Dynamic Demand Simulation for Automated Mobility on Demand

Doctoral Thesis

Author(s):
Hörl, Sebastian

Publication date:
2020-05

Permanent link:
https://doi.org/10.3929/ethz-b-000419837

Rights / license:
In Copyright - Non-Commercial Use Permitted
DYNAMIC DEMAND SIMULATION FOR AUTOMATED MOBILITY ON DEMAND

A dissertation submitted to attain the degree of
DOCTOR OF SCIENCES OF ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
SEBASTIAN HÖRL
MSc. Complex Adaptive Systems, Chalmers University of Technology
born on 12 July 1990
citizen of Germany

accepted on the recommendation of
Prof. Dr. Kay W. Axhausen, examiner
Prof. Dr. Klaus Bogenberger, co-examiner
Prof. Dr. Hani S. Mahmassani, co-examiner
Prof. Dr. Kai Nagel, co-examiner

2020
ABSTRACT

Automated Mobility on Demand (AMoD) describes a concept in which automated (self-driving) vehicles are used to transport passengers just like taxis today. As the vehicles are not human-driven, service cost are expected to decrease radically. At the same time, the removal of the ownership burden renders such services highly attractive. However, adverse effects such as an increase of road use due to empty movements of the vehicles or a general rise in travel (induced demand) are debated actively in the literature. While previous analyses and simulations focus on the control and coordination of robotic taxis, only few research exist that takes into account the attractivity of the service towards the customer.

This work aims at closing this research gap by making use of three components: a representative stated-choice survey yielding the mode choice trade-offs that are important between conventional and automated modes of transport; a detailed calculator for the cost of mobility services; and an agent-based transport simulation with dynamic demand. In such simulations, travellers choose their modes of transport based on, for instance, travel time, wait time and cost of the service. However, waiting times and cost depend recursively on the attracted demand.

The focus of the thesis is the development of the methodology and the necessary tools. Their application to the use case of Zurich, Switzerland, leads to new insights on AMoD systems when customer perception is considered. In particular, a demand curve can be obtained, which shows that very small or very large fleet sizes are not attractive to the customers. In the first case, waiting times explode, while in the latter one, the service becomes too expensive. The simulation study yields 7,000 vehicles as the fleet size that can attract the largest demand. With the fleet control policy chosen, a substantial increase in VKT can be observed, and a consistently higher level of moving vehicles throughout the day. The simulations indicate that care must be taken when setting up policies for future automated vehicle services.

In the last part of the thesis, similar simulations are presented for Paris to demonstrate the transferability of the approach.
Automated Mobility on Demand (AMoD) beschreibt ein Konzept, bei dem automatisierte (selbstfahrende) Fahrzeuge zur Personenbeförderung wie die heutigen Taxis eingesetzt werden. Da die Fahrzeuge nicht von Menschen gesteuert werden, ist zu erwarten, dass die Servicekosten radikal sinken. Gleichzeitig sind solche Dienste sehr attraktiv, da der Zwang ein Fahrzeug zu besitzen entfällt. Negative Auswirkungen wie eine Zunahme der Strassennutzung aufgrund von Leerfahrten der Fahrzeuge oder eine allgemeine Zunahme der Fahrten (induzierte Nachfrage) werden jedoch in der Literatur aktiv diskutiert. Während sich bisherige Analysen und Simulationen auf die Steuerung und Koordination von automatisierten Taxis konzentrieren, gibt es nur wenige Untersuchungen, die die Attraktivität des Dienstes für den Kunden berücksichtigen.

Die vorliegende Arbeit hat das Ziel, diese Forschungslücke zu schliessen, indem drei Komponenten genutzt werden: eine repräsentative Umfrage, die die wichtigen Tradeoffs bei der Verkehrsmittelwahl zwischen konventionellen und automatisierten Verkehrsmitteln aufzeigt; ein detaillierter Rechner für die Kosten von Mobilitätsdienstleistungen; und eine agentenbasierte Verkehrssimulation. In dieser Simulation wählen die Reisenden ihre Verkehrsmittel beispielsweise auf der Grundlage der Reisezeit, der Wartezeit und der Kosten aus. Gleichsam hängen aber die Wartezeiten und Kosten rekursiv von der angezogenen Nachfrage ab.

Der Schwerpunkt der Arbeit liegt in der Entwicklung der Methodik und der notwendigen Tools. Die Simulationen für den Anwendungsfall Zürich führen zu neuen Erkenntnissen über AMoD-Systeme, bei welchen die Kundenwahrnehmung berücksichtigt wird. Insbesondere kann eine Nachfragekurve erstellt werden, die zeigt, dass sehr kleine oder sehr grosse Flottengrössen für die Kunden nicht attraktiv sind. Im ersten Fall explodieren die Wartezeiten, während im zweiten Fall der Service zu teuer wird. Die Simulationsstudie ergibt 7.000 Fahrzeuge als die Flottengrösse, die die grössste Nachfrage anziehen kann. Mit dem gewählten Ansatz der Flottenssteuerung ist eine erhebliche Zunahme der Fahrdistanz und ein konstant höheres Niveau an aktiven Fahrzeugen über den Tag hinweg zu beobachten. Die Simulationen zeigen dass bei der Entwicklung von Richtlinien für künftige automatisierte Fahrzeugdienste Vorsicht geboten ist.
ACKNOWLEDGEMENTS

Little did I know what an amazing time the following years were to become when I first climbed Hönggerberg. They have passed in the blink of an eye - because of all the extraordinary and wonderful people I have met. This is the place to acknowledge them.

To begin, I want to thank Prof. Axhausen, who gave me the freedom and support to test good (and bad) ideas and to set up my own research agenda. Thanks for letting me take part in the IVT adventure and allowing me to connect with researchers from all over the world.

Of course, the past years would not have been such a fantastic memory without my fellow PhD students, many of who have transformed from strangers to colleagues to good friends. Thank you all for the productive discussions, coffee breaks and “enlightening” evenings. In particular, I want to mention Felix, who introduced me to the black magic of choice modelling, and Chris, who one day will write a research plan. Without you I would never have spent every dinner in the campus canteen - but also would never have had so much fun!

A big thanks goes to Milos, who never got tired discussing ideas with me; some of the better ones are documented below. Together, we not only went through the ups and downs of MATSim research but also travelled the world. Without you (and your bonus points) those trips to Washington, Porto, Athens, Sydney, Auckland, ... would not have been the same. I hope that many mighty years of working together are yet to come!

About a year after the start of my PhD, I met an ambitious researcher who was brave enough to build on my early work with MATSim. Since then, a large variety of joint research has emerged, much of which is referred to in this thesis. Thank you, Claudio, for the most brilliant remarks and discussions. Good luck on your next adventure!

To finish, I want to thank my parents Yvonne and Mario, my sister Lena and the rest of my family who do not see me as often as they surely deserve. Thank you for always supporting me on my never-ending journey far from home. Ihr seid die besten! Finally, thank you Maéva, for supporting me every day, for tolerating my hours of “just one more line of code”, and for lovingly reminding me from time to time that a coffee and a walk outside can be equally enjoyable. Je t’aime!
# CONTENTS

## 1 INTRODUCTION

1.1 Automated Mobility on Demand ........................................ 1
1.2 Operational challenges .................................................. 3
1.3 Automated mobility in Switzerland .................................... 4
1.4 Challenges and research goals .......................................... 6

## 2 DYNAMIC DEMAND SIMULATION

2.1 The MATSim Framework .................................................. 9
   2.1.1 Basic concepts .................................................... 10
   2.1.2 Decision-making in MATSim .................................... 13
2.2 Discrete Choice Modeling ................................................ 22
   2.2.1 Basic theory ...................................................... 22
   2.2.2 Econometric value ............................................... 25
   2.2.3 Choice sampling .................................................. 26
2.3 Discrete Mode Choice for MATSim ..................................... 28
   2.3.1 Model structure .................................................. 30
   2.3.2 Model compatibility .............................................. 36
   2.3.3 Further use cases ............................................... 38
2.4 Discussion and lessons learned ....................................... 38
   2.4.1 Approaches in comparison ....................................... 38
   2.4.2 Fixed error terms ................................................ 40

## 3 AUTOMATED MOBILITY SIMULATION

3.1 Automated vehicles in MATSim ......................................... 46
   3.1.1 Compatibility with discrete choice .............................. 49
   3.1.2 Further components .............................................. 51
3.2 The AMoDeus framework .................................................. 52
   3.2.1 Aspects of fleet control ......................................... 53
   3.2.2 Use cases ........................................................ 55

## 4 ZURICH MODEL

4.1 Scenario synthesis ...................................................... 58
   4.1.1 Available data ................................................... 59
   4.1.2 Algorithms ........................................................ 61
4.2 Cutting process .......................................................... 67
   4.2.1 Population ......................................................... 69
   4.2.2 Public transport and road network ............................. 71
   4.2.3 Households and Facilities ..................................... 73
4.3 Baseline model ................................................. 75
  4.3.1 Custom components ....................................... 75
  4.3.2 Travel behaviour .......................................... 76
  4.3.3 Validation ..................................................... 81
4.4 AMoD Model ..................................................... 84
5 RESULTS ............................................................... 89
  5.1 Fleet sizing ....................................................... 91
  5.2 Scenario analysis ............................................... 92
  5.3 Further context ................................................ 98
    5.3.1 Dispatching algorithms .................................... 99
    5.3.2 Pooling potential .......................................... 102
  5.4 Discussion ..................................................... 103
6 TRANSFERABILITY ............................................... 107
  6.1 A MATSim scenario for Île-de-France ...................... 110
  6.2 Case study: Automated taxis in Paris .................... 113
7 GENERAL DISCUSSION ........................................... 119
  7.1 Applicability of methods and tools ....................... 119
  7.2 Open data and software ..................................... 123
8 CONCLUSION ....................................................... 127

BIBLIOGRAPHY ....................................................... 129
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>MATSim Zurich</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>MATSim simulation loop</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Queue and buffer</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Discrete mode choice structure</td>
<td>31</td>
</tr>
<tr>
<td>2.5</td>
<td>Trip-based vs. tour-based mode choice</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Automated taxi schedule</td>
<td>48</td>
</tr>
<tr>
<td>4.1</td>
<td>Secondary location assignment</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Scenario cutter: Trips</td>
<td>69</td>
</tr>
<tr>
<td>4.3</td>
<td>Scenario cutter: Network routes</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>Map of Zurich scenario</td>
<td>74</td>
</tr>
<tr>
<td>4.5</td>
<td>Calibration: Travel times</td>
<td>82</td>
</tr>
<tr>
<td>4.6</td>
<td>Calibration: Travel distances</td>
<td>83</td>
</tr>
<tr>
<td>4.7</td>
<td>Calibration: Travel speeds</td>
<td>84</td>
</tr>
<tr>
<td>4.8</td>
<td>Calibration: Mode share by distance</td>
<td>85</td>
</tr>
<tr>
<td>4.9</td>
<td>Calibration: Mode share by time of day</td>
<td>85</td>
</tr>
<tr>
<td>4.10</td>
<td>Classification of road types</td>
<td>86</td>
</tr>
<tr>
<td>4.11</td>
<td>AMoD simulation structure</td>
<td>88</td>
</tr>
<tr>
<td>5.1</td>
<td>AMoD fleet sizing</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>AMoD scenarios: Moving vehicles by time of day</td>
<td>96</td>
</tr>
<tr>
<td>5.3</td>
<td>AMoD scenarios: Accessibility</td>
<td>97</td>
</tr>
<tr>
<td>5.4</td>
<td>Control policies: Waiting times</td>
<td>100</td>
</tr>
<tr>
<td>5.5</td>
<td>Control policies: Empty distance</td>
<td>101</td>
</tr>
<tr>
<td>6.1</td>
<td>General structure of the scenario pipeline</td>
<td>109</td>
</tr>
<tr>
<td>6.2</td>
<td>MATSim Île-de-France</td>
<td>111</td>
</tr>
<tr>
<td>6.3</td>
<td>Paris AMoD: Operating area</td>
<td>113</td>
</tr>
<tr>
<td>6.4</td>
<td>Paris AMoD: Demand by fleet size</td>
<td>116</td>
</tr>
<tr>
<td>6.5</td>
<td>Paris AMoD: Price by fleet size</td>
<td>117</td>
</tr>
<tr>
<td>6.6</td>
<td>Paris AMoD: Waiting time by fleet size</td>
<td>118</td>
</tr>
<tr>
<td>6.7</td>
<td>Paris AMoD: Fleet activity by time of day</td>
<td>118</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>MATSim runtimes and memory</td>
<td>68</td>
</tr>
<tr>
<td>4.2</td>
<td>Scenario cutter: Population characteristics</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>Scenario cutter: Schedule characteristics</td>
<td>72</td>
</tr>
<tr>
<td>4.4</td>
<td>Scenario cutter: Network characteristics</td>
<td>72</td>
</tr>
<tr>
<td>4.5</td>
<td>Scenario cutter: Households and facilities</td>
<td>73</td>
</tr>
<tr>
<td>4.6</td>
<td>Zurich mode choice parameters</td>
<td>78</td>
</tr>
<tr>
<td>4.7</td>
<td>Basic cost model for Switzerland</td>
<td>80</td>
</tr>
<tr>
<td>4.8</td>
<td>Subscription cost model for Zurich</td>
<td>80</td>
</tr>
<tr>
<td>4.9</td>
<td>AMoD mode choice parameters</td>
<td>87</td>
</tr>
<tr>
<td>5.1</td>
<td>AMoD scenarios: Mode shares</td>
<td>93</td>
</tr>
<tr>
<td>5.2</td>
<td>AMoD scenarios: Vehicle distance</td>
<td>94</td>
</tr>
<tr>
<td>5.3</td>
<td>AMoD scenarios: Unique vehicle counts</td>
<td>95</td>
</tr>
<tr>
<td>5.4</td>
<td>AMoD scenarios: Daily cost and travel time</td>
<td>96</td>
</tr>
<tr>
<td>6.1</td>
<td>Île-de-France mode choice parameters</td>
<td>114</td>
</tr>
</tbody>
</table>
ABBREVIATIONS

AMOD  Automated Mobility on Demand
ARE  Federal Office for Spatial Development (Switzerland)
AU  Adaptative Uniform Rebalancing Policy (Section 5.3.1)
AV  Automated vehicle
BAV  Federal Office of Traffic (Switzerland)
BFS  Federal Statistical Office (Switzerland)
BPE  Service and facility census for France
CHF  Swiss Francs
DMC  Discrete mode choice (Section 2.2, Section 2.3)
DRIEA  Regional representation of the Ministry of Ecology and Energy in Île-de-France
DRT  Demand-Reponsive Transit
DVRP  Dynamic Vehicle Routing Problem
EGT  Regional household travel survey for Île-de-France
ENTD  National household travel survey for France
EUR  Euro
GBM  Fluidic Feed-Forward policy (Section 5.3.1)
GA  General subscription (for public transport in Switzerland)
GBM  Global Bipartite Matching Policy (Section 5.3.1)
GIS  Graphical Information System
GTFS  General Transit Feed Specification
HAFAS  Digital public transport schedule for Switzerland
**HTS** Household Travel Survey (see MTMC, EGT, ENTD)

**INSEE** Statistical Office of France

**LBH** Load-Balancing Heuristic policy (Section 5.3.1)

**MATSIM** Multi-Agent Transport Simulation

**MTMC** Mobility and Transport Micro Census in Switzerland

**OD** Origin-Destination

**OSM** OpenStreetMap

**PRNG** Pseudo-Random Number Generator

**SP** Stated Preference (survey)

**STATENT** Enterprise census of Switzerland

**STATPOP** Population census of Switzerland

**SVI** Association of Swiss transport engineers

**VKT** Vehicle kilometres travelled

**VTTS** Value of Travel Time Savings (Section 2.2.2)
INTRODUCTION

In recent years the world has seen a substantial increase in new mobility solutions. The variety spans from car-sharing services to ride-sharing platforms and micro-mobility services which flood cities with electric bikes and scooters. All of these developments have in common that they are fostered by modern information technology, which establishes a quick and convenient communication gateway between customers and service providers.

Along with this, travel decisions are in the process of becoming more and more fast-paced and individual. Traditionally, the options for mid- and long-range trips consisted of car use (which was mainly governed by ownership) or use of more or less well-developed public transit systems. Today, additional on-demand options exist, which do not require ownership of a mobility tool, but merely membership with various private mobility providers, i.e. ride-sharing services such as Uber, Lyft or any of the numerous regional options. They make it easy to call a ride at any time and any location in the service area. Today, a chauffeur-driven vehicle would arrive soon after the request and bring the customer to their desired destination. Such services offer previously unknown flexibility in terms of space and time, especially for non-car owners.

1.1 AUTOMATED MOBILITY ON DEMAND

At the same time, technology for automated driving is developing at a quick pace. Therefore, studies predict that existing on-demand ride-sharing services will transform into a concept described as Automated Mobility on Demand (AMoD). Such a service would send robotic taxis to pick up and drop off customers.

One can divide the short-term effects of automated mobility and AMoD into three categories: access to mobility, infrastructure usage and demand effects.

Automated vehicles would allow previously impeded or excluded user groups to gain access to mobility (Litman, 2020). This is true for the elderly or young children (Truong et al., 2017) who depend on public transport...
or other people to drive them, especially in areas that are currently underserved by public transport. On the other hand, giving people access to such services could lead to increased individualisation of traffic, which, in turn, might further contribute to the decline of aggregated public transport services.

In terms of infrastructure, some studies predict a more efficient use of the built environment (Alessandrini et al., 2015; Soteropoulos et al., 2019). As many expect automated vehicles to be connected, higher reaction rates could lead to smoother traffic (e.g. Stern et al., 2018) and highly efficient intersections (Pereira et al., 2017; Rios-Torres and Malikopoulos, 2017). Effectively, road capacities would increase, although this may not be true in the early phase of technology adoption as automated vehicles may be required to drive very carefully (see a review in Livingston et al., Upcoming). The flip side, when looking at infrastructure usage, would be that automated vehicles could drive empty. An AMoD system would even depend on empty rides (just like taxi drivers need to go to their customers today) to offer the service. Consequently, the vehicle kilometres travelled (VKT) may increase (Litman, 2020) and if an AMoD trip replaces each of today’s private car trips an increase of the number of cars being on the road at any time of day seems likely. In case AMoD could not be operated as efficiently as commonly expected, transport planners may start to observe occupancy rates of less than one person per vehicle at times.

Economically, the crucial point for the feasibility of an AMoD service is its costs. While services such as Uber today are believed to be only kept alive by venture capital, the removal of driver’s wages from the operator’s cost structure may make on-demand ride-sharing services profitable (Bliss, 2018). The prices could then drop drastically, making an AMoD service competitive compared to private vehicle ownership or even public transport. In such a case, it seems likely that low costs will lead to an increase in individual road-based travel, which is commonly known as the problem of induced demand (Litman, 2017; Meyer et al., 2017).

As one crucial competitive advantage of AMoD over private car ownership, usually comfort is named (see Becker and Axhausen (2017) for a review of surveys). Automated vehicles may be designed very differently than today’s cars which aim at providing comfort while driving, not for relaxation. Since people can perform many activities other than driving in an automated vehicle, they may value the time spent in the vehicle much higher than today (Becker and Axhausen, 2017). While this argument seems to be inherent to automated mobility and AMoD, service quality (such as
waiting times and response times) and service costs are strongly dependent on the service provider.

The ecological aspects, welfare considerations and cost-benefit components of AMoD services are therefore governed by the interplay of efficiency gains and losses in infrastructure use, their relative attractivity compared to traditional modes of transport, and their potential to increase mobility in absolute terms.

1.2 OPERATIONAL CHALLENGES

From the perspective of the service provider, an AMoD system poses new challenges in fleet control. While traditional transport service planning considers the time scale of years, months or weeks, a fleet of automated vehicles can theoretically be controlled dynamically from minute to minute or second to second. It is essential to control vehicle movements, customer pick-ups and dropoffs, maintenance, and charging periods as optimally as possible to gain competitive advantages over other services. For that, the best possible knowledge about the current and future demand is paramount.

Contrary to existing on-demand services that still rely on human interaction (König and Grippenkoven, 2017), an efficient AMoD system can not be controlled on a per-second time horizon by human dispatchers. Therefore, powerful algorithmic fleet operating policies are vital for the success of an AMoD service. They can make the difference between waiting times of one or two minutes between two different fleet operators.

For the operator, two key questions are essential to assess the profitability and economic sustainability of the service: How much revenue does the service generate by attracting demand? Moreover, how much cost is due to capital and operational expenditures? While longer-term decisions determine capital expenditures (e.g. how many vehicles to include in the fleet), operational expenditures can be almost directly derived from the distance driven by the fleet vehicles. Likewise, the distance driven with a paying customer determines revenues. Therefore, to increase profit or offer lower prices, the empty distance must be minimised.

On the other hand, demand is not only attracted by low prices, but also by low waiting times. While waiting times can be decreased by longer-term decisions (as a higher number of vehicles increases the chances of having one available in due time for every request), this is also possible through short-term fleet management. If the operator can predict demand reliably, he can move idle vehicles towards areas where requests will be
likely to arrive in the future. While he can provide short waiting times for those customers, these vehicles need to be moved empty. Hence, low waiting times come to the cost of extra empty distance, which translates to additional cost for the operator. Fleet operating policies, therefore, need to consider the tradeoff between minimisation of empty distance and not pushing waiting times beyond a guaranteed or desired level.

1.3 AUTOMATED MOBILITY IN SWITZERLAND

In 2016, the project SVI 2016/001: Induced demand by automated vehicles (Hörl et al., 2019) was commissioned by the Association of Swiss Transport Engineers (SVI) and financed by the Federal Road Administration of Switzerland (ASTRA). The project was conducted at the Institute for Transport Planning and Systems at ETH Zurich and had the goal to shed light on several open questions around automated vehicles. The overarching question was how they would affect future mobility in Switzerland. This thesis draws from the results and methods of this project. While the project, in general, looked at both the effects of privately owned automated vehicles and Autonomous Mobility on Demand, the thesis focuses on the results of the latter. The results are complemented by additional research and studies that were inspired by the work in SVI 2016/001.

The project consisted of three parts: A detailed cost assessment of automated mobility services in Switzerland, a stated preference survey to estimate user behaviour for automated mobility, and a transport simulation to assess system impacts holistically. The following paragraphs shall give a short overview of the two former components of the project, while the latter part will be the focus of the thesis.

The cost assessment part of the project has lead to a well-received cost calculator of automated and electrified mobility services (Bösch et al., 2018). The approach of the study was to establish a bottom-up cost assessment of existing mobility options such as private car ownership, public transport or taxi services in Switzerland. The assessed cost components range from investment costs, depreciation and maintenance to cleaning and fuel costs. The resulting cost values have been validated against known numbers, for instance, from public transport operators. Afterwards, the impacts of automation and electrification on all cost components have been estimated based on the literature. As a result, new cost structures arise which, for instance, take driver wages out of the equation if taxi services replace human-driven cars by automated vehicles.
For Switzerland, the study finds that the full cost of owning and using a private car is 0.48 CHF/pkm. An automated private car, according to this study, is expected to be even more expensive with 0.50 CHF/pkm. Furthermore, a cost reduction of around 50% can be expected for bus services, while rail services would only be affected marginally. The most substantial cost reduction, however, is expected for taxi services. Today, they are costly in Switzerland with 2.73 CHF/pkm per passenger kilometre, while an automated taxi service could operate at 0.41 CHF/pkm. This number gets very close to the cost of driving a private car.

The second part of SVI 2016/001 started with a survey in the canton of Zurich with about 350 respondents. In the first phase of the survey, the study asked respondents to provide sociodemographic information about themselves such as age, gender and income and to define two trips they do regularly, one for a distance of less and one for a distance over 50 km. In the second phase of the study, all respondents obtained information on automated mobility services, and they had to perform various choice tasks. Those contained their regular trip as indicated before and alternative automated modes of transport such as a private automated car and an automated taxi service. Attributes like waiting times and costs (based on the calculator) were varied to explore how the respondents would opt for or against different automated mobility services in the constructed tradeoff situations. The survey results were subsequently used to estimate statistical models of the respondents’ choice behaviour. From such models, it is possible to calculate a Value of Travel Time Savings, which quantifies how much money a person would theoretically be willing to pay to reduce travel time by, for instance, one hour.

For private cars, the study yielded a VTTS of around 19 CHF/h while the public transport value is 9 CHF/h\(^1\). In comparison, travellers value reducing time in an automated taxi at about 13 CHF/h. Hence, the results indicate that people perceive automated taxis as more comfortable than private cars. Their VTTS are even closer to those of public transport than to the private car. However, in-vehicle time is not the only important factor. In terms of waiting time, the respondents perceive public transport waiting time at 22 CHF/h. In contrast, waiting for an automated taxi translates to around 35 CHF/h. Therefore, people seem to be much more sensitive to long waiting times in an AMoD service than in conventional public transport.

\(^1\) For distances of around 5 km
These results are valuable as they support various assumptions about AMoD. The survey part of the study supports the claim that travellers perceive the time spent in an automated taxi as more desirable than spending time in a private car. Adding the removal of the vehicle ownership burden, this renders an AMoD service very attractive compared to private vehicle ownership. Likewise, the cost part of the study confirms that the prices of an AMoD service would be close to car ownership. Hence, it becomes evident that there is a market for AMoD services once the technology is mature enough. Therefore, AMoD operators will likely emerge.

While these results show that AMoD may become an attractive business, it is not clear which impact these services will have on the transport system once they become widely adopted. They may improve or worsen traffic problems in cities, they may or may not be an ecological addition to the transport ecosystem (Taiebat et al., 2018; Wadud et al., 2016), and they may or may not make access to transport more equitable and affordable for everybody. The critical task for transport planning is to assess the potential impact of AMoD services and to study potential outcomes and policy measures (Milakis et al., 2017). Hence, the goal of research should be to inform policymaking on the topic of automated mobility. The third part of SVI 2016/001 describes simulations to advance such discussions.

1.4 CHALLENGES AND RESEARCH GOALS

As outlined above, AMoD services require highly dynamic decision making. On the side of the customer, requests are issued on demand, while the AMoD operator needs to react to those requests in due time. Accordingly, spatially and temporally detailed interactions need to be considered when assessing the value and threats of such a system.

