Simon Plakolb, BSc.

Towards Sustainable City Traffic
-
Agent based simulation of commuter impact on urban traffic emissions

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Environmental Systems Sciences/Natural Sciences - Technology

submitted to

Graz University of Technology

Supervisor

Univ.-Prof, Dr. phil. Manfred Fuellsack

Institute of System Sciences, Innovation and Sustainability Research

Graz, 03.2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Acknowledgements

First and foremost I want to thank Prof. Manfred Füllsack for the chance to work under his supervision and in his working group and moreover for providing computational ressources and help whenever troubleshooting or coordinational decisions had to be made while always sharing an open mind for new approaches. The systems theoretical thinking and deep knowledge of agent based modeling and network theory gained through his supervision lead my mind throughout the thesis.

I feel especially obliged to Georg Jäger and Christian Hofer who were the core developers for the model expanded in the course of this thesis and were despite being busy themselves always right at hand when support for their code was needed, some part of the program failed or if much more general problems occured. Christian Hofer with his remarkable knowledge of computer science and Georg Jäger with his unmatched mathematical and algorithmical clout both incouraged me to perform at my very best, which the good will of both only amplified even more.

Furthermore, I want to thank Marie Kapeller for having great patience with me whenever I couldn't find an overseen problem. For valuable suggestions and help regarding the parts of the thesis slipping into geographical informations systems I want to thank Laura Zilian. All in all, I felt very much cared for and always in good company within the working group. Thus, I want to thank all of its participants from the bottom of my heart!

Finally, during my whole studies I had great company, whom I want to thank for their pactience in hard times but especially for all the good and fun times. I am grateful for my family and friends, who are the corner stone in my life, for my studies and for my thesis as well.

Thank you all, indeed!

Abstract

Traffic models are used in a multitude of transportation planning applications and environmental modelling. Nevertheless, they have so far been expensive to conduct both in terms of computational cost and time consumation due to a large amount of data being immanently needed in order to support the models. A novel approach was introduced by Hofer et al. [2017] to remedy both of these shortcomings and subsequently succeeded in predicting the influence of various scenarios on CO_2-emissions due to traffic and congestion in the City of Graz.

However, the commuters taking part in this urban traffic scenario have so far only been implemented by hand, exploiting expert knowledge on the city in scope and its travellers. Therefore, although being reliant on less data the portability of the model to other cities was hampered by the desideratum for local knowledge about the traffic situation and the commuters.

Hence, to enable the adaptation of the model for other cities and diminish any bias introduced by human error, a commuter model serving as an extension to the original model of Hofer et al. [2017] is devised. It is fitted to the original commuter model and subsequently deployed to model the congestion arising in the City of Salzburg as a case study, proving that the gain in portability strived for was successfully accomplished.

Additionally for both, the City of Graz, for which the model was initially created and the City of Salzburg, the impacts of the commuters on the CO_2-emissions of the car traffic in the city are assessed and differences depending on the origins of the commuters are explored.

Kurzfassung

Verkehrsmodelle finden sich in einer Vielzahl von Einsatzmöglichkeiten der Transportplanung und der Umweltmodellierung wieder. Dennoch wohnen ihnen einige Hürden inne, welche eine weitverbreitete Nutzung erschweren, denn die Algorithmen traditioneller Verkehrsmodelle beruhen auf einer Vielzahl von Daten und benötigen des weiteren einen hohen Rechenaufwand.

Ein grundsätzlich differenter Zugang gewährte es Hofer et al. [2017] diese Hindernisse zu überwinden und ein agentenbasiertes Modell vorzustellen, welches mit einer geringen Datenmenge erfolgreich die Verkehrsbelastung der Stadt Graz und in Folge auch den Einfluss verschiedener Zukunftsszenarien auf die Emissionen, welche durch den Verkehr und Staus in der Stadt entstehen, abzuschätzen.

Allerdings umfasste dieses Modell PendlerInnen bisher in händisch implementierter Form, indem die Ortskenntnis der Autoren zum Tragen kam. Dies verhinderte die ansonsten durch das geringe Bedürfnis nach Daten erleichterte Portabilität des Models zu anderen Städten.

Auf Grund dessen, wird ein PendlerInnenmodell vorgestellt welches als Erweiterung zu Hofer et al. [2017]'s Modell dient und in Folge parametrisch an deren ursprüngliches PendlerInnenmodell angepasst wird. Daraufhin wird durch die Anwendung des erweiterten Modells zur Vorhersage der Stausituation in der Stadt Salzburg gezeigt, dass die benötigte Portabilität erreicht werden konnte.

Schlussendlich werden sowohl für die bereits zuvor implementierte Stadt Graz, als auch für die nun zur Analyse vorbereitete Stadt Salzburg einige Szenarien simuliert, für welche die Abhängigkeiten der CO_2-Emissionen von der Herkunft der PendlerInnenströme und deren Differenzen diskutiert werden.

Contents

Abbreviations

ABM	Agent-Based Modelling
CO_2	Carbon Dioxide
DTA	Dynamic Traffic Assignment
FP	Functional Programming
GHG	Green-House Gas(-es)
GPS	Global Positioning System
GPU	Graphics Processing Unit
IBM	Individual-Based System
I/O	Input and Output
MAP	Minimum Angular Path assumption (Turner [2000])
MAS	Multi Agent System
OD	Origin-Destination data
OSM	OpenStreetMap.org (OpenStreetMap contributors [2017])
RDI	Route Directness Index (Ciscal-Terry et al. [2016])
sMAP	simplified Minimum Angular Path
sRDI	simplified Route Directness Index

Introduction

One of the main challenges facing mankind nowadays is the pollution of the environment and the thus caused climate change and its mitigation. While the driving forces for climate change are manifold, the impact of CO_2-emissions is an influential and mainly anthropogenic one. The transportation sector contributed 27% to the total amount of green house gas (GHG) emissions of the European EU-28's countries in 2016 and within this sector the fraction of road transport makes up for 72.1% and of this partial quantity 44% can be traced back to car emissions. (European Environment Agency [2018])

Cities take a special place in the challenges caused by traffic emissions, as they have the biggest impact on carbon dioxide expelled due to transportation. (Gately et al. [2015]) Graz serves as a prime example for a city where the air quality is harmed by traffic. It has suffered from repeated exceedance of the limitation for fine particle concentration in the air. (Heiden et al. [2008]) Furthermore, noise pollution, road congestion and a big impact on public space availability are all promoted by city traffic. However, cities also take a special place in policy and decision making as they often have direct access to policy instrument. (Rosenzweig et al. [2010])

To assess such decision possibilities and to plan the development of the city's traffic system in order to mitigate the problems caused by the emissions, simulations are needed. These provide policy makers with the necessary information about the impacts of various policies and network properties and features. (Hofer et al. [2018b]) Such simulations are carried out by traffic models, of which a myriad of different approaches exist.

