scientist, ML/AI (NorthGravity)
- future: Assistant Professor at Machine Learning Group at Faculty of Mathematics and Computer Science, Jagiellonian University (Krakko) - under DigiWorld programme.

our group

Critical MaaS

Critical MaaS

ERC Starting Grant, 5 years,
PI (prof. Oded Cats + 4 PhDs, 1 PostDoc (me)
Civil Engineering, Transport & Planning, TU Delft (ranked 1st in Europe in Transport Engineering)

Scope

PhD students:

Supply	Peyman Ashkrof
Evolution	Arjan de Ruijter
Demand	Nejc Gerzinic
Interactions	Subodh Dubey

The CriticalMaaS research program

develops and tests theories and models to explain and predict the performance of (potentially fully-automated) flexible on-demand transport services offered by Mobility as a Service providers at strategic, tactical and real-time operation levels by identifying key determinants of level-of-service, including the consideration of travel demand patterns, traveller and operator behavioural preferences, service design and fleet allocation and management with a special focus on system-wide accessibility, efficiency and equity effects.

CRITICAL MAAS

TU Delft

lab Smart Public Transport

Outline

1. Intro
   - myself
   - our group
2. Two-sided mobility platforms
   - introduction
3. Ride-pooling
   - introduction
   - algorithm
   - ExMAS
   - results
4. Agent-based two-sided mobility platform simulator
   - MaaSSim
   - Agents

Two-sided mobility platforms

Two-sided platforms

Two-sided mobility platform:

- **two-sided** supply (drivers, vehicles) and demand (travellers)
- **platform** connects supply and demand
- **mobility** offering travellers to supply their mobility needs (reach a destination)

TU Delft Lab Smart Public Transport

Case-studies

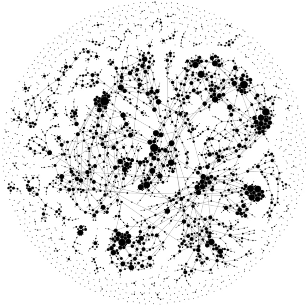
Today:

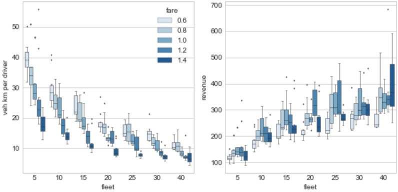
Kucharski, R., Cats, O., & Sienkiewicz, J. (2021). **Modelling virus spreading in ride-pooling networks**. Scientific Reports, 11(1), 1-11.

How do viruses spread in ride-polling networks and how to control it?

Kucharski, R., & Cats, O. (2020). **MaaSSim–agent-based two-sided mobility platform simulator**. arXiv preprint arXiv:2011.12827.

How individual agents make independent decision that yield complex system dynamics?





TUDelft
lab
Smart Public Transport

Case-studies

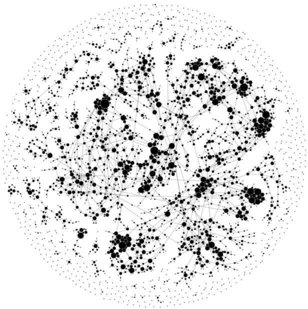
Today:

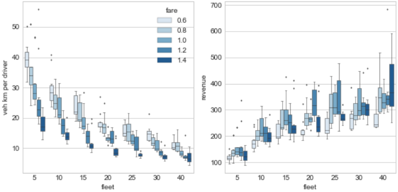
Kucharski, R., Cats, O., & Sienkiewicz, J. (2021). Modelling virus spreading in ride-pooling networks. Scientific Reports, 11(1), 1-11.

How do viruses spread in ride-polling networks and how to control it?

Kucharski, R., & Cats, O. (2020). **MaaSSim–agent-based two-sided mobility platform simulator**. arXiv preprint arXiv:2011.12827.

How individual agents make independent decision that yield complex system dynamics?





Ride-pooling

Modelling virus spreading in ride-pooling networks

an intermediate alternative for pandemic urban mobility

Context:

1. crowded public transport systems can be a major contributor to virus spreading.
2. but individual, private rides in cars will cause congestion and capacity issues.
3. what is in between?

