

## Simulating two-sided mobility platforms with MaaSSim

Kucharski, Rafal; Cats, Oded

**DOI**

[10.1371/journal.pone.0269682](https://doi.org/10.1371/journal.pone.0269682)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

PLoS ONE

**Citation (APA)**

Kucharski, R., & Cats, O. (2022). Simulating two-sided mobility platforms with MaaSSim. *PLoS ONE*, 17(6 June), Article e0269682. <https://doi.org/10.1371/journal.pone.0269682>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

## RESEARCH ARTICLE

# Simulating two-sided mobility platforms with MaaSSim

Rafal Kucharski<sup>1,2\*</sup>, Oded Cats<sup>1</sup>

**1** Department of Transport & Planning, TU Delft, Delft, The Netherlands, **2** Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

\* [rafal.kucharski@uj.edu.pl](mailto:rafal.kucharski@uj.edu.pl)



## Abstract

Two-sided mobility platforms, such as Uber and Lyft, widely emerged in the urban mobility landscape. Distributed supply of individual drivers, matched with travellers via intermediate platform yields a new class of phenomena not present in urban mobility before. Such disruptive changes to transportation systems call for a simulation framework where researchers from various and across disciplines may introduce models aimed at representing the complex dynamics of platform-driven urban mobility. In this work, we present MaaSSim, a light-weight agent-based simulator reproducing the transport system used by two kinds of agents: (i) travellers, requesting to travel from their origin to destination at a given time, and (ii) drivers supplying their travel needs by offering them rides. An intermediate agent, the platform, matches demand with supply. Agents are individual decision-makers. Specifically, travellers may decide which mode they use or reject an incoming offer; drivers may opt-out from the system or reject incoming requests. All of the above behaviours are modelled through user-defined modules, allowing to represent agents' taste variations (heterogeneity), their previous experiences (learning) and available information (system control). MaaSSim is a flexible open-source python library capable of realistically reproducing complex interactions between agents of a two-sided mobility platform. MaaSSim is available from a public repository, along with a set of tutorials and reproducible use-case scenarios, as demonstrated with a series of illustrative examples and a comprehensive case study.

## OPEN ACCESS

**Citation:** Kucharski R, Cats O (2022) Simulating two-sided mobility platforms with MaaSSim. PLoS ONE 17(6): e0269682. <https://doi.org/10.1371/journal.pone.0269682>

**Editor:** Xinwu Qian, The University of Alabama, UNITED STATES

**Received:** September 16, 2021

**Accepted:** May 25, 2022

**Published:** June 9, 2022

**Copyright:** © 2022 Kucharski, Cats. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** Software source code is stored at public github repository <https://github.com/RafalKucharskiPK/MaaSSim>, including input files to reproduce paper's content.

**Funding:** The research conducted by OC and RK was supported by the CriticalMaaS project (no. 804469) which is financed by the European Research Council and the Amsterdam Institute for Advanced Metropolitan Solutions. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## 1 Introduction

Two-sided mobility platforms (like Uber and Lyft) match supply with demand in urban transportation systems. Users submit travel requests in real-time and are matched with drivers, offering to take them to their desired destination. All parties are independent decision-makers acting according to their individual, heterogeneous preferences and learning from past experiences. Travellers are free to select among competing platforms and travel modes. Drivers choose whether to work for the available platforms, freely decide on their working hours and strategically select served requests to maximise revenues. Finally, platforms make strategical decisions to maximise their own profit while being attractive both for the supply (by offering

**Competing interests:** The authors have declared that no competing interests exist.

attractive profit for the drivers) and for the demand (by providing services with high reliability and attractive prices).

The emergence of such two-sided mobility markets disrupts the transport landscape. Conventional models for transport planning and operations are focused on top-down planning of service lines, timetables and traffic control measures which are not directly suitable for capturing the double-sided decision-making process and the dynamics of mobility on demand services. A wide array of research questions has consequently emerged, ranging from traffic flow, labour economics, real-time control and optimisation to travel behaviour.

This bursting stream of research calls for the development of a unifying simulation framework under which emerging models, algorithms and approaches may be integrated. Recent changes are disruptive enough to justify a new framework that explicitly accounts for both supply-side and demand-side dynamics, as well as their interaction with the intermediate matching platform.

In particular, in order to capture the bottom-up emerging order resulting from two-sided mobility, it is essential to revise the modelling approach of key elements of the transport system: demand (which is now inherently microscopic), supply (which has become a decision-maker) and a road network (which capacity and congestion are no longer a single pivot variable of assignment models) along with a new agent, the platform, which orchestrates supply and demand interaction and which might be subject to regulation.

Furthermore, the already interdisciplinary field of transportation science has recently gained increasing interest from various fields such as complex network theory, system dynamics, social networks, marketing economics and computational physics. This makes it particularly timely to support a fast learning curve by offering a quick, minimal setup to reproduce the basic dynamics of two-sided mobility platforms. To allow researchers to contribute to their domain, we introduce modular software that requires minimal knowledge of other modules while allowing to enrich the overall experimental analysis.

## 1.1 State of the art and contribution

Understanding and modelling two-sided mobility systems are challenging and require a broad set of interdisciplinary expertise [1–3]. While modelling the relations between the different actors involved in two-sided platforms proved to be non-trivial [4, 5], their manifestation in the context of a dense and congested urban mobility network induces additional complexity.

Since the emergence of the platform economy in the mobility context, empirical evidence has been collected and analysed, revealing new phenomena that call for further, model-based analyses. Prime examples of which include the comparison of services offered by public transport and ride-hailing platforms [6], driver acceptance behaviour [7], drivers' labour choices [8], Mobility as a Service (MaaS) adoption [9], mode choice [10] and spatio-temporal demand patterns [11].

This, in turn, opens a class of novel research problems arising from the emergence of two-sided mobility platforms including:

- dynamic interactions between supply and demand [12, 13];
- matching of drivers to requests with various optimisation criteria and algorithmic structures [14–17];
- travellers' mode and platform choices and attitudes towards emerging modes of transportation [18–20];
- drivers' strategical decision of participation, working shifts and platform choices [7, 21, 22];

- labour economics and the impact of decentralised platform operations [23–25];
- platform pricing strategies, including discriminatory data-driven pricing strategies [26];
- pooled rides, promising in terms of sustainability and efficiency, yet posing significant computational and organisational challenges [27–30];
- fleet operations and management of shared autonomous vehicles [31–34]; and
- re-positioning strategies of drivers and fleet operators [35, 36].

Each of these research domains gives rise to a series of significant and challenging research questions. Answering each of which is non-trivial, yet the main challenge lies in representing the complete system with its (inter)dependencies and feedback loops, non-determinism and adaptive evolution.

In this rapidly developing research field, multiple studies addressed emerging problems by means of simulation frameworks, typically agent-based. Starting from classical taxi operations [37] and extending to emerging modes of car-pooling [38], ride-sharing [39] and ride-hailing [29]. In the absence of an encompassing modelling framework, studies have often been limited to a single aspect. For example, [27] focuses on travellers' behaviour and neglects fleet operations, whereas [28] focuses on real-time fleet operations neglecting the travellers' decision process; and [21] relies on an abstract grid network to focus on income equity. Similarly, in STARS [40] travellers are matched with drivers to travel together towards a destination in space and time yet both demand and supply are fixed inputs; in [41] travellers may leave the system if they are unsatisfied, but this is a fixed condition rather than a full behavioural model; while in [42] the day-to-day evolution of both sides is simulated for a fixed within-day behaviour.

More complete solutions typically extend established simulation frameworks. This is most notably the case for *SimMobility* [43] and *MATSim* [44]. In [43], supply and demand agents are heterogeneous decision-makers, making daily decisions to participate in the system or not, yet within-day decisions (acceptance) and competition of multiple platforms are not considered. In *AMoDeus* [44] (which builds on top of *MATSim*—an urban mobility simulation framework) the focus is on autonomous mobility—while travellers are individual decision-makers, the supply is composed of a fully-controllable self-driving fleet. Both learning and adaptation can be indirectly implemented into *AMoDeus* via *MATSim* which allows more detailed demand models (e.g. activity plans) and traffic flow models (to reproduce congestion) but also requires significant effort to implement inside a complex *Java* environment.

Consequently, we introduce *MaaSSim*, which is the first modular, extensible framework that contains the fundamental representation of key unique and novel phenomena related to two-sided mobility platforms. *MaaSSim* is the first publicly available python simulation framework allowing to reproduce within-day and day-to-day dynamics of travellers, drivers and platforms. Key features of a two-sided mobility platform (non-deterministic, adaptive, heterogeneous behaviour of agents interacting with each other) are explicitly handled via user-defined and flexible python functions, allowing to reproduce the desired behaviour and to trace emerging complex dynamics. An extensive set of tutorials and sample experiments facilitates a fast learning curve for users of various backgrounds, while modular, extensible architecture allows for its seamless development.

Notably, *MaaSSim* is not intended for the complete modelling of transport systems for which there is an abundance of mature and developed frameworks, both commercial (like *PTV Visum*, *CUBE*, *Emme*) and open source (like *MatSim* [45], *SUMO* [46], *DynaMIT* [47], *SimMobility* [48] etc.). Instead, the explicit objective of *MaaSSim* is to support researchers

with modelling and reproducing the emerging novel phenomena taking place in the context of two-sided mobility platforms and analyse their disruptive potential for urban transport systems.