Recently, a model was brought up by Hofer et al. [2017] utilising a novel approach reminiscent of Monte Carlo simulations while combining agent based modelling with network techniques. While Hofer et al. [2017]'s traffic model exposes a lot of advantages compared to classical traffic models, it cannot yet make use of one of its most predominant features. Only relying on readily available data and being thus, in principle, applicable for many cities with low additional effort being taken, the model still calls for local expert knowledge for the implementation of commuters, which hinders the aforementioned application on different cities.

Therefore, the following research questions arise:

- How can for a given spatial network be deduced, where the exit and entry nodes, the sources and drains so to speak, are located?
- Can a simulation method be found that allows considering route choices of commuters inside a city road network while disregarding the outside network and can this method yield results corresponding to the performance of Hofer's model?
- Finally, can the model consecutively be used to assess different mid-sized Austrian cities?

In the course of the thesis, a model is devised to automate the process of commuter modelling and consequently extend Hofer et al. [2017]'s model, in order to lose the reliance on expert knowledge. Furthermore, with the extended model at hand, two case studies are carried out: One on the City of Graz to compare the model to the initial implementation and another one on Salzburg, in effect proving the newly gained applicability. Both of these case studies are also used to assess the impact of commuters on the CO_2-emissions in the cities in scope.

The thesis is structured as follows: The first chapter depicts a literature review on existing traffic models, as well as the behaviour of commuters and advances in modelling these. Furthermore, a closer look onto the initial model by Hofer et al. [2017] is taken.

The methodology used is then discussed in the second chapter, taking a closer look into agent based modelling and the Python programming language, as well as functional programming, which serves as a method for the extension of an existing code base, while prohobiting interfering with its working mechanisms.

Consequently, the formalisation of the commuter model is given in the consecutive chapter on the realisation of the research objective.

This model is then evaluated in two stages, first in a purely qualitative form by comparison of its results to graphs showing congestion, both from the original model by Hofer et al. [2017] and as given by Google Maps [2019] and second in a more quantitative way, coupling the extended traffic model with an emissions model proposed in Hofer et al. [2018b] and comparing the calculated CO_2-emissions results with the values stated in literature. Furthermore, the evaluation is also used to assess the impact of the commuters on the emissions in the cities the model is implemented for.

The thesis closes with a discussion of the results, as well as the implementation of the model and a concluding summary.

Chapter 1

Literature Review

To gain an oversight of traffic models in general and on how the model of Hofer et al. [2017] fits into the existing ecosystem of proposed modelling approaches, as well as on how the behaviour of commuters is structured and what can be learned about their choices, facilitating the subsequent task of modelling their decisions, a literature review on the fields of traffic and emissions modelling and on various studies on commuter behaviour has been conducted.

1.1 Literature on traffic and emissions modelling

Traffic systems constitute non-linear behaviour, which can be influenced by multiple external as well as internal factors and hence expose complex behaviour, in which the effect of humans is the main driving factor for the dynamics. (Zegeye et al. [2013], Erol et al. [2000])

Nowadays, traffic modelling is among the best studied topics in computer simulation. (Kotusevski and Hawick [2009]) Such models are used with a multitude of objectives ranging from emission and fuel consumption modelling (Jamshidnejad et al. [2017], Thonhofer et al. [2018], Zegeye et al. [2013]), over laboratory studies on driver behaviour (Chen and Mahmassani [1993], Mahmassani [1990]), to the assessment and planning of infrastructural changes (Kotusevski and Hawick [2009]). Thus, the working mechanisms of traffic models are plentiful and of high diversity. Nevertheless, they have been categorised into two classes depending on the approaches underlying their principles: Top-down or macroscopic models and bottom-up or microscopic models. (Kotusevski and Hawick [2009], Bert et al. [2006]) However, in between these two categories mesoscopic models have emerged. (Kotusevski and Hawick [2009], Thonhofer et al. [2018], Jamshidnejad et al. [2017])

In this section a closer look is taken onto existing traffic models of different scopes. Furthermore, emission models, which translate the input of traffic data into emissions data are discussed briefly, as one such model is needed in order to assess the environmental impact of the commuters, developing such a model nevertheless is beyond the scope of this thesis.

1.1.1 Bottom-up / microscopic models

For some usage scenarios bottom-up models are better suited, utilising the description of the behaviour of the traffic participants themselves and upon simulation depicting the emerging result of their movements and interactions. (Thonhofer et al. [2018]) In these models the task of creating the driving patterns in microsimulations is often referred to as dynamic traffic assignment (DTA). (Di and Liu [2016], Bert et al. [2006])

A prevalent fraction of such bottom-up models is based on the car following approach where, as the name suggests, cars are simulated driving on the streets of a road network obeying traffic laws and their exact movement is tracked. (Kotusevski and Hawick [2009]) The most used approach to implement this was proposed by Gipps [1981] and further refined by Morello et al. [2014]. (Samaras et al. [2018]) Such models are for instance the commercial software VISSIM (Fellendorf and Vortisch [2011]) or the open source distributed SUMO model (Lopez et al. [2018]). Some of these models reach an elaborateness to the extent that even a three dimensional live view is included, showing the traffic participants going about their business in a city like environment. (Kotusevski and Hawick [2009])

Nevertheless, with the high detail these models contain some severe detriments are introduced. As a first result of the exhaustiveness the computational cost is extensive. (Jamshidnejad et al. [2017], Thonhofer et al. [2018]) Additionally, there is a lot of data required to create the routes for the cars to travel on, which consist of an origin and destination and is thus called Origin-Destination (OD) data. (Hofer et al. [2017], Bonabeau [2002]) Collection of OD matrices can be done in various ways such as traditional surveys (Thériault et al. [1999]), or more novel gathering methods utilising mobile phone networks (Caceres et al. [2007], Larijani et al. [2015]) or their Bluetooth identification process (Carpenter et al. [2012], Barceló et al. [2012]). Therefore, the route can be given in various ways and often has to be manipulated in order to yield a usable basis for the simulation. This occurs for routes given in terms of landmarks as well as for GPS data which does often not directly correspond to the coordinates of the given road network. (Bierlaire and Frejinger [2008])

To estimate the destination choice of commuters the primary activity assumption can be used, where from a set of OD data one pair of origin and destination is chosen according to an activity previously defined as the primary for each separate agent. (Balmer et al. [2006]) An alternative way to generate the demand for bottom-up models is to use data on the surrounding area simulated utilising a macroscopic model which yields a good estimate. (Samaras et al. [2018])

Fontes et al. [2015] ascertained the most dominant pitfalls contained in microscopic traffic modelling and devised some best practices in order to avoid them. In effect, an upper and a lower limit to the length of links in a network was discovered, where at the lower limit some cars with high speed would not be detected due to moving over two links at one timestep and at the upper one, the spatial resolution was lost and high congestion at some intermediate stretches of the links was not depicted. Thus, Fontes et al. [2015] argue that a balance has to be made regarding link length and maximum speed travelled at road sections.