Traditionally, transport researchers use four-step models with which they simulate traffic on a macroscopic level by distributing vehicle flows over a network of roads. The underlying demand is usually defined over spatial zones from fine-grained grids to municipalities or even higher levels of aggregation. In any case, it is not easily possible to consider waiting times in the system, which depend on the specific locations and temporal order of customer requests. However, to allow for a fair assessment, waiting times should affect the choice between AMoD and other modes, just like travel times on congested roads affect the mode choice between public transport and the private car.
Therefore, agent-based transport models have been a common choice for simulations of automated taxis (see Narayanan et al. (2020) for a recent review). In such models, each traveller and each vehicle is a single entity in the transport network. Therefore, agent-based models appear to be the right choice for simulating emerging mobility modes. They are suited for the simulation of concepts such as automated vehicles, urban air mobility, car-sharing or micro-mobility where availability and attractivity of the service has a highly dynamic spatial and temporal dependency on demand itself. One such agent-based transport simulation frameworks is MATSim (Horni et al., 2016), which is developed by researchers at ETH Zurich and TU Berlin. MATSim was used as the simulation tool in SVI 2016/001.

The goal of the remaining project was then to extend and prepare MATSim for the simulation of automated taxi fleets. For that, the simulations made use of the cost analyses and surveys from the previous stages of the study.

The main research questions of the project were:

- How much demand would an automated taxi fleet attract in Zurich?
- How would such a fleet affect the transport system in Zurich?

While many simulation attempts for automated taxis existed already at the beginning of the project, none of them took cost structures and user preference into account in a consistent way. Therefore, besides the transport planning-related questions, the development of a methodology to make use of the cost analyses and choice models in MATSim was the technical focus of the project.

The first chapters of this thesis follow the development of the simulation platform. In Chapter 2 MATSim will be introduced. A comparison between the traditional approach of decision making in MATSim and a novel extension that considers discrete choice models is made. Chapter 3 covers new simulation components for MATSim that make it possible to simulate automated vehicles. It furthermore gives details on the AMoDeus simulation framework that emerged from this work and serves as a comparison platform of fleet control algorithms.

While the development of new simulation components was necessary to reach the project goals, it was equally important to generate a realistic population of artificial agents that represent the travel demand in the study area of Zurich, Switzerland. The synthesis of this population is described in detail in Chapter 4. Chapter 5 then presents results based on SVI 2016/001 by analysing how a fleet of automated taxis would affect the transport
system of Zurich. The chapter presents further studies that have partly been done in parallel or after this work to put the obtained results into perspective.

The development of the simulation components and the population synthesis raised many questions and points for discussion. One major point is how far a methodology for agent-based simulation is specific to one use case and whether it can be generalised. Chapter 6 covers this topic by showcasing the development of a synthetic agent population for Île-de-France, followed by a simulation of automated taxis in Paris. Finally, Chapter 7 discusses the applicability of the generated results and tools. The thesis finishes with a conclusion in Chapter 8.
DYNAMIC DEMAND SIMULATION

The goal of the simulations in this thesis was to explore the interplay between the characteristics of conventional and new mobility services with traveller preferences. This process is inherently circular. Given a specific demand, the AMoD operator can provide a certain level of service. If demand increases and the operator does not increase fleet size or uses the vehicles more efficiently, the level of service will worsen. Hence, there is an upper limit of how large the demand can become before waiting times become bad enough that no new customers can be attracted. On the way to this upper demand bound self-selection effects can lead to the phenomenon that different user groups are attracted to the service, depending on the specific configuration of waiting times and prices. Therefore, an important part of the transport simulation should be to simulate such customer behaviour reliably.

The following two sections will give an overview of the MATSim simulation framework in general, with a focus on how the simulation considers decision-making. Afterwards, a new decision-making component for MATSim is introduced that makes use of discrete choice models, followed by a discussion of advantages and disadvantages of both approaches. Potential pathways of combining the approaches will be provided.

2.1 THE MATSIM FRAMEWORK

MATSim (Horni et al., 2016) is an agent-based transport simulation framework. While it can be used as a stand-alone tool with a simple GUI and well-documented configuration file, its real value lies in the fact that it is provided as an open-source programming framework. It provides a rich toolkit to set up very customised and individualised agent-based transport simulations. Occasionally, institutes around the world add new developments to the core code of MATSim as contributions.
2.1.1 Basic concepts

MATSim simulates traffic by treating each traveller and vehicle in the transport system as a unique entity. The simulation is usually performed on a time scale of seconds, which makes it possible to follow locations and states of those entities (agents) in detail throughout one simulation day (see Figure 2.1 for an example). The goal is then to set up the simulation in a way such that the emerging traffic patterns from interactions of these agents resemble what users observe in reality.

MATSim runs are usually full-day simulations. Each of the simulated agents holds sociodemographic attributes and a daily plan that contains activities and trips. Activities have a specific type, end times and desired durations while trips carry their transportation mode and additional information on the planned route through the transportation network. The basic idea of MATSim is then to simultaneously simulate all of these daily plans in one capacitated transportation network. Traffic jams emerge if too many agents use a particular road in the network. Also, trains can become crowded if too many agents want to use them. From these mobility
Figure 2.2: The general simulation loop of MATSim with an initial population state, mobility simulation, decision-making and analysis.

Simulations, information like the observed travel times, delays and costs for each agent can be recorded. Based on this data, agents can alter their daily plans at the end of the day, for instance in terms of the departure times from activities, their chosen transport modes, the routes through the road network or connections in public transport. Even locations of activities (Horni et al., 2009) or the existence of certain activities themselves may be subject to change (Feil, 2010). Afterwards, MATSim simulates the new daily plans with new outcomes.

The two main components of MATSim (as shown in Figure 2.2) are a detailed agent-based mobility simulation and a decision-making process. While the latter will be covered in detail further below, a short overview of the mobility simulation shall be given in the next paragraphs.

The transport simulation part of MATSim comprises two components: the activity queue and the mobility simulation. Together both components are known as the QSim. At the beginning of one simulation day, all agents rest in activity state. Usually, the first activity in an agent’s plan is a “home” activity. As each of the agents may have different departure times from the first activity, all agents are put in an ordered queue dependent on their planned activity end time. The QSim then runs iteratively through one full day and checks second by second (or at another configurable interval) which agents should end their current activity. The simulator sends those
agents to update their state. In case they had another activity planned at the same location, the QSim would add their new activity back to the activity queue, and would wake up the agent again at the activity end time. However, usually agents would connect activities by trips. Dependent on the desired mode of transport, agents are sent to specific simulators for public transport, slow modes, car travel, and other modes. An important simulator is the Netsim, which simulates car traffic in a queue-based fashion.

The Netsim is often characterised as a mesoscopic traffic simulator. It does not consider second-by-second movements of vehicles using a car-following model, which would be called microscopic in this context. However, it considers agent interactions on a detailed temporal and spatial level, which goes beyond more aggregated (macroscopic) modelling approaches that rely on volume-delay functions and similar approaches. Internally, the Netsim models each link (road) in the network as a “queue with buffer” (see Figure 2.3). The first part (queue) receives vehicles from upstream, while the second part (buffer) sends vehicles to downstream links. During the simulation, the Netsim considers each link individually in two phases for each simulation step.

• In the first phase, vehicles are moved from the queue to the buffer. Whenever a vehicle enters a link, the Netsim calculates its earliest exit time, based on the traversal speed and the length of the link. It then adds agents to a queue in the order of arrival. Agents at the top of the queue whose exit time is reached are moved further to the buffer. However, certain restrictions exist. In general, all links are characterised by a flow capacity value which defines how many vehicles are allowed to traverse the link per time step. Hence, the link simulator meters how many vehicles it moves to the buffer. Note that this metering is independent of the earliest exit time.

• In the second phase of the Netsim, it moves vehicles from upstream buffers into downstream queues. This movement is only possible if there is space available on the downstream links. This condition is characterised by the storage capacity, which depends on the length of the link. It describes how many vehicles are allowed to reside on it at the same time. Note that this includes vehicles inside the queue and buffer alike. Hence, traffic is not only limited by the flow capacity, but also by congestion on downstream links.

Agarwal et al. (2018) describe these traffic dynamics in more detail and extend them with additional elements. For instance, the concept of
holes propagating back from the end of each link allows for more realistic modelling of shock waves inside of the traffic dynamics.

Researchers have applied MATSim to many uses cases around the world in terms of study areas and application domains. There exist well-documented simulation scenarios for Berlin (Ziemke et al., 2019) and Santiago de Chile (Kickhöfer et al., 2016) and studies range from minibus systems (Neumann et al., 2015) to public transit network design (Manser et al., Under Review), rail interruptions (Leng and Corman, 2020), freight simulation (Bean and Joubert, 2018), and parking (Bischoff and Nagel, 2017) to car-sharing (Ciari et al., 2015; Balac et al., 2019a) and Urban Air Mobility (Balac et al., 2019b).

2.1.2 Decision-making in MATSim

The decision-making process in MATSim is iterative in that always new information is gained from sequential traffic simulations that inform the available choices for the agents. Furthermore, it is based on the theory of utility maximisation. Throughout one day, every activity and trip is valued with a certain score in MATSim. This score increases if activities are
performed for the right duration and at the right time of day. It decreases if agents are too late or too early or while they spend time in traffic.

A typical score for a trip in an agent’s plan follows the function:

\[
S_i = \beta_{ASC,m} + \beta_{travelTime,m} \cdot x_{travelTime} + \beta_{distance,m} \cdot x_{distance} + \beta_{cost} \cdot x_{cost}
\]

Here the \( \beta \) are scoring parameters for the chosen mode \( m \), e.g. \( m \in \{\text{car}, \text{public transport}, \text{bicycle}, \text{walk}\} \), and \( x \) are attributes estimated or derived from the transport simulation. While the parameters in Equation 2.1 are standard in MATSim and users can easily set them up through a configuration file, in theory arbitrarily complex scoring functions can be defined in the code.

Note that the \( \beta \) parameters are usually smaller or equal to zero to penalise time spent in transit. Further standard parameters for specific modes, such as a \( \beta_{\text{numberOfTransfers}} \) for public transport are available.

The typical way of scoring activities in MATSim is

\[
S^+ = \beta_{\text{performing}} \cdot \theta_{\text{typicalDuration}} \cdot \log \left( \frac{x_{\text{duration}}}{\theta_{\text{zeroUtilityDuration}}} \right)
\]

if the activity is performed for longer than a duration \( \theta_{\text{zeroUtilityDuration}} \). This value may be defined per activity type via configuration but even more fine-grained via code. For shorter durations, the function is linearly extended with the slope at this time, i.e.

\[
S^- = -\left(\theta_{\text{zeroUtilityDuration}} - x_{\text{duration}}\right) \cdot \frac{\partial S}{\partial x_{\text{duration}}} (0)
\]

Further scoring components can be applied out-of-the-box. For instance, there is a linear penalty on the time that an agent is arriving too late or leaving too early from an activity. Some of these components are furthermore dependent on attributes of the selected location, for instance, opening times of a shop and others. The total score of a realised plan in the transport simulation is then the sum of all trip and activity scores.

After the simulation, it is possible to modify the plan. For instance, an agent may choose a new mode of transport. With \( i \) describing a certain realisation of the daily plan, one can then check how the scores \( S_i \) of
different realisations relate to each other. In general, utility maximization suggests that a person would prefer plan \( i \) over plan \( j \) if \( S_i > S_j \).

Assuming that an agent only has access to using the car or public transport and the simulation only considers mode choice, we arrive at a countable set of possible plans. Assuming that the plan has \( N \) trips, there are \( 2^N \) alternative plans. Assuming further that everything else except the agent’s mode choices in the system stays constant, we could simply iterate through all options \( i \in \{1, ..., 2^N\} \) and find the best plan as

\[
i^* = \arg \max_i S_i \tag{2.4}
\]

In MATSim, many agents are simulated in the transport system. Hence the choice of one agent in the fight for slots in space and time influences the observed score of every other agent. Let \( l \in \mathbb{N}_{\geq 1} \) define the index of an agent and \( L \in \mathbb{N}_{\geq 0} \) the number of agents. Let furthermore \( \mathcal{I}_l \) be the set of all possible plans for agent \( l \) such that \( i_l \in \mathcal{I}_l \). We can then define

\[
I^* = (i^*_1, ..., i^*_l, ..., i^*_L) \tag{2.5}
\]

with

\[
S_{i^*_l} \geq S_{i_l} \quad \forall i_l \in \mathcal{I}_l \quad \forall l \in \{1, ..., L\} \tag{2.6}
\]

as a selection of plans in which the system is in user optimum state. In this state, none of the agents can improve their score by unilaterally changing their plan.

There are cases in which finding a user optimum is quite easy. Consider the example from above, only that the road network has a limited capacity. One would start by letting all agents use uncapacitated public transport. Subsequently, one agent after another would switch to the car alternative as it is valued slightly better. At some point, the road would reach the capacity limit. For all remaining agents, the public transport alternative would then provide a better score. This assignment process is independent of the order of agents if they all have the same perception (i.e. same scoring function). Hence, there is a large number of possible user optima \( I^*_u \) with \( u \in \mathbb{N}_{\geq 0} \) describing the different permutations.

Defining \( S^*_{l,u} \) as the score of agent \( l \) in user optimum \( u \) this example would yield that there is a constant value \( S^*_{l,u} = C^+ \) for all states \( u \) in which agents \( l \) are able to use the car, and a constant value \( S^*_{l,u} = C^- \) in case they choose the second-best option.
A metric on the scores of all agents can be defined, for instance the total score \( S_u^S = \sum_i S_{l,u}^* \) or the mean \( \bar{S}_u = \frac{1}{L} \sum_i S_{l,u}^* \). Again, in the presented example one would arrive at

\[
S_u^S = \xi C_1 + (1 - \xi) C_2 = C_3 \quad \text{and} \quad \bar{S}_u = C_3 / L \quad (2.7)
\]

where the capacity of the road influences \( \xi \). Nevertheless, both metrics are independent of the specific equilibrium state \( u \).

Now consider that some of the agents value public transport less than others (maybe because it is not easily accessible for the elderly). In this case, it matters which agent is assigned a specific plan alternative. If all penalised agents need to take public transport, there is the lowest possible \( S_u^S \). If all of them can take the car, the best possible state is achieved. There are many mixed states in which only a share of those agents can use the car. Note that in any case, all of these states are user equilibria in which agents can not unilaterally improve their score. To gain a better score, two of the “mismatched” agents would need to switch, which is not a unilateral decision. The last considerations add another component to the choice process. From all user equilibrium states \( u \), is it possible to find the \( u^* \) with

\[
 u^* = \arg \max_u S_u^S \quad ? \quad (2.8)
\]

Such a user equilibrium maximises the cumulative or average score of the population. Note that this is still different from a system optimum in general terms. In a system optimum some agents would have the possibility to switch to another plan to unilaterally improve their score but they would not do it or they would be prevented from doing so to have an overall better system state. A system optimum is therefore inherently linked to some overall objective (such as minimisation of emissions or low congestion levels) that is policy-defined.

The decision-making process in MATSim has the goal to find a state close to \( u^* \). The aim is to find plan configurations for all agents such that the cumulative population score are as high as possible. Assuming again the toy example from above, where each agent has \( 2^N \) plan alternatives, the full set of possible configurations of the population would consist of \( 2^{N \cdot L} \) alternatives as one can combine any plan with any other set of plans. Suppose the plans have three trips, and there are only one hundred agents, there would already be \( 2^{300} \) alternatives to evaluate.

The goal of decision making is to give agents freedom in at least two dimensions: departure time choice and mode choice. As outlined above,
given $M$ different modes of transport and $N$ trips, an agent may have $M^N$ alternatives to assign modes to trips in their plans. For departure time choice, consider that the agent may choose any hour of the day, so there would be $24^N$ alternatives. Each of them could be combined with each mode choice alternative, leading to $(24 \cdot M)^N$ options. More fine-grained temporal choice options would further increase this number. Assuming the usual number of four modes in a MATSim simulation and plans of a length of around three trips, this leads to 884,736 alternatives, which is closer to what would be the case in a realistic simulation. Now consider that MATSim intends to simulate thousands or even millions of agents, which would mean one needs to evaluate $(884,736)^{1,000,000}$ alternatives to find the optimal user equilibrium. A further explosion of alternatives can be observed when there are thousands of locations that serve as alternative destinations for the activities in the plans.

Evaluating all possible configurations is not an option for all practical applications of MATSim. Therefore, an approach is used that has been coined as the “co-evolutionary algorithm” of MATSim. The algorithm consists of four major parts: (1) scoring agents’ plans as described above, (2) the plan memory, (3) plan selection, and (4) innovation strategies.

At the beginning of a MATSim simulation, each agent usually starts with one initial plan that is scored in the subsequent mobility simulation. Afterwards, this plan may be modified, but modifications are not applied directly to the existing plan. Instead, the algorithm copies the plan and applies changes to this copy. Both the original plan and the newly generated one are then part of the agent memory. The memory has a limited size in the magnitude of around three to five plans and tracks the observed score for each plan. If a new plan exceeds the memory limit, the algorithm chooses one of the plans to be removed.

Copying and modifying an existing plan is only one option that can happen after the mobility simulation. MATSim defines a set of replanning strategies which define the decision-making of the agent. Each of the strategies $k$ is assigned a certain weight $w_k$, which defines the probability of calling this strategy in the replanning step of MATSim. For that purpose the weights are normalized such that $\sum w_k = 1$.

There are selection strategies and innovation strategies. The former consider the memory of an agent and apply a specific selection procedure to choose one of the plans to be executed in the next iteration. Note that the “remembered” score of a plan might not be up to date anymore because the

---

1 In fact, in MATSim any second of the day could be chosen.
conditions in the transport system may have changed. This way, the score of the selected plan is updated after the next mobility simulation run.

The innovation strategies have the task to apply modifications to a copied plan. The typical modules are TimeMutator, which changes the activity end times in the plan with a random offset \( \delta \) sampled from a uniform distribution \( \Delta \sim U(-T, T) \), Reroute, which updates a trip’s route through the network, and SubtourModeChoice. The latter searches for all trip sequences in a plan that start and end at the same location. Then, the strategy changes the transport modes of these trips to randomly chosen alternatives. For modes that are not constrained by mobility tools (like making sure that the private car arrives back home), further fine-grained randomisation of modes along the sequence is possible in recent versions of MATSim.

Structurally, the replanning step of MATSim looks as follows:

\[
\begin{align*}
\text{if memory exceeds limit then} & \quad \text{Choose plan for removal} \\
& \quad \text{if removed plan was selected plan of agent then} \\
& \quad \quad \text{Uniformly select new plan} \\

\text{Choose replanning strategy based on weights} & \\
\text{if a selection strategy is chosen then} & \quad \text{Apply strategy to select a plan for execution} \\
\text{if a innovation strategy is chosen then} & \quad \text{Uniformly select one plan from memory} \\
& \quad \text{Add a copy of the plan to memory} \\
& \quad \text{Apply changes to it acc. to innovation strategy} \\
& \quad \text{Select new plan for execution}
\end{align*}
\]

Two design decisions strongly influence the behaviour of the algorithm: the selection strategy for plan removal and the plan selection strategy. The standard configuration of MATSim always removes the plan with the worst score. The set of plans in memory is therefore implicitly biased towards plans with higher scores.

One common choice for plan selection is to use the logit formula. Denoting the plans in an agent’s memory by \( P_1, \ldots, P_N \) the probability of selecting plan \( P_i \) is
\[ P(P_i) = \frac{\exp(\sigma S_i)}{\sum_{i'=1}^{N} \exp(\sigma S_{i'})} \] (2.9)

According to the formula, plans with higher scores are selected with higher probability. Note that usually \( \sigma \) is set to 1. Together with the magnitude of \( S \) it has considerable influence how “random” the selection is. In case \( \sigma = 0 \) the selection would be uniform over the available plans, while \( \sigma \to \infty \) would make the selection deterministic for the plan with the highest score\(^2\).

The usual plan selector in MATSim mimics the behaviour of the logit formula, but only considers a choice between the currently selected plan \( P^* \) with score \( S^* \) and one uniform randomly selected plan from memory. The probability of switching from the currently selected plan to the proposed plan \( P' \) with score \( S' \) is then:

\[ P(P^* \to P') = \eta \exp(\beta (S' - S^*)) \] (2.10)

One can interpret the formula as follows: If \( S^* = S' \) the probability of performing the transition is \( \eta \). If \( S^* > S' \) the probability is smaller than \( \eta \) and if the proposed plan has a better score, the probability of the switch is larger than \( \eta \). Currently, \( \eta \) is fixed to \( \eta = 0.01 \) in MATSim, but should probably be made available through configuration in the future. Theoretically, this selection process converges to the same results as the logit formula (Nagel and Flötteröd, 2016). The choice of \( \eta \) does not affect this statement; rather, the speed of convergence is dependent on it.

Other plan selectors are available, such as a uniform random selector and a “maximum score selector”, which always selects the best plan that is available in memory. It will become of interest further below.

The search dynamics from iteration to iteration have two major influences: The repeated removal of the worst plan, and preference of selecting plans with high scores. As the innovation part proposes changes at random, the selection and removal parts are responsible for filtering out those proposals that are not beneficial to increase the agent’s daily score. The reason for not always selecting the plan with the maximum score and keeping a memory of past plans is to be able to escape local maxima. While a particular plan may have been updated multiple times with increasing score, it may reach a limit eventually. In that case, it makes sense to jump back to a plan from

---

\(^2\) This is only true in theory. In practice large numbers of \( \sigma \) would lead to loss of numerical precision and overflow.
before and “try” a different path of modifications that may lead to better results.

This search for the user equilibrium comes with artefacts: Imagine a search process as described above that runs for infinite time, but there are no capacity restrictions in the system (i.e. anybody can choose between car or public transport). In such a case plans with the slightly better scored car choice would accumulate in all agents’ memories over time. Now and then, the innovation step would produce a plan where the agent has to use public transport. As MATSim would call the innovation strategy with replanning weight $w$ (for instance in 10% of the cases, $w = 0.1$) there would be a 50/50 chance to create a “car” or a “public transport” plan. Hence, in expectation, one would see at least 5% of agents in the simulation taking public transport.

The recommended remedy for such a search artefact is to turn off innovation strategies after a sufficient number of iterations. After that, only selection is active on all agents’ memories. For some of them, a sub-optimal plan may still exist, but dependent on the scores of the optimal plans the rate of occurrence after selection might be low. ³ However, for practical purposes, simulations cannot be run for such large numbers of iterations that optimal plans sufficiently populate agents’ memories. Accordingly, innovation turn-off will “freeze” the current memory of an agent and afterwards selection is performed on this set of plans.

The situation becomes more complicated if there are capacities in the simulation. Then each agent’s optimal score is dependent on the plans of all other agents. Having 10% of agents innovating their plans heavily biases the perceived situation in the transport system. Therefore, turning off innovation has another objective in this, more practice-relevant, case: By removing 10% of strongly random (and potentially irrational and unintuitive) travel decisions, traffic conditions are stabilised substantially. Accordingly, it becomes easier to bring the agents into an equilibrium state, given the plans that are available to them after turning off innovation. However, even then, one can observe variability in the choices from iteration to iteration as the selection process remains stochastic.

To summarise, MATSim applies a search algorithm which uses random plan changes to generate plan proposals in a user equilibrium. The selection and removal part chooses those proposals that are promising to lead to a high score equilibrium. As the search space is vast and computational

³ It will likely not be zero, except maximum score selection is used.
resources are limited, the process stops once the population reaches a sufficiently stable score.

At the end of this analysis, the focus should be put again on the initial question: How does decision-making in MATSim work? MATSim strongly revolves around the concept of a user equilibrium in which every agent finds a daily plan with the best possible score. One can criticise this approach from different perspectives. First, one can argue that reality is inherently not in equilibrium. Constantly, decisions and conditions are changing. However, we observe phenomena such as traffic jams that occur every morning at the same spots in a city. It is therefore also possible to argue that some constrained situations, in reality, are close to a user equilibrium. Indeed, the selfish behaviour of travellers plays an active role in causing congestion. Second, assuming that one can sufficiently approximate reality by a user equilibrium, which one is the correct one? Is it the one with the highest possible “cumulative score”, the one with the lowest, or somewhere in between? While these are interesting questions, they are of lesser practical relevance in setting up MATSim simulations.

The reason is that it is inherent to the process that one needs to calibrate MATSim. Usually, one does this by setting scoring parameters in such a way that specific metrics obtain a good fit with reference data measured from reality. The most common of such metrics is the overall mode share of the system. In the toy example, we would measure that in reality, 70% of travellers take the car and 30% of them take public transport. Interestingly, if the system does not have any capacities, turning off innovation at the right point becomes crucial even to be able to achieve such a modal split. Other calibration objectives are vehicle counts on specific links for an average weekday or by the hour, boarding and alighting counts at public transit stations or travel time distributions, to give some examples.

At the time of writing this thesis, multiple approaches for automatic calibration of MATSim simulations have been proposed (Flötteröd et al., 2012) or are under development (Makarova et al., Under Review). The recently published Opdyts algorithm (Flötteröd, 2017) makes use of the iterative structure of MATSim as system metrics and scores can be approximated by a linear increase between two or few iterations. Interestingly, innovation turn-off poses a problem in such cases as it introduces a strong non-linearity into the model dynamics. It is therefore important to discuss potential alternatives.

In the future, it will be essential to study some of the phenomena mentioned above and to describe the search process of MATSim more thor-
oughly: How does the variance within one simulation depend on the chosen parameters? How does the variance across simulations with different random seeds behave? How does turning off innovation at different iterations affect the final score and variance? Some of such analyses have recently performed on specific use cases (Guggisberg Bicudo, 2020; Paulsen et al., 2018), but not formalized in a general way.

2.2 DISCRETE CHOICE MODELING

Discrete mode choice models (Train, 2009) are an essential part of transport research. It is their goal to statistically describe the decisions that people make, for instance, when they are faced with the task to decide whether to take a car to go to work or to take the bus. The nature of most of the decisions of interest is discrete, i.e. there are two or more options available that comprise specific attributes. The following section will give a brief introduction to set the field for understanding how a discrete mode choice model has been integrated into MATSim.