## 2 Software description

MaaSSim is an agent-based simulator reproducing the dynamics of two-sided mobility platforms in the context of urban transport networks. It models the behaviour and interactions of two kinds of agents: (i) travellers, requesting to travel from their origin to destination at a given time, and (ii) drivers supplying their travel needs by offering them rides. The interactions between the two agent types are mediated by the platform, matching demand and supply. Both supply and demand are microscopic. For supply, this pertains to the explicit representation of single vehicles and their movements in time and space (using a detailed road network graph), while for demand this pertains to the exact trip request time, origin and destination defined at the graph node level. Agents are decision-makers. Specifically, travellers may reject the incoming offer or decide to use another mode than those offered by the mobility platform altogether (opt-out). Similarly, the driver may opt-out of the system (stop providing service) or reject/accept incoming requests. Moreover, drivers may strategically re-position while being idle.

All of the above behaviours are modelled through **decision modules**, the core functionality of MaaSSim. By default agents' decisions are deterministic and ubiquitous but designed to be easily replaced with user-defined functions representing desired behaviour—presumably probabilistic, representing agents' taste variations (heterogeneity), their previous experiences (learning) and available information (system control).

MaaSSim allows to replicate simulations (to obtain meaningful distributions of random variables), explore multidimensional parameter grids in parallel (e.g. various travellers' value-of-time and fleet size combinations) or simulate day-to-day evolution until convergence (as we illustrate in the case study of the next section). Independent simulation runs may be executed in parallel, distributing computation load over multiple threads. Each simulation run (day) outputs a sequence of recorded space-time locations and statuses for simulated vehicles and travellers. These outputs are further synthesised into agent-level and system-wide KPIs for in-depth analyses. Fig 1 provides an overview of MaaSSim usage.

### 2.1 Software architecture

MaaSSim is a lightweight, modular, scalable and extensible python library. The main class is called with a configuration file (.json file) allowing to control the input (travel demand, fleet supply, road network) and simulation (e.g. simulation time or event duration and their variability). External decision functions to reproduce desired agents' behaviour are passed by reference and can be user-defined (Fig 2). A simulation corresponds to a single day, during which routines of interacting agents are processed (Fig 3) with a `Simpy` discrete-event simulation framework [49]. The simulation outputs raw .csv logs where spatio-temporal stamps of consecutive events are stored for each agent, as well as aggregated reports with key performance indicators.

**2.1.1 Input.** MaaSSim is controlled via an editable .json file which configures the simulation and specifies the input. Users may define e.g. supply and demand levels, simulation time, travel speeds and duration of respective simulation events (e.g. transaction time and its variability). A full list of parameters with default values is presented in the Table 1.

To run the MaaSSim simulation following input is required:

```

1 from MaaSSim.simulators import simulate, simulate_parallel
2 from MaaSSim.utils import get_config, load_G
3 from MaaSSim.utils import prep_supply_and_demand, collect_results
4
5 sim = simulate() # run MaaSSim simulation
6 sim.runs[0].trips # access the results
7 params = get_config('default.json') # load configuration
8 params.city = "Nootdorp, Netherlands" # modify it
9 inData = load_G(params) # load different network graph
10 params.nP = 50 # modify number of travellers
11 inData = prep_supply_and_demand(inData, params) # regenerate supply and demand
12 sim2 = simulate(inData, params) # rerun the simulation with new data and parameters
13 print('Simulated wait times: {}s and {}s.'.format(sim.res[0].pax_exp['WAIT'].sum(),
14         sim2.res[0].pax_exp['WAIT'].sum())) # compare some results
15
16 space = {nP=[5,10,20], nV = [5,10]} # define the search space to explore in experiments
17 simulate_parallel(inData, params, search_space = space) # run parallel experiments
18 res = collect_results(params.paths.dumps) # collect results from so mparallel experiments
19
20 def my_function(**kwargs): # user defined function to represent agent decisions
21     veh = kwargs.get('veh', None) # input
22     sim = veh.sim # access to the simulation object
23     if len(sim.runs)==0 or sim.res[last_run].veh_exp.loc[veh.id].nRIDES > 3:
24         return False # if I had more than 3 rides yesterday I stay
25     else:
26         return True # otherwise I leave
27
28 sim = simulate(inData,params, f_driver_out = my_function) # run MaaSSim with user-defined function

```

Fig 1. MaaSSim usage at glance: Starting from a single simulation, through modifying input and configuration, up to parallel experiment computation and user-defined decision function.

<https://doi.org/10.1371/journal.pone.0269682.g001>

- an urban road network graph (an instance of `networkX DiGraph` imported for the simulated urban area with `OSMnx` [50]),
- travel demand (`pandas DataFrame` with the origin and destination nodes and departure time for each trip request) and
- supply specifications (drivers with their initial locations).

Both supply and demand may come from external data sources or be generated using internal MaaSSim procedures (e.g. to create synthetic demand patterns following predefined

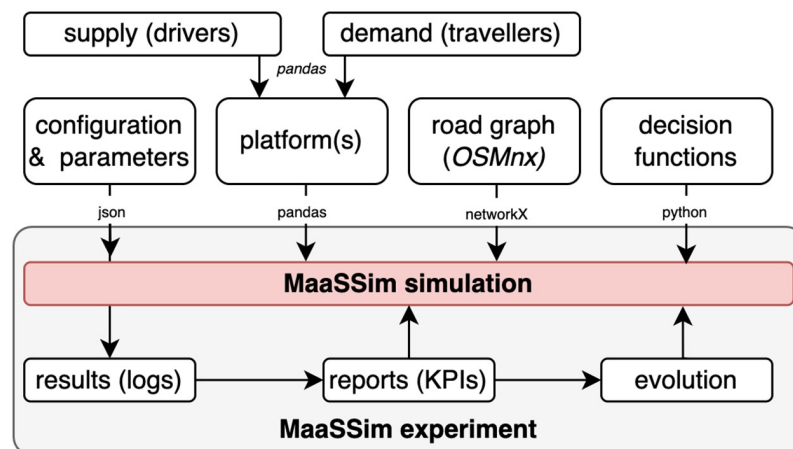
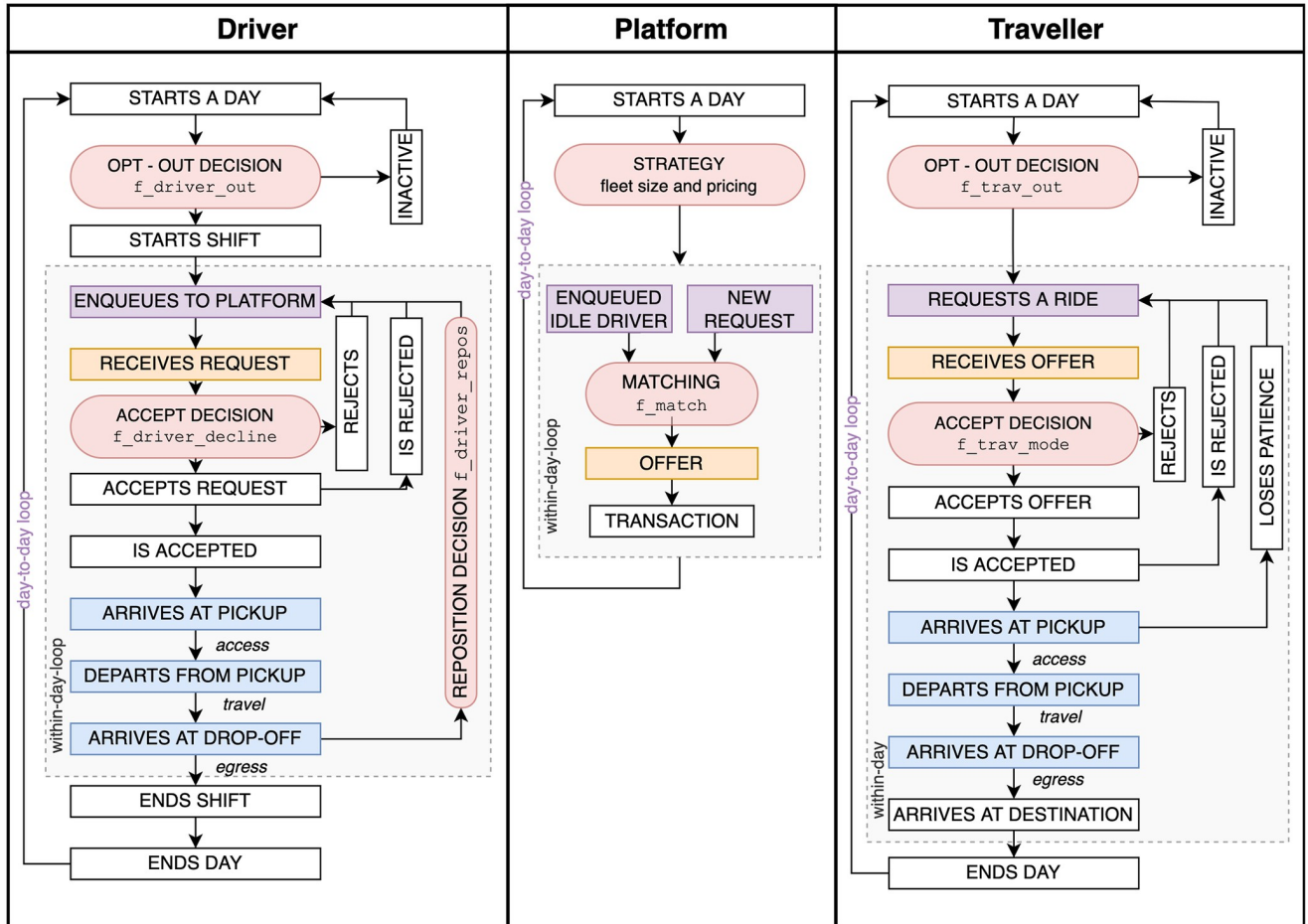


Fig 2. Input and output of MaaSSim workflow.