1.1.2 Top-down / macroscopic models

Top-down models utilise an equation-based description of the behaviour of the system dependent on various parameters and hence incorporate average driver behaviour for their modelling efforts. (Zegeye et al. [2013]) Often these models are based on the continuity equation, which can be exploited due to the fact that the number of vehicles is inherently constant. (Helbing et al. [2001]) Recent models mostly deploy differential equations from fluid dynamics. (Jiang et al. [2018], Thonhofer et al. [2018])

Macroscopic models are computationally faster and thus used in scenarios where quick results are necessary, e.g. in traffic control applications. (Zegeye et al. [2013], Jamshidnejad et al. [2017]) In addition, less input data are required than in microscopic models. (Jiang et al. [2018]) Nevertheless, still a need for OD data can be existent. (Thonhofer et al. [2018]) Like in microscopic models, concurrent computation can also be performed in macroscopic ones operating on a traffic network, by calculation of the equation system for each road link in parallel. (Thonhofer et al. [2018])

However, the macroscopic modelling approach yields no detailed behaviour of vehicles and hence, it is not possible to assess simulations on the microscopic scale. (Kotusevski and Hawick [2009]) The output of such a model consists predominantly of mean speed and traffic volume for each separate road link. (Samaras et al. [2018]) Recently, a model was devised omitting the road network for the simulation completely in favour of a two-dimensional continuous plane. Accordingly, the traffic network and its population are assumed to be dense enough to make such an abstraction. (Jiang et al. [2018])

1.1.3 Mesoscopic models

The middle ground between the formerly mentioned two categories is populated by mesoscopic traffic models. (Kotusevski and Hawick [2009]) These models emerged out of the strive for a faster realisation of simulations, while maintaining some of the detail and spatial resolution microscopic models comprise. (Jamshidnejad et al. [2017], Thonhofer et al. [2018]) Thus, instead of separate vehicles, groups of these are simulated (Jamshidnejad et al. [2017], Mahmassani [2001]), hence incorporating a variance reduction method reminiscent of implicit photon capture used in Monte Carlo simulations of light transport. (Wang et al. [1995], Kahn and Harris [1951]) The packets of vehicles travel in accordance with aggregated data of the vehicles they are composed of. (Thonhofer et al. [2018]) This can be beneficial especially in urban environments where macroscopic models do not provide acceleration and speed data and in contrast mesoscopic ones can do so, while simultaneously high computational speed can be guaranteed. (Jamshidnejad et al. [2017])

1.1.4 Emissions models

Emissions models use the information given by traffic models and their dynamic traffic assignment and provide information on fuel consumption and various emissions upon these datasets. They can analogously be divided into macro- and micro-simulation approaches underlying their working principles. (Zegeye et al. [2013]) Microscopic emissions models are also known as instantaneous emissions and fuel consumption models and thus do provide such instantaneous information. (Samaras et al. [2018], Jiang et al. [2018]) In contrast, macroscopic emissions models are average speed based. (Zegeye et al. [2013])

As one might expect, macroscopic traffic models are a good source for the data needed on macroscopic emissions models and microscopic DTA is better suited for instantaneous ones. Nonetheless, hybrid approaches combining models with different scopes started to emerge. (Samaras et al. [2018], Jiang et al. [2018]) The most attractive current avenue of hybridisation is to use a macroscopic or mesoscopic traffic model for DTA and couple it with an instantaneous fuel consumption and emissions model. This might also be due to the assumption that macro emission models are less accurate. (Zegeye et al. [2013]) Nevertheless, in a comparison of macro- and microscopic emissions models, Borge et al. [2012] found both to yield results of comparable inaccuracy. An overestimation of NO_x is subject to all modelling efforts evaluated so far. (Borge et al. [2012], Smit et al. [2010])

The task of preparing the data provided by models without detail on single driver behaviour for such an emissions model is non-trivial. Nonetheless, recently multiple frameworks for this process have been devised and shown to yield estimates accurate enough for their purpose. (Jamshidnejad et al. [2017], Zegeye et al. [2013]) It is shown that even with macroscopic data independent of an underlying network a translation for microscopic emission models can be conducted. (Jiang et al. [2018]) However, such simulations are only useful for assessment of the average car as they do not represent each single car individually. (Samaras et al. [2018])

A lack of models calculating instantaneous air quality measures was found by Fontes et al. [2015] and interpreted to be caused by the need for data on a link by link basis, thus spatially bound, where most microscopic models expose their data on a car by car basis. While macroscopic models do provide fitting data, air quality models incorporating these have to be based on average speed and thus lack the required detail.

1.1.5 Common shortcomings

As comes clear from this short overview on modelling approaches, nearly all of them incorporate a traffic network consisting of nodes where junctions, traffic signals or other structures intervene the traffic flow and edges depicting the roads between these nodes. Furthermore, to the author's knowledge none of these models is able to operate faster than real time during simulation runs, as even those devised with the scope on quick computation time only reach real time. (Jamshidnejad et al. [2017]) Nevertheless, the model proposed by Zegeye et al. [2013] is only evaluated using a single freeway stretch but performs with a hundredfold decrease in computation time and could thus yields faster results than that on a city network.

Moreover, most of the models rely on origin-destination data and can thus only be adapted to a city with foregone extensive data collection. However, the novel approach of the traffic model by Hofer et al. [2018a] which is to be extended by the commuter model presented in chapter 3 counteracts these limitations and yields close to microscopic results an order of magnitude faster than real time, without the need for origin-destination data.

1.2 The initial model

As apparent from section 1.1 there was a gap for a model which incorporates on the one hand no dependence on origin-destination data while on the other hand still being able to describe the traffic behaviour on the micro scale in a computational less demanding way. Hence, Hofer et al.

[2017] have proposed such a model, based on a blend of agent based simulation, Monte Carlo methods and some network techniques, combined into an efficient traffic simulation suitable for mid-sized cities.

1.2.1 Networks as basis

Most traffic models build upon a network comprising the streets as edges and their intersections and some other points of interest as nodes and so does the model by Hofer et al. [2017]. The data for the network was gained from OpenStreetMaps (OSM) (OpenStreetMap contributors [2017]) which has been shown to yield a useful source for map data. (Zilske et al. [2011]) The underlying network can thus be updated at any time given that a connection to OSM is abundant. To convert the map data into a valid network of nodes and edges, the Python library OSMnx by Boeing [2017] is utilised.