2.2.1 Basic theory

Most discrete choice models are based on the concept of utility maximization. While MATSim is focused around the notion of maximizing a score, discrete choice models assume that decision-makers want to maximize the utility of their decision. Similar to scoring functions in MATSim, those systematic utilities are expressed analytically:

\[ v_{i,\text{car}} = \beta_{\text{ASC,car}} + \beta_{\text{travelTime,car}} \cdot x_{\text{travelTime,car},i} + \beta_{\text{cost}} \cdot x_{\text{cost,car},i} + ... \]

\[ v_{i,\text{pt}} = \beta_{\text{ASC,pt}} + \beta_{\text{inVehicleTime,pt}} \cdot x_{\text{inVehicleTime,pt},i} + \beta_{\text{waitingTime,pt}} \cdot x_{\text{waitingTime,pt},i} + \beta_{\text{lineSwitch,pt}} \cdot x_{\text{lineSwitches,pt},i} + \beta_{\text{cost}} \cdot x_{\text{cost,pt},i} + ... \]  

(2.11)

Here, \( v_i \) describes the utility of an alternative for decision \( i \), \( x \) describe the choice attributes and \( \beta \) are the model parameters that define the tradeoffs between the attributes and alternatives. Each alternative has a special \( \beta_{\text{ASC}} \), which is the alternative specific constant. Note that parameters can appear multiple times, like \( \beta_{\text{cost}} \), which describes how monetary costs are perceived.
All $\beta$ can be thought of as being given in the unit of utilities $[\text{utils} / \cdot]$ and are combined in the vector $\beta$.

Assume that there are $N$ persons and each of them has been tracked for one day. The trip of interest is their morning commute. Each morning commute then represents a choice situation $i$, here between two different modes “car” and “public transport”. Furthermore, the attributes $x$ are known for all chosen and alternative options. Given trip $i$ and alternative $k$, we can define a deterministic utility $v_{i,k}(\beta)$.

Defining $k^*_i$ as the selected alternative of all respondents, we can try to find a set of $\beta$ that fulfil the following constraint:

$$k^*_i = \arg \max_k \{v_{i,k}(\beta) \mid k \in \{\text{car, pt}\}\} \quad \forall i \in \{1, \ldots, N\} \quad (2.12)$$

Generally, this will not easily be possible as some choices by different respondents may be contradictory in the sense of Equation 2.12. Such differences exist because people have variable tastes in their decision making, and the utility functions defined by the modeller may only approximate the choice process in all its subtlety. Therefore, slack is added to the utilities by explicitly modelling uncertainty about the choice into the equations. We define a random utility for each choice situation $i$ and alternative $k$ as:

$$u_{i,k} = v_{i,k} + \sigma \epsilon_{i,k} \quad (2.13)$$

Here, $\epsilon_{i,k}$ is a i.i.d. random variable and $\sigma \in \mathbb{R}_{\geq 0}$. We now require that:

$$k^*_i = \arg \max_k \{v_{i,k}(\beta) + \sigma \epsilon_{i,k} \mid k \in \{\text{car, pt}\}\} \quad \forall i \in \{1, \ldots, N\} \quad (2.14)$$

The selected choice $k^*_i$ can now be expressed as a random variable $K^*_i$ that follows a specific distribution, dependent on the error distribution. Some alternatives are more likely to be chosen than others. Note that if $\sigma = 0$ the random term vanishes and deterministic choices are made, but if $\sigma \to \infty$ the random term becomes dominant. Hence, choices would be completely random. It is possible to formally write down the probability that a certain alternative $k_i$ is the actual choice $k^*_i$ in choice situation $i$:

$$P(K^*_i = k_i \mid \beta) = P(u_{i,k_i} \geq u_{i,1} \wedge \ldots \wedge u_{i,k_i} \geq u_{i,k} \mid \beta) \quad (2.15)$$

The probability of $k_i$ being the selected option is the probability of its associated utility being larger or equal to the utility of any other choice.
Now viewing the parameters $\beta$ as the dependent variable, it is possible to write down a likelihood:

$$L(\beta) = \prod_{i=1}^{N} P(k^*_i \mid \beta)$$  \hspace{1cm} (2.16)

In principle, one could try to find a maximum likelihood estimate for $\beta$. Unfortunately, this is not easy as there is no general expression for $P(K^*_i = k_i \mid \beta)$. A common assumption is to define $\epsilon_{i,k} \sim EV$, i.e. that the model errors follow an Extreme Value distribution (such as a standard Gumbel distribution). In this case, it has been shown (McFadden, 1974; Train, 2009) that there is an analytical expression for the choice probability:

$$P_{i,k} = \exp\left(\frac{v_{i,k}}{\sigma}\right) \sum_{k'} \exp\left(\frac{v_{i,k'}}{\sigma}\right)$$  \hspace{1cm} (2.17)

Here, the abbreviation $P(K^*_i = k_i \mid \beta) = P_{i,k}$ is used. Equation 2.17 shows that the probability of choosing alternative $k$ is the exponential of its deterministic utility scaled by $\sigma$ and divided by the sum of the exponentials of all scaled alternative utilities. By defining the error to be i.i.d. EV distributed, the initial random utility maximization model is transformed into a more specific multinomial logit model that has an analytical expression for the choice probability. It is hence possible to write down Equation 2.16 analytically and derive a log-likelihood function. It has been shown that the maximum log-likelihood estimator of a multinomial logit model exists (Train, 2009). In fact, the model provides analytical derivatives $\frac{d \log P}{d \beta}$ such that $\beta$ can be estimated very fast by applying a simple gradient descent optimization.

Looking at Equation 2.17 one can see that the formulation is consistent with the pattern that has been observed above. On the one hand, if $\sigma \to \infty$ all exponentials become 1 and $P_{i,k} = \frac{1}{K}$ for all alternatives, i.e. the selection is uniform random. On the other hand, if $\sigma \to 0$, the distribution becomes increasingly spiked for the alternative with the maximum utility.

It should be noted though, that in practice only $(\sigma \beta)$ can be estimated via maximum likelihood, resulting from the structure of the equations. Hence, $\sigma$ itself is unidentifiable as it is implicitly included in the resulting values for $\beta$.

These considerations conclude a basic introduction to the multinomial logit model that will be used throughout the thesis. Many differently structured and more elaborate model formulations exist, from multinomial probit
to *nested* and *mixed* logit models and beyond. Train (2009) provides an outstanding and detailed introduction.

### 2.2.2 Econometric value

Today, where processing power is abundant, one may define (to stick with the conventional notation) a vector of attributes $X$ and a vector of chosen alternatives $Y$ and search for a mapping $f$ that fits well the equation $Y = f(x)$. From artificial intelligence and machine learning many approaches exist, such as neural networks, that can easily construct a function $f(\cdot)$ *without* the initial work of modelling adequate utility functions and thoroughly cleaning the data set. One, therefore, might question what is the value of a discrete choice model if such possibilities exist?

The reason is that, for instance, a neural network is commonly defined as a black-box model. Some training input $X^T$ goes into the network $f$ with output $Y^T$. Then $f$ is adjusted such that $Y^T$ replicates well reference data and ideally $f$ also replicates well another validation set $(X^V, Y^V)$ which it did not “see” in the training phase. Hence, the focus of approaches like neural networks is to optimize the prediction rate. It is (primarily) *not* a tool for researchers to understand the process of decision making, but merely to replicate it. On the other side, a discrete choice model involves substantial modelling effort when designing the utility functions and interactions between attributes.

An important concept in discrete choice models is the following. Consider the two utility functions from Equation 2.11 for the choice of car versus public transport. As defined previously, all terms in those sums are given in the unit of $[\text{util}]$. However, parameters, such as $\beta_{\text{cost}}$ are defined such that they convert values of the respective attribute into utilities. For instance, we have $\beta_{\text{cost}}$ given in $[\text{util}/\text{CHF}]$ for the case of Swiss Francs (CHF) and $\beta_{\text{travel} \text{Time, car},i}$ in $[\text{util}/\text{min}]$. A common concept in choice modelling is to take the cost parameter $\beta_{\text{cost}}$ and divide all equations by this value:
\[
\frac{v_{i,\text{car}}}{\beta_{\text{cost}}} = \frac{\beta_{\text{ASC,car}}}{\beta_{\text{cost}}} + \frac{\beta_{\text{travelTime,car}}}{\beta_{\text{cost}}} \cdot x_{\text{travelTime,car},i} + x_{\text{cost,car},i} + \ldots
\]

\[
\frac{v_{i,\text{pt}}}{\beta_{\text{cost}}} = \frac{\beta_{\text{ASC,pt}}}{\beta_{\text{cost}}} + \frac{\beta_{\text{inVehicleTime,pt}}}{\beta_{\text{cost}}} \cdot x_{\text{inVehicleTime,pt},i} + x_{\text{cost,pt},i} + \frac{\beta_{\text{lineSwitch,pt}}}{\beta_{\text{cost}}} \cdot x_{\text{lineSwitches,pt},i} + x_{\text{cost,pt},i} + \ldots
\] (2.18)

The utility functions are now given in the unit of [CHF] and therefore represent a perceived monetary cost. Note that the estimation process from above is still valid for these equations, only that also the model error would need to be given in monetary units. Furthermore, the same transform could directly be applied to the probability of choosing an alternative in Equation 2.17.

Looking at Equation 2.18 one can derive the following insight: A saving of car travel time by 5 minutes would lead to a change of the perceived monetary cost of that mode by five times \( \frac{\beta_{\text{travelTime}}}{\beta_{\text{cost}}} \). This value is therefore commonly known as the Value of Travel Time Savings (VTTS):

\[
VTTS_{\text{car}} = \frac{\beta_{\text{travelTime,car}}}{\beta_{\text{cost}}}
\] (2.19)

VTTSs are important and useful as they make it immediately possible to grasp the value, for instance, of an infrastructure project that is predicted to change travel time by a certain value. If it is known how many users are affected by that change, a social benefit can be derived from the VTTS. On the other side, one can know the infrastructure investment and maintenance costs. Comparing both values with each other leads to the basic principle of a cost-benefit analysis.

The VTTS is also often known as the Value of Time (VOT) or more specifically when it comes to other choice dimensions, such as IVTT (In-Vehicle Travel Time) or VWT (Value of Waiting Time).

### 2.2.3 Choice sampling

As this thesis focuses on simulation, the main interest is not how to estimate a discrete choice model, but how to use it for making decisions. Given
attributes \( x' \) for a choice situation and the estimated parameters \( \beta \) the idea is to \textit{sample} decisions.

For the following, the choice situation index \( i \) will be dropped as only one specific alternative will be regarded. Assume one has a completely new trip, with new attributes \( x' \). It is possible to use Equation 2.17 to calculate the probability of using transport for this trip: \( P_{pt}(x' \mid \beta) \). Likewise, one can use the distribution \( P(K) \) to sample 1,000 decisions for a specific trip. Assume that for this trip \( P_{car} = 0.7 \) and \( P_{pt} = 0.3 \). Within these 1,000 samples, there should be 700 decisions for car and 300 choices of public transport in expectation. Conceptually, applying such a sampling procedure is carried out in three steps:

1. Assume the alternatives are referenced by an index \( k \in \{0, ..., K - 1\} \).
2. Calculate \( P_k \) for all alternatives \( k \) given attributes \( x \) and parameters \( \beta \).
3. Construct an empirical CDF as \( C_k = \sum_{k' = 0}^{k} P_{k'} \).
4. Sample a uniform random value \( u \sim U(0, 1) \).
5. Select \( k^* = \sum_{k' = 0}^{K - 1} [u \geq C_k] \).

In principle, also Equation 2.14 can be used to sample such decisions. For that, one would sample \( 2 \cdot 1,000 \) error values, apply them to the deterministic utilities \( v_k \) and always select the alternative with the highest random utility \( u_k \). Formally, the choice process would look as follows:

1. Assume the alternatives are referenced by an index \( k \in \{0, ..., K - 1\} \).
2. Calculate \( v_k \) for all alternatives \( k \) given attributes \( x \) and parameters \( \beta \).
3. For each \( k \), sample an error \( \epsilon_k \sim EV \).
4. For each \( k \), calculate \( u_k = v_k + \sigma \epsilon_k \).
5. Select \( k^* = \arg \max_k \{u_k \mid k \in \{0, ..., K - 1\}\} \).

While \( \sigma \) has a strong influence on the choice probability, it is usually not known. There is a procedure called “resampling” in which one would repeat the choice process for the reference data over and over again with different \( \sigma \) to find the one that provides the best fit with the probabilities

\[ [\cdot] \text{ being the Iverson bracket, i.e. it is 1 if the contained expression is true, otherwise its value is 0.} \]
observed from Equation 2.17. As this is a costly process, usually Equation 2.17 is used directly for sampling. Nevertheless, the notion of resampling shall be mentioned here as it will become relevant for section 2.4.2.

2.3 DISCRETE MODE CHOICE FOR MATSIM

The content of this section was partly published in two peer reviewed conference contributions:


Comparing the choice process of MATSim and the concept of Discrete Choice modelling, several parallels are evident. The approaches are initially based on the maximization of scores/utilities. While both describe the decision-maker as an entity that seeks the best possible alternative for a particular choice situation, discrete choice models acknowledge that there may be unmodelled taste variations and uncertainty by construction.

In the context of choice modelling, the multinomial logit formula can directly be derived from random utility maximization. In contrast, it is a computational vehicle in MATSim to occasionally select between alternatives (plans) that do not have the current highest perceived utility.

Because of this parallel, MATSim has often been conceived as making use of a multinomial logit model in the discrete choice sense, but it is not. The examples of Section 2.1.2 have shown that even though there is a multinomial selection process involved, the system dynamics are focused on finding the deterministic maximum utility. A considerable influence in this behaviour is the removal of the worst plan from memory. Furthermore, consider agents that only have one trip and three elements in memory. There are four different mode options. The choice at any memory state would be biased as at least one possible alternative would be missing.

In the project SVI 2016/001 surveys have been performed that allowed estimating a discrete mode choice model including future automated mobility modes in Zurich. From the considerations above, it became clear, that
parameters from the model cannot be transferred directly into the scoring logic of MATSim.

The first alternative to still make use of the data would be to calibrate the scoring parameters of MATSim, such that the simulation yields the same choice behaviour as estimated from the survey. Consider a set of scoring parameters $\beta$ for MATSim. Denoting MATSim itself as $M$ we intend to measure some value from the simulation (such as a more share):

$$y = M(\beta)$$  \hspace{1cm} (2.20)

As shown, MATSim is inherently stochastic, so looking at one single outcome is usually not of interest. Rather, the distribution $P(Y)$ should be looked at, or at least the expectation $E[Y]$, which can be approximated by a mean $\hat{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$ of $N$ outcomes.

The first aim would then be to reproduce the mode shares that are observed in the choice model. For that, one would need to consider all trips done by agents in MATSim, feed their attributes $X$ into the choice model $D$ and obtain calibration reference shares $y'_m = D(X)$. It is then possible to define a calibration objective, e.g. $J(\beta) = \|y'_m - E[Y]\|_2$, that should be minimized. This minimisation is a tricky task as multiple simulations are needed to estimate the expectation, each of them being costly.

In any case, it is not clear whether this procedure would lead to overfitting. By definition, the choice model is sensitive to changes in the attributes and those sensitivities are what researchers are interested in when performing simulations: What happens if costs increase by a certain percentage? What happens if travel time on a specific connection can be reduced considerably? Hence, we would additionally be interested in reproducing elasticities such as $\frac{\partial D_k}{\partial x_{\text{travelTime,car}}}$ in the first order and, ideally, also in higher orders (i.e. joint derivatives in multiple attributes).

Practically, this would mean introducing such changes to the simulation, feeding trip characteristics back to the choice model, obtaining reference shares and then making sure that the initial objective is minimised. Additionally, the offset between the elasticity values from the choice model compared to those estimated from MATSim must be similar. This process requires an almost impossible computational effort, not even taking into account that established automatic calibration tools for MATSim do not exist at the time of writing.

Here, we, therefore, explored the idea of explicitly integrating the discrete choice model into MATSim and, temporarily, deviated from the standard approach of scoring. These developments have lead to a new packaged-
up open-source extension for MATSim that is available online (Hörl and Balac, 2020a). The following sections describe this extension in detail and summarise the lessons learned.

2.3.1 Model structure

The extension has been set up in a modular way, such that every step in the process of generating agent decisions can easily be customised. In that sense, it has evolved from the initial goal of using a multinomial logit choice model in MATSim to a general framework to customise mode choice in MATSim. Figure 2.4 shows all components that compose the process.

The mode choice component itself is integrated into MATSim as an innovation strategy. As such, it is called with a certain probability whenever replanning is performed on an agent. The aim is to update the transport modes of all trips in their plan. Once the strategy is called on an agent, the choice process unfolds as is shown in Figure 2.4:

1. The relevant choice situations are obtained.
2. Choice alternatives for each situation are constructed.
3. Alternatives are dropped because of structural and mode-related constraints.
4. Utilities are estimated for all choice candidates.
5. One candidate is selected as the final choice.

These steps and their respective implementational components will be covered in detail in the next sections.

2.3.1.1 Definition of choice situations

In the first step of the process, all choice situations are obtained. One obvious option is to treat every trip in an agent’s plan as one choice situation in which a specific decision must be made. However, this approach carries a flaw as decisions of daily mobility are heavily correlated. Consider, for instance, somebody who goes to the bakery in the morning, which is one block away and then starts his or her daily commute to work, which takes about 30 minutes by car. From a trip-based choice perspective, choosing to walk to the bakery might be the best alternative. This decision would mean, however, that the agent does not have the car available once he or she is
Figure 2.4: Schematic information flow inside of the Discrete Mode Choice extension for MATSim.
out of the bakery. Logically, this alternative should then not be available anymore as the vehicle remained at home. As decisions are pre-planned for a whole day in MATSim, mode choices must, therefore, take into account future trips.

A usual approach is to consider tour-based choices instead of trip-based choices. A tour is defined as a trip sequence that starts and ends at the same location (for instance, at home). A choice situation is therefore not only asking to assign one single mode for a trip but a sequence of modes for a sequence of trips.

The default mode of operation of the Discrete Mode Choice (DMC) extension is to define choice situations as home-based tours. However, a trip-based or plan-based choice process is possible as well (though the latter may become inhibitively complex, as shown later). How a plan is segmented into tours (or choice situations) is defined by the TourFinder component. The standard implementation of it is hence the HomeBasedTourFinder.

In some cases, not all available choice situations for a plan should be considered. This may be the case if the modeller explicitly wants some elements of the plan to stay constant independent of mode choice. A good example will be shown later in this thesis, where we cut smaller scenarios out of bigger ones and want to keep mode decisions constant at agents’ points of entry into the smaller scenario. The relevant component is called the TripFilter or TourFilter.

### 2.3.1.2 Choice set construction

Once all relevant choice situations are known, alternatives for these situations must be generated. This process is also called choice set construction.

In the trip-based case, the process is clear: Given a set of $M$ different available transport modes, one alternative is created per mode for each trip. For tours with more than one trip, the choice set consists of all combinations of modes that are possible. Figure 2.5 visualizes this concept. There is one tour (choice situation) from the first to the fourth activity and another tour from the fourth to the sixth activity. While the first tour consists of three trips, there are $M$ choices for the first trip on that tour. Since these $M$ choices can be combined with $M$ different alternatives on the second trip, we have $M^2$ alternatives, and so on.

In general, a tour of length $L$ has $M^L$ potential alternative assignments of transport modes. This number can get very large quickly, so the extension provides the means of reducing this set. First, there is the ModeAvailability component. Given an agent (and its person- or even household-level at-
Figure 2.5: Trip-based versus tour-based choice situations in the DMC extension.
tributes) this component decides which modes are available. For instance, driving a car would never be provided as an alternative for agents that are only five years old.

Next, constraints are applied to the generated mode sequences. If a constraint is not fulfilled, the alternative is dropped from the set of options. Note that contrary to the choice situation filtering from above, here specific alternatives are filtered out, leaving all remaining alternatives available in the situation. Constraints can either be applied in the form of TourConstraints considering the whole chain of modes, or as TripConstraints which invalidate the whole alternative if they fail on any trip along the tour. Constraints have available all information on all trips of a tour, for instance, the types of all activities, and their planned locations.

The most important TourConstraint is the VehicleContinuityConstraint. It makes sure that certain vehicle-bound modes (such as private car and bicycle) are only assigned to trips in a way that if they are moved from one point along the tour, they are moved back to that point. For instance, consider a daily plan consisting of a trip from home to work, and then a trip back from work to home. In such case, the assignment of “private car” to both trips is valid - leaving the car at work or departing with the car from work while using another transport mode on the way there is not. The condition that a vehicle needs to be moved back to its initial position also covers cases such as a sub-tour to the restaurant in between two separate work activities.

The framework provides several general-purpose constraints. One example is the ShapefileConstraint, which makes sure that certain modes can only be used in places that are defined by a shapefile. Such shapefiles are commonly used in GIS (Geographical Information Systems) and define the extent of regions and areas. The extension makes it easy for the modeller to add custom constraints such as prohibiting the use of a transport mode at certain times of day if the agents have specific attributes and more.

Hence, the final size of the choice set for each tour alternative has an upper bound of $M^L$ but is commonly much smaller.

2.3.1.3 Utility calculation

Each alternative is then assigned a utility. For that, the attribute values $x$ of an alternative need to be obtained, which usually involves routing the trip on the road or public transport network or otherwise creating the necessary information to perform the choice. This process can become rather costly as the set of alternatives can be large at times. However, it is possible to
cache the provision of $x$ intelligently. Consider again Figure 2.5. The choice of public transport on the first trip of the first tour will be possible in many of the alternatives, but the governing characteristics (point of departure and point of arrival) stay constant, regardless of the subsequent modes. If this generally holds, each mode only needs to be routed once for each trip, i.e. one has $L \cdot M$ routings. This is important as routing is the most expensive step in the choice process.

The modeler can define custom utility functions $u_c(x, \beta)$ for each chain alternative $c$. Note that this is different from before as in Section 2.2, mainly trip-based models were discussed. On the other hand, it is difficult to consistently estimate tour-based models as they would be required here because tours vary in length. We propose to define the utility of a chain alternative $c$ as the cumulative utility of all trips

$$v_c(x, \beta) = \sum_{t=1}^{L_t} v_{c,t}(x, \beta)$$

with $t$ referencing the trips along chain $c$ and $v_{c,t}(\cdot)$ being the respective utility function for the chosen mode of trip $t$ on chain $c$. The specific utility calculation process is handled by the Trip/TourEstimator component.

After alternatives have been routed and assigned a utility, there is another chance for the Constraints to drop specific alternatives. Such constraints should be reliant on routing information. For instance, choosing the public transport alternative should be forbidden, if the router finds that there is no public transport at all in the area of the trip.

### 2.3.1.4 Candidate selection

At this stage, the process has identified choice alternatives $c$, which are potential candidates for selection. Each of them is assigned a utility $v_c$ that is dependent on the model parameters and characteristics of the alternative.

It is possible to set up various ways of performing the choice. Approaches can be flexibly implemented as a UtilitySelector component. The most simple approach is the RandomUtilitySelector. It does not even need a sophisticated utility model, as it does not take $v_c$ into account. Instead, it uniformly samples one of the $C \in \mathbb{N}$ available alternatives $c$:

$$c^* \sim \text{Cat}([1, \ldots, C])$$

Another possibility is to select the candidate with the maximum utility, which is implemented as the MaximumUtilitySelector:
Finally, the \texttt{MultinomialLogitSelector} is available, which selects one candidate by sampling it from the distribution of available alternatives following the procedure introduced in Section 2.2.3.

The selected alternative is then implemented into the agent plan by updating the mode attributes of every relevant trip. If routes are passed along with the $v_k$ from the estimation step, they are implemented directly into the plan. Otherwise, a re-routing happens after the update. In the end, the agent plan consistently carries new information about the modes and routes of all processed trips.

\subsection*{2.3.2 Model compatibility}

For the remainder of the thesis, the usual set-up of MATSim with the DMC extension is as follows: MATSim is configured to keep only one plan per memory and scoring is deactivated. Whenever replanning happens, with a probability of $\zeta \in [0, 1]$ mode choice is performed for an agent; otherwise, the previous plan is kept constant. DMC is configured to use multinomial logit selection.

In essence, this transforms MATSim into a traffic simulator (providing information such as travel times or congestion levels) that is called in a loop with a discrete mode choice model. The principle is that attributes such as travel times or waiting times for automated taxi services can be measured from one iteration and used as the basis for mode choice in the next iteration. By performing this loop long enough, decisions stabilise. However, they have the variability and tradeoffs as they can be observed in the survey data and in the estimated choice model.

It is interesting to discuss how far parameters from the choice model can be directly used in this set-up. While the process resembles the dynamics of the choice model strongly, one difference is that a tour-based choice is performed instead of a trip-based choice like in the survey. For that, it is interesting to understand which effect summing up trip-based utilities has. While a much more analytical and detailed analysis would be possible and should be carried out in the future, some hints and practical experiences will be presented in the following.

From the theoretical introduction above, it can be stated that the trip utilities $u_t = v_t + \epsilon_t$ are random variables that follow certain distribu-
tions. Denoting the tour utility as \( u \), we therefore intuitively claim in our implementation that

\[
   u = \sum_{t}(v_t + \sigma \epsilon_t) = v + \sigma \sum_{t} \epsilon_t .
\]

Assuming that \( \epsilon_t \) are i.i.d. standard Gumbel-distributed, the question becomes what the distribution of \( \sum_{t} \epsilon_t \) is? It can be shown that the sum of multiple i.i.d. Gumbel variables is not Gumbel. Hence, using \( v \) in the multinomial logit formula is analytically not valid.

While the multinomial logit model is a commonly used model in choice modelling, there are alternatives, such as the multinomial probit model. In this model formulation, the \( \epsilon_t \) are not EV-distributed, but follow a standard normal distribution \( \epsilon_t \sim N(0,1) \). In such a case, the sum of \( N \) multiple standard normally distributed \( \epsilon_t \) would follow a \( N(0,N) \) distribution. A couple of conclusions can be drawn from this observation. Often, EV errors are understood to be helpers to make random utility maximisation models analytically feasible and to approximate a normal random variable. Therefore, summing up utilities may not be a horrible idea after all from that perspective.