<https://doi.org/10.1371/journal.pone.0269682.g002>





**Fig 3. Routines of the three kinds of agents in MaaSSim.** Boxes in violet denote an interaction with the platform, orange displays matching between drivers and travellers, and blue refers to the joint part where a traveller is transported by the driver. Places where agents make a decision are marked with red rounded boxes and their decision protocols can be replaced by user-defined python functions.

<https://doi.org/10.1371/journal.pone.0269682.g003>

spatial distributions). Each input DataFrame may store additional information in separate columns, interpreted within user-defined modules (e.g. traveller-specific value-of-time or drivers' working shifts).

**2.1.2 Agents.** Three kinds of MaaSSim agents (implemented as SimPy processes) interact with each other during the course of their daily routines (Fig 3):

- **Travellers** may be assigned to multiple platforms and submit a request to them to choose the best offer amongst those. A traveller unsatisfied with previous experience may opt-out before requesting. 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 the traveller's daily routine.
- **Drivers** operate in a loop, enqueueing to the platform and serving matched requests until the end of their shift (Fig 3). Drivers may decide to opt-out before starting a shift and not enter the platform at all. When they start their shift, they accept or reject the incoming requests.

Table 1. Default parameterization of MaaSSim and parameters' description.

parameter	name	type	default	description
general	city	String	'Nootdorp, Netherlands'	query for Open Street Map to download a new graph
	nP	Int	20	demand level (number of trip requests to be generated)
	nV	Int	5	supply level (number of vehicles to be generated)
	nD	Int	1	number of days to be simulated
	t0	String	'17:00'	simulation start (string interpretable as <code>pd.Timestamp</code> )
	simTime	Int	1	hours of simulation
times			-	duration of respective events
	request	Int	15	making a request via app
	transaction	Int	20	accepting the request and match
	pickup	Int	30	entering the vehicles
	dropoff	Int	10	leaving the vehicles
	patience	Int	600	maximal time to wait for match before leaving the system
demand density		Float		spatial distribution
	origins		-0.0003	of origins
	destinations		-0.001	of destinations
speeds		Float		mean speeds [m/s]
	walk		1.2	of pedestrians
	ride		10	of vehicles
paths		urls or paths		input files
	G	.graphml	'Nootdorp.graphml'	osmnx graph
	skim	.csv	'Nootdorp.csv'	node x node distance matrix
parallel				control parallel simulations
	nThread	Int	1	number of threads
	nReplications	Int	1	number of replications

<https://doi.org/10.1371/journal.pone.0269682.t001>

When accepting, they serve the matched request: arrive at the pick-up point, wait for the traveller, travel to their drop-off point and may reposition after becoming idle.

- **Platforms** operate in an infinite loop during simulation. Every day platform may update its control variables e.g. fare for single and shared rides or a commission fee collected from the drivers. Whenever a trip is requested or a driver becomes idle (starts a shift or completes a previous request), the platform matches a two-sided queue of travellers on one side and drivers on another. By default, incoming travel requests are matched with the nearest idle vehicle. Such a generic algorithm can be easily replaced with a user-defined function (e.g. batching requests—as illustrated with reproducible experiments on the public repository).

**2.1.3 Decision modules.** The central functionality of MaaSSim lies in representing agents' individual decision processes (marked with round boxes in Fig 3). To this end, we introduced an interface where default functions may be overwritten with user-defined modules and integrated within MaaSSim simulations. User-defined functions receive as input: the main simulation MaaSSim object and a decision-maker (traveller or a driver along with their individual attributes) which can be used to reproduce the desired behaviour of:

- drivers:
  - leaving the system (`f_driver_out`),
  - accepting requests (`f_driver_decline`) and



- re-positioning (`f_driver_repos`),
- travellers
  - leaving the system (`f_trav_out`) and
  - selecting among platforms and modes (`f_trav_mode`),
- platform
  - updating control variables (fares and commission)
  - matching requests to drivers (`f_match`)

**2.1.4 Computation times.** MaaSSim has been developed to facilitate the research and assessment of system operations rather than real-time applications. The focus of its development has therefore been on code clarity and its accessibility for a broad community. Notwithstanding, it remains efficient and allows running real-size computations within a reasonable time. For instance, simulating 1000 travellers and 50 drivers over 4 hours for the city of Delft, the Netherlands takes ca 70s. It requires 28 minutes to simulate the Amsterdam network for 8 hours with 5 000 travellers and 200 drivers, whereas simulating 50 travellers and 5 drivers for the small network of Nootdorp takes less than 2 seconds on MacBookPro 2019. Parallel computations on multiple threads allowed to run 20 000 simulations of an experiment from Fig 5 in less than two hours.

The complexity grows with the number of travellers, drivers and platforms (while each agent adds a new routine, the number of possible interactions between travellers and drivers within the platform follows a quadratic pattern). Surprisingly, network size does not affect the computations, as long as the pre-computed distance matrix fits into memory. The computation times may of course increase significantly if complex decision modules are introduced (and executed along with each agent's routine).

### 3 Results

We present MaaSSim modelling capabilities through a series of illustrative experiments where we simulate various system settings under a range of configurations, followed by an extensive experimental scheme of more than 1000 simulations of 200 days of supply and demand evolution.

#### 3.1 Illustrative examples

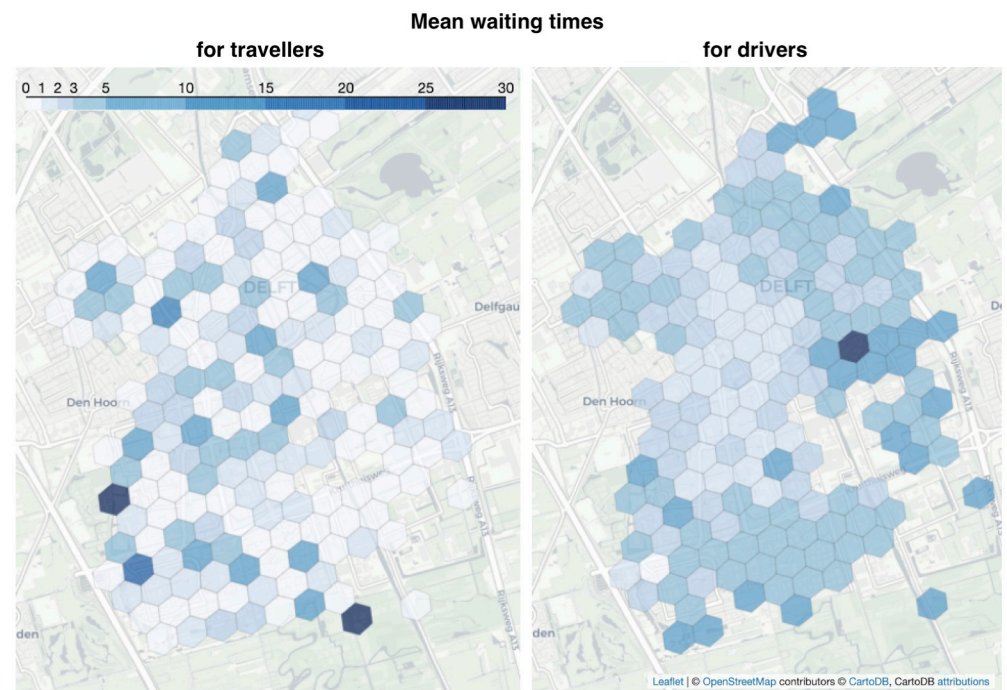
The following examples are stored on public repository (<https://github.com/RafalKucharskiPK/MaaSSim/tree/master/docs/Experiments>) to be reproduced and adapted. Below we use the detailed OSMnx network of Delft, the Netherlands (city of ca. 100 000 inhabitants) downloaded from OpenStreetMap with [50]. We set the speed to 36 kilometres per hour across the network. Likewise, the duration of events (transaction time, pick-up time, etc.) and travel speeds are deterministic and fixed. Each simulation day starts with drivers randomly positioned at network nodes. Trip requests connect origins with destinations, both assigned to the randomly selected network nodes. Spatial distributions are set to reproduce a pattern typical to the morning commute, i.e. origins are dispersed, while destinations are concentrated around the centre. Demand is uniformly distributed throughout the analysis period. The ride-hailing platform matches incoming requests with the nearest vehicles. Unless otherwise stated all decisions are deterministic and ubiquitous: travellers and drivers do not opt-out and do not

reject incoming matches. Notwithstanding, we assume that travellers leave the system if they have not been matched with a driver after waiting for 10 minutes.