This software library has the ability to download the map area of a specified city if the boundaries are contained within OSM and exposes a network in the fashion of a MultiDiGraph class defined in the Networkx (Hagberg et al. [2008]) Python library. This kind of network is ideal for the representation of a road network in that it is directed, i.e. an edge points from one node to another in a given directionality, and it allows for multiple edges between two nodes, such that there can be two or more roads connecting two intersections and consequently, lanes in both directions or just in one, expressing one-way streets. Properties of the roads, crossings and other points of interest are stored in the corresponding class attributes of the edges, or respectively nodes. Such data can comprise e.g. speed limits, road width and length or the geographical location of nodes. (Boeing [2017], Zilske et al. [2011])

1.2.2 Agents as car drivers

With a network representation of the city's streets existent, there is the need for cars to populate them. These cars, or to put it better their drivers, are depicted by agents who travel among the edges from node to node to satisfy their necessities. (Hofer et al. [2017])

However, here Hofer et al. [2017]'s model distinguishes itself from other bottom-up traffic models discussed in section 1.1.1, in that it does not need any form of origin-destination data to accomplish the simulation of the agents.

Instead, a way was found to generate these from much more basic travel information gained in a survey by the Austrian government, in which the participants recorded their travel distances, times and purposes and their used vehicle on two days in a year. In this survey called "Österreich Unterwegs" (Tomschy et al. [2016]) some 18000 participants took part, which left the generation of OD data with a dataset of over 36000 days of travel behaviour. Such a survey can be done more easily than one on actual origin-destination data and is for example readily available for the UK with about 20000 participants (Department For Transport [2017]) as well. Note that only distances and purposes were recorded and not the actual origin or destination and therefore, the data can also be transferred to and adapted for other regions which would be impossible with OD data which is inherently and stringently bound to the geographical area it was collected for.

The data can be further refined by picking only inhabitants of cities or the corresponding area for the generation of the OD data, however, while the difference between rural and urban inhabitants might be of bigger importance, the travel behaviour of Austrians could well be comparable to that of other European people of developed countries, as in a study on the travel behaviour of inhabitants of the area of Geneva (Palma and Rochat [1999]) no significant difference could be found to that of an earlier study conducted in Brussels (Khattak and Palma [1997]).

Additionally, the survey participants can be split into age groups, as there are sizeable behavioural differences which can be shown for these. (Hofer et al. [2017])

1.2.3 The Monte Carlo OD generation

Generation of origin-destination data is then carried out by creating an agent for every citizen of the assessed city and assigning to it an archetype depending on its age group assignment in Tomschy et al. [2016] and placing it on a random node corresponding to the population and age group distribution in the city's districts. (Hofer et al. [2017])

Consequently, the agents are sent on their journey, where they pick a random target in the distance assigned to them by their archetype's behaviour at the same daytime as the archetype would decide to undertake their travelling. To facilitate computation speed, every node has sets of nodes for given distances stored, in effect no shortest path search has to be performed. Note that the route between them is not determined yet. (Hofer et al. [2018a])

The node randomly picked then becomes the starting point for the next journey and the process of picking another random node is again done by distance matching. This process is repeated in steps until the agent, i.e. its archetype, ultimately decides to go home and returns to the initially selected node. (Hofer et al. [2017])

Of course, while the foregone description is based on one single agent to ease the understandability, this computation is done analogously for all the agents depicting inhabitants of the city in each step, representing one hour in the modelled world.

1.2.4 Generated OD data as simulation basis

After the OD generation, the simulation of the traveller routes works by simply picking the fastest route between the nodes in the dataset. The congestion of streets can then be calculated by finding the hourly traffic capacity for each road section C_h (Höfler [2004]) which is given by:

$$C_h = L_{eff} * 750 \frac{cars}{hour} \tag{1.1}$$

where L_{eff} denotes the effective number of lanes and 750 depicts the conversion factor yielded by the conversion to an hourly capacity. L_{eff} can either be given by the actual number of lanes, or if the lane count is not given in the edge properties it can be approximated from the road width with according to Höfler [2004]:

- $Width > 7.5m : 2.6$
- $5.5m < Width < 7.5m : 2.0$
- $Width < 5.5m : 0.8$

Finally, to evaluate the congestion of one simulated hour the load quotient a can be found by setting the number of vehicles on the corresponding road section in that particular hour V_h into relation to its hourly traffic capacity (Höfler [2004]):

$$a = \frac{V_h}{C_h} \tag{1.2}$$

Dependent on the value of a different levels of congestion can be interpreted. Below a load quotient of $\frac{3}{4}$ the traffic is free flowing and above $\frac{9}{10}$ stop and go traffic is prevalent. A value between these boundaries can be interpreted as constrained traffic flow. (Höfler [2004])

This calculation of congestion serves as a tool for comparison to real world traffic jam emergence and as a first evaluation of the model's fit. As can be seen in figure 4.1 such a simple combination of generated origin-destination data in combination with fastest route picking yields a result comparative to the average congestion shown by Google Maps. (Google Maps [2019]) Consequently, different scenarios could be carried out and their impact on CO_2-emissions of traffic in the mid-sized Austrian City of Graz were evaluated. (Hofer et al. [2018b])

1.2.5 The missing link: Commuters

Nevertheless, the model lacks the ability to assess the behaviour of commuters in a meaningful way. These are so far implemented in a rather trivial manner, assigning manually picked entry and exit nodes to the various counties the commuters are originating from. The agents depicting the commuters then choose randomly from this set of entry and exit nodes with a uniformly distributed probability over all the nodes. Furthermore, they choose a destination inside the network randomly and travel only twice a day: Once entering the city in the morning and once exiting it in the evening. The route in between these nodes is subsequently calculated as it is for the other agents which depict city inhabitants. (Hofer et al. [2018a])

Therefore, the simulation could so far only be carried out for the City of Graz, as this was the only city for which extensive expert knowledge was abundant to classify nodes as entries or respectively exits and assign sets of those to the different counties. Nevertheless, disregarding this commuter related problem, the model itself would be perfectly transferrable at least to other similarly sized Austrian cities, if not even to many European cities in general. Thus, a mechanism is needed to automate the simulation of commuters and in the course of doing so enhance the model, such that decisions or scenarios carried out on the investigated road network do influence the commuter agents.

1.3 Literature on commuter behaviour

Commuters can be considered a special case in traffic simulation, especially when looking at the traffic system emerging in and around a city. What sets them apart from the usual traffic participant is, that they can be modelled as taking only two trips a day, with one going into and one going out of the city. (Quarmby [1967]) Therefore, all of their journeys intersect the city boundary and thus, the boundary of the system in scope, if only the city interior is assessed as it is the case in the model presented by Hofer et al. [2018a].

This makes for a difficult modelling task as the simulation of commuters somewhat smears the system boundary. However, it is convenient to introduce a second system boundary and hence, create a coupled simulation with two interacting models each with its separate starting and boundary conditions, the latter of which of course have to meet at the shared system boundary as subsequently discussed in section 3.2.

Nevertheless, the commuters travel within both subsystems and thus, a means is needed to describe their behaviour in a traffic network with the information given by itself, as well as with the limited knowledge of just the geographical distance and the data given at the boundary of the network. Therefore, a literature review was conducted to gain an overview of the most important factors for commuter behaviour, their route choice and to find a path to the development of a simplified route algorithm.