The example of normally distributed errors shows nicely that the scale of the error changes by applying the sum operation. Intuitively this makes sense the more trips per tour there are the more uncertain we are about what would be the selected option as many different decisions are involved. However, it also makes intuitive sense that decisions along a tour are correlated in reality. If a person uses the car to go to work in the morning, one can be almost entirely sure that the person will also take this car to go back home in the evening. Hence, in such a tour, one would not observe “double uncertainty” about using the car.

Practically, the model formulation, as presented above, has proven to be a valuable tool in our simulations. As will be shown later, the summed up utility functions of a trip-based multinomial logit model used in a tour-based selection lead to excellent model fits in simulation with only little further calibration. While there is no rigorous proof, we believe that the combination with vehicle-continuity and other constraints provides the “missing bit” of certainty. While it exceeds the scope of this thesis, further, more analytical, work should be pursued in this direction.
2.3.3 Further use cases

Apart from the projects covered in this thesis, the DMC extension has been successfully used in a couple of research projects around new mobility modes since it has been developed.

- Becker et al. (2020) assess the welfare impact of shared mobility and Mobility as a Service (MaaS) in Zurich by extending the model from SVI 2016/001 with ride-hailing and other services.
- Balac et al. (2019a) model multiple free-floating carsharing operators in Zurich by integrating car-sharing into the discrete mode choice model.
- Balac et al. (2019b) use the DMC extension to estimate the demand for Urban Air Mobility in Switzerland.

Furthermore, the extension has recently been updated to (experimentally) support nested logit models.

2.4 Discussion and lessons learned

While our approach of using discrete mode choice models in MATSim has now been used in many use cases and studies, the approach had an experimental component from the beginning. Accordingly, developing the method was also a process of discovering the advantages and disadvantages of the approach. Notably, comparison with the conventional way of decision-making in MATSim was a continuous goal. The following section will draw a comparison between the two approaches. Afterwards, a potential path of combining them, based on our lessons learned, will be discussed.

2.4.1 Approaches in comparison

It needs to be acknowledged that the focus has always been on mode choice in the specific simulations of this thesis and all accompanying research. Hence, the DMC extension has not been adapted to allow for departure time or destination decisions. While there are potential pathways, looking into them in detail remains for future research. The overarching advantage of the MATSim co-evolutionary approach is that it is impressively flexible. Even if another choice component - for instance, synchronisation between household members (Dubernet, 2017) - becomes of interest, all that is
required is to find a process that can randomly create proposals for the choice of interest and to define how to score it.

Likewise, the approach of MATSim is straightforward to implement. While the DMC component has been well-designed code-wise and refactored multiple times such that setting up models has become increasingly accessible, there is still added complexity compared to MATSim’s scoring. Especially, predicting values for specific attributes is not trivial in many cases (as will be shown for predicting automated taxi waiting times further below). One of such attributes is the detour factor experienced in ride-hailing services. While MATSim can naturally score out plans in which rides with long detours occur, a DMC model would need to predict them. Due to the potentially high spatial and temporal variability, not only depending on the origin but also the destination, predicting such rides poses a challenge with DMC. On the other hand, one may argue that choices should be based on the “worst-case” that usually needs to be communicated by the operator.

One disadvantage of MATSim is that there is no established theory on setting or calibrating the parameters. While it is always possible to achieve a precise fit with reference data by fitting scoring and other model parameters, the question remains open which predictive ability such a calibration procedure provides. Now that large-scale mobility data sets become available, it would be interesting to see MATSim simulations that are jointly calibrated for two or more states in time. A compelling use case would be, for instance, if GSM or GPS traces were available before and after construction work.

Furthermore, “innovation turn-off” needs to be performed in MATSim. Otherwise, a share of agents would always try random travel choices which bias all others. The way we use the DMC extension, no innovation turn-off is needed as the model replicates the variability that is observed in the survey data and thus provides a realistic level of variation.

For the choice models, traveller behaviour is derived directly from variations in attribute values (primarily when they are based on stated preference data) without the need for a costly calibration process. As our experiments have shown (see below), only little fine-tuning is necessary when used in our agent-based simulations. The choice models allow for a tremendous speed-up when setting up new simulations. Lastly, a discrete choice model from stated preference data can take into account future services. These services can reach from a train line that has not been built yet up to a future automated taxi service. While the credibility of such models depends
heavily on how respondents have been introduced to the alternatives and how the survey is set up, tradeoffs between new and traditional alternatives are factored in naturally. In a standard MATSim simulation assumptions on those future alternatives must be made. While it is challenging to find scoring parameters that fit (or, even overfit) today’s state of the traffic system, it is not possible to fit parameters to something that is not there yet. Usually, case studies would assume parameters from today’s modes that are similar to the future alternatives or adjust parameters relatively, with the sensitivities found in discrete choice models. Additionally, sensitivity studies would be performed.

To summarise, developing the DMC extension was a valuable decision as it allowed us to directly make use of the choice models that have been estimated from survey data. Nevertheless, we acknowledge that this is a specific use case as we do not consider changes in departure times or locations. While we strongly encourage further research and experiments with the flexible and versatile co-evolutionary algorithm of MATSim, we want to underline that it is not the case that parameters from an arbitrary discrete mode choice model can be used inside of MATSim right away without carefully validating the outcomes.

2.4.2 Fixed error terms

While the differences between the co-evolutionary algorithm and discrete mode choice have been discussed, there might be a way to couple both approaches with each other. No experiments in that direction have been performed yet; hence the following shall be seen as an outlook and theoretic formulation of the approach.

As shown, discrete mode choice is initially based on the idea of random utility maximisation. Also, the purpose of the co-evolutionary algorithm is to maximise the overall utility of all agents in the population. The big difference between both approaches is that the discrete choice approach explicitly models an uncertainty about choices such that agents are likely to choose options with a high utility, but sometimes may also favour another alternative. This uncertainty can be explained by indifference about options (i.e. non-substantial difference in utilities), varying tastes across the sample of respondents, and elements that govern the decision process but have not been included in the model.

In the co-evolutionary algorithm, agents always are driven to use the best possible option. However, what if also their options were affected by offsets
to their score that reflect the variability of choices? The idea would be to model these errors explicitly in MATSim. For every trip, one would add an error term $e(\cdot)$:

$$S_{\text{Person,Index,Mode}} = \beta_{\text{ASC,Mode}} + \beta_{\text{travelTime,Mode}} \cdot t + e(\text{Person, Index, Mode})$$ (2.25)

This error term is deterministic given a combination of person identifier, trip index in the daily plan and chosen mode. Apart from that, the process would be as before; one would run the co-evolutionary algorithm to find the best user equilibrium possible. However, assuming an uncapacitated scenario as shown before, not all agents would choose the alternative with the (deterministically) best score. However, their best plan would be affected by the error terms. Hence, given the right way of producing those error terms, one would be able to construct the same outcome (mode share in our case) as a discrete choice model on the population level. For that, two conditions need to be met:

1. Looking at all possible trip alternatives of all agents, the process that generates $e(\cdot)$ must make sure that these values are distributed according to the underlying distribution of the choice model. It may be an EV distribution for a multinomial logit model or a normal distribution for a multinomial probit model. Presumably, this is not an easy task. The challenge is to maintain a distribution of errors that is close to the desired distribution. At the same time, one needs to make sure that errors are not correlated with each other. Good starting points for further research on the topic would be the theories on hash functions and pseudo random number generators (PRNGs).

2. The scale of the error needs to be adjusted. As described above, multinomial logit models are usually estimated without knowing the scale of the error. While the scale may be resampled with survey data, it could become another calibration parameter for MATSim.

Such an approach could make it indeed possible to directly transfer utility parameters estimated from surveys or travel diaries into MATSim. The method could potentially remove much calibration work and would allow MATSim users to operate more consistently with “new” modes of transport.
AUTOMATED MOBILITY SIMULATION

The introduction to this chapter is based on a peer-reviewed conference contribution:


In the thesis one goal is to model a realistic interplay between supply and demand for Automated Mobility on Demand. Furthermore, the idea is that the way a fleet is operated would govern the service level and the price at which rides can be offered. For that purpose, one crucial element is a component for MATSim that simulates a fleet of automated taxis dynamically controlled by a central unit inside of the simulation. While a versatile and easily configurable extension to MATSim has been developed during the project, other solutions and related studies exist.

One of the first large-scale analyses was performed by Spieser et al. (2014) for Singapore. The study finds that the whole transport demand of the city could be covered by one-third of today’s vehicle fleet if it entirely would consist of automated single-occupancy vehicles.

Subsequently, a series of studies have been performed for the case of Austin, Texas. In (Fagnant and Kockelman, 2014) a grid-based simulation for the city is introduced. For an artificial demand based on real-world trip generation rates and randomly assigned destinations, it is found that Austin’s demand in private car trips could be served by an automated vehicle fleet that is reduced by 90% compared to today. The use case is further extended by Chen et al. (2016), who assume electric charging infrastructure. Further studies introduce a more detailed demand for the scenario, based on static trips from the regional household travel survey (HTS). Levin et al. (2017) introduce congestion to the simulation and find that this has a substantial impact on fleet size. Liu et al. (2017) first consider the demand side by applying a choice model, though only in a post-processing step. Based in an approach that had been previously applied for Zurich (Bösch et al., 2016) they feed a discrete choice model with information about travel and waiting times to analyze potential mode shares.
in Austin. The utility function is entirely based on literature; no model is estimated from stated preference (SP) data. Finally, Fagnant and Kockelman (2018) extend the Austin case with a ride-sharing component and find that pooling could reduce wait times for the customers and mitigate the increase of VKT due to empty rides. However, customer preferences for the service are not taken into account.

For the case of Berlin, Bischoff and Maciejewski (2016b) use a static travel demand from the regional HTS to create a MATSim simulation with automated taxis. All car trips within the city boundaries are replaced by the service, leading to a scenario where one-tenth of all vehicles replace the existing fleet if every agent in the simulation was to use the service with acceptable wait times. In the same scenario, Bischoff and Maciejewski (2016a) find that also allowing public transport users to use the service leads to a linear increase in needed fleet size. Finally, Maciejewski and Bischoff (2017) introduce congestion to the simulation showing that without significant gains in road capacity due to automation, a fleet of automated taxis serving all of the city’s demand would worsen congestion dramatically. Naturally, agent-based simulations are suited to analyze availability of limited infrastructure resources (such as parking space, Axhausen (1989)). Simulations of automated taxis with limited parking capacity have shown (Bischoff et al., 2018b), that automated vehicles are likely to mitigate parking search problems in the city.

Simulations based on Aimsum have been performed for Munich (Dandl et al., 2017; Dandl and Bogenberger, 2018) where HTS data and real-world car-sharing data is used to establish a demand that is to be covered by automated taxis. In a follow-up study the addition of ride-pooling is analyzed (Engelhardt et al., 2019). The authors show that while low demands lead to an increase in VKT, a demand from at least 5% of today’s car trips leads to reduced VKT. However, the study finds that the potential for pooling more than two people is low.

Hyland and Mahmassani (2018) present a simulation of automated taxis with stochastic taxi demand patterns for Chicago. They compare advanced linear programming fleet control strategies with simpler heuristic approaches. The results indicate that valuable gains in fleet efficiency and service level are possible at low cost by applying more intelligent algorithms. The framework is extended with with two-person ride-sharing capabilities in (Hyland and Mahmassani, 2020). The results for New York taxi data show that even small detours of 5% of direct travel distance allow for substantial improvements of wait times and fleet cost. Both studies
are simulated on a Manhattan grid for the respective use cases without considering congestion or customer behaviour. Dandl et al. (2019b); Hyland et al. (2019) add demand forecasts to the New York model and analyze the impact of different spatial and temporal demand resolution levels on system performance. While forecast quality decreases with finer resolution it is shown that fleet performance is improved. This track of research is further extended with a dual-horizon planning approach to take into account day-to-day dynamics (Dandl et al., 2020). Furthermore, the framework is applied in a context where operators may perform dispatching decisions although users may not have accepted a certain offer, thus replicating the uncertainty of operator-customer interaction in the real world (Dandl et al., 2019a).

Several studies exist for the French context. Poulhès and Berrada (2017) propose a simulation framework of automated campus busses in Palaiseau close to Paris. Demand is static and based on trip-generation rates at well-defined spots in the system. The simulation is extended in (Berrada et al., 2019), where a choice model is introduced to establish a dynamic demand. Utilities for the choice model are based on literature values.

The first simulation with automated vehicles in Paris is presented by Kamel et al. (2018). Based on census and HTS data the characteristics of a static fleet of 17,000 vehicles are examined serving a detailed dynamic demand based on a classic MATSim simulation. No SP data is used, but a heuristic that, based on literature, models the attractivity of an automated taxi service for different sociodemographic and income-based groups. As a first study in the field, the impact of such a fleet on sociodemographic groups is studied in detail. The framework is further applied to the city of Rouen (Vosooghi et al., 2019b), where a fleet sizing given diverse survey-based user group preferences is performed (Vosooghi et al., 2019a). It is found that user preferences may have a substantial impact on the utilization of such a service. Subsequently, the framework is used by (Vosooghi et al., 2020) to assess the impact of varying electric charging infrastructure placement.

Martinez and Viegas (2017) introduce a simulation platform for Lisbon, where HTS data is used statically, but where the choice for various automated mobility services is determined by a heuristic model based on expert opinion. A dynamic demand approach is shown in (Wen et al., 2018) for a not further specified European city. Not using an agent-based transport model, the simulation is based on trip generation rates, which are in turn dependent on a discrete choice model that is fed with travel and
waiting times from a previous simulation of the generated trips. Hence, an equilibrium of supply and demand can be achieved.

Following the path of jointly simulating user behaviour and AMoD, the AV extension for MATSim has been developed and released as open-source software (Hörl, 2020). An introduction will be given in the next section. Afterwards, AMoDeus, a flexible simulation platform to assess the performance of fleet control algorithm, will be introduced. Based on the AV extension, it has been used for various research projects around the world.

3.1 AUTOMATED VEHICLES IN MATSIM

The development goal of the AV extension for MATSim (Hörl, 2017) was to create a toolkit to implement automated vehicle services in a comfortable and feature-rich way. The idea was to simulate each automated vehicle individually as an agent that drives to pick up and drop off people. At the end of the simulation, this allows measuring a multitude of relevant metrics on the service, for instance, empty distance and occupancy rates.

The big difference between the automated vehicle agents and traveller agents in MATSim is that the latter have predetermined plans when a new day is simulated. From the replanning phase, all activities and trips are known and are just executed one by one in the simulation. For the automated vehicle fleet, though, it is necessary to control the agents time step by time step to resemble the operational side realistically. A real operator would not create fixed plans for all vehicles but react flexibly to incoming requests which are not known a priori. Therefore, this also needs to be possible in simulation.

For the simulation of dynamic (supply side) agents, an extension for MATSim had been developed by TU Berlin (Maciejewski et al., 2017). While it is called DVRP (Dynamical vehicle routing problem) its functionality has become more versatile and universal rather than just being focused on DVRPs. The basic structure is as follows. Each of the dynamic agents has a schedule, which somewhat follows the concept of daily plans of conventional agents. However, these schedules are comprised of tasks. These tasks can be flexibly defined by the programmer and represent a sequence of instructions to the dynamic agent.

During the simulation, the agent goes through these tasks step by step. Whenever one task of the schedule is finished, the next one is started. A component called ActionCreator translates whatever the task is into a dynamic version of either an activity or a leg as they are commonly used in
MATSim. The difference is that those dynamic activities are queried every second to determine whether they should be ended (for instance, to check whether all passengers are on board). Vehicles can be diverted while they are on a dynamic leg.

The schedules of dynamic agents are constructed and updated by the optimizer. This optimizer has knowledge about all vehicles and their state, and it receives all customer requests to process them further. The bookkeeping is one of the main tasks of the DVRP extension, to track whenever a traveller agent wants to depart from an activity with the mode of transport that is associated with the dynamic fleet. In that case, the traveller agents are kept at their location until they are picked up, and they will be passed back to the activity simulation once they are dropped off.

The AV extension makes use of the components of the DVRP extension and adds further elements to make it easy to define a custom automated mobility on demand (AMoD) service. The main components are:

The **Generator** defines where (and how many) AMoD vehicles are located in the network at the beginning of the day. Default implementations are random distribution in the operating area, and distribution according to population density.

The **Dispatcher** defines the operator behaviour in terms of assigning vehicles to customer requests and micro-managing the movements of the whole fleet in terms of modifying their schedules. It builds directly on the optimizer from the DVRP package.

These two structures are defined *per operator*. Each of these operators can easily be defined through the MATSim configuration file. Additional components that can be controlled via configuration are how trips with the AMoD service are scored (including wait time perception and pricing), and how they are routed, including information on predicted waiting times.

Another essential component is the operating area, that can be defined per operator. This means that not always the whole MATSim scenario must be served, but that fleet operation can be restricted to specific regions. These can either be defined by customized rules or through shapefiles, which are commonly used in GIS (Geographical Information Systems).

The schedules of all AMoD vehicles then look as shown in Figure 3.1 if they are operated as single-occupancy taxis. Initially, they are in Idle state. Eventually, the dispatcher can send them on an empty Drive task to go to some location in the network. This task is translated into a dynamic leg performed on the road network. While empty movements are possible,
which lead to a new Idle activity, the Drive task can also be followed by a Pickup task, which carries information about which agent should be at the specified location. The vehicle will then wait at this location until the passenger has arrived. Afterwards, a new Drive task is executed, leading to a Dropoff task and potentially another Drive or Idle task. Durations of pickup and dropoff activities can be flexibly defined in the framework.

On the operational side, the logic would look as follows. At some point, the operator receives a request, which is saved in a list. As the operator also knows which vehicles are available, it would pick one of those vehicles and send it to the customer, by adding a chain of Drive - Pickup - Drive - Dropoff tasks to the vehicle schedule. The AV extension makes it easy for programmers to define and implement their dispatching and assignment strategies. The default one is the one by described by Bischoff and Maciejewski (2016b). Every $N$ seconds, the following logic is executed:

1. A list $V$ of all idle vehicles is obtained.
2. A list $R$ of all unmatched requests is obtained.
3. If $|V| \geq |R|$ the situation is called over-supply as more vehicles are available than requests. In such a case, the algorithm would go through all requests and assign the closest vehicle. This leads to requests being served in the whole operating area.
4. If $|V| < |R|$ the situation is called under-supply as vehicles are scarce. The algorithm would then loop through all available vehicles and assign the closest customer requests. This may lead to longer waiting times in remote areas but helps to process the high demand efficiently.
Furthermore, an extension of this approach is available by default that considers point-to-point pooling. In this algorithm, the dispatcher would check every incoming request against a list of already existing “primary” requests. If one of these primary requests has an origin and destination that are within a distance of only $\theta_D$ of the new request, respectively, the new request is assigned as a “secondary” request to the primary one. Otherwise, the new request is registered as a new primary request. Whenever the dispatching period $N$ is reached, the algorithm from above is executed on the primary requests. The major operational difference is that the vehicle remains at the location of the primary request until all other persons have arrived. Additionally, it rests a minimum time $\theta_T$ to wait for potential future requests that are also compatible. This way, trips are pooled in an origin-destination manner. A comparison of the algorithms and a first feasibility study of the framework has been presented in (Hörl, 2017).

It should be mentioned that since the development start of the AV extension, the DVRP extension has been further applied at TU Berlin in their DRT (Demand Responsive Transit) extension (MATSim contributors, 2020). While the two implementations had a very different feature set at the beginning of development, they have converged strongly towards each other in terms of functionality. Soon, both approaches should be merged together.

3.1.1 Compatibility with discrete choice

It is relevant to think about how this simulation component can be made compatible with discrete mode choice. The main components of the utility function of the AMoD transport mode are travel time, waiting time, and price. These attributes must be predicted for every trip.

For travel times, the trip can be routed through the road network. This serves as a predictor for the actual drive time in the simulation and works the same as, for instance, private cars: A Dijkstra or A* algorithm is performed on the road network to obtain the overall travel time of a trip, based on per-link and per-time-bin travel times from the previous iteration.

For prices, the standard functionality of the AV extension considers the routed distance and applies a fixed price per kilometre and a base fare to it. These can be defined via the configuration file. However, the detailed cost model from Bösch et al. (2018) was used. Hence, a new price structure was implemented that dynamically takes into account passenger vs empty distance from the previous iteration, fleet size and occupancy of the vehicles. In the end, this still translates to a base fare and a price per kilometre for
the customer, although the underlying process is more complicated than before. Usually, to smooth out large demand-dependent changes in prices, they are averaged by calculating the current estimated price $p_i$ in iteration $i$, but then applying the smoothed price $p' = \frac{1}{H} \sum_{i-H}^{H} p_i$ to the simulation with $H$ being the averaging horizon.

The most interesting component regarding DMC is the waiting time. The challenge is to predict the waiting time of a trip for which the origin location and departure time is known. Hence, the prediction should be spatially and temporally detailed as demand and supply patterns differ across regions of the operating area and between different times of the day. For that, the operating area is covered by $N_S$ spatial zones and the day is divided into $N_T$ time bins. It is then possible to track all AMoD trips during one simulated day and capture two numbers: The number of trips in spatial bin $z \in \{1, ..., N_S\}$ and temporal bin $t \in \{1, ..., N_T\}$ in iteration $i$ as $n_{i,z,t}$ and the cumulative waiting time of all such trips as $T_{i,z,t}$. For one iteration it would then be possible to calculate the mean for one combination of spatial and temporal bin as $T_{i,z,t} / n_{i,z,t}$. However, experience has shown that this way of estimating waiting times can become extremely unstable, especially in zones and at times of low demand. The mean could then be comprised of only one or two observations, and outliers can have a strong influence. While one approach would be to define zones and time bins adaptively, another approach has been chosen here. An estimation horizon $H$ is defined to gain more stable means, and the values of previous iterations are taken into account:

$$T''_{i,z,t} = \frac{\sum_{i-10}^{i} T_{i,z,t}}{\sum_{i-10}^{i} n_{i,z,t}}$$

(3.1)

Based on these numbers, waiting times are estimated for the discrete mode choice models.

Another component for the interplay with discrete mode choice are constraints. The most important one is the operating area constraint that only allows the DMC component to even consider a trip with AMoD if the origin and destination of the trip in question are within the operating area. Note that this is different from defining the operating area as the entirety of roads where the service can drive as explained above. Furthermore, the DMC component allows to flexibly pose additional constraints such as that the service should not accept trips under a certain distance.
3.1.2 Further components

A multitude of additional features has been developed in the framework of this work and related projects. Unfortunately, many have not been documented in literature yet as they were driven by projects that did not leave much time to process the results scientifically. While the functionality is already available, studies should be performed in the future.

First, MATSim provides the concept of flow efficiency factors per vehicle type. Such a factor says that in traffic simulation, a vehicle with a factor 2 is treated as if only half a vehicle would have traversed the link. In that sense, the factors represent the inverse of what is widely known as a passenger car unit although only in terms of flow, not in terms of space used by the vehicle. For SVI 2016/001 and a follow-up project, this functionality has been extended to flexibly allow the modeller to define such factors dependent on the combination of vehicle and link. This made it possible to define different flow efficiency factors depending on whether an AMoD vehicle is operated on a highway or a residential road.

Second, the extension has been used with first/last mile simulation. For that, a combined AMoD feeder mode has been defined. When a trip is routed, a set of potential origin and destination transit stations is created. In ongoing experiments, those are restricted to the closest station from the origin/destination of the trip, per "rail" line (i.e. no bus stops). It is then possible to perform routing with the AMoD service to and from any of these stations and find the best public transport connection between them. By minimizing any combination of access and egress trip and public transit connection in terms of total travel time, one specific route can be chosen. This process has been applied in a project that assesses the value of AMoD in a Swiss urban environment.

Third, spatial constraints have been implemented for AMoD simulations, but have now been generalized to become a component of MATSim. It is now possible to define in detail how vehicles are processed when they intend to exit the road network at a link, i.e. what happens before they can start their following activity at that location. Parking and pickup/dropoff constraints have been implemented using this functionality. For pickup/dropoff constraints, the subject of research is spillback effects at heavily frequented locations. This makes the component especially attractive when used in combination with feeder services as train stations would be such spots of interest. Also, studying parking capacity is a relevant subject as the infrastructural availability of parking space may strongly
impact the efficiency of fleet operation. A related study in the AMoDeus framework (see below) has recently been conducted.

3.2 THE AMODEUS FRAMEWORK

Based on the AV extension, a fruitful interaction with the Institute for Dynamical Systems and Control (IDSC) at ETH Zurich has been started. The result of this collaboration is a continuously improving framework to study fleet control. The aim of the framework is thereby to give control engineers easy access to the simulations. The idea is that they can focus on implementing the control algorithms in a terminology that is close to the one of their field. The code has been published under the name AMoDeus (Ruch et al., 2020) as open-source software.

The use cases are supposed to be somewhat simpler than those for a full MATSim simulation. Usually, only static demand is considered, which means that only one “day” is simulated. During this day, only the requests for the AMoD service are relevant. Hence no detailed synthetic population is necessary (although it can be used). Vice-versa, the framework is structured in a way that all contained algorithms can be used in a full MATSim simulation as well.

Often, this is not practical, because one of the research goals of AMoDeus is to provide a platform on which most of the fleet control algorithms known from the literature are implemented. By being able to run all of them in the same simulation platform, the framework intends to provide a way to compare these algorithms consistently. However, most of these algorithms cannot be easily used in a full MATSim simulation as only simulating one day in medium-sized scenarios may take many hours or even days. The reason for that is that many algorithms from the literature are based on iteratively solving hard optimization problems and are not designed for fast runtime performance.