**3.1.1 Spatial patterns.** We start with a single simulation, where 10 drivers serve 200 trip requests. A single platform matches travellers to their nearest idle drivers. They meet at the pick-up point and travel together to the destination. We replicate non-deterministic trip requests generation to obtain meaningful results. We report MaaSsim results in terms of waiting times—a key performance indicator for both supply and demand—in the respective parts of the city (hexagons). The obtained spatial patterns reveal conflicting trends (Fig 4). While traveller waiting times (left) are low in the Eastern parts and high in the Western parts of Delft, for drivers the opposite trend prevails (right). This reveals an interesting interaction between agents and potentially conflicting interests in the two-sided mobility market, similar to the ones observed empirically e.g. in Philadelphia [51] or in Beijing [11].

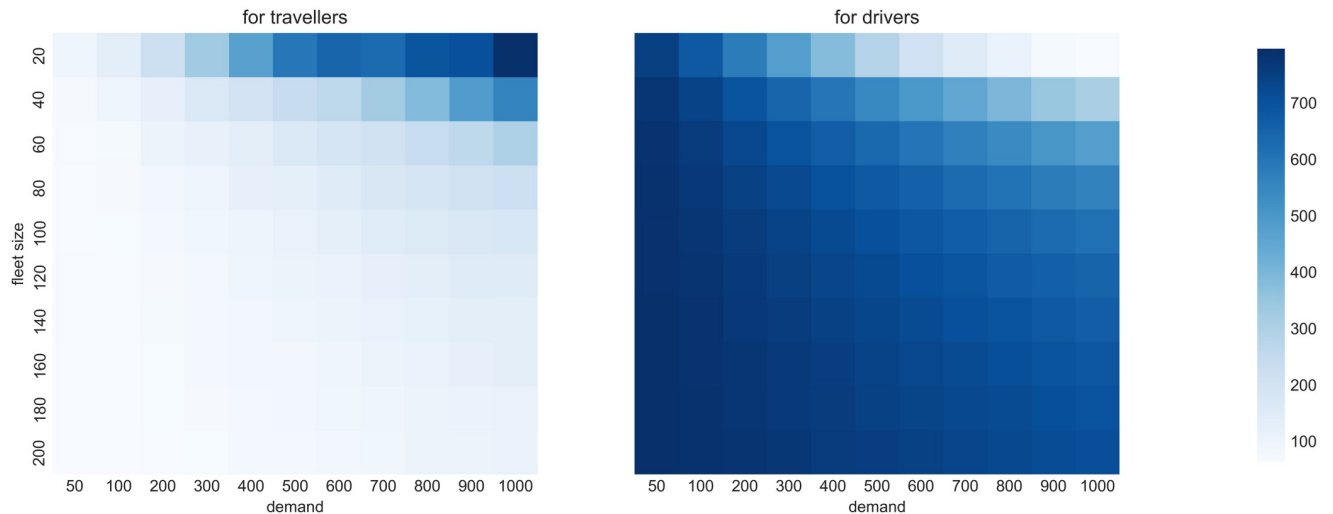
**3.1.2 Supply and demand interactions.** Next, we examine the supply and demand interactions in various settings. We explore the scenario grid varying from low to high demand (trip requests per hour) and supply levels (vehicles serving requests). The waiting time for travellers (Fig 5 left) is low when there are few travellers and many drivers. Conversely, drivers' idle times decrease if the fleet size is low (Fig 5 right). While this overall trend is expected (and supports empirical findings e.g. from [52] and [53]), the magnitude and the sensitivity of these relations and their potential to result in feedback loops on both the demand and supply sides of the two-sided market would not have been possible without detailed modelling with MaaSsim.

**3.1.3 Platform competition.** Such interaction between the supply and demand typically leads to questions about the strategic behaviour, reinforced learning and system equilibria, like



**Fig 4. Average waiting time for travellers until the driver arrives (left) and for the driver, until they get requested (right).** Dark denotes longer waiting times. Results from 20 replications of a four hour period simulation with 200 travellers and 10 vehicles in Delft, the Netherlands ©OpenStreetMap contributors.

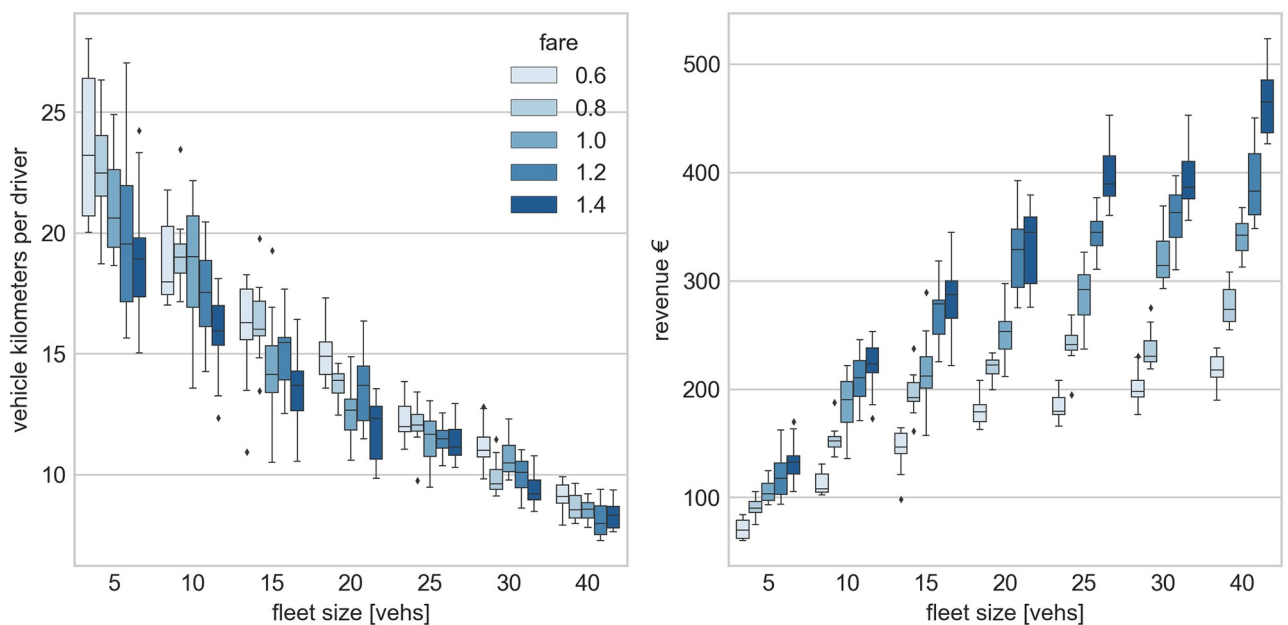
<https://doi.org/10.1371/journal.pone.0269682.g004>



**Fig 5. Service performance for various demand and supply levels.** Average waiting times for travellers (left) and drivers (right), which follow mirroring diagonal trends.

<https://doi.org/10.1371/journal.pone.0269682.g005>

in the empirically observed platform competition between Uber and Lyft in the United States [54]. We illustrate how MaaSSim supports answering those questions by means of a platform competition experiment. We consider a system where an existing platform with 20 drivers offers its travellers a trip fare of 1.0 units/km. We explore potential strategies for a new platform entering the market in terms of two key variables: fleet size (varying from 5 to 40 drivers) and a fare (varying from 0.6 to 1.4 units/km). Fig 6 shows the mileage per driver (left) and the



**Fig 6. Searching for optimal platform competition strategy: A platform enters a market with a competitor operating a fleet of 20 vehicles and offering a trip fare of 1.0 unit/km.** We report average vehicle kilometres per driver (left) and total platform revenues (right) resulting from varying fleet size (x-axis) and fare (per-kilometre) combinations for 20 replications.

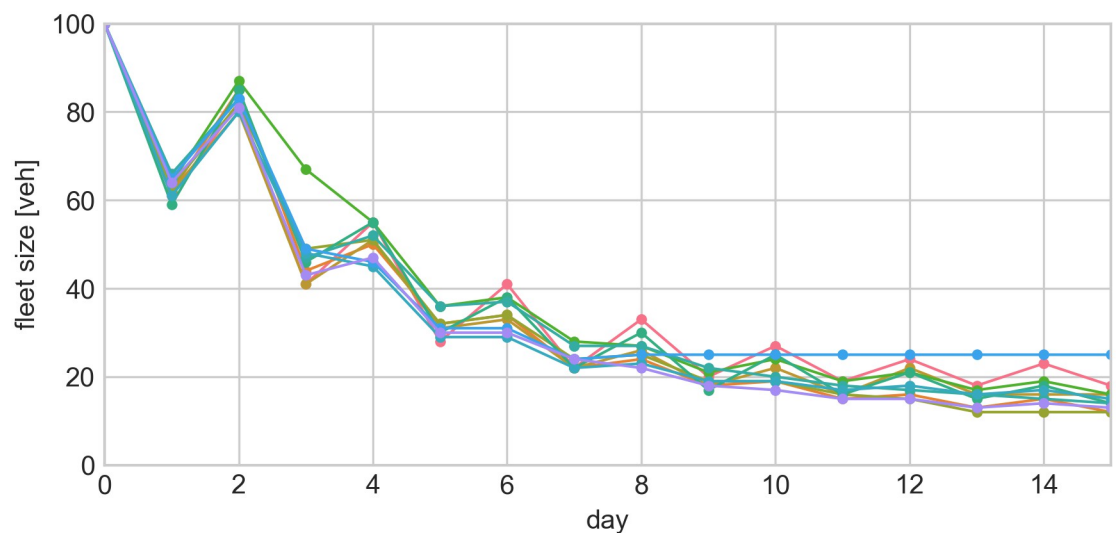
<https://doi.org/10.1371/journal.pone.0269682.g006>

total platform revenues (right) resulting from the combination of various strategies. The box plots denote means and distributions resulting from 20 replications.