Extensive research has been carried out in this area and the decision process of commuters has been looked into in multiple ways. Especially the behaviour of commuters under live traffic information systems is a well studied topic and will consequently be discussed in section 1.3.1. Another well discussed part among commuter decision processes is their mode choice which will be looked into more closely in section 1.3.3.

Furthermore, route choice modelling is a topic where sizeable efforts have been made to find descriptions of not just commuter, but driver behaviour in general. In section 1.3.2 an attempt is made to sieve out those aspects of the findings in this field important to the modelling of commuters.

However, in the proposed model neither a simulation of live traffic information systems, nor a possibility of a mode choice or varying route choice are existent, as it is a vast simplification of the system in scope.

Nevertheless, some literature on such immensely simplified formalisations is abundant and will be encapsulated in section 1.3.4.

Finally, gender specifics have been a thoroughly investigated topic. While Spyridakis et al. [1991] found men to fork off earlier than women, Mokhtarian et al. [2011] found female commuters to be more likely to change route and Matthies et al. [2002] came to the conclusion, that they are also more likely to switch their travel mode in order to reduce car use, with the most important aspect being ecological beliefs. However, Hess [2001] did not find any mode preferences due to gender. Furthermore, females were found to have their homes significantly closer to their working places than men. (Spyridakis et al. [1991]) Nevertheless, currently no gender specifics are included in the model expanded upon in this thesis.

1.3.1 Commuter behaviour under traffic information systems

Nowadays traffic information systems are overly present in our lives, more so than one would have probably expected before the advent of smartphones and highly sophisticated infotainment systems inside cars. However, the introduction of such systems and their influence on the behaviour of traffic participants has been an important research topic at the end of the last century. Thinking of the problem from the perspective of game theory, this can be interpreted as a transition from a game with incomplete information to one with complete information. (Klügl and Bazzan [2002])

Conquest et al. [1993] by cluster analysis of the behaviour under traffic information systems find four distinct groups of commuters. While some will stick to their predefined route completely, others decide upon their travel mode, departure timing and route choice before starting their trip, but do not change anything during their journey. The third and fourth group consist of commuters willing to only change their route while in effect having to adjust their time of departure as well. Interestingly, pre-route and en route changing commuters travel less distance on freeways and the latter also have shorter trip distances. The groups also differ in other aspects like importance of commute safety and enjoyment and length of both travel time and distance. (Conquest et al. [1993])

The number of drivers using more than one route to work varies from 15.5% (A. Abdel-aty et al. [1993]) to 33% with at least two routine route choices. (Li et al. [2005]) For these, the alternative route is used on 20% to 40% of the weekdays. (A. Abdel-aty et al. [1993]) Papinski et al. [2009] find routes for home-work travelling relatively fixed.

Nevertheless, Srinivasan and Mahmassani [2002] find 82% of commuters value a time saving of more than 15 minutes important enough to switch route and 43% think so about savings from 5 to 15 minutes. Furthermore, 81% of the participants in their study rate incidents as very important for route choice and 68% do so for congestion. Their willingness to change departure time depends among avoidance of lateness (61%), time savings (58%) and congestion (59%), on the possibility of undertaking other tasks on their way (46%).

The percentage of drivers changing their route due to congestion is situated around 30% for different scenarios, converging to that level with higher congestion (Srinivasan and Mahmassani [2002]), which is comparable to the findings of Palma and Rochat [1999]. This is higher than the factor of 0.1 found to be fitting the best in Hofer et al. [2018a]. However, this could also be caused by regional peculiarities, since congestion was found to have the most impact on the route choice of swiss commuters (Palma and Rochat [1999]), while in Ontario the most influencing factor is found to be travel time. (Papinski et al. [2009])

With the prominent availability of live traffic information systems, the adjustment of routes will no longer only be based on day-to-day experience, but dependent on the live congestion as well. (Srinivasan and Mahmassani [2002]) While in laboratory experiments still 50% of the drivers would not choose the best path if it differed from the current one (Srinivasan and Mahmassani [2000]), this could have an impact on the figures above in the future.

Others however, additionally find higher probability of route change with making stops on the commute, which could be related to accomplishing other tasks en-route. (Li et al. [2005], Bhat and Sardesai [2006]) It also increases with more flexibility in the required time of arrival and furthermore, an increase of alternative route choice with age and income has been discovered. (Li et al. [2005], A. Abdel-aty et al. [1993]) For drivers in the Francisco Bay area, it could be shown that the demand for a stop on a single day of the week during the commute journey, leads to an increase in the choice of the auto mode on all weekdays and additionally increases the probability of travelling alone. (Bhat and Sardesai [2006]) The average number of stops on a home-work trip of commuters in Geneva was found to be 1.8. (Palma and Rochat [1999])

Moreover, when people change their route, they are also very likely to change their departure time and the latter is more likely to be changed than the route. (Mahmassani [1990]) It is assumed, that the chosen departure time is in accordance with a strive for the minimisation of the total cost of late arrival and actual travel time. (Arnott et al. [1991])

One important driver in the strive for a fast commute to work is, that for people with fixed work times arriving late is perceived to have some five times the cost per hour compared to the value of early arrival. With flexible working hours this figure decreases drastically to about 1.5 times of what travel time savings would be worth. (Asensio and Matas [2008]) Thus, commuters can be modelled to take into account some contingency time to counteract uncertainties in their time of arrival. (Hensher [2001])

1.3.2 Route choice modelling

While route choice modelling does not focus on commuters per se, choices of such are of course modelled in this field as well and the literature reviewed here is mainly focused on home-work travelling and amplified by more general studies on traveller behaviour in route choice scenarios.

In route choice modelling often only perfect rationality, where the forecast choice of a driver is equal to the path with the highest utility, is considered. (Di and Liu [2016]) However, multiple papers suggest that the actual choice is often inferior to this optimal one, as drivers choose longer routes to avoid city centres (Ciscal-Terry et al. [2016], Thomas and Tutert [2015]), make less turns to travel a perceived shorter, more direct path (Thomas and Tutert [2015], Turner [2000]), or simply make mistakes in their choice by not being familiar enough with the road network, or for other reasons. (Manley et al. [2015], Di and Liu [2016], Xu et al. [2011])

This is accounted for differently. Bounded rationality considers not the route with the highest utility but the one yielding the highest satisfaction for the traveller as the most likely choice. Thus, there is a possibility with low probability that an inferior route is chosen. (Di and Liu [2016])

The imperfect choice may be implemented by using heuristics including a certain standard deviation to mimic the human factor (Manley et al. [2015]), splitting the utility into a random and a deterministic part (Ben-Akiva and Bierlaire [1999]), applying cumulative prospect theory to capture feelings and cognition of travellers (Xu et al. [2011]), or by considering social agents (Bazzan et al. [1999]), among others. Some go to the extent of considering the learning effect daily travelling has on commuters (Di and Liu [2016]). Kluegl and Bazzan [2004] could even discover the emergence of a stable state incorporating simple learning by adjusting the route choice depending on the commuters' experiences. Such adapting agents also enable the assessment of irreversible equilibria observable in traffic networks undergoing temporary changes. (Guo and Liu [2011])