Looking for alternative urban mobility mode for pandemic cities

Ride-pooling

Shared ride:

1. two or more travellers can be matched into a pooled ride and travel in the same ride-hailing vehicle.
2. vehicle picks them up from origins and drops-them off at their destinations,
3. both pickup and travel times deviate from the desired or minimal ones,
4. this inconvenience needs to be compensated with a lower fare compared to an individual ride,
5. service provider can now:
    - better utilise its capacity
    - charge several users for a ride
    - while paying a single driver commission.

Ride-pooling

Shared ride:

1. two or more travellers can be matched into a pooled ride and travel in the same ride-hailing vehicle.
2. vehicle picks them up from origins and drops-them off at their destinations,
3. both pickup and travel times deviate from the desired or minimal ones,
4. this inconvenience needs to be compensated with a lower fare compared to an individual ride,
5. service provider can now:
   - better utilise its capacity
   - charge several users for a ride
   - while paying a single driver commission.

Ride-pooling

Shared ride:

1. two or more travellers can be matched into a pooled ride and travel in the same ride-hailing vehicle.
2. vehicle picks them up from origins and drops-them off at their destinations,
3. both pickup and travel times deviate from the desired or minimal ones,
4. this inconvenience needs to be compensated with a lower fare compared to an individual ride,
5. service provider can now:
   - better utilise its capacity
   - charge several users for a ride
   - while paying a single driver commission.

Ride-pooling

Shared ride:

1. two or more travellers can be matched into a pooled ride and travel in the same ride-hailing vehicle.
2. vehicle picks them up from origins and drops-them off at their destinations,
3. both pickup and travel times deviate from the desired or minimal ones,
4. this inconvenience needs to be compensated with a lower fare compared to an individual ride,
5. service provider can now:
   - better utilise its capacity
   - charge several users for a ride
   - while paying a single driver commission.

Ride-pooling

Shared ride:

1. two or more travellers can be matched into a pooled ride and travel in the same ride-hailing vehicle.
2. vehicle picks them up from origins and drops-them off at their destinations,
3. both pickup and travel times deviate from the desired or minimal ones,
4. this inconvenience needs to be compensated with a lower fare compared to an individual ride,
5. service provider can now:
   - better utilise its capacity
   - charge several users for a ride
   - while paying a single driver commission.

Model

how virus can spread in ride-pooling

1. it is quite obvious (and safe to assume) that you infect your co-riders.
2. but what happens beyond a single vehicle?

Everyday we apply the SIQR model with transitions taking place when:

1. infected travellers infect their susceptible co-riders ($S \to I$),
2. infected travellers are quarantined after the incubation period ($I \to Q$),
3. travellers recover after the quarantine and acquire complete immunity to the virus ($Q \to R$).

The loop terminates when all the infected travellers are quarantined (there are no active infections).

Model

how virus can spread in ride-pooling

1. it is quite obvious (and safe to assume) that you infect your co-riders.
2. but what happens beyond a single vehicle?

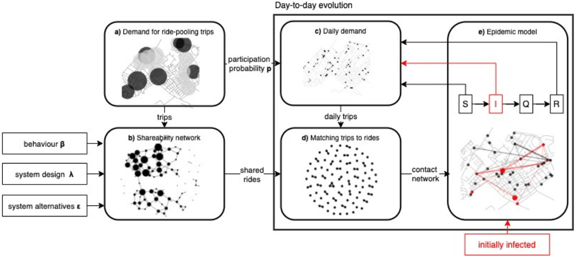
Everyday we apply the SIQR model with transitions taking place when:

1. infected travellers infect their susceptible co-riders ($S \rightarrow I$),
2. infected travellers are quarantined after the incubation period ($I \rightarrow Q$),
3. travellers recover after the quarantine and acquire complete immunity to the virus ($Q \rightarrow R$).

The loop terminates when all the infected travellers are quarantined (there are no active infections).