Currently, the ever-growing list of implemented algorithms contains:

- Global Bipartite Matching Policy (Kuhn, 1955) (see (Maciejewski et al., 2016) in the context of AMoD dispatching)
- SQM algorithm based on Pavone et al. (2010)
• Demand-supply-balancing dispatching heuristic (Bischoff and Maciejewski, 2016b)

• First Come First Served Strategy with Grid Rebalancing based on Fagnant et al. (2015)

• Feedforward-time-varying rebalancing policy based on Pavone et al. (2012)

• The +1 Method (Ruch et al., 2019)

• DFR algorithm (Albert et al., 2019)

• Algorithms from Arsie et al. (2009)

While the algorithms mentioned above are for single occupancy taxis, AMoDeus has recently been extended for the simulation of pooling services. The implemented algorithms are:

• A pooling-extended version of the approach from Bischoff and Maciejewski (2016b)

• Dynamic Ride Sharing Strategy by Fagnant and Kockelman (2018)

• T-Share (Ma et al., 2013)

• High-Capacity ride-sharing algorithm by Alonso-Mora et al. (2017)

3.2.1 Aspects of fleet control

In the context of AMoDeus, those algorithms are described as fleet operational policies. This term shall reflect that fleet control is not only about dispatching or vehicle-customer assignment, but covers a large variety of different, complementary and interlinked problems and questions.

The standard task of a fleet operational policy is vehicle assignment. The idea is that the operator receives requests and vehicles need to be sent to those requests. Determining which vehicle to send to which request strongly influences the waiting time experienced by the customer.

However, many times throughout the day, vehicles are not used because the fleet must be sized for peak hours with high demand. Then the question arises where to send empty vehicles. This component of a fleet operational policy is usually called rebalancing or relocation. Again, the aim is to improve the customer experience by already having a vehicle readily available when
the future demand pops up. A very successful rebalancing strategy can even be seen as an important part of the assignment policy as it would already send a vehicle to a location before the request has even arrived. In such an idealized case, the service would produce zero waiting time. Note that while a low waiting time is beneficial to the operator as customers are attracted, rebalancing induces additional costs because vehicles are moved empty.

Likewise, a parking strategy may strongly influence the service. While parking can be seen similarly as customer assignment, because vehicles need to be assigned to a finite set of possible destinations, it is further restricted by infrastructural constraints. On the one hand, the operator needs to assign parking locations intelligently such that no vehicle arrives at a full parking lot and has to add more costly empty distance. On the other hand, parking dispatching should be intelligent enough to optimally use the available infrastructure and not send vehicles far from where the demand arises. Also, note how parking is strongly connected with rebalancing as an excellent parking strategy may, at the same time, be an excellent rebalancing strategy and vice versa.

Another popular topic is ride-pooling. It is interesting as it has the potential to make the system more efficient when two or more people are on very similar routes in terms of time and space. However, the challenge is to not “over-optimize” pooling algorithms such that the induced detours lead to a decline in the efficiency of the system. While for the case of single-occupancy there are clear objectives for optimization (minimization of waiting time while keeping the empty distance constrained, or the other way round), the trade-off becomes more complicated, especially if vehicle sizing becomes part of the equation. As the decision of operating larger or smaller vehicles can not be made on a second-by-second basis, the control algorithms must be aware of the availability of different vehicle types. The assignment process then becomes considerably more complicated as not only the question of which vehicle to send to which request must be answered but rather which vehicle of which size should be sent to which set of requests.

Finally, routing plays an integral part in fleet operation, especially when we imagine a future where an AMoD operator serves a substantial amount of a city’s mobility demand. In most simulations (also in the standard version of AMoDeus) vehicles are routed on per-link and per-time-bin predicted travel times through the simulation. However, if 1,000 vehicles are routed for the same departure time, it is not taken into account how
many vehicles are routed through the same links. Therefore, it is possible, that the AMoD system causes congestion on itself. At latest at this point, it becomes necessary to think about more intelligent routing strategies in AMoD systems that evenly distribute traffic in the system (see, for instance, Levin, 2017).

3.2.2 Use cases

As shown, fleet operation consists of many different components and continues to be a rich field of research with many open questions. During the past years, a couple of interesting studies using the AMoDeus framework have been performed, which shall be presented in the following. Another example is presented in Chapter 5 in the context of the AMoD simulations for Zurich.

Sieber et al. (2020) consider the replacement of rural train services by AMoD systems. For that, several case studies in Switzerland are identified. These cases are based on an assessment of regional train lines that are heavily subsidized. They represent connections of multiple small communities in spatial situations that resemble valleys where not many lateral public transport connections exist. In all cases the subsidized line is (besides cars and active modes) the only option to move between these communities. The study found that AMoD services can be a viable replacement for these train lines and offer better service levels at lower costs for the operator. However, not all use cases show a positive analysis result for the AMoD service. Mainly, connections with low demands may be a natural use case for these services.

Ruch et al. (Under reviewa) look at how parking influences operations of an AMoD system. The simulation is carried out for Zurich with three different scenarios of parking availability: a uniform distribution across all roads in the city; the available parking spots in Zurich; and the parking facilities of an existing car-sharing service. Especially the latter analysis is interesting as car-sharing services might be candidates for transforming into AMoD operators in the future with dedicated infrastructure already in place. The demand of Zurich was then served with a specific fleet size, and parking spots were sampled from the three parking spot distributions to arrive at scenarios with different vehicles to parking spots ratios (starting from 1 spot per vehicle and more). Three strategies for parking search are proposed: one that searches the network randomly until the first free parking spot is found, one that has local knowledge about free spots in the
direct vicinity of the vehicle, and an intelligent strategy that has knowledge about all available parking spots in the whole operating area. The results show that the latter strategy is able to park vehicles with comparably little cost in terms of empty distance while only leading to small increases in waiting time.

A Master’s thesis (Lu, 2019) looks at the effect of intelligent routing in large AMoD fleets. The thesis considers several strategies from metering vehicles going into specific zones of the city (while keeping waiting times bounded) to reduce congestion. Also, vehicles are routed such that they do not slow down each other. The main findings of the thesis shed new light on the fear that automated vehicle fleets may worsen congestion because they would be driving slower, there would be more vehicles on the road at any time of the day. The simulations show that intelligent coordinated routing of AMoD vehicles could mitigate congestion. Experiments with an AMoD fleet in co-existence with selfish utility-maximizing car users even show that small shares of automated vehicles may lower travel times for everybody involved.
Parts of this chapter are based on the final report for the project *SVI 2016/001:*


The previous chapters have covered the simulation components necessary to set up an AMoD simulation in MATSim. While those are valuable outcomes of this research, gathering and processing the data needed to set up a simulation is equally important and challenging. The following describes the process of constructing a *MATSim scenario* in detail. Note that in the following a *MATSim scenario* describes the entirety of data that is used to run a simulation in MATSim. This is different from a *policy scenario*, which describes different parameter configurations for a simulation to inform policymakers and transport planners on different simulation outcomes.

A MATSim scenario for Switzerland had already been developed before this work. Unfortunately, the project showed that the previous scenario had been set up, but never tested and validated in the actual simulation. Further problems that occurred while trying to reliably replicate the process to create the scenario lead to the decision to re-implement the synthesis process in a replicable way. The process presented here is therefore partly a re-implementation, partly an improved version of the scenario developed by Bösch et al. (2016). On the implementational and software design side, the focus has been put on setting up a pipeline that synthesizes a MATSim scenario starting from raw data up to a final validated simulation. The synthesis pipeline has become so versatile that various other MATSim scenario pipelines for Sao Paulo, Jakarta, Los Angeles and San Francesco have been set up in the meantime with it. Especially one scenario for the case of Île-de-France has seen active development and substantial further improvement over the past year. As the pipeline for Switzerland is likely to be updated with the lessons learned from the French case soon, only a brief overview of the pipeline components will be given in the following.
The first section of this chapter gives an overview of the synthesis process, including an introduction to the structure of a MATSim simulation, available raw data sets in Switzerland, and the algorithms used. Next, a new cutting method for MATSim scenarios is introduced, which makes it possible to obtain a MATSim scenario for Zurich from the larger Switzerland scenario. Afterwards, additional simulation components are described that have been developed for the simulations, followed by a section on calibration, simulation and validation of the Zurich scenario.

4.1 Scenario Synthesis

A MATSim scenario consists of several components. Some are mandatory, some are optional and merely improve the level of detail of the simulation. The following elements are usually included in a full-stack simulation:

The population file contains all agents and their initial daily plans. These daily plans consist of activities and legs. Usually, agents start at an activity of type “home” and arrive back at such an activity with the same location in the evening. Every activity has a specified end time. It defines when the agent finishes an activity in the activity simulation. The location of an activity can either be defined as a coordinate, a link in the network, or a facility (see below). Every leg has a specified transport mode which tells the simulation which simulator is responsible for moving the agent. These legs can have additional information such as the chosen route through the road network or the chosen connections in public transport. Furthermore, all agents have custom attributes such as age, gender, or whether they have a driving license.

The households file is optional and groups agents into households. Each household can have specific attributes such as the household income or the number of available cars.

The facilities file describes all physical locations at which activities can take place. These can be homes of the people, shops, restaurants, workplaces, and more. They are characterised by a coordinate and a road network link to which they are associated. Furthermore, they hold a list of activity types that can be performed at each facility. Though not used in the simulations presented in this work, capacities can be defined for each facility in a standard MATSim simulation.
The **network** file contains nodes and links. While nodes are characterised by a certain location in a specified Euclidean coordinate system\(^1\), links are defined by a start node and an end node. Each link has additional attributes such as the flow capacity, free speed or a list of allowed modes.

The **transit schedule** file contains a digital representation of the public transport schedule. It consists of stop facilities located on links and coordinates, and it contains transit lines. Transit lines are further divided into specific routes. A route is comprised of a sequence of stops at their respective stop facilities. Thus, a line describes a certain service as it is known in the schedule, while different routes are for instance altered stop sequences when coming from or going to the depot in the morning or in the evening. Each route has furthermore several departures at the first stop, which define the frequency of the service.

A further component which is not used in the scenarios presented in this thesis is the **vehicles** file which could assign specific attributes and characteristics to private and public transport vehicles. For the latter, the passenger capacity would be an important parameter, but it is not used in the simulations presented (see below, Section 4.3.1).

The task of population synthesis is to generate the files that make up a MATSim scenario. While some parts can be almost directly generated from existing data sets, especially generating the population can become a complicated and time-consuming task, dependent on data availability and quality.

### 4.1.1 Available data

In Switzerland, many data sets are available to research which allow creating a MATSim scenario for the whole country. The data sets used in our synthetic population and scenario shall be introduced in the following.

The **STATPOP** (BFS, 2016) data set is a comprehensive census data set with around eight million person observations with sociodemographic attributes such as age and gender. It represents the resident population of Switzerland in 2012. Each person is assigned to a household with additional household-level attributes. Furthermore, each household is

---

1 For Switzerland, coordinates in the EPSG:2056 projection are commonly used.
annotated with a home location by coordinate. Therefore, STATPOP is a rich data set that makes it easy to synthesize a population of artificial persons and households.

The **Structural survey** (BFS, 2017b) is a household survey that contains further attributes about households and their members in Switzerland. In this study, the surveys from 2010, 2011, and 2012 are used. For the synthesis process, the structural surveys are a valuable source of information as they provide home, work and education municipalities of all respondents. This allows for estimating commuting patterns from the data. For statistical treatment, the data set provides a statistical weight for each person.

The **MTMC** (Mobility and Transport Microcensus, BFS and ARE (2018)) is a classical household travel survey in which households provide information about their mobility behaviour. Specifically, it provides sociodemographic attributes for each surveyed household and its members. Furthermore, the daily activity and travel decisions are provided for one reference person of each household. Such activity chains contain information about the types of activities, durations, and transport modes and departure times of connecting trips. Statistical weights are provided by the authorities. The data is valuable for population synthesis as it allows to link daily mobility patterns to sociodemographic attributes of persons. The latest MTMC, which is used in this study, has been conducted during 2015. The survey is repeated every five years.

**STATENT** (BFS, 2019c) is a comprehensive enterprise census including all registered companies in Switzerland by coordinate. Furthermore, information on full-time equivalents and business sector are provided. The data is useful for synthesis as it can be used to select specific locations for the activities of all synthetic persons. This study uses the STATENT data set for the year 2014.

**Spatial data** (BFS, 2020) is provided from many different sources, including official municipality borders and categorisations of municipalities into rural and urban areas. These data sets are valuable for creating a connection between coordinate-based and municipality-based data sets.

**OpenStreetMap** (OSM, The OpenStreetMap contributors (2020)) is a community-driven open data set containing geographical information
about roads, buildings and other infrastructure in most places of the world. Especially in Europe, the quality of the data is usually of a high standard and up to date. In this research, a data dump of Switzerland from Geofabrik (Geofabrik, 2020a) is used to create the road network.

Finally, **HAFAS** (BAV, 2018) is a digital public transport schedule for Switzerland, which is provided as open data. It provides information on the routes and time tables of most of the transport providers in Switzerland.

As mentioned, the population synthesis for **SVI 2016/001** was based on a previously developed simulation scenario for Switzerland. Hence old data sets have been reused. By now, newer versions are available, such as STATPOP and the structural surveys up to the year 2017. As the synthesis pipeline is implemented flexibly, it should be straightforward to update the scenario with the most recent data sources. Even better, by now, all data sets are available for the year 2015, which allows us to set up a scenario that consistently uses data from that year.

In contrast to the scenario of Île-de-France, which will be introduced in Chapter 6, most of the data sets for the Switzerland scenario are not open data. Although it is comparably easy for research to obtain the data, it is only available on request from the authorities. One reason is that the data sets are not anonymised and therefore make it, theoretically, possible to reconstruct the identity of individual persons. Only OpenStreetMap data and HAFAS are publicly and openly available free of charge.

### 4.1.2 Algorithms

Several algorithms and transformations are applied to the data sets to bring them into the format of a MATSim scenario. To recap: the aim is to generate a population with plans and households; facilities; and a road network and public transport schedule.

For the transport infrastructure, data merely needs to be converted. Fortunately, a well-established and tested open-source tool exists, which can do this task. The **pt2matsim** (Poletti, 2016, 2020) tool is used to convert the OSM data into a MATSim-compatible road network file. Also, it converts the HAFAS data set into a MATSim-compatible public transit schedule. While not necessarily important for our simplified public transport simulation component (see 4.3.1) it maps train lines, buses, trams and other public transport modes to the network generated from OSM data in a third step.
The STATENT data set is used to generate the file of facilities. For each observation in the enterprise census, one facility is created at the given coordinate and attached to the closest link in the road network. The detailed business categorization is used to reduce the main activity type of a facility to work, education, shopping, leisure, and other. All these facilities are furthermore marked as work places. During population synthesis (see below) one additional facility with activity type home is added per household.

The complicated part of scenario synthesis is population synthesis. Here, the goal is to create a set of persons that resemble the sociodemographic and socioeconomic structure of the real population. Furthermore, the relations between sociodemographics, home location, work location, and the structure of their daily mobility patterns should be realistic. While a discussion on what exactly “realistic” means in this context must be answered below (but see Chapter 6 for some hints) the basic building blocks of the Switzerland population synthesis pipeline shall be outlined here.

4.1.2.1 Persons and households

For the Switzerland scenario, persons and households can be directly generated from STATPOP data. For each household in the data set, the identifier and the coordinate is kept. At the same time, persons are copied along with information on their person identifier, household identifier, age, gender and marital status.

As full-scale simulations are computationally expensive (see below), usually down-sampled versions of the population are used in practice. For that, households are removed from this population with a configurable probability.

Afterwards, synthetic persons and households are enriched with data from the MTMC. This enrichment is performed using a statistical matching algorithm. For preparation, the head of household in all synthetic households is identified using a heuristic approach. The oldest household member is chosen, and if multiple persons have the same age, a male one is assigned, as was proposed initially by Bösch et al. (2016).

A first statistical matching run is then performed on all persons who are head of household. For the algorithm, some mandatory and optional attributes need to be defined. In the case of the head of household matching, these are age, gender, and marital status (mandatory), and household size and municipality type of the home location (optional). For each synthetic person, the algorithm then searches for all persons in the MTMC that match in all mandatory attribute values, and it prefers MTMC observations which
match additionally in as many optional attributes as possible. The criterion to select the active set of optional attributes is that at least 20 MTMC observations should be available. If 20 observations cannot be found for the mandatory attributes, the household of the synthetic person is dropped. If a set of matching observations from the MTMC can be found, one of them is sampled according to their MTMC weight. The attributes \textit{income class}, \textit{number of cars}, and \textit{number of bikes} are copied from the sampled MTMC household to the synthetic household.

Afterwards, a second statistical matching run is performed on all persons in the population. Matching is performed with the previously imputed attributes as additional optional matching criteria. This way, further attributes are imputed per person. These are: \textit{employment status}, whether the person has a \textit{driving license}, \textit{car availability}, and \textit{ownership of various public transport subscriptions} in Switzerland.

4.1.2.2 \textit{Activity chains}

The statistical matching step does not only allow to impute attributes from the MTMC to the synthetic population, but it is also used to attach the entire daily trip chain from the MTMC observation to the synthetic persons. For each trip the attributes \textit{departure time}, \textit{arrival time}, \textit{transport mode} and the \textit{purpose} are kept. While the MTMC allows for a more detailed distinction, transport modes are aggregated to \textit{car}, \textit{public transport}, \textit{bicycle}, \textit{walking}, and \textit{car passenger}. The trip purposes are also aggregated to be compatible with the activity types of the facilities. Implicitly, this defines a chain of activities with \textit{departure times}, \textit{durations}, and \textit{activity type}. Activities with types \textit{home}, \textit{work} and \textit{education} are called \textit{primary}, while \textit{shopping}, \textit{leisure}, and \textit{other} are called \textit{secondary} activities.

Furthermore, for each trip chain, one trip is selected as the \textit{commute trip}. The selection is performed heuristically by choosing the work or education trip with the longest Euclidean distance between the original MTMC home and work/education location. The Euclidean distance is saved as the \textit{commute distance} and the transport mode of that trip is noted as the \textit{commute mode} of the synthetic person.

Note that through the statistical matching trips and activities are attached to each synthetic person that fit their sociodemographic and regional characteristics.
4.1.2.3 Primary destinations

At this stage in the process, only the location of the household home is known. No other spatial information has been imputed so far. As there are few activity chains in the MTMC that happen with very high probability, they are representative of specific user groups. Unfortunately, the data set does not provide enough observations to allow for a realistic spatial distribution of activities in Switzerland. For that reason, locations for primary and secondary activities need to be assigned through a different process.

For work activities, the structural survey is used. Since it provides the origin municipality, destination municipality and weight for each respondent in the working population, it can be aggregated to an origin-destination (OD) matrix by commute mode.

The assignment process then iterates through all populated municipalities and counts synthetic working persons per commute mode. Then, destination municipalities are sampled according to that count from the respective OD matrix. All sampled destinations are ordered by Euclidean distance of the municipality centroids, and the synthetic persons are ordered by commute distance. Based on the position in each ordered list, destination municipalities are assigned to the synthetic persons. These municipalities are defined as the work municipalities for those persons. Afterwards, a facility is sampled for each person in their respective municipality as the primary work location. The sampling is weighted by the full-time equivalents of the underlying observation from the enterprise census.

Agents are first classified by their age to assign commuting destinations for education based on results published by Bösch et al. (2016). For agents under seven years old, the closest kindergarten is selected from the enterprise census. For agents between 7 and 12 years old, the closest primary school is assigned. For agents between 13 and 16 years one of the five closest secondary educational facilities is chosen, and for agents over 16 years, one of the ten closest tertiary education facilities is selected as the agent’s primary education location.

Afterwards, the primary work and education locations are attached to the commuting work and education activities in the agents’ activity chains. However, additional work and education activities may exist in those plans. Those “sub-primary” activities are assigned locations by interpolating home and primary locations of an agent with structural information about the activity chain from MTMC. The final result is a set of synthetic activity chains that have well-defined coordinates and facilities assigned to all primary (and sub-primary) activities.
For the process of assigning secondary locations not much data is available. Estimating an OD matrix for secondary activities such as shopping and leisure would be possible in principle from MTMC, but it is not detailed enough to create solid OD matrices. Furthermore, one would need to discuss whether these matrices should be dependent on the time of day, and so on.

The initial population by Bösch et al. (2016) also did not include locations for secondary activities. The idea was to perform a simulation-based assignment of such locations through the location choice module of MATSim. Unfortunately, running such a simulation takes much time and would need to be performed many times to calibrate the parameters of the location choice module correctly. Therefore, a novel data-driven approach for assigning secondary locations has been developed, which is only called once when the daily plans of the population are generated in the population synthesis process.

In a nutshell, the algorithm operates on partial activity chains that are framed by fixed activities (in this case, the primary ones). This way, partial sequences of activities emerge that for instance, have a home activity at the beginning of the chain, followed by a shopping activity, a leisure activity and then end again in a home activity. Such an example is shown in Figure 4.1. Furthermore, a reference distance distribution for the whole population must be given, which is estimated from the MTMC. For each mode and travel time, a distribution of likely Euclidean distances is obtained. The algorithm, which is described in detail and more technically in (Hörl and Axhausen, 2020) performs two steps repeatedly: relaxation and discretisation. In the relaxation phase, distances are sampled from these distributions such that they fit the trips, given transport mode and travel time. The secondary activities are then moved in Euclidean space using a gravity model such that the distances between all activities resemble those that have been sampled. Afterwards, discretisation is performed by finding the closest facility, which offers the particular activity type for each activity. The activity location is then updated with the facility location. Between the sampled distances and the distances between the discretised locations, an error metric can be defined. If this discretisation error is sufficiently small, the set of locations is accepted, if it is too large, another round of the algorithm is performed. The process is repeated for all persons and all partial chains in the synthetic population.
Figure 4.1: Example of the secondary location assignment process described by Hörl and Axhausen (2020).
After that, all information is ready to convert the population into the MATSim-readable format. Note that this approach is deliberately called “location assignment” as no behavioral “choice process” is taking place that would back up the assigned locations by preferences. The algorithm has merely the aim to assign discrete locations to the daily plans such that the distribution of distances is matched between the resulting synthetic population and the reference data.

4.2 Cutting Process

Since the movements of millions of agents need to be simulated in MATSim (at least in models of the scale of a country like Switzerland), simulations are computationally expensive. Two strategies are commonly applied to reduce computational cost. First, the population can be scaled down (as described above) such that, for instance, only 10% or less of all households are simulated. While only few systematic results exist on the the influence of down-sampling on the final simulation results (Llorca and Moeckel, 2019), such considerations are currently part of the research on the Île-de-France case (see Chapter 6). Second, if possible, only the region of interest should be simulated. While it is in principle possible to just create a scenario for a city or region from scratch, cutting them out from a larger scenario makes sense as interactions with the surroundings can be considered in the cutting process. The following will describe a new approach for scenario cutting that has been applied to cut a MATSim scenario for Zurich from the upper-level Switzerland scenario.

Traditionally, cutting is done by dropping agents from the initial population. For that, an area is defined, and then agents are deleted according to specific criteria. It is possible to drop all agents that do not have their home location inside of the area. This leads to a substantial reduction in agents but also removes all background traffic. For the case of Zurich, the scenario contains important national highway connections which would show substantially lower traffic if all through-traffic is eliminated. On the other hand, it would be possible to drop all agents that never touch the area of interest at any point during their daily plans. This would keep a larger number of agents in the scenario, but their arrivals and departures to and from the study area may become skewed as congestion outside of the study area would be lowered. While still keeping many agents in the population, this problem can be resolved by artificially fixing travel times on all roads outside of the area.
The method described in the following is a compromise between the mentioned approaches. It is not based on dropping agents from the population, but on cutting agents’ plans in detail. This adds considerable complexity to the cutting process, not only in terms of defining the respective procedures but also in implementing the approach. As all elements of the scenario shall be cut, it is an implementational challenge to keep plans, network, and other parts of the MATSim simulation consistent. None of the resulting data sets should reference items (such as agents, links or facilities) that are no longer existant after cutting.

In the case of Zurich, a circle of 30 kilometres is drawn around the city center\(^2\). The inside is defined as the study area, which is cut out from the Switzerland scenario. Table 4.1 shows a comparison of run times and memory use to motivate further the importance of cutting MATSim simulations to smaller areas.

\(^2\) Bürkliplatz

<table>
<thead>
<tr>
<th>Sample size</th>
<th>0.1%</th>
<th>1%</th>
<th>10%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime [per it.]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switzerland</td>
<td>80 sec</td>
<td>200 sec</td>
<td>22 min</td>
<td>55 min</td>
</tr>
<tr>
<td>Zurich</td>
<td>11 sec</td>
<td>20 sec</td>
<td>75 sec</td>
<td>400 sec</td>
</tr>
<tr>
<td>Memory used</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switzerland</td>
<td>30 GB</td>
<td>51 GB</td>
<td>73 GB</td>
<td>92 GB</td>
</tr>
<tr>
<td>Zurich</td>
<td>6 GB</td>
<td>14 GB</td>
<td>35 GB</td>
<td>40 GB</td>
</tr>
<tr>
<td>Scenario size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switzerland</td>
<td>55 MB</td>
<td>110 MB</td>
<td>630 MB</td>
<td>1.5 GB</td>
</tr>
<tr>
<td>Zurich</td>
<td>6 MB</td>
<td>18 MB</td>
<td>130 MB</td>
<td>320 MB</td>
</tr>
<tr>
<td>Events file</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Switzerland</td>
<td>22 MB</td>
<td>230 MB</td>
<td>2 GB</td>
<td>5 GB</td>
</tr>
<tr>
<td>Zurich</td>
<td>3.4 MB</td>
<td>40 MB</td>
<td>370 MB</td>
<td>885 MB</td>
</tr>
</tbody>
</table>

Table 4.1: Technical comparison between the Switzerland and Zurich scenarios at different sample sizes. Measurements have been taken on a machine with a 24 core CPU, Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz, and 376 GB available memory.
4.2 Cutting Process

The basic idea of the process is to cut all agents’ plans in detail at the correct points in time and space. For that, the routes through the road and public transport networks of all trips in their daily plans are examined. Trips that take place entirely outside of the study area shall be removed, and trips that take place entirely inside are kept. All remaining trips that cross the scenario border at one or multiple points in space and time along their route are processed in detail. Figure 4.2 shows all possible cases with up to two crossing points. While routes with more crossing points are not common, they do exist and are treated correctly by the cutting algorithm.