Bounded rationality can be implemented using thresholds. (Mahmassani [2001]) Thus, two ways of trip duration and cost optimization can be described: In the first case commuters simply choose the best route available, in the second case, only if a sufficient time saving is possible due to switching routes, the switch will be performed. (Chen and Mahmassani [1993])

Nonetheless, the most difficult task in route choice modelling is considered to be the creation of the choice set itself. (Kazagli et al. [2016], Ben-Akiva and Bierlaire [2003]) It can be shown, that a sizeable impact on the quality of a route choice prediction comes from the choice set and therefore its generation. (Prato and Bekhor [2007]) While Frejinger et al. [2009] argue, that all connections between a destination and an origin have to be considered when looking for the best route prediction, some also find that there exists a certain threshold on travel time. (Di and Liu [2016])

One way to catalyse the generation of the choice set, is to move the focus away from the microscopic details of single roads towards a more generalized depiction. A means to accomplish this is to split a network into separate layers and weighing each accordingly and hence, allowing a stepwise choice of regions, then nodes and finally edges to be used. (Manley et al. [2015]) Another pathway chosen for such an abstraction are mental representation items, which subsidise paths by considering and utilising the impression drivers have of their surroundings. (Kazagli et al. [2016])

Among the factors included to describe route choice behaviour the most used one in the papers reviewed is the minimisation of travel time. (Bazzan et al. [1999], Xu et al. [2011], Manley et al. [2015], Prato and Bekhor [2007], Thomas and Tutert [2015], Ciscal-Terry et al. [2016], Palma and Rochat [1999], Bert et al. [2006], Thériault et al. [1999], Rossetti et al. [2002], Ren et al. [2014]) Which correlates with the findings of Papinski et al. [2009] and Palma and Rochat [1999], suggesting that minimising travel time is among the most important influencing factors for the home-work route among minimisation of congestion and traffic lights. Traffic lights are found to be immensely negative correlated to speed. (Thomas and Tutert [2015]) Thus, intersections can be penalised by increasing travel time. (Thériault et al. [1999]) Furthermore, also other time measurements are used like bound time (Xu et al. [2011]) or free-flow time (Bierlaire and Frejinger [2008], Prato and Bekhor [2007]), slowed down and start/stop time (Hensher [2001]).

Additionally, many models incorporate the strive for minimisation of the distance travelled to model route choice. (Manley et al. [2015], Prato and Bekhor [2007], Thomas and Tutert [2015], Ciscal-Terry et al. [2016], Ren et al. [2014]) Fuel consumption (Bert et al. [2006]) and monetary cost (Xu et al. [2011], Hensher [2001]) are used as well and somewhat related to distance and can be included trivially by additional factors. In a cost based radiation model where cost can quickly be depicted by any factor, a comparison of cost based on travel distance to one based on travel time shows, that the latter is able to predict the measured behaviour more precisely. (Ren et al. [2014]) While traffic lights are stated to be important before the actual ride, afterwards minimisation of the distance appears to be much more important to commuters. (Papinski et al. [2009])

Finally, the maximisation of directness apparently has a big impact on route choice (Thomas and Tutert [2015], Ciscal-Terry et al. [2016], Papinski et al. [2009]) as paths with less turns are perceived shorter. (Turner [2000]) Thus, directness is becoming more frequently included in route choice modelling (Manley et al. [2015], Thomas and Tutert [2015]), by introduction of turns into the impact factors (Ciscal-Terry et al. [2016]), or the minimum angular path theory. (Turner [2000])

Various researchers have examined GPS-data to find behavioural structures in route choice. (Li et al. [2005], Ciscal-Terry et al. [2016], Manley et al. [2015], Kazagli et al. [2016], Bierlaire and Frejinger [2008], Papinski et al. [2009]) Findings from such studies show, that drivers will prefer the ability to travel at a higher speed despite the resulting additional length of the path being twice that of the shortest alternative. (Ciscal-Terry et al. [2016]) Also other studies using revealed preference data support this behaviour. (Thomas and Tutert [2015]) In addition to these findings, drivers also appear to value an increase in travel time differently on freeways compared to alternative, smaller and mostly slower roads. (Bierlaire and Frejinger [2008])

Furthermore and rather unsurprisingly, non-linear attractions are found for some locations (Manley et al. [2015]), suggesting that the inclusion of areas with increased appeal to the agents could facilitate the model's accuracy.

Revealed preference data is often argued to be difficult to handle and to fit to a traffic network. (Kazagli et al. [2016], Bierlaire and Frejinger [2008]) Using stated preference has the advantage that it allows for more control on the experiment. (Thomas and Tutert [2015]) However, revealed preference circumvents the shortcomings of stated preference data in that survey participants cannot skew the results by being unfamiliar with or biased towards some questions. (Asensio and Matas [2008]) By comparison of the revealed preference data from GPS tracks to the stated preference data from surveys pre- and post-ride, clear differences can be found in the stated influencing factors for route choice and how they affect the actual route driven. (Papinski et al. [2009])

1.3.3 Commuter mode choice

One of the decisions a commuter has to make in order to fulfil the wish for a journey into a city is which mode to choose. Especially the choice of taking a car over using public transport or shared rides is a much discussed topic. (Washbrook et al. [2006], Bhat [1997], Asensio [2002], Bhat and Sardesai [2006], Collins and Chambers [2005], Quarmby [1967], Frank et al. [2007])

An often stressed approach is a highly simplified model, boiling down the overall monetary cost versus time and convenience to find an equilibrium where the utility is maximised for all participants. (Arnott et al. [1991], Hess [2001], Hensher [2001], Quarmby [1967], Frank et al. [2007], Washbrook et al. [2006])

Like in route choice analysis and modelling, in the research on travel mode choice the most important factor appears to be travel time. (Quarmby

[1967], Frank et al. [2007], Asensio [2002], Collins and Chambers [2005]) Asensio [2002] finds that especially for the choice of car mode, low travel time and a better access by road to the central area of the city in scope are key elements.