Since in the above experiments the simulations are independent (unlike the reinforced learning, where agent decisions are dependent on their experience learned from previous simulation runs) we use parallel computations on multiple threads and collect results in a single file for analysis (which can be replicated with publicly available code).

**3.1.4 Drivers learning.** We illustrate the strategic learning of agents with a scenario where 100 drivers serve 200 travellers in a sequence of day-to-day simulations (Fig 7). Apparently, the initial supply level is too high, resulting in short waiting times for travellers and low revenues for drivers. Unsatisfied drivers will opt out due to bad previous experiences (low income), adjusting the pool of drivers which decreases until, eventually, some drivers decide to return to the system (as they observe high revenues in the adjusted system). This decision process is modelled via a user-defined python function (see example in Fig 1) which can introduce any generic formulas (e.g. discrete-choice model), parameters (also from `.json` configuration file) and agent-specific attributes (e.g. individual value of time). Such behavioural adaptation leads to system stabilisation, which may vary due to non-deterministic simulation settings (demand distributions and initial vehicle positions). A similar process was reported e.g. by [55] where drivers successfully adapted to maximise their revenues.

**3.1.5 Ride-pooling.** Finally, we demonstrate how ride-pooling is embedded with simulations in Fig 8, where we show a trajectory of a vehicle serving non-shared, private rides (left) and pooled-rides (right). Shared rides are here pre-computed with external the ExMAS [27] algorithm. ExMAS is fully integrated in MaaSSim so that ride-pooling can be easily reproduced, as demonstrated in the online tutorial: [https://github.com/RafalKucharskiPK/MaaSSim/blob/master/docs/tutorials/07\\_Shared\\_rides\\_with\\_ExMAS.ipynb](https://github.com/RafalKucharskiPK/MaaSSim/blob/master/docs/tutorials/07_Shared_rides_with_ExMAS.ipynb), relying on behavioural and system parameters to optimally match travellers into attractive pooled rides. In this study, we use a 20% discount for a shared ride and 1.2 penalty for the shared in-vehicle time (so-called willingness to share multiplier), further detailed in [27]). With such a setting, one can reproduce the impact of ride-pooling on system efficiency, revealed e.g. in Chicago [56].



**Fig 7. Fleet size evolution for 10 non-deterministic replications of driver learning behaviour.** Drivers make daily decisions to opt-out or stay in the system based on previous experience and expected outcomes.

<https://doi.org/10.1371/journal.pone.0269682.g007>



**Fig 8. Traces of rides for a single simulated vehicle without (left) and with pooled ride services (right).** Blue—single traveller on-board, Green—several travellers sharing a ride, Black—empty vehicle trip.

<https://doi.org/10.1371/journal.pone.0269682.g008>

### 3.2 Case study

To demonstrate the full complexity of two-sided market dynamics and MaaSsim capabilities to reproduce it, we report here a comprehensive simulation experiment. We simulate demand and supply evolution governed by the platform decisions in the city of Delft, the Netherlands. Drivers start each day at their fixed locations and travellers enter the system with the same travel demand (origin, destination and departure time)—as illustrated with Fig 9. Travellers are matched by the platform to the nearest drivers and together they traverse a detailed road network graph.

Every day travellers make travel choices. Based on their past experience they select between three alternatives: private ride-hailing (*rh*), ride-pooling (*rp*) or other alternative modes (e.g. public transport, car or bike). Which depends on their expected quality of service, i.e. waiting time, travel time and trip fares. Similarly, the drivers every day decide whether to participate in the system or not, relying on their past experiences. They compare their expected wages with the so-called reservation wage and make rational decisions (analogously to [57]).

Eventually, the learning process terminates as the system converges to a stable state. The evolution depends not only on the supply and demand levels but is also controlled by the platform strategy. The platform controls the behaviour of drivers by determining the share of collected fare that stays with a driver  $c$  and influences the behaviour of travellers by setting the discount rate for ride-pooling  $p$  (as an incentive to induce pooling)—both are treated in the experiments as exogenous variables, directly controlled by the platform.

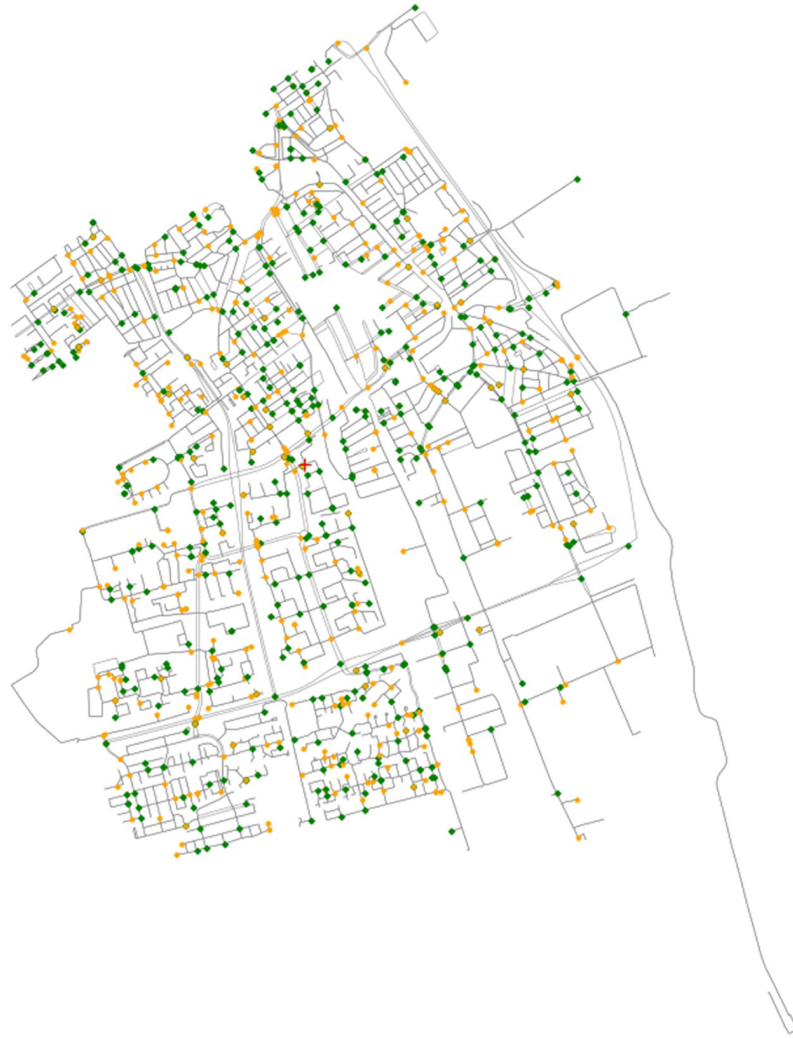
The day-to-day decisions are modelled using microscopic probabilistic discrete choice models, in which agents (travellers/drivers) estimate their utility associated with each alternative and make subjectively optimal decisions. These decisions are implemented as user-defined python functions passed to MaaSsim at the initialisation and controlled via a `.json` configuration file.

Drivers update their expected profit (wage  $w$  of driver  $d$  on the day  $i$ ) based on their experiences:

$$\bar{w}_{i,d} = f(\bar{w}_{i-1,d}, w_{i-1,d}) \quad (1)$$

Their expectation ( $\bar{w}$ ) is updated based on the actual experiences ( $w$ ) from previous days and used to make subsequent travel decisions (similarly to [55]). The probability of participating





**Fig 9. Road network of Delft with origins marked in green, destinations in orange.** ©OpenStreetMap contributors.

<https://doi.org/10.1371/journal.pone.0269682.g009>

(driving) in the system is then expressed as:

$$P_{i,d} = \frac{e^{\beta_d \bar{w}_{i,d}}}{e^{\beta_d \bar{w}_{i,d}} + e^{\beta_d W_i}}, \quad (2)$$

where the expected wage is compared with a so-called reservation wage  $W_i$  in a binary probabilistic choice model with a sensitivity parameter  $\beta_d$  (set to 0.1 in the experiment). The reservation wage  $W_i$  is user-dependent and drawn for each driver from a normal distribution with a mean of 10€ per simulation hour and a standard deviation of 2.5€. We store it as the extra column in the input DataFrame and use in further calculations.