Algorithm

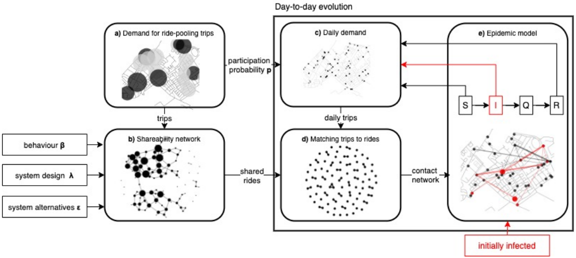
Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

Algorithm

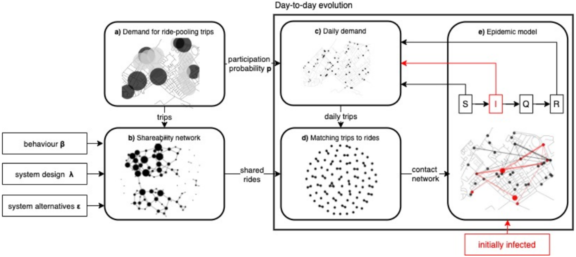
Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

Algorithm

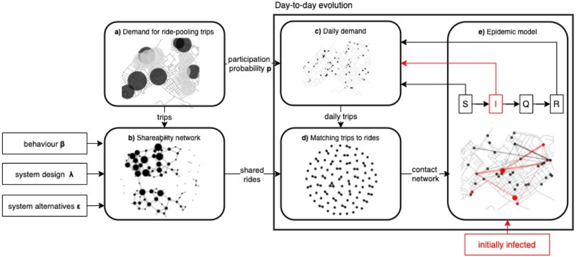
Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

Algorithm

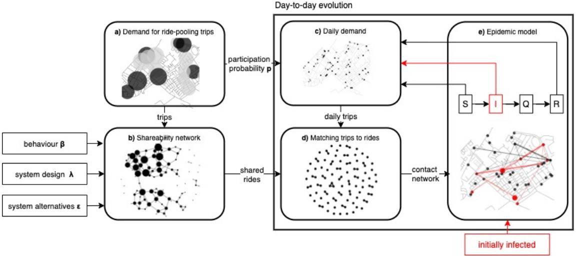
Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

Algorithm

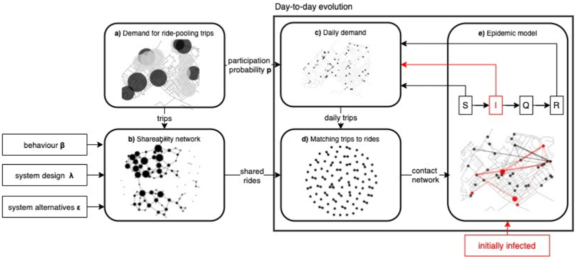
Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

Algorithm

Methodology at glance:



1. We consider travel demand for ride-pooling trips (a),
2. for which we compute a shareability network (b) with a given behavioural parameters β, system design λ and alternatives' attractiveness ϵ.
3. We simulate the day-to-day evolution of spreading until the virus is halted.
4. Each day we obtain the daily demand (c), consisting of those who want and can travel (decided to travel with probability p and are not quarantined).
5. Daily trip demand is optimally assigned to shared rides,
6. which forms the contact network (d) on which virus spreading is then modelled (e).

ExMAS

python based open-source package applicable to general networks and demand patterns

ExMAS [1], for a given:

1. network (`osmnx` graph),
2. demand (microscopic set of trips $q_i = (o_i, d_i, t_i)$)
3. parameters (behavioural, like willingness-to-share and system like discount for shared rides)

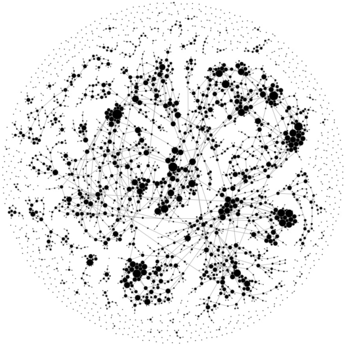
computes:

1. optimal set of **pooled rides** (results of bipartite matching with a given objective)
2. shareability graph
3. set of all feasible rides

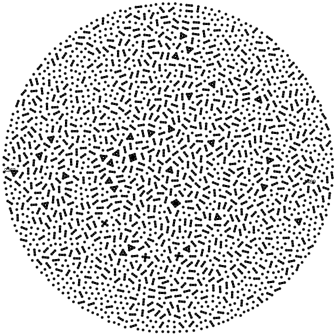
[1] https://github.com/RafalKucharskiPK/ExMAS

Contact network

shareability network:
travellers (nodes) are linked if the can share a ride together (are compatible), node size proportional to degree



actual matching:
each clique is a vehicle and travellers within vehicle are isolated from others and fully connected with co-travellers.