In addition to travel time, when investigating mode choice, excess travel time becomes an important feature, where transfer time, waiting time or the time spent looking for a parking lot are valued more than the time moving. (Quarmby [1967], Washbrook et al. [2006], Frank et al. [2007], Asensio [2002]) Collins and Chambers [2005] find that time in public transport modes is valued differently and equals about 1.25 times the time spent travelling by car. Furthermore, reliability is a highly influential parameter for the use of public transport modes. (Bhat and Sardesai [2006])

In contrast to travel choice modelling, distance from home to work is not a commonly included factor, only being chosen to be implemented by Asensio [2002]. However, cost is far more important, used in almost all the papers reviewed and shown to have great impact on mode choice. (Collins and Chambers [2005], Washbrook et al. [2006]) Nevertheless, some researchers find travel time to be the more influencing factor in modal choice situations (Asensio [2002], Collins and Chambers [2005], Frank et al. [2007]), while others see an indication of the monetary cost to have a better leverage on people's choice how to conduct their journey (Washbrook et al. [2006], Hess [2001], Bhat [1997])

A different perception of cost is given due to the income, where, as one would suggest, higher income leads to less response on cost increase, but has the opposite effect on an increase of travel time. (Asensio [2002]) Furthermore, people with higher income tend to prefer travelling by train over bus rides and overall like to travel alone by car the most. (Bhat and Sardesai [2006], Bhat [1997]) Some non-linearities and counter-intuitivities can be found in the response to additional charges for road use. Variable pricing lead to a shift in the timing of trips, rather than to bus mode in a field study in Norway (Polak et al. [1991]) and an increase of the road toll above a certain level lead to an increase in the road use in a study conducted in the area of Greater Vancouver. The latter was interpreted to be due to the hope of drivers, such an expensive road would be less congested. (Washbrook et al. [2006])

Additionally, access to public transport modes is another regularly used variable (Collins and Chambers [2005], Asensio [2002]), however, not useful in the context of this thesis. The most important insight gained from the review of literature on commuter behaviour in mode choice scenarios is, that monetary cost and the overall time as well as its variance are regarded as the most important factors in the decision making process. Thus, it would be beneficial to achieve an influence of these variables into the commuter simulation as well.

1.3.4 The idealised round city

A very interesting idea is, to depict a city in an idealized manner. One such concept has been proposed with a city taking the form of a circle in which working places were distributed evenly and commuters lived outside the city boundary by Smeed [1964].

Surprisingly, there exists a study on route choice behaviour which was done in a principally round city by Thomas and Tutert [2015] conducted in a midsized Dutch city. The findings from this licence plate survey correlate with the proposal of Haight [1964] who took a closer look into Smeed [1964]'s round city, in that the Hypothesis that a route along the orbit is more attractive than traversing the city centre is observed to be true to some extent. A similar behaviour is also reported by Ciscal-Terry et al. [2016] for the city of Reggio Emilia, where drivers are willing to drive on average 5 km more in order to circumvent the centre. An increase of 10 km is found to be at the upper boundary of convenience, especially for short routes around 5 km, which seems comprehensible. Nevertheless, also in such essentially round cities directness is found to be of importance. (Ciscal-Terry et al. [2016], Thomas and Tutert [2015])

While a round city of course denotes a vast simplification and can in no way be generalized for cities grown in a natural environment often hindered in their growth into one direction by rivers, mountains or other landscape phenomena, it is worthwhile looking into such a form of traffic model disregarding any road infrastructure. Intriguingly, one can find that circling the city outside its boundaries can yield an advantage due to less congestion (Smeed [1964]), which could in turn lead to travel time savings. However, this also increases the distance travelled (Smeed [1964]) and thus the (perceived (Quarmby [1967])) cost of the journey as a sizeable part of the cost is given by running costs. The average distance for commutes ("Cross-Cordon journeys") along this hypothetical road was found to be 1.904 times the radius of the city. (Holroyd [1966])

Upon this model, Haight [1964] applied methods of basic probability and delved into their distributions for various simplified ways of reaching a destination in the round city interior and in effect deduced commuter preferences.

Logically, the direct and polar choices are the most direct ones, among the rest of the proposed ways the preference depends on the destination coordinates. (Haight [1964]) The direct and the radial possibilities are both ruled out by their viability and their high effect on congestion in the city centre. (Smeed [1964]) As it exploits the radial variant for destinations further than two radians from the entry (Haight [1964]), the polar variant which uses the shortest path possible in a polar coordinate system can also be omitted.

Hence it can be concluded that among the choices deemed as realistic, in the half circle of evenly distributed destinations up to $2.4rad$ from the entry, the shortest path choice is the inner variant, travelling along the radius to the point where the distance from the centre equals the distance of the destination to the centre, then proceeding along a circular way. Is this angle exceeded, the shortest path along a rectangular coordinate system is the most attractive one. (Haight [1964]) However, in this case the rectangular option resembles most likely a principally straight path through the city centre reminiscent of the polar and radial variant. Thus, the last viable option becomes that of the outer circle.

Chapter 2

Methods

While the methodology is in principle predetermined by Hofer et al. [2017]'s model, an overview of the methods put to use is provided. Agent based modelling is the key principle of the model and its implementation is based on the Python programming language, where the functional programming paradigm serves as a means to enhance the compatibility between the original model and its extension.

2.1 Agent based modelling

Agent based models (ABM) utilise a multitude of entities called agents and their interactions to describe emergent phenomena arising out of these. (Macal and North [2007]) Through the swivel of the scope from the former used top-down view on describing the dynamics arising in a systems (Borshchev and Filippov [2004]), to the characterisation of the behaviour and interactions of the agents and thus a bottom-up modelling advance, a more natural way of modelling can be reached, facilitating the human interpretability of the underlying mechanisms in such systems. (Macal and North [2007], Chen [2012])

However, even the modellers themselves cannot fully understand how the low level definitions lead to the exhibited behaviour on the higher system level. (Galan et al. [2009]) Hence, assessment of agent based models is difficult and they have to be checked thoroughly in order to stipulate correctness. (Chen [2012], Galan et al. [2009], Niazi and Hussain [2009])

Furthermore, while such models might yield good qualitative results, quantitative results can only be reached with accurate data on the agents themselves. (Bonabeau [2002])

An often stressed criticism of efforts to model human decision making is the sometimes irrational behaviour of human beings. ABM has the advantage of being able to implicate this behaviour in its working principles. (Bankes [2002], Bonabeau [2002], Ringler et al. [2016]) Note however, that no one agent can directly be mapped onto depicting one specific person or other entity in reality, but in the aggregate form of all agents a portion of these behave similarly to a portion of the individuals they aim to imitate.