The cutting process varies between transport modes. For car trips, routes through the network are described link by link in the daily plans. The algorithm, therefore, examines which of the traversed links are inside or outside of the study area and segments the whole route in “inside” sequences and “outside” sequences. The car trip is then split into multiple trips that are separated by an activity of the type “outside“, as shown in Figure 4.3. Therefore, the resulting plan contains a larger number of trips than before. All trips that are now outside of the study area, are not performed with the “car” transport mode, but with an artificial “outside” mode. The resulting “outside” activities all define an activity end time, which is set to the exact time when the agent enters the area in the large scenario.

**Figure 4.2:** Schematic representation of the cases that occur during cutting of agents’ plans.

4.2.1 Population

...
Figure 4.3: Visualization of the cutting process for road-based routes.

For public transport the cutting is similar. The spatial resolution is coarser as the routes are cut at stops along the selected public transit lines. When cutting a transit connection, only the earliest/latest stops inside of the study area are kept as a “public transport” trip, while all other segments are converted to “outside trips”. The activities performed at the first/last stop inside the area are assigned the “outside” activity type and their end time is set according to the departure/arrival time on the chosen route.

All other modes (walking and bicycle) are simulated as “teleported modes” in our MATSim simulations. This means that their Euclidean distance is obtained and multiplied by a “beeline distance factor” to account for detours along the way. After combining the resulting distance with a constant walking/cycling speed, the travel time is obtained. While more detailed results on age-specific walk speeds in Switzerland are available (Dobler, 2013), they were not used in this work, but should be added to the model in the future. In the simulation, agents are teleported from the trip origin such that they appear at the destination after this time span. This makes it easy to cut those trips, as only intersection points between a straight line for the trip and a polygon for the study area need to be found. Afterwards, the trips can be split into smaller segments and departure times from their entry points can be calculated by linearly interpolating the travel time along the line.

After all trips and activities have been processed this way, all daily activity chains are reduced by collapsing sequences of outside activities into single outside activities. However, any time when an agent enters the area and any time when leaving it, one distinct outside activity must mark this event. Assume the activity chain (home, work, home), but the workplace is outside of the study area. The resulting activity chain after cutting the trips then looks like (home, outside, outside, outside, home). While the two outside activities next to “home” are the cutting points, the middle one is the actual
work activity that was converted. After collapsing the plan, this one can be removed: (home, outside, outside, home). Automatically, the transport mode between the two outside activities is the “outside mode” as defined by the cutting process. In the simulation, these “outside” trips are teleported instantly. The time of entry is determined solely by the calculated end time of the entering “outside” activity.

Table 4.2 compares the population for Switzerland with the smaller Zurich cut-out.

4.2.2 Public transport and road network

After cutting the population, the public transit network can be reduced. Different to agents’ plans, it was decided that cutting transit lines and replacing them by multiple smaller ones would be feasible but was not seen as adding much value to the cutting process. This is especially true as public transport is simulated deterministically according to schedule in our simulation, rather than detailed on the rail/road network (see below).

The cutting procedure iterates through all public transit lines and finds the most extended sequences of stops outside of the region from the start and end of each line, respectively. All stops inside these sequences are deleted, and line departures are adjusted such that they depart from the

<table>
<thead>
<tr>
<th></th>
<th>SWITZERLAND (10%)</th>
<th>ZURICH (10%)</th>
</tr>
</thead>
<tbody>
<tr>
<td># Agents [x1000]</td>
<td>823</td>
<td>158</td>
</tr>
<tr>
<td># Home inside</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td># Has activity inside</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td># Only through-traffic</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td># Trips [x1000]</td>
<td>2,230</td>
<td>606</td>
</tr>
<tr>
<td># Inside area</td>
<td>514</td>
<td></td>
</tr>
<tr>
<td># Entering / Leaving</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td># Crossing</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td># Activities [x1000]</td>
<td>2,935</td>
<td>1,023</td>
</tr>
</tbody>
</table>

Table 4.2: Characteristics of the Zurich population after cutting in comparison to the Switzerland population.
first stop inside the area when they are planned to do so according to the initial schedule.

Table 4.3 summarises the reduction in public transport lines and stops.

Cutting the road network follows a simple approach: all network links that are included in any remaining traveller or public transport route are kept, together with those that are located entirely (with their start and end coordinate) inside of the area. While this procedure sounds easy, it is in fact quite complicated, mainly because the underlying graph of the road network is unidirectional in MATSim. During the simulation, routing is frequently performed, and all implemented routing algorithms (such as Dijkstra or A*) require a consistent graph. This means that no sources (nodes that only have outgoing links), sinks (nodes that only have incoming links) and islands (disconnected sub-graphs) can exist in the network. Following the cutting procedure for car trips from above, a high number of sinks and sources are generated, which break the routing algorithms.

All sources, sinks and islands in the reduced network are identified to fix the network. A routing to and from each of those links, respectively, is then performed from/to a reference link in the centre of the scenario. All links on those routes are then added to the fixed network. The result is a network in which any link can be reached from any other link, such that no sinks, sources or islands exist.
Table 4.5: Characteristics of the Zurich households and facilities after cutting in comparison to the Switzerland scenario.

Table 4.4 shows statistics on the road network of the Switzerland and Zurich scenarios and Figure 4.4 shows the resulting road network and all remaining public transit stops for rail services.

4.2.3 Households and Facilities

Finally, households and facilities are reduced. For each household, the remaining household members are counted. Those households that have no members left are removed from the scenario.

For facilities, all observations inside the study area are kept. All others, including household facilities of removed households, are removed. Additionally, new “outside” facilities are created and attached to all “outside” activities in the population. This is important as a number of structural validation criteria have been defined for the resulting scenario:

1. All activities need to hold information about the coordinate at which they are performed, the road network link that is assigned to the activity and the facility at which the activity is taking place.

2. The coordinates of an activity need to be equal to the coordinates of the assigned facility.

3. The link assigned to an activity needs to be equal to the link assigned to the respective facility.

These rules enforce the following spatial structure. Facilities can be located anywhere in space, but they are connected to a road. This makes sure that routes to the facility end at a well-defined location in the road network. Therefore, activities happen at facilities, but it is also known how to reach those facilities via the road network.

Table 4.5 summarises the cutting process of households and facilities.
Figure 4.4: Road network and public transit stops for rail services after cutting the Zurich scenario.
4.3 Baseline Model

The following describes the baseline simulation scenario for Zurich. First, a number of custom components are described that have been developed for the specific simulation scenario. Afterwards, calibration results and comparisons to reference data are shown.

4.3.1 Custom components

For the project, two custom components have been developed that make simulations faster and easier to handle, and help for calibration of the scenario.

The first is a schedule-based public transport simulation component. In general, MATSim makes it possible to simulate public transport in great detail. Each bus or train is simulated as one agent that can pick up and drop off passengers along a predefined route. While this allows us to study in detail transit delay and crowding in public transport, it also poses a substantial computational burden. Often, simulations in MATSim are scaled-down such that for instance, only a 10% sample of the whole agent population is simulated. However, such downscaling is not very easy for the public transport supply. Even if vehicle capacities are scaled-down, all public transport vehicles need to be simulated to model the correct frequencies on the transit lines. For a whole scenario of Switzerland or Zurich, this is a great computational effort in itself. Therefore, it was decided to simulate agents as if public transport would function without delay. A component has been developed that is activated once an agent wants to depart from an activity by public transport. It is pre-calculated which public transport lines to take to get to the destination. At this point, the trip can be routed through the transit schedule, and travel times without delays are used. The simulator then sets the agents into “travelling” mode and moves them to their destination after this period has elapsed. While still providing much detail as interaction with the public transit schedule is possible, this approach allows for drastically reduced runtimes. Note that such functionality is now included by default in newer versions of MATSim.

Second, a “crossing penalty” has been introduced to the simulation. By default, MATSim calculates the (uncongested) traversal time for a link as
with $L$ being the length of the link, $v_f$ the free speed attribute of the link and $v_{\text{veh,max}}$ the maximum speed of the vehicle. From experience, it is also known that the traffic dynamics in MATSim usually underestimate node dynamics, i.e. agents pass intersections rather quickly. While extensions for the simulation of traffic lights are available (Kühnel et al., 2018), it is challenging to apply it as detailed information needs to be available. To introduce a sense of “right of the way” into the simulations, we calculate the link traversal time as

$$\min \left\{ \frac{L}{v_f} , v_{\text{veh,max}} \right\}$$  \hspace{1cm} (4.1)

with $c$ being the “crossing penalty”. The variable $\sigma \in \{0, 1\}$ defines whether the link is subordinate at the downstream intersection. The value is determined by examining the flow capacities of all downstream links and checking whether any of them has a higher capacity than the current one. If this is the case, we set $\sigma = 1$. By that, we add a penalty on small roads that lead to larger roads. The value of $c$ itself then becomes a calibration parameter for the simulation. After several experiments, we have noted that a value of $c = 3 [s]$ strongly improves the fit with reference data in terms of travel times and resulting mode shares.

### 4.3.2 Travel behaviour

The Zurich baseline scenario uses a discrete mode choice model to simulate the behaviour of the population. In SVI 2016/001 the choice model was estimated from a survey with about 350 respondents in the canton of Zurich. Besides their traditional modes of transport, automated taxis were also available in their choice set. While complex and detailed models were estimated based on the data (Hörl et al., 2019), a relatively simple Multinomial logit model was defined that is only based on variables that are compatible with the MATSim simulation. Specifically, all attributes in the model can either be looked up in the sociodemographics or the daily plans of the agents, or calculated during the simulation.

\[^{3}\text{The choice modelling part of SVI 2016/001 and afterwards was done by Felix Becker.}\]
The utility function for the conventional transport modes are defined as follows:

\[
\begin{align*}
    u_{\text{car}} &= \beta_{\text{ASC,car}} \\
    &+ \beta_{\text{travelTime,car}} \cdot x_{\text{travelTime,car}} \\
    &+ \beta_{\text{travelTime,car}} \cdot \theta_{\text{parkingSearchPenalty}} \\
    &+ \beta_{\text{travelTime,walk}} \cdot \theta_{\text{accessEgressWalkTime}} \\
    &+ \beta_{\text{cost}} \cdot \left( \frac{x_{\text{euclideanDistance}}}{\theta_{\text{averageDistance}}} \right)^\lambda \cdot x_{\text{cost,car}} \\
    u_{\text{pt}} &= \beta_{\text{ASC,pt}} \\
    &+ \beta_{\text{numberOfTransfers,pt}} \cdot x_{\text{numberOfTransfers,pt}} \\
    &+ \beta_{\text{inVehicleTime,pt}} \cdot x_{\text{inVehicleTime,pt}} \\
    &+ \beta_{\text{transferTime,pt}} \cdot x_{\text{accessEgressWalkTime,pt}} \\
    &+ \beta_{\text{accessEgressTime,pt}} \cdot x_{\text{accessEgressTime,pt}} \\
    &+ \beta_{\text{cost}} \cdot \left( \frac{x_{\text{euclideanDistance}}}{\theta_{\text{averageDistance}}} \right)^\lambda \cdot x_{\text{cost,pt}} \\
    u_{\text{bicycle}} &= \beta_{\text{ASC,bicycle}} \\
    &+ \beta_{\text{travelTime,bicycle}} \cdot x_{\text{travelTime,bicycle}} \\
    &+ \beta_{\text{age,bicycle}} \cdot \max(0, a_{\text{age}} - 18) \\
    u_{\text{walk}} &= \beta_{\text{ASC,walk}} \\
    &+ \beta_{\text{travelTime,walk}} \cdot x_{\text{travelTime,walk}}
\end{align*}
\]

(4.3) (4.4) (4.5) (4.6)

All parameters $\beta$ and $\lambda$ can be estimated from survey data, while all $x$ are trip-specific attributes that can be obtained and calculated adaptively while running the simulation. Agent-specific attributes (here only age) are denoted as $a$. The constant $\theta_{\text{averageDistance}}$ is a model constant obtained from the survey data, and all other $\theta$ are calibration constants that have been added to the model to provide a good fit with reference data. Note that the original choice model (estimated using Biogeme) was set up accordingly without the terms involving $\theta_{\text{parkingSearchPenalty}}$ and $\theta_{\text{accessEgressWalkTime}}$. Additionally, the parameter $\beta_{\text{travelTime,bicycle}}$ was adjusted during calibration. The final parameter values are shown in Table 4.6.

One crucial component of the choice model is the cost of driving a car and using public transport. After calibration and comparison with typical
<table>
<thead>
<tr>
<th>Transport Mode</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>$\beta_{ASC,car}$</td>
<td>0.827</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,car}$</td>
<td>-0.0667</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td>Public transport</td>
<td>$\beta_{ASC,pt}$</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>$\beta_{numberOfTransfers,pt}$</td>
<td>-0.17</td>
</tr>
<tr>
<td></td>
<td>$\beta_{inVehicleTime,pt}$</td>
<td>-0.0192</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{transferTime,pt}$</td>
<td>-0.0384</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{accessEgressTime,pt}$</td>
<td>-0.0804</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td>Bicycle</td>
<td>$\beta_{ASC,bicycle}$</td>
<td>-0.1</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,bicycle}$</td>
<td>-0.0805</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{age,bicycle}$</td>
<td>-0.0496</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[a]$^{-1}$</td>
</tr>
<tr>
<td>Walk</td>
<td>$\beta_{ASC,walk}$</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,walk}$</td>
<td>-0.141</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]$^{-1}$</td>
</tr>
<tr>
<td>Others</td>
<td>$\beta_{cost}$</td>
<td>-0.126</td>
</tr>
<tr>
<td></td>
<td>$\lambda$</td>
<td>-0.4</td>
</tr>
<tr>
<td></td>
<td>$\theta_{averageDistance}$</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[km]$^{-1}$</td>
</tr>
<tr>
<td>Calibration</td>
<td>$\theta_{parkingSearchPenalty}$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]</td>
</tr>
<tr>
<td></td>
<td>$\theta_{accessEgressWalkTime}$</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[min]</td>
</tr>
</tbody>
</table>

Table 4.6: Parameters for the choice model in the Zurich baseline simulation.
values in Switzerland, car costs are calculated as 0.26 CHF/km based on driven distance. Note that these are the *variable costs* of travelling that are taken into account for choice making, not the *full costs* of owning and using a car.

For public transport, a detailed ticket-based cost model has been developed in parallel to SVI 2016/001. In principle, it is possible to obtain the fare structure of all transport authorities in Switzerland, but not in digital format. Therefore, all available data needed to be automatically transcribed from PDF documents, converted into the right formats and further processed. It is then possible to divide a public transport trip into segments between transfers. The condition for a realistic price estimate is then that the traveller needs a set of tickets that covers all legs during the trip, ideally the cheapest one. Given all available tickets $t$ and their price $p_t$, it is then possible to set up the following combinatorial optimisation problem:

$$\begin{align*}
\text{minimize} & \quad \sum_t x_t \cdot p_t \\
\text{subject to} & \quad \sum_t c_{st} \cdot x_t > 0 \quad \forall s \in S \\
& \quad x_t \in \{0, 1\} \\
& \quad c_{st} \in \{0, 1\}
\end{align*}$$

(4.7)

Here $S$ denotes all the segments on a traveller’s trip, referenced by index $s$. The parameter $c_{st}$ defines whether ticket $t$ covers stage $s$ in terms of space (tariff zones) and time (validity). The variables $x_t$ describe whether the traveller should buy ticket $t$. The aim of the optimisation is to find a set of purchase decisions that minimises the overall cost of the trip while having a valid ticket at any time and place. The formulation in equation 4.7 is generally known as a “cutting stock problem” (Kantorovich, 1960) and can be solved by common tools such as GLPK or Gurobi.

While the data is available and the solver is operational, the detailed ticket-based model was not included in the final simulations due to time constraints. Furthermore, upon finalisation of the approach, Swiss Railways have announced to stop the distribution of tariff documents as of 2020. While currently not publicly accessible, a new online API has been set up for the same purpose. Pathways for future collaboration between Swiss Railways and research for using the data of the digital ticketing API have been discussed.

As a quicker (and easier to calculate) solution, the following cost model is used. Table 4.7 shows the full price per (in-vehicle) distance that establishes
the basis of the public transport cost calculation with a minimum price of 2.70 CHF. As public transport subscriptions are common in Switzerland, they have been included in the model. In case a person owns a “Generalabo” (GA), they have paid upfront for a month or a year to use any public transit line in Switzerland. Therefore, the variable costs used in the choice model are set to zero. Users with a half-fare subscription (“Halbtax”) pay half the price as shown in Table 4.7, but a minimum price of 2.30 CHF per trip. Persons who own a regional subscription (“Verbundabo”) pay according to the pricing scheme shown in Table 4.8. The values in both Table 4.7 and Table 4.8 are based on trip information form the surveys of SVI 2016/001.
4.3.3 Validation

To validate (and calibrate) the scenario, reference data needed to be obtained. For the project we decided that three dimensions are of major interest: travel time distributions, travel distance distributions for the car and public transit modes, and mode shares for multiple distance bins. Other dimensions, such as vehicle flows at count stations, were considered less important. The reason is that the work was a simulation-based mode choice experiment to estimate the attractiveness of automated mobility services. While infrastructural analyses were also performed, the focus was put on analysing mode adaptation patterns of the population which are mainly dependent on the trip characteristics. More detailed analyses regarding flows on public roads are considered in a follow-up project (Livingston et al., Upcoming).

Travel time distributions were essential to replicate because the valuation of time spent in or waiting for a vehicle is a crucial component of the choice model. Likewise, a good fit in travel distances is important as distances directly influence the costs for the travellers.

Many MATSim simulations have traditionally been calibrated to resemble well the overall mode shares in a study area. For this work, such an approach was seen as too weak to compare conventional transport modes with future options reliably. Therefore, mode shares have been calibrated such that they fit not only in the total number of trips but also trips per distance bin compared to reference data.

As mentioned above, calibration to achieve a good fit for travel time distributions, travel distance distributions, and mode shares by distance, was performed by adjusting the crossing penalty introduced before and, by adjusting the price structure and adding calibration constants to the choice model. Those constants account, for instance, for parking search which is not modelled explicitly in the MATSim simulations presented\textsuperscript{4}.

Figure 4.5 shows the travel time distributions per mode for the baseline simulation of Zurich. To set up this run, first, a full simulation for Switzerland was performed, then the Zurich scenario was cut out, run again and analysed. Reference data comes from the MTMC which has been analysed for the same region that was defined as the study area. For the sake of comparability, only trips that take place entirely inside of the 30km radius around Zurich are taken into account in simulation and reference data.

\textsuperscript{4} Respective extensions exist (Bischoff and Nagel, 2017), but including them was not the focus here. Furthermore, any additional component in the simulation increases computation time, especially when it is as dynamic as parking search.
respectively. The plots show that travel time distributions can be reasonably reproduced, the same is true for travel distance distributions (Figure 4.6) and speeds (Figure 4.7). Note that the active modes walking and cycling are simulated with constant speed\(^5\). Hence a comparison of speed distributions is omitted.

Figure 4.8 shows that the calibrated baseline case reproduces well the correct share of transport modes at different distance classes. The distance bins are defined as the twenty quantiles from the distance distribution over all modes up to an upper value of 20 km. Although not explicitly the aim

\(^5\) Bicycle 10 km/h, Walking 5 km/h
Figure 4.6: Calibrated travel distance distributions for the Zurich scenario.
of calibration, Figure 4.9 also shows a good resemblance of mode share by the time of day.

All simulations are run with a 10% sample of households to keep run times in an acceptable, feasible range (see Table 4.1).

4.4 AMoD Model

For the AMoD simulations, the baseline scenario was extended with additional assumptions and model components. Most importantly, the AV extension was added for the simulation of automated taxis. The vehicle fleet and surrounding scenario need to be configured carefully to render a realistic future picture of automated mobility. Three decisive components are available from the model set up: road capacity effects, the choice model, and the operator model.

In terms of capacity effects, a literature review can be found in (Hörl et al., 2019). For the final experiments, classification of roads in two categories was performed. One category contains urban roads, which are likely to impose larger constraints on the capacity benefits of automated vehicles. For these, a capacity gain of $+40\%$ is assumed. Other roads outside of the city or higher-order roads traversing the city are assigned a capacity gain of $+80\%$. Figure 4.10 shows all roads with $+40\%$ in red, all others fall into the $+80\%$ category.
Figure 4.8: Calibrated mode share by Euclidean distance for the Zurich scenario.

Figure 4.9: Calibrated mode share by time of day for the Zurich scenario. Each data point denotes the mode share for the following hour.
Figure 4.10: Classification of roads to determine the capacity gain for automated vehicles. All roads in the “urban network” receive a capacity gain of +40%, all others +80%.
Table 4.9: Additional parameters for AMoD in the Zurich choice model.

<table>
<thead>
<tr>
<th>Transport Mode</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMoD</td>
<td>$\beta_{ASC,AMoD}$</td>
<td>0.162</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,AMoD}$</td>
<td>-0.0527</td>
</tr>
<tr>
<td></td>
<td>$\beta_{waitingTime,AMoD}$</td>
<td>-0.0802</td>
</tr>
</tbody>
</table>

To model the use of automated taxis, not only conventional modes were considered in the choice model, but also an AMoD alternative. The survey focused on the trade-off between travel times, waiting times and costs compared to other modes of transport. The corresponding utility function is defined as

$$u_{AMoD} = \beta_{ASC,AMoD} + \beta_{travelTime,AMoD} \cdot x_{travelTime,AMoD} + \beta_{waitingTime,AMoD} \cdot x_{waitingTime,AMoD} + \beta_{cost} \cdot \left( \frac{x_{euclideanDistance}}{\theta_{averageDistance}} \right)^\lambda \cdot x_{cost,AMoD} \quad (4.8)$$

with the parameters shown in Table 4.9. The prices are calculated directly from the price calculator of future mobility services by Bösch et al. (2018). The main influences of price in the calculator are fleet size, fleet vehicle distance, customer distance. While the first two components translate to costs for the operator, the latter translates into revenue. Given the operational characteristics of the fleet from simulation, the calculator yields a price per passenger-kilometre that covers the cost of the operator. Such a price, with a profit margin of 3%, is used in the following simulation. With such a profit margin, the service should merely be imagined as being offered by a public actor such as the public transport authority of Zurich. The operating area of the service is the (municipal) city area of Zurich (see Figure 4.10). While this decision has been made for the simulations shown in this dissertation, a more detailed refinement would be interesting in the future. Even dynamic approaches of defining the service area of a dynamic transport service could be applied (Bischoff et al., 2018a).

Figure 4.11 shows the conceptual structural of the AMoD simulation loop. Starting from a mobility simulation (one iteration in MATSim), various values can be measured that are relevant for the customer choice and
the cost calculation. Attributes such as travel times and waiting times are estimated and made available to the mode choice model. On the operator side, utilisation, empty distance, customer distance and fleet size are fed into the cost calculator. From the cost calculator, a new price is obtained that can be used in the discrete choice model. Finally, traveller agents make their choice based on all provided information, and another mobility simulation is started. This loop is run until mode shares stabilise. So far, no widely accepted metric is available to determine when a MATSim simulation is finished (as constantly small variations can be observed). However, by visually assessing the evolution of mode shares over many cycles, a specific number of iterations can be defined at which mode shares merely fluctuate in a stationary distribution.
This chapter is based on the project report of SVI 2016/001:


The project SVI 2016/001, which is the basis of this thesis, was one of the first projects set up to explore the impacts of automated mobility in the context of Switzerland. While the project contained parts on private automated mobility and Automated Mobility on Demand alike, the following will only focus on the on-demand results as they are generally more robust and less affected by relatively rough assumptions about topics such as intra-household mobility.

The AMoD simulations developed offer a multitude of possibilities for analysis and shaping the simulation experiments. Therefore, it is important to first think about which experiments are interesting, how they can be achieved using the parameters that are available and which aspects of the simulation results should be analyzed.

Four scenarios have been defined. The first is the baseline case in which the status quo of the transport system is simulated. In the second scenario, we assume that automated driving technology has advanced and public transport has become widely automated. In terms of input parameters, the prices for light public transport (including busses, trams, and ferries) are halved according to the results of the cost calculator (Bösch et al., 2018) while prices for rail connections stay the same. For the Swiss scenario, comparing the baseline case with conventional or automated public transport has an additional aspect. As today’s public transport is subsidized by around 50%, the “conventional” case can also be seen as a future with automated public transport, but in which cost savings from automation have led to a subsidy reduction. The aim of both scenarios is therefore to explore how travel decisions in Zurich would shift if future price structures are applied to public transport.

To study the effect of AMoD, both conventional and automated public transport scenarios are paired with a fleet of automated vehicles in Zurich.
In these cases the impact is not only driven by a different price structure, but by the addition of a completely new mode of transport. The attractiveness of the AMoD system is driven by its price and offered waiting times. As by definition of our cost model the service needs to be profitable with a profit margin of 3% both dimensions are heavily driven by the fleet size that is chosen by the operator. Therefore, one specific case needs to be chosen for scenario comparison. In this study, various fleet sizes are tested and the one that attracts the maximum demand in the “conventional public transport” case is chosen.

The resulting scenarios can be evaluated in a number of aspects. For the study at hand the following have been chosen:

The **mode share by trip distance** defines how much a certain transport mode is used and how much it contributes to the overall movement of people.

The **vehicle distance** in the system defines how much road infrastructure is used and deteriorated by vehicle use. It furthermore directly translates into a range of other externalities such as warm and cold emissions, accidents and noise.

The **number of vehicles** that occupy roads at any point during the day gives insight on how many vehicles the system needs to handle during one day. Also, it indicates how many vehicle owners there are and how much space is needed for parking. Therefore, the number relates to the amount of infrastructure that needs to be dedicated to road-based mobility in a city. A different metric is the number of vehicles that are concurrently travelling the system at any time of the day. It relates to the general perceived level of traffic and local emissions.

Finally, the spatial impact of the AMoD fleet shall be analyzed by looking at how accessibility to workplaces changes in Zurich if such a service is available.