TU Delft lab Smart Public Transport

Control variable

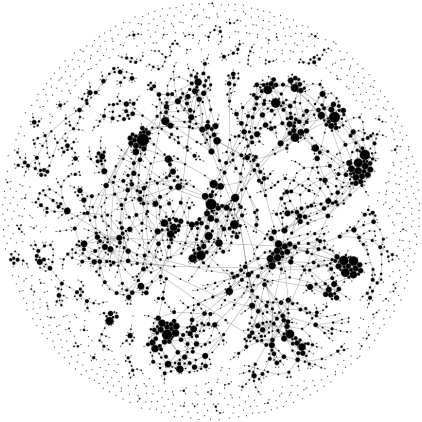
Demand stability

We identify an effective **control measure** allowing to halt the spreading before the outbreaks without sacrificing the efficiency achieved by pooling.

p demand-stability

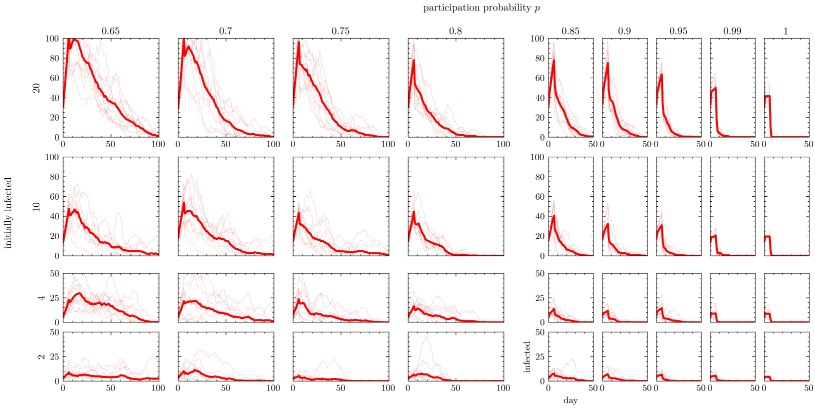
Each traveller participates in the system with probability of p.
Total number of travellers (nodes) remains the same $2000 =$ sample$(2000|2000/p)$

Fixed matches among co-travellers disconnect the otherwise dense contact network, encapsulating the virus in small communities and preventing the outbreaks.



Results

Number of infected in time

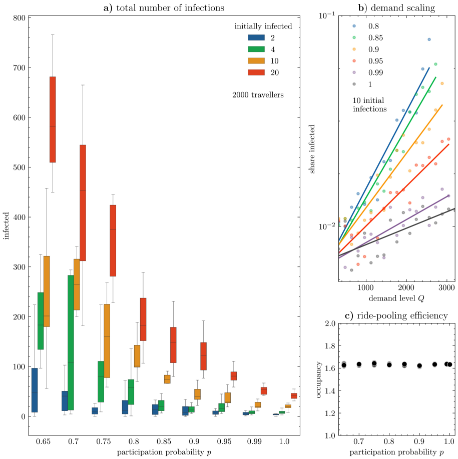


Number of infected travellers over the course of epidemic outbreaks, with various settings of initially infected (rows) and demand stability (columns), bold lines denote averages over all experiments (shown individually using thin lines)

TU Delft
Lab Smart Public Transport

Results

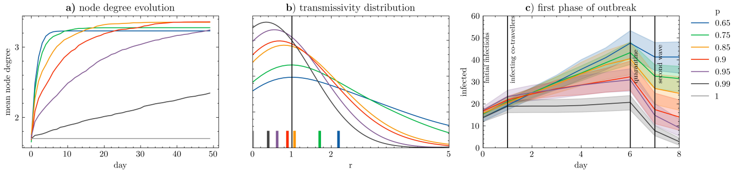
Scaling



a) Number of eventually infected travellers for varying demand stability p and initially infected travellers.
(b) share of infected travellers changing with a demand for various p's.
Importantly, stabilizing the demand does not reduce the efficiency, as we report in **(c)**, where the mean occupancy (key efficiency indicator of ride-pooling) remains stable as demand stabilizes.