Travellers choose between three alternative travel modes, following a multinomial logit model:

$$P_{i,d}^m = \frac{e^{U_{i,d}^m}}{e^{U_{i,d}^h} + e^{U_{i,d}^s} + e^{U_{i,d}^o}}, \quad (3)$$



where the probability that individual  $i$  chooses mode  $m$  on day  $d$  is calculated based on the utility of this mode, relative to the utilities associated with alternative modes (similarly to [10]). Here, we consider three modes: private ride-hailing ( $rh$ ), shared ride-pooling ( $rs$ ) and other, non-platform based modes (bike, public transport and car). Mode utilities are based on travellers' preferences (expressed with  $\beta$ 's in the utility formulas) and travel attributes (travel time  $t$ , cost  $c$  and waiting time  $w$  of respective mode  $m$ ):

$$U_i^m = \beta_{w,i} w_{i,m,d} + \beta_{t,i} t_{i,m,d} + \beta_{c,i} c_{i,m,d} \quad (4)$$

The attractiveness of platform-based modes is controlled by the price, set by the platform. The dynamics in the choices are induced by accumulating acquired experience which is then used to form expectations regarding travel times  $t$  and waiting times  $w$  of both private and pooled rides. The waiting times depend mainly on the supply (available vehicles) which, in turn, is controlled by the platform through the commission fee (a low commission fee will attract more drivers). Even though travel times are assumed fixed (fixed network-wide speed) in these series of experiments, the detour induced due to ride-pooling may vary and is unknown to the travellers. We specify the behavioural parameters ( $\beta$ 's) based on a recent stated-preference study [10], tuned to induce greater market shares (assuming critical mass for efficient platform-based operations is already reached) and greater sensitivity to waiting time. Under this setting, the share of platform-based modes varies from ca. 50% for null waiting times, to 10% when the average waiting time equals 5 minutes.

Travellers are initialised with optimistic expectations of travel and waiting times. In the course of their learning, they gain experience and update their expectations accordingly.

Agents update their individual expectations based on their recent experience using the following formula:

$$\bar{t}_{i,d} = (1 - \omega_{i,d}) \cdot \bar{t}_{i-1,d} + \omega_{i,d} \cdot t_{i,d} \quad (5)$$

The weight  $\omega_{i,d}$  depends on the number of experiences that the traveller has acquired  $H_{i,d}$  and is bounded by a so-called look-back window ( $\omega_{max}$ —the number of days used to update the experience):

$$\omega_{i,d} = 1 / \min(H_{i,d}, \omega_{max}) \quad (6)$$

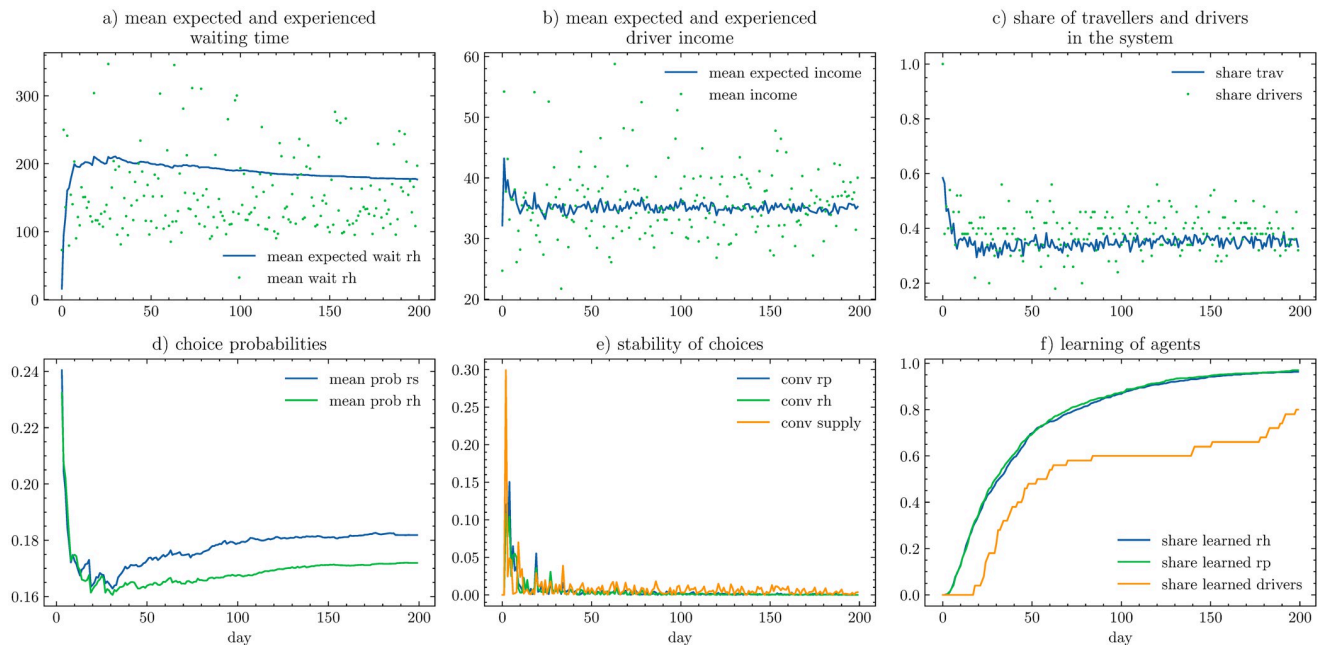
We explored the grid of the following parameters in the experiments:

- demand 300, 500, 700, 900 and 1100 trips per 4 hours of simulation.
- supply 10, 30, 50, 60 vehicles
- share of fares collected by the driver 10, 30, 50, 70, 90, 110%
- pooling discount 0, 5, 10, 20, 30, 4%

and let the system evolve until the stabilisation (when agents finish their learning process). The results are presented and discussed with the sample of the evolution process in Fig 10 and the platform profits on the search space grid in Fig 11.

## 4 Conclusion

The overarching objective underlying the development of MaaSSim is to allow researchers to focus on their partial models and integrate them within the simulation framework, allowing a group of interdisciplinary researchers to share expertise from their fields. For instance, MaaSSim has been instrumental in the scientific discovery process of supply-side dynamics and the



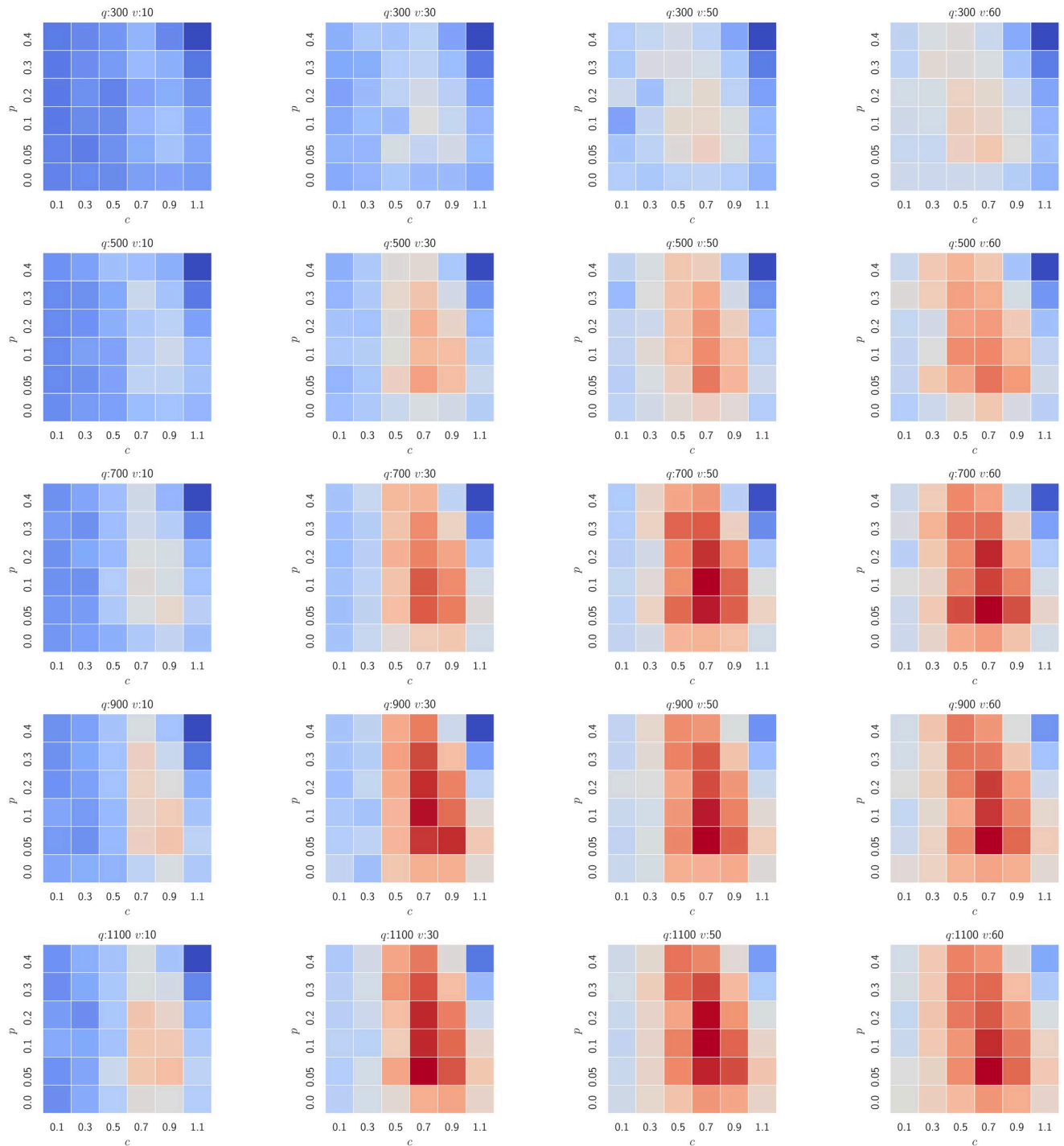
**Fig 10. 200 days of supply and demand evolution for the case study experiment with 900 travellers and 50 drivers.** Travellers expected waiting time (a) stabilises over consecutive days (blue line) despite the high day-to-day variability of actually experienced waiting times (green dots). Similarly for the drivers, whose mean expected incomes (b) remain stable around 35€, with day-to-day variability (green dots) ranging from 27 to 50€. The share of travellers in the system (c) remains stable around 37% (blue line) yet the number of drivers (green dots) may range between 10 and 30%. The initially high choice probabilities for MaaS modes (d) happened to be over-optimistic and quickly drop from 24% to 16%. Yet, as agents learn, the pooled alternative (blue—*rs*) becomes more popular than private ride (green—*rh*) stabilising around 18%. Agent day-to-day choice evolution stabilises and the variability stays within the 2% range already after ca. 40 days of evolution (e). After 100 days 90% of travellers have gained enough experience to learn the system and stabilise their choices (blue and green in the panel f). However, drivers learn slower with only 60% of them finishing the learning process after 100 days (orange on panel f).