Note that the shown simulations in this thesis differ from those in the report of SVI 2016/001 as there an additional base fare of 3 CHF was added to the price of each trip. For the sake of consistency with the cost calculation, this additional fee is omitted in the following simulations. While the base fare was deemed sensible to avoid many trips of very short distance in the project report, further research has shown that this behaviour is consistent with the discrete choice model and should, therefore, be kept. Generally, a higher number of (short) trips leads to additional empty distance and therefore higher prices.
5.1 FLEET SIZING

For fleet sizing, various simulations were performed where AMoD services with different fleet sizes were offered to the population. Prices and waiting times are obtained dynamically from the current use of the fleet. The only restrictions that are imposed are that the origin and destination of any AMoD trip need to be inside of the operating area and that the trip must have a Euclidean distance longer than 200m.

Figure 5.1 shows the results of the fleet sizing simulations. The most interesting metric is shown in the upper left corner. The plot shows the number of trips that are attracted by the AMoD fleet dependent on the number of available vehicles. A clear maximum at around 7,000 vehicles can be seen. For all fleet sizes that are below this level higher waiting times render the service less attractive (see plot on the upper right), and
for all fleet sizes above 7,000 vehicles, prices are high enough to suppress additional demand (lower left plot).

It must be emphasized that only a dynamic demand simulation makes it possible to produce such an analysis. In a static demand analysis, increasing fleet sizes usually would lead to lower waiting times (as seen here). However, there would be no response to those waiting times, and there is especially no response on higher prices for larger fleets.

Looking at the demand-optimal fleet size of 7,000 vehicles one can see that on average a waiting time of less than five minutes is measured and that waiting times of around seven to eight minutes are accepted if they do not exceed this limit in 90% of the cases. For the price, the fleet with 7,000 vehicles leads to a value of 0.60 CHF/km, which is below, but close to the full cost of owning a car in Switzerland of around 0.70 CHF/km.

It must be noted that all fleet configurations in Figure 5.1 are economically feasible as they are always based on prices that lead to a profitable service. Hence, the choice of 7,000 vehicles is arbitrary. If an operator, for instance, is constrained to 5,000 vehicles, he could ask a lower price, but waiting times would be higher. On the other hand, looking at the lower right plot, this configuration would be the worst case in terms of empty distance brought into the system.

The plots can, therefore, also be interpreted from the perspective of policymakers. There is an interesting trade-off, because the city may require the operator to drive no more than 20% of the distance empty. In that case, however, the service would become more expensive, leading to a less attractive offer.

5.2 Scenario Analysis

Based on the fleet size of 7,000 vehicles, the defined future scenarios can be compared. All analyses include all trips taking place inside of the city area of Zurich, i.e. through-traffic, is filtered out for a more focused analysis.

First, a look from the transport system perspective can be taken at the simulation results. Table 5.1 shows the mode shares by trip distance in the four scenarios. Already in the baseline case, Zurich shows a high share of public transit use. A large share of 43.56% of the distance is covered by public transport. Automation of busses and trams would lead to an even higher share of almost 51%. Regarding the change of mode shares, primarily former car users would be attracted by lower fares.
An AMoD fleet of 7,000 vehicles and the discussed price and waiting time characteristics would attract around 31% of the demand (Table 5.1), which is a substantial share of the overall covered distance by the population of Zurich. Demand is mainly drawn from bicycle users with more than half the distance being covered by AMOD, but also from public transport and car users. The attraction is strongest for the latter where a drop from a 44% mode share to 25% is simulated. Hence, it can be stated that the AMOD service does attract demand from public transport, but it does so even more for car traffic.

In the case where automation is applied to the public transport system shares of public transport use stay constant. As some demand is drawn from former car users, the share of private car usage is less than half compared to the baseline scenario. These results indicate that an AMoD system may be a viable solution to reduce private vehicle ownership with all attached costs while not deteriorating public transport service levels. This, however, requires that investments in the automation of the public transport system are taking place in parallel.

If the AMoD service is able to decrease car ownership, the question becomes whether it is also the more ecological alternative. This can be answered by looking at Table 5.2. Without any automation in public transport, adding the AMoD service to the system would increase VKT (vehicle kilometres driven) by an enormous 50%. Such an outcome may be shocking for infrastructure planners as this distance translates directly into maintenance costs. A private operator that is responsible for such an increase in road use may be charged accordingly, which could potentially alter the cost structure. Such an analysis would be an interesting point for future research.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>AMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conv. Pt</td>
<td>Aut. Pt</td>
</tr>
<tr>
<td>Car</td>
<td>43.71</td>
<td>37.66</td>
</tr>
<tr>
<td>Public transport</td>
<td>43.56</td>
<td>50.96</td>
</tr>
<tr>
<td>Bicycle</td>
<td>4.95</td>
<td>3.73</td>
</tr>
<tr>
<td>Walking</td>
<td>7.78</td>
<td>7.66</td>
</tr>
<tr>
<td>AMoD</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.1: Mode share by travel distance in the scenarios of the Zurich case study.
Table 5.2: Vehicle distance in the scenarios of the Zurich case study.

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th></th>
<th>AMOD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conv. PT</td>
<td>Aut. PT</td>
<td>Conv. PT</td>
<td>Aut. PT</td>
</tr>
<tr>
<td>Cars</td>
<td>1060.64</td>
<td>927.06</td>
<td>600.35</td>
<td>501.43</td>
</tr>
<tr>
<td>Public transport</td>
<td>150.9</td>
<td>150.09</td>
<td>150.09</td>
<td>150.09</td>
</tr>
<tr>
<td>AMoD (Customer)</td>
<td>-</td>
<td>-</td>
<td>792.65</td>
<td>745.79</td>
</tr>
<tr>
<td>AMoD (Empty)</td>
<td>-</td>
<td>-</td>
<td>296.64</td>
<td>265.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(27.23%)</td>
<td>(26.22%)</td>
</tr>
<tr>
<td>Total</td>
<td>1210.73</td>
<td>1077.15</td>
<td>1839.73</td>
<td>1662.35</td>
</tr>
<tr>
<td></td>
<td>(-11.03%)</td>
<td>(+51.95%)</td>
<td>(+37.30%)</td>
<td></td>
</tr>
</tbody>
</table>

In the scenarios with automated public transport, there is a decrease in total VKT if no AMoD system is in place. However, the automated transit services are not able to counteract the strong increase of VKT of the AMOD fleet. Even if automated public transport is available, VKT is increased by 37%.

In ecological terms, an increase of 37% or even 50% in VKT is only acceptable if automated vehicles are substantially more ecological than today’s cars, maybe powered by electricity from regenerative energy sources. Therefore, it is worth looking at the total number of vehicles that are used in each scenario. Table 5.3 counts the number of unique vehicles that are used at least once during the simulated day. On the one hand, vehicles that are missing are an indicator of decreasing car ownership. On the other hand, a fleet of 7,000 vehicles is added on top in case the AMoD service is available.

In all analyzed scenarios, a reduction of the vehicle count can be observed. The automated public transport system and the AMoD service alike have the power to make car users give up their vehicle. The comparably small number of 7,000 fleet vehicles does not change this fact. These are good news in ecological terms if a full life cycle analysis is the basis for the assessment. As much fewer vehicles are needed (and produced) the ecological footprint of the mobility system becomes considerably smaller. Table 5.3 shows that potentially almost 40% fewer vehicles would be needed if public transport was automated, and an AMoD service existed in Zurich.
### Table 5.3: Number of vehicles in the scenarios of the Zurich case study.

<table>
<thead>
<tr>
<th></th>
<th>Baseline [1000 veh.]</th>
<th>AMod</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conv. PT</td>
<td>Aut. PT</td>
</tr>
<tr>
<td>Car</td>
<td>107.73</td>
<td>98.92</td>
</tr>
<tr>
<td>AMoD</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>107.73</td>
<td>98.92</td>
</tr>
<tr>
<td></td>
<td>(-8.18%)</td>
<td>(-31.31%)</td>
</tr>
</tbody>
</table>

It is easy to misinterpret the numbers in Table 5.3 as they provide information on the number of vehicles in general, but not about how many vehicles are driving at any time of the day. Figure 5.2 shows the amount of concurrently driving private cars and automated taxis for the baseline scenario (black line) and the AMoD / automated public transport scenario with the strongest decrease of unique vehicles in Table 5.3. Even in this case, the plot shows that the future scenario consistently has more moving vehicles on the roads than today. At noon even twice as many vehicles are moving. The general reduction of the overall vehicle fleet caused by the AMoD service does therefore not generally mean that there is a reduction in traffic. On the contrary, because the vehicles can drive empty and because they attract demand from public transport, traffic flows are increasing.

Table 5.4 analyzes the scenarios from the user perspective. The upper part shows the daily expenses of travellers if they spend any money on mobility. This means that persons who only walk, cycle or do not move at all are not included in these statistics. First of all, it can be stated that costs stay rather constant. This is expected as agents adapt their mobility behaviour according to the cost-sensitive choice model. Compared to the baseline scenario, the largest increase is surprisingly seen for the scenario with automated public transport and AMoD. While this result may indicate self-selection effects among the agents and trips, it can also stem from the stochastic fluctuation of the simulation outcome (see discussion below).

In terms of daily travel times, Table 5.4 shows that the AMoD system generally leads to an increase. This is also expected as agents value the time spent in the automated taxi higher than time spent driving their car. Nevertheless, the increase in travel time stays rather small.
Figure 5.2: Number of vehicles on the road in Zurich in comparison between the baseline case and the scenario with AMoD and automated public transport.

<table>
<thead>
<tr>
<th></th>
<th>BASELINE</th>
<th>AMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CONV. PT</td>
<td>AUT. PT</td>
</tr>
<tr>
<td><strong>Daily cost</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean [CHF]</td>
<td>4.31</td>
<td>4.05</td>
</tr>
<tr>
<td>Median [CHF]</td>
<td>2.92</td>
<td>2.69</td>
</tr>
<tr>
<td><strong>Daily travel time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean [min]</td>
<td>34.10</td>
<td>34.67</td>
</tr>
<tr>
<td>Median [min]</td>
<td>25.40</td>
<td>26.13</td>
</tr>
</tbody>
</table>

Table 5.4: Scenario analysis from the user perspective. Cost metrics do not include persons who do not spend any money on mobility during the simulated day.
5.2 Scenario Analysis

Figure 5.3: Comparison of accessibility in the baseline and automated scenario with automated public transport and AMoD. The accessibility is based on FTEs in enterprises in each zone and the fastest centroid-to-centroid travel time among all modes of transport.
Figure 5.3 shows a comparison of accessibility between the baseline scenario and the scenario with AMoD and automated public transport. The accessibility of one zone is defined as

\[ A_i = \sum_j o_j \cdot [\tau_{ij} < T] \] (5.1)

with \( i \) and \( j \) denoting zones in the city, \( o_j \) the number of FTEs at workplaces in each zone and \( \tau_{ij} \) the fastest travel time among all modes to go from the centroid from zone \( i \) to the centroid of zone \( j \). The threshold \( T \) is defined as 10 and 15 minutes in Figure 5.3. The map is based on travel times at off-peak (15:00) as almost no differences can be seen for peak hours when the system is under stress. However, at the shown time accessibility is increased in most parts of the city due to the presence of automated public transport and AMoD.

For the case of 15 minutes, a gap in accessibility can be seen in the south-eastern part of Zurich because the hilly region is not easily accessible by public transport. As the AMoD fleet still provides a service there, accessibility is increased. The AMoD system improves access to mobility for people living in this area.

To summarise the results:

- An AMoD system in Zurich has the potential to reduce the share of private cars on the roads but may also attract demand from aggregated public transport. Introducing automation and cheaper prices for public transport can counteract the shift from transit users to the AMoD system.

- While car ownership will drop with an AMoD system, VKT will increase by a large amount, as well as the number of vehicles driving at any point during the day.

- Although daily travel times increase slightly in the presence of an AMoD fleet, the overall accessibility to workplaces does not deteriorate. Even more, the system has the potential to improve access to mobility in areas underserved by public transport.

5.3 Further Context

After and during the initial results were obtained, additional studies with slightly different approaches and perspectives were added. The following
shall give an overview of those studies to put the previous results into perspective.

5.3.1 Dispatching algorithms

In (Hörl et al., 2019b), a study on the performance of different dispatching algorithms was performed based on the AMoDeus framework:

**Global bipartite matching** (GBM) solves a matching problem every decision epoch (usually around 120s of time in the simulation). For that, a list of open (i.e. not picked up) requests is obtained, including their coordinates. Furthermore, a list of all available vehicles, which do not have a customer on board is constructed. Afterwards, a cost for each vehicle to each request is calculated. The aim of the algorithm is then to match vehicles and requests such that the cumulative cost of the matching is minimal. The problem can efficiently be solved using the Hungarian algorithm (Kuhn, 1955). While one could perform routing and use an accurate network distance as the cost, the study uses the Euclidean distance. Hence, the matching can also be called *Euclidean global bipartite matching*. The matching problem can efficiently be solved using the Hungarian algorithm.

The **Feedforward Fluidic Optimal Rebalancing** (FF) strategy calls the Euclidean global bipartite matching step repeatedly after a *dispatching epoch*. On top, rebalancing instructions are executed every *rebalancing epoch*. Those instructions are based on request information from previous iterations. For that, the operating area is divided into zones, and for each zone and time bin, the incoming requests are tracked. The strategy then expects the same number of requests to arrive in any space/time bin as in the previous iterations and accordingly moves vehicles empty to those zones.

The **Adaptive Uniform Rebalancing Policy** (AU) works similarly as the previous strategy, only that no historical information is available. Rebalancing is based on the current number of open requests in each zone but distributes vehicles uniformly over all zones in times of low demand.

The GBM strategy is especially interesting when studying the performance of fleet control algorithms as it provides an approximative lower bound in operational costs by minimising the empty distance of the fleet.
Figure 5.4: Waiting times for various fleet control policies. Source: Hörl et al. (2019b).
On the other hand, FF tends to provide lower waiting times, and AU additionally distributes waiting times more evenly in the system. Note that the three algorithms are designed such that vehicle-request assignments are not binding, i.e. vehicles can be reassigned in each decision epoch.

The fourth strategy assessed in (Hörl et al., 2019b) is the Load-balancing heuristic (LBH) approach by Bischoff and Maciejewski (2016b) which was also used above. Note that this algorithm works with binding vehicle-request assignments.

The study was performed on an uncongested network. Hence the results presented above are not directly comparable. Furthermore, a static demand case is examined in which the AMoD service serves all viable trips by car and public transport in Zurich. Nevertheless, (Hörl et al., 2019b) sheds light on the relative performance of the LBH algorithm used. It shows that for off-peak times the LBH works similarly as the GBM, while it provides waiting times that are slightly shorter when the system is under stress (Figure 5.4). However, the LBH does not reach the waiting time performance of the algorithms with rebalancing steps.

In terms of empty distance (Figure 5.5), the LBH produces naturally more empty distance than GBM and performs between the two rebalancing algorithms. Hence, the LBH is an easy to grasp and easy to implement strategy, that provides a good trade-off between waiting time and empty distance.
distance. However, (Hörl et al., 2019b) shows that the FF algorithm can outperform the LBH in both dimensions. It is therefore definitely possible to reach better waiting times at a lower empty distance with a more robust algorithm. While it would be interesting to repeat the analyses above with other fleet control policies, such simulations remain for future research. One substantial obstacle for this research is that the algorithms mentioned repeatedly need to solve rather substantial optimization problems. While the LBH is a very performant heuristic that can be called at many time steps in many iterations of a MATSim simulation, the other algorithms were so far only used in a static demand context, where simulating one single day is sufficient.

5.3.2 Pooling potential

A recent study (Balać et al., 2020) assessed the potential of ride-pooling in Zurich. Again, a static demand case is assumed where all car trips within the city of Zurich are transformed into trips with an automated shuttle system. The study is not based on MATSim simulations but uses the daily travel demand from the synthetic population. First, all car trips in the population between 16:00 and 18:30 (the evening peak) are selected and assigned into seven-minute time bins, dependent on their departure time. Furthermore, their start and end locations are assigned to spatial bins in a hexagon grid that is spanned over Zurich. The hexagons have a radius of 350m.

The idea of the approach is then that people need to walk to predetermined departure hubs of the service, which are at the centroids of those hexagons. This means that in the worst case, people need to walk for 350m. Furthermore, the shuttles always leave at fixed times, in seven-minute intervals and all passenger trips are served without detour. By using real-world travel times for Zurich, it can be determined for each trip from which origin zone to which destination zone and from which origin time bin to which destination time the trip occurs. By aggregating trips from the same spatial and temporal origins and destinations, flows in a space-time graph can be obtained. It is then possible to define a minimum flow optimization problem that determines the minimum number of vehicles to serve this demand. The full methodology is explained in detail in (Balać et al., 2020).

The study finds that a fleet of 10,607 individual taxis can serve the evening peak for Zurich with an increase in VKT of 3.5% compared to the baseline case with 162,648 cars. In case pooled two-seaters are used, 9,568 taxis
would be necessary, and VKT could be reduced by 2.5%. Hence, pooling people together would lead to lower numbers of cars on the road and also help reduce infrastructure deterioration. Finally, the methodology also makes it easy to calculate optimal fleet mixes. It concludes that a fleet of 9,152 two-seaters, 229 five-seaters, and nine ten-seaters could serve the demand with a VKT decrease of 3.2%.

While these results show that there is a potential for pooling, there is also a limit to how much efficiency can be gained, even if everybody is forced to use the service. In the light of the results above the question becomes how attractive such a system would be if decision-making users respond to the offer. Considering expected wait times of around seven minutes, Figure 5.1 shows that already much demand is lost at that point. A large fleet, as calculated in (Balać et al., 2020), could not be sustained at that demand level. It would, therefore, be interesting, to find a fleet control policy that can replicate the method from (Balać et al., 2020) dynamically in the simulation. Such a study is currently in preparation.

5.4 DISCUSSION

To put the results obtained for an AMoD system in Switzerland into perspective, it makes sense to return to the initially defined categories of infrastructure usage, demand effects, access to mobility, and ecological questions.

Regarding infrastructure use, the studies show that there is a substantial increase in VKT and flow, but there is also a reduction in the number of cars. The former is the most severe problem as every driven kilometre poses externality costs. In 2016, the Federal Statistical Office estimated that one kilometre driven on public roads leads to 0.06 CHF/km in infrastructure costs (BFS, 2019b). According to this number, the worst case (increase in VKT of around +50%) would generate infrastructure costs of 18,000 CHF per day. In case these costs have to be recovered to 100% by the AMoD prices, an increase in the cost of around 0.22 CHF/km would be necessary. Experiments that take such a policy into account would be interesting to conduct.

The studies on dispatching algorithms have shown that the chosen algorithm is not optimal w.r.t. empty distance. It would, therefore, be interesting, to repeat the study, for instance, with the Global Bipartite Matching algorithm. This would lead to lower availability and less equitable access to the system. It is hence interesting to see how the decision-makers would
respond in such a case. Especially the case where infrastructural (or ecological) costs are recuperated from the price asked from the customer this would lead to interesting trade-off situations.

A recent study performed with MATSim and AMoDeus shows that a centrally controlled fleet of vehicles is able to mitigate traffic by intelligently choosing routes that avoid self-inflicted congestion (Lu, 2019). The study shows that this holds even in the presence of selfish car drivers. Hence, the increase in flow does not need to be regarded as an infrastructural challenge. Similar studies have shown that automated vehicles are able to help to smoothen traffic substantially (Rios-Torres and Malikopoulos, 2017).

The overall decrease of cars in the city is good news for urban planners as their hope for a future with automated vehicles is that much public space can be reattributed to active mobility and recreation for greener and more liveable cities. Still, parking space needs to be provided for the remaining vehicles. Interestingly, one of our studies (Ruch et al., Under review) indicates that parking can be combined intelligently with rebalancing. While a restriction of the simulation to support parking adds ever more empty distance to the system, the increase is predicted to be low while having almost no impact on waiting times, even if parking infrastructure is limited to only a few strategically placed locations in the city.

In any case, an interesting additional point of analysis would be to look at the emissions caused by the fleet in different configurations with electric or combustion engine vehicles. This should include not only CO2 or particular matter emissions, but also noise and congestion. An interesting exercise would then be to internalize these costs into the AMoD service price. Automated mobility poses an excellent opportunity to make people aware of the full cost of their travels as substantial long-term investment costs for car ownership are removed from the equation. Including the reduction of cars all together into a full life cycle analysis could further inform policymakers how ecologically friendly an automated on-demand taxi fleet would be.

In terms of demand effects, the simulations show a steady shift from private car use to the AMoD system. This can be seen as favourable if a way can be found to operate the system with a smaller overhead in externalities. The shift from public transport to AMoD is more problematic as it indicates a shift from aggregated welfare-based public transport system to further individualized traffic. A full cost-benefit analysis exploring the trade-off between busses, trams and trains in comparison to an AMoD fleet is still pending. However, Sieber et al. (2020) show that in some rural
cases in Switzerland, a small AMoD fleet may be the more cost-efficient transport alternative to a train while providing a better level of service. For high demand as in urban environments, however, the study indicates that conventional public should be better.

Regarding public transport, it must be noted that no intermodal travel was simulated in the simulations so far. As the respective components are now available, it would be interesting to investigate how the system metrics would change if people had the possibility (or even were restricted) to use the automated taxis to reach public transport facilities. Generally, such a system promises to substantially increase accessibility to remote areas. It would be interesting to study the environmental effects and economic feasibility of such a service using the simulations presented above. The topic can furthermore be extended by not only viewing AMoD as an addition to the existing public transport system, but even by designing both systems in a complementary way (Pinto et al., 2019).

From the discussion, several pathways for future research have been identified:

- Dynamic demand simulations with different fleet operating policies to show the effect of different optimization objectives.
- Intermodal simulations with restricted and unrestricted first/last mile access to public transport facilities.
- Recovery of infrastructural costs and environmental and societal externalities through pricing schemes.
The first two sections of this chapter are based on the conference paper

Hörl, S., C. Tchervenkov and M. Balac (Under review) A generalizable pipeline for agent-based transport models in France, paper submitted to the 9th Symposium of the European Association for Research in Transportation, Lyons, September 2020

The case study on Paris is based on a peer-reviewed conference contribution:


The previous two chapters of this thesis have covered the development and use of agent-based transport simulation. As is typical with MATSim and other agent-based transport models, the simulation is strongly tied to a specific use case - in this example, Zurich. It is, therefore, interesting to pose the question of how adaptable the whole process is to other use cases.

The pipeline for Switzerland was developed in a very extensible way that makes it easy to change and replace certain parts of the modelling and simulation pipeline. For a project around Urban Air Mobility, it has been adapted for the case of Île-de-France. Since the available data sets are similar to those that are available for Switzerland, setting up a pipeline in the French context was relatively quick. From this work, a powerful scenario synthesis pipeline has emerged that is continuously evolving and improving. Various models have been created so far or are currently under development:

- Sao Paulo (Sallard et al., Under review)
- San Francisco and Los Angeles (Balac, Under review)
- Jakarta (Ilahi et al., 2019; Ilahi and Axhausen, 2019)
• Lagos (Kagho and Axhausen, 2019)
• Montreal and Quebec City
• The French region Occitanie

The central idea of the population/scenario synthesis pipeline is to allow researchers to go from raw data to full agent-based transport simulations in one consistent stream of models and transformations (see Figure 6.1). Ideally, all elements in this process should be published as open-source code and data should be open and publicly available as well. This way, it would be possible for anybody interested to gather the publicly available data, run the code and reproduce a synthetic population and mobility scenario that has been used in research elsewhere.

Such a process has many advantages. First, research becomes reproducible, and results can be verified. While this should be the standard, it is often not possible when it comes to agent-based transport simulation. The same applies to applied planning projects, which could be performed more transparently if the entire process of setting up the required simulations were open.

Second, researchers usually work on rather specific elements of such a pipeline. For instance, there are many algorithms for population synthesis, such as the examples described by Saadi et al. (2016) or Sun and Erath (2015); Sun et al. (2018). Those publications usually focus on a specific part of the pipeline. However, it is not clear to which extent the algorithms can improve the overall quality of full-stack transport simulations. Hence, a pipeline, as proposed by us, would allow researchers to test their algorithms in an integrated environment.

Third, building a mobility scenario can become a rather complex project with many steps and dependencies between models and data sets, which makes it challenging to keep track of changes and intermediate alterations of algorithms. A pipeline as shown in Figure 6.1 therefore makes it possible to apply integration tests to the pipeline, which will alert the user in case small changes of parameters or algorithms lead to (drastic) changes in the output data.

The next section introduces the MATSim scenario for Île-de-France that is based on such a pipeline including an overview of available data sets and methods used. Afterwards, a case study for an AMoD service in Paris is presented.
Figure 6.1: General structure of the scenario pipeline.
6.1 A Matsim Scenario for Île-de-France

In the following, the scenario synthesis pipeline for Île-de-France shall be introduced. While here a broad overview of the process is given, upcoming work will describe all steps in detail and perform a thorough analysis of synthesis errors. Furthermore, this upcoming research will explore the variance of population attributes across multiple realizations of the synthetic population and how it evolves when downsampling the population to allow for faster simulation.

The pipeline aims to start with raw data sets, to transform them, to apply further models, and arrive at a final running agent-based transport simulation. We intentionally use straight-forward algorithms in the current phase to establish a baseline against which future implementations of more complex and elaborate algorithms can be tested.

The proposed pipeline for Île-de-France consists of many steps that each draw from a specific data set and apply a particular algorithm to make use of the data in a synthetic transport scenario. In France, a detailed census data set is made publicly available every year by the statistical office (INSEE, 2018b). The data set contains sociodemographic information about persons grouped into households for the whole country. Households and persons are given as single observations, but they can only be geolocalized at the level of statistical zones. Such zones are defined such that each of them contains at least 200 inhabitants to allow for sufficient anonymization. Further anonymization measures are applied to areas with even lower population density. As the data set is not comprehensive, we use the household weights that are provided by the statistical office to scale up the population. It is then possible to generate artificial persons in all Île-de-France with sociodemographic attributes such as the number of vehicles, household size (on the household level), and age, gender, socio-professional category, and employment status (on the personal level). Home coordinates are sampled at random within the statistical zone of each household.