Results

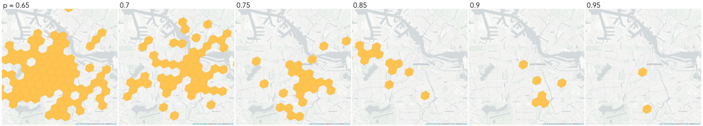
Network properties



a) Average node degree in the evolving contact networks. b) Mean transmission rate r (number of new infections per infected) distributions. c) Insights into the first phase of the epidemic outbreak in the case of 10 initial infections.

Results

Spatial spreading



An illustration of the spatial extent of epidemic outbreaks originating from two initial infections. A major part of Amsterdam becomes infected for spontaneous demand (left), while it remains spatially contained as the demand stabilises (right). For stable demand ($p > 0.8$) the geographical boundaries are confined, while otherwise, the virus crosses the river Ij and reaches also the north parts of Amsterdam.

Spreading

summary

1. complex system
2. highly stochastic
3. highly connected shareability network
4. highly disconnected its daily realisation (assignment to the vehicles)
5. massive spreading under unstable demand
6. effective halting with stable demand

Limitations/Assumptions/Caveats

one trip per-day

driver is not a spreader

Agent-based two-sided mobility platform simulator

MaaSim

Agent-based two-sided mobility platform simulator

MaaSSim

open source · python · lightweight · agent-based · simulator

The **why's**:

motivation emerging service, disruptive to urban mobility landscape

new to focus on phenomena central to two-sided platforms and not well-studied traffic flow, route choice, congestion, etc. Faster learning curve than well-established full-stack `MatSim`, `SUMO`, etc.

challenging independent decision makers: heterogenous, individual, adaptive, strategic

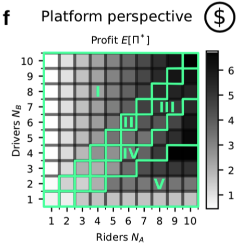
complex system dynamics driven by multiple agent classes

MaaSSim

Use-case

Let's simulate a system in which:

1. travellers choose among public transport, ride-hailing (Uber) and ride-pooling
2. drivers decide whether to work for the platform or not
3. platform sets a fare and commission for drivers



MaaSSim

Agents

Travellers

1. may be assigned to multiple platforms and submit request to all of them to choose the best offer amongst those.
2. A traveller unsatisfied with previous experience may opt-out before requesting.
3. When receiving an offer the traveller makes a decision whether to accept it or not.

While accepting she/he walks to the pick-up point, waits until the driver arrives, travels to the drop-off point and walks to the final destination, which terminates traveller's daily routine.

Drivers

operate in a loop, queuing to the platform and serving matched requests until the end of their shift.
Along their routines, drivers decide whether to:

1. opt out before starting a shift and not enter the platform at all;
2. accept or reject the incoming requests
3. re-position after becoming idle.

Platforms

matches a two-sided queue of travellers on one side and drivers on another.
For a given price (traveller) and commision (driver).

MaaSSim

Agents

Travellers

1. may be assigned to multiple platforms and submit request to all of them to choose the best offer amongst those.
2. A traveller unsatisfied with previous experience may opt-out before requesting.
3. When receiving an offer the traveller makes a decision whether to accept it or not.

While accepting she/he walks to the pick-up point, waits until the driver arrives, travels to the drop-off point and walks to the final destination, which terminates traveller's daily routine.

Drivers

operate in a loop, queuing to the platform and serving matched requests until the end of their shift.
Along their routines, drivers decide whether to:

1. opt out before starting a shift and not enter the platform at all;
2. accept or reject the incoming requests
3. re-position after becoming idle.

Platforms

matches a two-sided queue of travellers on one side and drivers on another.
For a given price (traveller) and commision (driver).

MaaSSim

Agents

Travellers

1. may be assigned to multiple platforms and submit request to all of them to choose the best offer amongst those.
2. A traveller unsatisfied with previous experience may opt-out before requesting.
3. When receiving an offer the traveller makes a decision whether to accept it or not.