<https://doi.org/10.1371/journal.pone.0269682.g010>

fleet size attained in equilibrium in a decentralised bottom-up mobility platform context. This research resulted in a novel service supplier (i.e. driver) learning and choice model [57] which can be now reused for studying platform pricing strategies, re-positioning algorithms or travellers' mode-choices.

MaaSIm provides an extensible, easy-to-use simulation platform allowing for user-defined representation of a two-sided mobility platform that can support a variety of research interests. MaaSIm is an open-source library available through the `pip` installer as well as from the public repository, where it comes along with a set of tutorials and applicable use-cases. With the above set of experiments, coupled with reproducible `jupyter notebooks` stored on the repository, the reader can get an impression of the range of MaaSIm applications and start developing their own experiments. In particular, explore their own networks (queried from OpenStreetMap with [50]) and run the experiments with tailored configurations. Making it capable to support researchers in exploring future research directions in the field of two-sided mobility platforms.

As always, the realism of the obtained simulation outputs strongly depends on the quality of the input parameters. MaaSIm can be parameterized to reproduce a variety of system representations, yet the empirically valid settings remain largely unknown due to the limited availability of data and evidence from the rapidly evolving ecosystem of two-sided mobility platforms. While the active stream of research contributes to a better understanding of the



**Fig 11. The total commission collected by the platform in the series of over 1000 experiments (blue cells denote deficit, red ones denote profit).** Each grid denotes a single supply-demand setting with a fixed number of trip requests ( $q$ ) and a fixed number of potential drivers ( $v$ ). Every single cell therein represents a stable state of the evolution process depicted in Fig 10. The platform explores the best setting of a ride-pooling discount ( $p$ —rows) and a share of fares remaining with a driver ( $c$ —columns) to maximise its profit. For a low demand the system is not profitable (top left), yet as the number of drivers increases, the platform starts generating profits (top right). As both supply and demand increase, the central parts of the grids become profitable, i.e. commissions between 0.5 and 0.7. Notably, a too low commission for drivers ( $<0.5$ ) is not profitable for the platform, as it leads to fewer participating drivers and thus lower revenues, and so does a too high commission for drivers (when drivers stay within the system but the platform does not generate profit). At the high supply and demand levels, the platform may start providing pooled rides as it remains profitable even for high discounts offered ( $p > 0.2$ ). Nonetheless, under this setting the platform profit is always maximised at low discounts ( $p = 0.05$ ) and introducing lower prices for pooling would result in lower profits.

<https://doi.org/10.1371/journal.pone.0269682.g011>

underlying phenomena, additional empirical analysis is needed to better underpin some of the simulator parameters.

## Acknowledgments

We thank the team of CriticalMaaS project at TU Delft, specifically the PhD students—Arjan de Ruijter, Subodh Dubey, Nejc Geržinič and Peyman Ashkrof—whose research programmes were closely tied with the MaaSsim development. This was a mutual intellectual exchange: MaaSsim was tailored to support their research programmes and their programmes were tailored to benefit from the capabilities of MaaSsim.

## Author Contributions

**Conceptualization:** Rafał Kucharski, Oded Cats.

**Data curation:** Rafał Kucharski.

**Funding acquisition:** Oded Cats.

**Methodology:** Rafał Kucharski, Oded Cats.

**Software:** Rafał Kucharski.

**Validation:** Rafał Kucharski.

**Visualization:** Rafał Kucharski.

**Writing – original draft:** Rafał Kucharski, Oded Cats.

**Writing – review & editing:** Rafał Kucharski, Oded Cats.

## References

1. Kim HJ, Suh CS. Spreading the sharing economy: Institutional conditions for the international diffusion of Uber, 2010-2017. *Plos one*. 2021; 16(3):e0248038. <https://doi.org/10.1371/journal.pone.0248038> PMID: 33690676
2. Xu K, Sun L, Liu J, Wang H. An empirical investigation of taxi driver response behavior to ride-hailing requests: A spatio-temporal perspective. *PloS one*. 2018; 13(6):e0198605. <https://doi.org/10.1371/journal.pone.0198605> PMID: 29883478
3. Hu B, Kong Y, Sun M, Dong X, Zong G. Understanding the unbalance of interest in taxi market based on drivers' service profit margins. *PloS one*. 2018; 13(6):e0198491. <https://doi.org/10.1371/journal.pone.0198491> PMID: 29912890
4. Casey TR, Töyli J. Dynamics of two-sided platform success and failure: An analysis of public wireless local area access. *Technovation*. 2012; 32(12):703–716. <https://doi.org/10.1016/j.technovation.2012.08.003>
5. Graham M, Hjorth I, Lehdonvirta V. Digital labour and development: impacts of global digital labour platforms and the gig economy on worker livelihoods. *Transfer: European Review of Labour and Research*. 2017; 23(2):135–162. <https://doi.org/10.1177/1024258916687250> PMID: 28781494
6. Cats O, Kucharski R, Danda SR, Yap M. Beyond the dichotomy: How ride-hailing competes with and complements public transport. *Plos one*. 2022; 17(1):e0262496. <https://doi.org/10.1371/journal.pone.0262496> PMID: 35030219
7. Ashkrof P, de Almeida Correia GH, Cats O, van Arem B. Understanding ride-sourcing drivers' behaviour and preferences: Insights from focus groups analysis. *Research in Transportation Business & Management*. 2020; p. 100516. <https://doi.org/10.1016/j.rtbm.2020.100516>
8. Xu Z, AMC Vignon D, Yin Y, Ye J. An empirical study of the labor supply of ride-sourcing drivers. *Transportation Letters*. 2020; p. 1–4.
9. Alonso-González MJ, Hoogendoorn-Lanser S, van Oort N, Cats O, Hoogendoorn S. Drivers and barriers in adopting Mobility as a Service (MaaS)—A latent class cluster analysis of attitudes. *Transportation Research Part A: Policy and Practice*. 2020; 132:378–401. <https://doi.org/10.1016/j.tra.2019.11.022>