Unfortunately, census data does not provide income information, which is essential to realistically model mobility behaviour and allows for equity analyses in the final simulations. Therefore, we use the publicly available Filosofi data set (INSEE, 2018a), which aggregates tax data for all France and provides decile-based household income distributions for all statistical zones and municipalities. Currently, we use the data to sample household incomes dependent on the municipality for each household at random. In
Figure 6.2: Example of a MATSim simulation of Île-de-France. Visualization created in Simunto Via.

The future, the data set would even allow further refinements as distributions are also provided per household type.

Third, we use household travel surveys to attach activity chains to the synthetic population. For Île-de-France, the pipeline can use the publicly available national household travel survey (ENTD, Ministère de la Transition écologique et solidaire (2010)), which, unfortunately, is rather sparse as less than 5,000 person activity chain observations are available for Île-de-France. Alternatively, the regional household travel survey (EGT, DRIEA (2013)) can be used with around 35,000 activity chains. However, the EGT is not publicly available and must be obtained on request from the relevant authorities. Both data sets are rather old with the former being conducted around 2010 and the latter around 2012. It is possible to choose either of the data sets in the pipeline code. The chosen data set is then used in a statistical matching procedure to attach a daily activity chain to each of the synthetic agents. This is done by comparing the age, gender, socio-professional category, household income and car ownership attributes from the synthetic agents to the persons in the household travel survey. We then sample one chain dependent on weights given in the HTS from the set of all persons that match those attributes.
Fourth, we use origin-destination commuting data which is provided along with the census data (INSEE, 2018c,d). The flows are given on the level of municipalities and for work and educational commutes. Given the home municipality of each synthetic person, we sample a destination zone from the resulting OD matrix. In the future, this could be further refined by mode of transport as the data allows for that. A specific location is chosen with the help of the BPE (INSEE, 2019), which is a publicly available data set on all enterprises in France. Given the destination zone, a random observation is drawn from this data set to assign a specific coordinate to the "work" or "education" activities in a synthetic person’s activity chain. All other activities (secondary activities) are assigned the location of an observation from the BPE where, for instance, shopping, is available and in such a way that the distance distributions in the synthetic population follow the reference data from the household travel survey. This procedure was described in more detail above (see section 4.1.2.4) and in (Hörl and Axhausen, 2020).

Finally, the synthetic population is converted to the MATSim format. Additionally, we use an extract of Île-de-France from OpenStreetMap (Geofabrik, 2020b) to create a MATSim-format road network and also the digital public transport schedule for Île-de-France is converted to the simulation format. This schedule is made public by the regional public transport authority (Île-de France mobilités, 2020). At the end of this pipeline, the simulation can be run. Figure 6.2 shows a snapshot of such a large-scale agent-based simulation of Île-de-France.

This simulation is set up the same way as for the Switzerland case. Therefore, a choice model is needed to run it. Unfortunately, the regional household travel survey that was available for this research does not contain detailed start and end locations of the trips. Therefore, it is not easily possible to generate transport mode alternatives for all recorded trips, which would make it possible to estimate a discrete mode choice model for Île-de-France. For that reason, a Zurich model (different to the one presented previously) was applied to the Île-de-France scenario and adjusted to arrive at the correct mode share by Euclidean distance classes that are observed from reference data. Surprisingly, only few adjustments were necessary. This is probably due to the very general structure of the model (only a few variables are used) and the fact that travel decisions may mainly be affected by infrastructure availability and mobility tool ownership. The calibrated model for Île-de-France can be seen in Table 6.1. As a first sanity check, it can be stated that the VTTS in the calibrated model of around 19.40
6.2 Case Study: Automated Taxis in Paris

The aim of the future scenarios is to obtain an idea about the demand for AMoD travel in Paris. For that purpose, first, an operating area needs to be chosen. For the sake of simplicity, only the city of Paris within the highway ring is served by a fleet of automated single-occupancy taxis (see Figure 6.3). Future studies may be based on a more sensible choice of operating area as will be explained below.

EUR/h compares well to the numbers that have been reported previously for Île-de-France (Meunier and Quinet, 2015).

In general, we observe that the pipeline is mainly based on data that is available for the whole country: census, household travel survey (in case the national one is used), tax data, enterprise census, and OpenStreetMap. Only public transport schedules in GTFS or similar format are not available everywhere in the country, and preferably a more detailed and up-to-date household travel survey than the ENTD should be used. Several French cities such as Toulouse and Lyon provide such data sets, meaning that our open-source synthesis pipeline could easily be used to create agent-based simulations for these cities.

![Operating Area Map](image-url)

**Figure 6.3:** Operating area for the AMoD system in Paris. In the background the MATSim road network for Île-de-France is shown.
<table>
<thead>
<tr>
<th>Transport Mode</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Car</strong></td>
<td>$\beta_{ASC,car}$</td>
<td>1.35</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,car}$</td>
<td>-0.0667 [min]$^{-1}$</td>
</tr>
<tr>
<td><strong>Public transport</strong></td>
<td>$\beta_{ASC,pt}$</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>$\beta_{numberOfTransfers,pt}$</td>
<td>-0.17</td>
</tr>
<tr>
<td></td>
<td>$\beta_{inVehicleTime,pt}$</td>
<td>-0.017 [min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{transferTime,pt}$</td>
<td>-0.0484 [min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{accessEgressTime,pt}$</td>
<td>-0.0804 [min]$^{-1}$</td>
</tr>
<tr>
<td><strong>Bicycle</strong></td>
<td>$\beta_{ASC,bicycle}$</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,bicycle}$</td>
<td>-0.15 [min]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\beta_{age,bicycle}$</td>
<td>-0.0496 [a]$^{-1}$</td>
</tr>
<tr>
<td><strong>Walk</strong></td>
<td>$\beta_{ASC,walk}$</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td>$\beta_{travelTime,walk}$</td>
<td>-0.09 [min]$^{-1}$</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>$\beta_{cost}$</td>
<td>-0.206 [EUR]$^{-1}$</td>
</tr>
<tr>
<td></td>
<td>$\lambda$</td>
<td>-0.4</td>
</tr>
<tr>
<td></td>
<td>$\theta_{averageDistance}$</td>
<td>40 [km]$^{-1}$</td>
</tr>
<tr>
<td><strong>Calibration</strong></td>
<td>$\theta_{parkingSearchPenalty}$</td>
<td>4 [min]</td>
</tr>
<tr>
<td></td>
<td>$\theta_{accessEgressWalkTime}$</td>
<td>4 [min]</td>
</tr>
</tbody>
</table>

Table 6.1: Parameters for the choice model in the Île-de-France baseline simulation.
Three experiments are set up. In the first experiment, a best-response model is configured in such a way that all trips that can be served by the automated taxi service must be served by it. That means regardless of waiting times or travel costs, any trips in the daily plans of all travellers that are not otherwise constrained are converted to the AMoD transport mode. The experiment assumes that the service is so attractive that it is used in any possible case. The simulation yields, therefore, an estimate of the fleet size that is needed to serve the "maximum static demand scenario".

Second, the simulation is run with a multinomial logit model with the additional AMoD transport mode. The parameters of the utility function are set to the same values as for the public transport alternative since at the time of conducting the experiments no stated choice data or similar was available for Paris (and waiting time is valued like "transfer time" from the public transport alternative). The price per distance of the service is set to various constant values. The experiment, therefore, has the purpose of exploring the resulting demand in the system given the choice dependence on the level of service, fixed fleet size and a fixed price level.

The third experiment makes use of the detailed cost model of automated mobility. It has recently been applied to a range of cities worldwide (Becker et al., 2019). Unfortunately, to date, no specific assumptions for Paris are available. Therefore, the model specification for Berlin is used, which should be closest to the case of Paris among the available city-specific parameter sets. In the present case, the cost model is configured such that the service should always have zero net costs, i.e. no profit margin is included.

It is important to mention that in all experiments trips that are performed using the "walk" or "bike" transport mode in the baseline scenario are never converted to AMoD trips. Furthermore, constant travel times within 5-minute bins on all network links are enforced as measured from the baseline scenario. Therefore, effects in travel time changes due to fewer or more vehicles on the road are deliberately ignored here. This way, we avoid motivating additional assumptions on flow efficiency effects of automated vehicles for the case of Paris. All simulations are run using a 10% sample of the full agent population. This way, feasible runtimes are achieved.

The "maximum demand scenario" yields 2.3M trips that must be served by the AMoD fleet. This number can be compared to the waiting time-dependent cases, as depicted in Figure 6.4. The demand that is attracted once the choice model is in place and waiting time is considered, never reaches this maximum value. In fact, with a maximum of around 1.4M trips,
it is only slightly more than half the number of possible trips that could be converted to the AMoD transport mode.

The dependence of demand on fleet size is as expected: With increasing fleet sizes, more demand is attracted until it goes into saturation. The low number of trips for small fleet sizes are a direct consequence of the increased waiting times that are produced by smaller AMoD fleets. Also, the travellers’ reaction to different prices that are imposed on AMoD travels makes sense: The lower the price, the more demand is attracted in these simulations.

Finally, in the last set of simulations, the price is not predefined but calculated with the cost model. The "dynamic pricing" graph in Figure 6.4 shows that there is a demand maximum if the level of service and prices are considered: Larger fleet sizes lead to higher costs and, therefore, to less demand. In these simulations, we arrive at an optimal fleet size of around 25k vehicles which would be able to attract around 1.2M trips during one day. This is one-half of all trips that would structurally be possible.

The price at the demand optimal fleet size of around 25k vehicles is around 0.27 EUR per passenger kilometre (Figure 6.5). In comparison (Pelletier, 2018) a price of 0.3 EUR/km is listed as the cheapest full cost per vehicle kilometre for car ownership in France. Therefore, such a service could be highly attractive for today’s car users if their travels are mainly
confined to the proposed operating area. Unsurprisingly, today’s minimum taxi fare of 1.07 EUR/km (Service Public, 2019) in the city exceeds the AMoD tariff by far.

In terms of waiting times, the optimal fleet size yields an average of 3 minutes. Figure 6.6 demonstrates the influence of dynamic decision making on the demand side. For the static maximum demand case, much more vehicles would be needed to reach this number. To only push the median waiting time down to the level of the dynamic demand, 10,000 more vehicles would be needed.

Figure 6.7 shows how the fleet is used over the course of a day for different fleet configurations. Only the case of 15,000 vehicles goes into saturation during peek hours where every vehicle is busy driving with a customer or empty (which is shown in the figure) or picking or dropping off a customer. For the demand-optimal case of 25,000 vehicles the morning peek almost reaches the available amount of vehicles. Interestingly, the constrained case of 15,000 vehicles shows relatively less empty distance during peek times, mainly due to the undersupply mode, into which the dispatcher switches at those times.
Figure 6.6: Dependency of the waiting times on different fleet configurations in Paris.

Figure 6.7: Dependency of the number of active and empty driving vehicles for different fleet configurations in Paris.
GENERAL DISCUSSION

While the specific elements of the research around this thesis and beyond have been discussed in the respective chapters, a more general viewpoint shall be assumed in this last chapter of discussions.

First, a look at the applicability of the developed set of methods and tools should be taken, including pathways for future research and applications. Second, a more in-depth look at the topic of open data and open source software is provided as our projects around automated mobility, and transport simulation in general, are based on both.

7.1 APPLICABILITY OF METHODS AND TOOLS

The set of tools and assets developed during this research can be summarised as follows:

- A semi-standardised framework for population synthesis was created that has the aim to make agent-based transport models more reproducible and reliable. By defining standard components of processing input data, transparency about which analyses make sense for different use cases with different data availability and chosen approaches is achieved. The code for the scenario pipeline is publicly available\(^1\), together with a detailed description of the Île-de-France scenario (Hörl and Balac, 2020b).

- Several very similar agent-based transport scenarios have been created for cities around the world. While Switzerland and Zurich, as well as Île-de-France and Paris, fall well into the scope of this thesis, other cases were set up for Sao Paulo, San Francisco, Los Angeles, and others such as Jakarta, Montreal and Lagos are in preparation. An open data version of the Île-de-France case will be made public soon.

- An extension to simulate automated vehicles for MATSim was developed and used above. While it initially had a different feature set

\(^1\) https://github.com/eqasim-org/ile-de-france
from the DRT contribution of MATSim, both approaches have converged in functionality. It remains a task for the future to merge both approaches to avoid double maintenance work. On top of this extension, the AMoDeus framework was created that makes it possible to quickly implement and benchmark new fleet dispatching algorithms with a limited feature set of MATSim.

- A new extension to make use of discrete mode choice models in MATSim was set up and repeatedly refactored to now provide an easy interface to implement arbitrarily complex multinomial logit models for mode choice in MATSim.

How can these tools and scenarios be applied outside of the scope of this thesis?

The scenarios and their respective pipelines will be the basis for a multitude of future studies. These do not necessarily need to revolve around future modes of mobility, such as automated vehicles or Urban Air Mobility. Even interesting from the perspective from the author would be to analyse and tackle problems that exist today and for which various solution methods are proposed. While studies on road pricing, superblocks and infrastructure planning and reattribution exist, agent-based models only now start to highlight different sociodemographic, socio-professional and spatial aspects. Even more, the synthetic populations do not necessarily need to be used in a transport simulation. Already in their basic form, they allow for a multitude of static analyses how certain user groups and areas can take part in mobility, where infrastructure such as hospitals or schools need to be built to have the most substantial impact on selected user groups, and so on. By putting the synthesis of such populations on a solid foundation, we aim to make research in this direction more transparent and reliable.

Some very concrete examples for potential future use cases with the Île-de-France scenario would be to add the Grand Paris Express project (Société Grand Paris, 2020) to the scenario, analysing the ever growing network of cycle express ways in the city, segmenting the city into superblocks like in Barcelona (Baussells, 2016), or testing (currently politically dismissed) plans for cordon pricing. It will be interesting to combine such projects with existing capabilities of MATSim such as emission analysis (Kickhöfer and Kern, 2015), congestion pricing (Kaddoura and Nagel, 2019), tolls (Zockaie et al., 2015), and internalization of transport externalities (Kaddoura and Nagel, 2018; Kaddoura et al., 2020). Another interesting topic is transport behaviour in rare events (Frei et al., 2014).
Work remains to be done in the synthesis process. In a currently ongoing follow up project (Livingston et al., Upcoming) population predictions (BFS, 2015b) were used to scale today’s synthetic population to a future state in many relevant directions such as the population per canton and age distributions. As with all the components that have been put in place at the beginning of the thesis maturing and generalising this functionality takes time but will be an interesting project and valuable addition to the synthesis pipeline.

The second topic for the simulation that shall be mentioned is freight and delivery traffic. While many studies and simulations look at the impact of AMoD systems for passengers, much less research exists on freight and delivery. For MATSim, extensions focusing on freight traffic exist (Schröder et al., 2011, e.g.) and are currently gaining much traction. While data for personal mobility is increasingly becoming available, data on commercial traffic is still scarce. For Switzerland, several data sets exist, such as the freight traffic survey (BFS, 2019a) and the delivery vehicle survey. Furthermore, a data set on cross-border freight traffic (BFS, 2015a) exists, along with another one for personal cross-border traffic (BFS, 2017a). While partly already implemented in the follow-up project it remains to consolidate and standardise the process.

The third large topic regarding future work on the synthesis pipelines is household mobility. While here no car passengers were simulated, the follow-up project (Livingston et al., Upcoming) currently considers them although they are assigned statically and are not allowed to perform mode choice. A true implementation of car passengers would require, on the one hand, a sensible simulation of social networks such that workers can synchronise to take a car together (Dubernet, 2017), and realistic household dynamics, on the other hand. Currently, the daily plans of household members do not react to each other. However, this would be necessary to estimate the impact, for instance of private automated vehicles as their daily schedules may likely be adapted to the whole daily plan of the household (see (Wadud and Huda, 2019) for a real-world experiment). Vice versa, household mobility would adapt to the new possibilities that an automated vehicle provides; see the research by (Xu et al., 2019b,a; Xu, 2019) for a methodological framework for household-level activity-scheduling with automated vehicles.

The topic of household mobility is strongly related to decision making. One future pathway for the discrete mode choice extension is, therefore, to not only look at per-person decisions but at decisions of the whole
household. It will be challenging and interesting to find ways to generate whole household plans that are synchronised to simulate joint trips reliably (Xu et al., 2017). Presumably, much inspiration can be drawn from recent research on the inclusion of social networks into MATSim (Dubernet, 2017). Besides household mobility, the DMC component still has a list of potential improvements and extensions. Only recently nested logit models have been added as an experimental feature, but in principle also even more complex model structures could be used. One interesting pathway could be to not rely on closed-form choice models but to add the possibility to use arbitrary choice models through sampling and simulating choice probabilities. While such an approach is still highly hypothetical, it would be interesting to test its feasibility in terms of run times and modelling performance.

On the simulation side, we see various use cases of the AMoDeus framework outside of the present study. Only recently research has been performed on real-world taxi data for Chicago and San Francisco (Ruch et al., Under reviewb). The study showed that a taxi system that is not based on individual selfish decision-makers (the taxi drivers) but instead is controlled by a central unit would be much more efficient and reliable than today’s system. While a detailed cost analysis is pending, these results could change how (human-driven) on-demand systems will look like in the future. Related to that Sieber et al. (2020) show with AMoDeus that certain rarely frequented rural train lines in Switzerland may well be replaced by an on-demand taxi service at a lower level of subsidies than today.

AMoDeus has been used in several programming challenges at machine learning conferences and therefore already provides a flexible networking API that is independent of the backend to control the vehicles. Likewise, an API in the other direction could be set up in which the readily available algorithm implementations of AMoDeus are used to control vehicles in another simulation or even in reality. A fleet operator that has equipped all drivers with GPS devices and apps could, therefore, start using intelligent dispatching algorithms for the vehicle fleet right away.

From another perspective, AMoDeus could be used, like in the study on Chicago and San Francisco, to assess the performance of fleet operators in a city. By simulating an operator’s requests with sensible benchmark algorithms, such as empty distance-minimising Global Bipartite Matching, one could assess how much externalities could potentially be saved. This could become a tool for cities and policymakers to assess ride-hailing services and incentivise more ecological and equitable operation of the service.
Traditionally, transport planning tools have been closed software. One needed to buy a license from a company that may have charged for support and additional components. The advantage of this model was that quite sophisticated planning tools have emerged with ever-larger feature sets. Many of these tools have been tested over the decades and are very mature and widely accepted in transport planning today. However, the striking disadvantage is that eventual bugs can not be fixed easily by anybody as the source code is not accessible. Also, it is not possible to verify from the outside whether the offered algorithms and methods are implemented correctly or if there are undiscovered bugs.

Over the past years, agent-based transport models have gained increased interest due to their ability to assess emerging dynamic mobility concepts better as described in the introduction of this thesis. Interestingly, most of the known frameworks are open source\(^2\), which may also be connected to the fact that they are still mostly in academic use and therefore bound to open access research practices. Unfortunately, this also has the side effect that the tools are slow in becoming stable and ready for productive use. However, the open structure of those projects makes it possible to verify and validate the models and their components, and they can be used freely by anybody. Given lower learning curves in the future, this widens the circle of potential users of transport planning tools.

MATSim is a good example of such an open-source agent-based transport simulation. Over the years, the framework has seen constant improvement and consolidation by many people around the world. Therefore, the framework has become a feature-rich and popular tool inside of the transport research community. As the code is open and anybody can use it, more and more universities and institutes are joining the user base. Increasingly, companies such as the Swiss Federal Railways (Scherr et al., 2018) or Airbus (Rothfeld et al., 2020) have become interested in the software. By that, they start to take part in the sharing-based open-source environment.

Interestingly, the quick development of MATSim also poses a risk as can be seen with the parallel development of the DRT contribution and the AV extension that has been discussed in this thesis. There is a general lag in the documentation compared to the actual codebase, and extensions frequently become abandoned once projects are finished or the academically employed

---

\(^2\) For instance, MATSim (Horni et al., 2016), SUMO (Lopez et al., 2018), or Polaris (Auld et al., 2016)
developers part ways with MATSim on their career path. It is therefore vital
to institutionalise the development and management of the software such
as has been the case with known open source products such as Ubuntu
or Mozilla. For that reason, a foundation to support the development and
maintenance of MATSim is currently being set up. This will potentially lead
to more structured updates, documentation and teaching material and to
an entity that can follow current developments around the software. After
all, the software itself stays open-source such that there can be a breathing
ecosystem of experimentation and testing of new components and following
consolidation of promising and well-developed extensions which can be
led by the foundation. This institution will sustain the idea of open and
reproducible transport research. It may likewise step by step transform
transport planning to become more transparent and less dependent on
proprietary software.

The situation is different on the data side. Traditionally, transport-related
data was only available to authorities, companies and entities perform-
ing transportation studies. Only in recent years, relevant data sets, such
as census data, household travel surveys, road vehicle counts, passenger
frequencies, have become openly accessible. The underlying idea in most
cases is that data that has been obtained by the action of and about citizens
should be made public by the respective authorities.

However, there is not yet much experience with working with those data
sets systematically. This is also the case because data sets have been indi-
vidually structured, which made it difficult to find standardised processes
and methods of working with them. Furthermore, it was not clear which
attributes are commonly available in similar data sets. In recent years, how-
ever, the design of surveys and resulting data, for instance for household
travel surveys, has seen a somewhat informal standardisation as similar
questions are now asked for different cities and countries.

These good practices do not leave room for any transport research acting
in the dark about the origin and form of the data used. Wherever possible,
use should be made of open transport data, and alternatives to proprietary
data sets should be sought. Only such practice can make transport research
on agent-based models truly reproducible and transparent. Standards have
to be set on how to work with open data for agent-based models. The
pipeline developed in this research is meant as a first step into that direction.
Eventually, by having high-quality and useful open simulation scenarios
and open simulation tools emerge from the research community, also the
practice of applied transport planning could become more open, transparent and participative.

As this study focuses on two different use cases, it is interesting to put the considerations from above into perspective as the data situation in France and Switzerland is very different. This is not because one country would share more data than the other, but rather how their privacy policies are defined.

In France, a vast amount of open data sets is available ranging from a very detailed census data set to household travel surveys, tax registries, count data for all vehicle counting locations in Paris. This made it easy to create a MATSim model for Île-de-France, and it even makes it possible to publish code that reproducibly can create the scenario from these raw data sets. We hope that this way Île-de-France can become a test field for new transport simulations, all giving new insights on how to improve the transport system of Paris. This can be a unique value for the region, and hence it shows that it is a smart decision of the authorities to take action in open data. From the census data, not only transport research is profiting, but many more actors can get involved to make good use of the data. The French statistical office furthermore publishes guidelines on how to structure surveys and provides a list of common sociodemographic attributes that should be included in any survey conducted in France.

The situation in Switzerland is even better for researchers, as detailed, often un-anonymised data is available that exceeds the level of detail of the French data by far. For instance, the STATPOP census data set contains the age, gender and home coordinates of all Swiss residents. In sparsely populated areas it is therefore theoretically possible to identify individual persons. As no anonymisation is performed, it is impossible to publish results below a certain level of aggregation, which is undoubtedly violated by a synthetic population as the one used in this research. Clearly, this also prohibits the release of an open agent population of Switzerland. As a research institution, one would need to prove that published data is sufficiently anonymised. The problem with that is that rigorous anonymisation techniques (see, for instance, Samarati and Sweeney (1998), and Fourie (2017) in the context of MATSim) are costly and often impossible to verify on such large data sets. Hence, the decision about whether a data set can be considered anonymised is political. The French approach, where an already anonymised data set is made available to everybody (and not only to researchers), is to be favoured in most cases. In that regard, Swiss authorities could try to learn from their neighbours and assess whether the
anonymisation methods applied to the French data would be suitable in the Swiss context as well.

In any case, France, as well as Switzerland, make it possible to work on agent-based transport simulations in principle. Many other countries in Europe do not provide such rich sources of information for research, either because respective data sets do not exist at all, or because more restrictive sharing policies are in place (like in Germany, for instance). Another remarkable example of excellent open data policy is Sao Paulo, where we found that it is also possible to build a MATSim model entirely from open data (Sallard et al., Under review).
CONCLUSION

In this thesis, an agent-based transport simulation for Zurich was introduced, and a future scenario of the city with an Automated Mobility on Demand (AMoD) was simulated. The simulations made use of a detailed cost study on automated mobility services in Switzerland and a large-scale survey on potential adoption and usage patterns of people in the canton of Zurich.

The simulations show that an AMoD fleet of around 7,000 vehicles would attract the highest demand, leading to generally lower numbers of cars that are used in the city. Two scenarios, one where automation-related price reductions for public transport were considered, and one where prices are defined as today, show that the AMOD fleet would increase the vehicle kilometres travelled (VKT) on the roads by a large percentage of 30% or even 50%. Likewise, at any time of day, more vehicles would be populating the roads. While the reduction in overall cars is good news for urban planning as parking space can be reallocated, the strong increase in infrastructure use indicates that operators should take financial part in maintaining this infrastructure, especially if they make a profit from using it.

The same methodology was applied, in less detail, though, to the case of Paris, showing the adaptability and versatility of the approach.

Throughout the research project, many new components for the MATSim simulation framework have been developed, as well as accompanying tools. These are a component for the use of discrete mode choice models in MATSim, which are commonly used in more traditional transport simulation models; a component to simulate automated taxis in MATSim, together with a novel framework for control engineers to easily test fleet control algorithms; and a software package to set up pipelines for the creation of MATSim scenario data.

Ironically, while a strong focus in all the developed components was put on open software and open data, the simulations for Zurich are not easily replicable themselves as Swiss data privacy policies apply. Nevertheless, all tools are provided online, and the example of Île-de-France shows how to set up a simulation only with open data. Therefore, I hope that this thesis contributes to more reproducible, verifiable and robust practices in agent-based transport modelling.


Hörl, S. and M. Balac (2020b) Reproducible scenarios for agent-based transport simulation: A case study for Paris and Île-de-france.


Ruch, C., S. Hörl, R. Ehrler, M. Balac̆ and E. Frazzoli (Under reviewa) How many parking spaces does a mobility-on-demand system require?, Working paper, IDSC, ETH Zurich, Zurich.