While accepting she/he walks to the pick-up point, waits until the driver arrives, travels to the drop-off point and walks to the final destination, which terminates traveller's daily routine.

Drivers

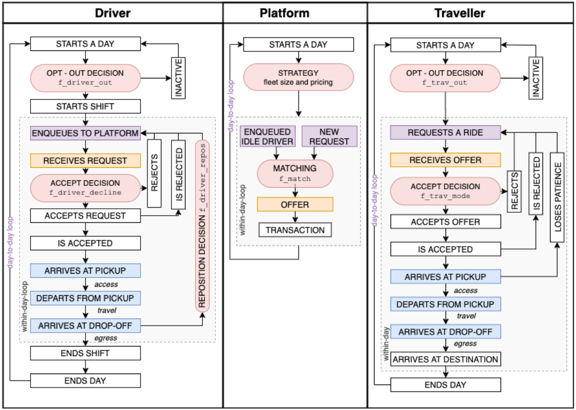
operate in a loop, queuing to the platform and serving matched requests until the end of their shift.
Along their routines, drivers decide whether to:

1. opt out before starting a shift and not enter the platform at all;
2. accept or reject the incoming requests
3. re-position after becoming idle.

Platforms

matches a two-sided queue of travellers on one side and drivers on another.
For a given price (traveller) and commision (driver).

MaaSSim Agent routines



drivers

· leaving the system · accepting requests · re-positioning

travellers

· accepting offers,
· selecting platforms and modes,
· leaving the system

platform

setting prices matching request

Smart Public Transport

MaaSSim

Usage

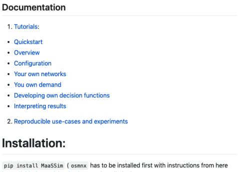
```
from MaaSSim.simulators import simulate, simulate_parallel
from MaaSSim.utils import get_config, load_G
from MaaSSim.utils import prep_supply_and_demand, collect_results

sim = simulate() # run MaaSSim simulation
sim.runs[0].trips  # access the results
params = get_config('default.json')  # load configuration
params.city = "Nootdorp, Netherlands" # modify it
inData = load_G(params)  # load different network graph
params.nP = 50 # modify number of travellers
inData = prep_supply_and_demand(inData, params)  # regenerate supply and demand
sim2 = simulate(inData, params) # rerun the simulation with new data and parameters
print('Simulated wait times: {}s and {}s.'.format(sim.res[0].pax_exp['WAIT'].sum(),
      sim2.res[0].pax_exp['WAIT'].sum()))  # compare some results

space = {nP=[5,10,20], nV = [5,10]} # define the search space to explore in experiments
simulate_parallel(inData, params, search_space = space) # run parallel experiments
res = collect_results(params.paths.dumps) # collect results from so mparallel experiments

def my_function(**kwargs): # user defined function to represent agent decisions
    veh = kwargs.get('veh', None)  # input
    sim = veh.sim  # access to the simulation object
    if len(sim.runs)==0 or sim.res[last_run].veh_exp.loc[veh.id].nRIDES > 3:
        return False # if I had more than 3 rides yesterday I stay
    else:
        return True # otherwise I leave

sim = simulate(inData,params, f_driver_out = my_function) # run MaaSSim with user-defined function
```

Smart Public Transport

MaaSSim

public repository

1. public repository
2. open, short code
3. module, rather than a software
4. tutorial, examples, jupyter notebooks



Documentation

1. Tutorials:

- Quickstart
- Overview
- Configuration
- Your own networks
- You own demand
- Developing own decision functions
- Interpreting results

2. Reproducible use-cases and experiments

Installation:

`pip install MaaSSim` (`osmnx` has to be installed first with instructions from here

`https://github.com/RafalKucharskiPK/MaaSSim`

Questions

Discussion

Thank you!

Rafał Kucharski,
PostDoc @ Critical MaaS,
Tansportation and Planning TU Delft,
r.m.kucharski@tudelft.nl[2]

2 This research was supported by the CriticalMaaS project (grant number 804469), which is financed by the ERC and Amsterdam Institute of Advanced Metropolitan Solutions.