10. Geržinič N, van Oort N, Hoogendoorn-Lanser S, Cats O, Hoogendoorn S. Potential of on-demand services for urban travel. *Transportation*. 2022;. <https://doi.org/10.1007/s11116-022-10278-9> PMID: 35382447
11. He Z. Portraying ride-hailing mobility using multi-day trip order data: A case study of Beijing, China. *Transportation Research Part A: Policy and Practice*. 2021; 146:152–169.
12. Xu Z, Yin Y, Ye J. On the supply curve of ride-hailing systems. *Transportation Research Part B: Methodological*. 2020; 132:29–43. <https://doi.org/10.1016/j.trb.2019.02.011>
13. Nourinejad M, Ramezani M. Ride-Sourcing modeling and pricing in non-equilibrium two-sided markets. *Transportation Research Part B: Methodological*. 2019;.
14. Cui Y, Makhija RSMS, Chen RB, He Q, Khani A, et al. Understanding and Modeling the Social Preferences for Riders in Rideshare Matching. *Transportation*. 2020; p. 1–27.
15. Zuniga-Garcia N, Tec M, Scott JG, Ruiz-Juri N, Machemehl RB. Evaluation of ride-sourcing search frictions and driver productivity: A spatial denoising approach. *Transportation Research Part C: Emerging Technologies*. 2020; 110:346–367. <https://doi.org/10.1016/j.trc.2019.11.021>
16. Sühr T, Biega AJ, Zehlike M, Gummadi KP, Chakraborty A. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; 2019. p. 3082–3092.
17. Haliem M, Mani G, Aggarwal V, Bhargava B. A distributed model-free ride-sharing algorithm with pricing using deep reinforcement learning. In: *Computer Science in Cars Symposium*; 2020. p. 1–10. <https://doi.org/10.1145/3385958.3430484>
18. Welch TF, Gehrke SR, Widita A. Shared-use mobility competition: a trip-level analysis of taxi, bike-share, and transit mode choice in Washington, DC. *Transportmetrica A: transport science*. 2020; 16(1):43–55. <https://doi.org/10.1080/23249935.2018.1523250>
19. Wang Y, Wang S, Wang J, Wei J, Wang C. An empirical study of consumers' intention to use ride-sharing services: using an extended technology acceptance model. *Transportation*. 2020; 47(1):397–415. <https://doi.org/10.1007/s11116-018-9893-4>
20. Alonso-González MJ, van Oort N, Cats O, Hoogendoorn-Lanser S, Hoogendoorn S. Value of time and reliability for urban pooled on-demand services. *Transportation Research Part C: Emerging Technologies*. 2020; 115:102621. <https://doi.org/10.1016/j.trc.2020.102621>
21. Bokányi E, Hannák A. Understanding inequalities in ride-hailing services through simulations. *Scientific reports*. 2020; 10(1):1–11. <https://doi.org/10.1038/s41598-020-63171-9> PMID: 32300117
22. Calderón F, Miller EJ. A conceptual framework for modeling the supply side of mobility services within large-scale agent-based travel demand models. *Transportation Letters*. 2021; p. 1–10.
23. Berger T, Chen C, Frey CB. Drivers of disruption? Estimating the Uber effect. *European Economic Review*. 2018; 110:197–210. <https://doi.org/10.1016/j.euroecorev.2018.05.006>
24. Fielbaum A, Tirachini A. The sharing economy and the job market: the case of ride-hailing drivers in Chile. *Transportation*. 2020; p. 1–27.
25. Allon G, Cohen M, Sinchaisri WP. The impact of behavioral and economic drivers on gig economy workers. Available at SSRN 3274628. 2018;.
26. Ke J, Yang H, Li X, Wang H, Ye J. Pricing and equilibrium in on-demand ride-pooling markets. *Transportation Research Part B: Methodological*. 2020; 139:411–431. <https://doi.org/10.1016/j.trb.2020.07.001>
27. Kucharski R, Cats O. Exact matching of attractive shared rides (ExMAS) for system-wide strategic evaluations. *Transportation Research Part B: Methodological*. 2020; 139:285–310. <https://doi.org/10.1016/j.trb.2020.06.006>
28. Alonso-Mora J, Samaranyake S, Wallar A, Frazzoli E, Rus D. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*. 2017; 114(3):462–467. <https://doi.org/10.1073/pnas.1611675114> PMID: 28049820
29. Bischoff J, Maciejewski M, Nagel K. City-wide shared taxis: A simulation study in Berlin. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*; 2017. p. 275–280.
30. Veve C, Chiabaut N. Estimation of the shared mobility demand based on the daily regularity of the urban mobility and the similarity of individual trips. *PLoS one*. 2020; 15(9):e0238143. <https://doi.org/10.1371/journal.pone.0238143> PMID: 32941487
31. Fagnant DJ, Kockelman KM. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation*. 2018; 45(1):143–158. <https://doi.org/10.1007/s11116-016-9729-z>
32. Engelhardt R, Dandl F, Bilali A, Bogenberger K. Quantifying the benefits of autonomous on-demand ride-pooling: A simulation study for Munich, Germany. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE; 2019. p. 2992–2997.



33. Hörl S, Ruch C, Becker F, Frazzoli E, Axhausen KW. Fleet operational policies for automated mobility: A simulation assessment for Zurich. *Transportation Research Part C: Emerging Technologies*. 2019; 102:20–31. <https://doi.org/10.1016/j.trc.2019.02.020>
34. Vazifeh MM, Santi P, Resta G, Strogatz SH, Ratti C. Addressing the minimum fleet problem in on-demand urban mobility. *Nature*. 2018; 557(7706):534–538. <https://doi.org/10.1038/s41586-018-0095-1> PMID: 29795256
35. Holler J, Vuorio R, Qin Z, Tang X, Jiao Y, Jin T, et al. Deep Reinforcement Learning for Multi-driver Vehicle Dispatching and Repositioning Problem. In: 2019 IEEE International Conference on Data Mining (ICDM); 2019. p. 1090–1095.
36. Afeche P, Liu Z, Maglaras C. Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. *Columbia Business School Research Paper*. 2018;(18-19):18–19.
37. Maciejewski M, Bischoff J. Large-scale microscopic simulation of taxi services. *Procedia Computer Science*. 2015; 52:358–364. <https://doi.org/10.1016/j.procs.2015.05.107>
38. Martinez LM, Correia GH, Viegas JM. An agent-based simulation model to assess the impacts of introducing a shared-taxi system: an application to Lisbon (Portugal). *Journal of Advanced Transportation*. 2015; 49(3):475–495. <https://doi.org/10.1002/atr.1283>
39. Agatz N, Erera AL, Savelsbergh MWP, Wang X. Dynamic Ride-Sharing: a Simulation Study in Metro Atlanta. *Procedia—Social and Behavioral Sciences*. 2011; 17:532–550. <https://doi.org/10.1016/j.sbspro.2011.04.530>
40. Ota M, Vo H, Silva C, Freire J. Stars: Simulating taxi ride sharing at scale. *IEEE Transactions on Big Data*. 2016; 3(3):349–361. <https://doi.org/10.1109/TBDATA.2016.2627223>
41. Nourinejad M, Roorda MJ. Agent based model for dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*. 2016; 64:117–132. <https://doi.org/10.1016/j.trc.2015.07.016>
42. Djavadian S, Chow JY. An agent-based day-to-day adjustment process for modeling ‘Mobility as a Service’ with a two-sided flexible transport market. *Transportation research part B: methodological*. 2017; 104:36–57. <https://doi.org/10.1016/j.trb.2017.06.015>
43. Nahmias-Biran Bh, Oke JB, Kumar N, Basak K, Araldo A, Seshadri R, et al. From traditional to automated mobility on demand: a comprehensive framework for modeling on-demand services in SimMobility. *Transportation Research Record*. 2019; 2673(12):15–29. <https://doi.org/10.1177/0361198119853553>
44. Ruch C, Hörl S, Frazzoli E. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2018. p. 3639–3644.
45. Axhausen W K, Horni A, Nagel K. *The multi-agent transport simulation MATSim*. Ubiquity Press; 2016.
46. Lopez PA, Behrisch M, Bieker-Walz L, Erdmann J, Flötteröd YP, Hilbrich R, et al. Microscopic Traffic Simulation using SUMO. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE; 2018. Available from: <https://elib.dlr.de/124092/>.
47. Ben-Akiva M, Bierlaire M, Koutsopoulos H, Mishalani R. DynaMIT: a simulation-based system for traffic prediction. In: *DACCORD Short Term Forecasting Workshop*. Delft The Netherlands; 1998. p. 1–12.
48. Adnan M, Pereira FC, Azevedo CML, Basak K, Lovric M, Raveau S, et al. Simmobility: A multi-scale integrated agent-based simulation platform. In: *95th Annual Meeting of the Transportation Research Board* Forthcoming in *Transportation Research Record*; 2016.
49. Team SimPy. *SimPy—discrete event simulation for Python*; 2021. Available from: <https://pypi.org/project/simpy/>.
50. Boeing G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*. 2017; 65:126–139. <https://doi.org/10.1016/j.compenvurbsys.2017.05.004>
51. Shokoohyar S, Sobhani A, Ramezanpour Nargesi SR. On the determinants of Uber accessibility and its spatial distribution: Evidence from Uber in Philadelphia. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*. 2020; 10(4):e1362.
52. Chen L, Mislove A, Wilson C. Peeking beneath the hood of uber. In: *Proceedings of the 2015 internet measurement conference*; 2015. p. 495–508.
53. Widita AA, Diwangkari A. How disruptive is a disruption? The association between TNCs and vehicle ownership in urbanizing Indonesia. *Case Studies on Transport Policy*. 2022; <https://doi.org/10.1016/j.cstp.2022.01.017>
54. Jiang S, Chen L, Mislove A, Wilson C. On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi. In: *Proceedings of the 2018 World Wide Web Conference*; 2018. p. 863–872.



55. Djavadian S, Chow JY. Agent-based day-to-day adjustment process to evaluate dynamic flexible transport service policies. *Transportmetrica B: Transport Dynamics*. 2017; 5(3):281–306.
56. Taiebat M, Amini E, Xu M. Sharing behavior in ride-hailing trips: A machine learning inference approach. *Transportation Research Part D: Transport and Environment*. 2022; 103:103166. <https://doi.org/10.1016/j.trd.2021.103166>
57. de Ruijter A, Cats O, Kucharski R, van Lint H. Evolution of labour supply in ridesourcing. *Transportmetrica B: Transport Dynamics*. 2022; p. 1–28.