2.1.1 What is an Agent?

The central element of an ABM is the agent. Agent based models have been utilised in a multitude of fields and applications, all of which base their model on the notion of an agent. (Niazi and Hussain [2011])

Agents can be found in the domain of artificial intelligence where they comprise the term multi-agent systems (MAS) and are in effect viewed as rather intelligent, with the ability to learn and adapt to their environment. (Ringler et al. [2016], Macal and North [2007]) Consequently, from a scientometrics point of view, the field of Ecology, where bottom-up modelling has a long tradition serves as a link between MAS and ABM, as it has brought forward models of many individuals and their interactions, the so called individual based models (IBM). (Niazi and Hussain [2011])

However, there is no clear and certainly no definite definition of what makes an agent. (Ringler et al. [2016]) Coming from the domain of ecology, Grimm [2005] tries to answer the question of what the key features of an agent are with the question:

> "What makes James Bond an agent?"
> - [Grimm, 2005, Page 987]

While at first glance this might be seen as an exploitation of a mere coincidental similarity in nomenclature, the answer to the question reveals insight into the essentials of an agent in an ABM, as the mentioned film character strives for a clear goal and decides individually on how to reach it, reacting to his ever changing environment. (Grimm [2005]) This matches the principles of agents found in other papers, thus it can be concluded that an agent is individually identifiable with an initial state, reacts to its environment autonomously and depending on its own and the state of the environment, is clearly situated somewhere in the environment and somewhat in contrast to the popular movie character is also defined through its social behaviour interacting with other agents. (Ringler et al. [2016], Macal and North [2006], Grimm [2005], Chen [2012])

A proved and intelligible way to find these properties of the agents is pattern oriented modelling, where the thought process is based on observed patterns on different levels in space, time and hierarchy and the key features of agents leading to these are then implemented in the agents, (Grimm [2005]) reminiscent of how the modelling of the entry and exit node choice was done in section 3.2.

2.1.2 Tools

Agent based models can basically only be used for simulations with computational assistance as they require a vast amount of computational power. (Richmond and Romano [2008], Erol et al. [2000]) As a consequence of the various fields where ABM has been put to use, many tools have been devised focusing on different applications and ranging from a scope on minimalist models to that of the development of sophisticated decision support systems. (Abar et al. [2017], Macal and North [2006])

As in traffic modelling the opposing approach to macroscopic models is the bottom-up approach where individuals in the traffic systems are going about their day and congestion emerges out of their interaction with each other and the road network, ABM is an obvious choice for such microscopic models. However, in contrast to previous individual based models where the data underlying the basic model components gave a definite description of their behaviour and left no room for autonomy of the entities, microscopic traffic models utilising actual ABM like TRANSIMS (Smith et al. [1995]) and MATSIM (and [2016]) allow a degree of freedom and reaction to changes in the environment like congestion. (Bonabeau [2002])

Though tools like NetLogo and Mason allow for quick, integrated development of ABMs, (Abar et al. [2017], Niazi and Hussain [2009]) for novel advances like the model by Hofer et al. [2018a], a general purpose programming language is often more suited due to its flexibility and depth of control over computation itself, which might be a reason the Python programming language was chosen to develop their model.

2.2 ABM using the Python programming language

Python is a general purpose programming language developed by Guido van Rossum in 1990. (Sanner [1999]) It incorporates the principles of many paradigms such as object orientation (Sanner [1999]), imperative programming (Buitinck et al. [2013]) and some functional programming (Mertz [2015]) and therefore enables one to choose the preferred way of coding. As an interpreted language it supports an interactive coding style and provides great extensibility and in effect facilitates high level programming. (Sanner [1999]) Therefore, Python and other interpreted languages have become popular in the scientific community and many scientists report their perception of being more productive with such languages. (Cai et al. [2005])

Using an idiom like the Python programming language for agent based modelling, it is convenient to utilise the object orientation for the depiction of agents. (Macal and North [2006]) The notion of an agent in a model has previously been interpreted as an additional layer of abstraction similar to objects or functions, yet on a higher level of abstraction. (Abar et al. [2017]) Therefore, in the model of Hofer et al. [2017] an agent is defined as a class which combines its state with two functions, one for general movement in the network as a city citizen and the other for the movement of commuters, where the agent itself knows whether it is a commuter or not.

However, within this framework given by the model of Hofer et al. [2017] the model presented in this thesis is meant to be an extension and thus has to fit into the model it relies on. Moreover, caution has to be taken not to interfere with it in order to keep it from causing malfunction upon simulation. A means to ensure such fit in terms of robustness and facilitating expansion of an underlying code base is the paradigm of functional programming (FP). (Hughes [1989])

2.3 Functional Programming in Python

Stemming from mathematics programs based on the FP paradigm do not rely on mutable state and consequently do not mutate any state themselves, rather they only expose a function in the pure mathematical sense such that upon the same input it returns the very same output. (Hughes [1989]) This concept is known as referential transparency. (Quine [1960])

Programming paradigms tend to be a controversial topic amidst the coding community. (Hudak [1989]) The history of programming languages has been accompanied by the strive for the most efficient way to denote a problem in a machine-readable way. However, while for the machine clearly binary code is the most efficient format, it is de facto unreadable to humans. For centuries mankind has turned to mathematical notation in order to enable a compact yet universally standardised means of problem description. Thus, the utilisation of mathematics for programming seemed reasonable and was subsequently proposed by Backus [1978]. Nowadays, functional programming languages are utilised in mobile network systems with Ericsson's Erlang (Wadler [1999]) and of course still researched with the research driven language Haskell (Hudak et al. [2007]), which has interestingly also been used to renew Facebook's spam-filter Sigma (Marlow [2015]).

Most of the more traditional programming languages utilise an imperative concept. (Hudak [1989]) These languages like C, Java and Python dominate the top ten of the Tiobe index (www.toibe.com [2019]) and therefore, likely enforce the imperative paradigm amongst the programming community and teaching. The form of this denotation is akin to Assembler language and thus, directly inherited from binary machine code. (Hudak [1989])

Imperative programming has however multiple shortcomings all derived from so called side effects, which are not present in mathematical functions. (Hughes [1989]) Side effects are all instructions which write or read from some form of mutable state. Mutable state comes in many flavours: memory, storage, I/O, networking and even looping over a variable are considered as such and therefore, do not occur in pure functional languages like Haskell. (Launchbury and Peyton Jones [1995], Hughes [1989]) The problems arising from extensive use of mutable state and the total control over it come clear when considering the memory management of C or C++ which has to be done manually and hence lead to the introduction of garbage collection of which, due to the vast abstraction of memory, functional languages make extensive use. (Sansom and Jones [1993])

2.3.1 Purity, laziness and equational reasoning

It is important to distinguish between pure and impure code and languages. Impure functional languages exhibit functional programming styles but fall back to side-effects for at least some purposes. (Peyton Jones and Wadler [1993]) These formalisms subsequently lose referential transparency. (Hudak [1989]) Pure functional programming languages do not have any side-effects, they only evaluate expressions to their result without anything else happening. (Hughes [1989])

Such a pure functional language can make use of its gained referential transparency by emphasizing the ability to evaluate each expression only as soon or late as it is needed. This so called lazy evaluation, or laziness, can in fact hardly be conducted with impure languages. (Hudak et al. [2007]) However, with the loss of transparency at what exact time each expression is to be evaluated, some general view over the control flow has to be traded for this feature. (Abelson and Sussman [1996], Hudak et al. [2007]) Nevertheless, it enables the introduction of infinite lists and other constructs that might not finish evaluation in traditional languages, such as erroneous and incomplete functions. (Hudak [1989], Abelson and Sussman [1996])

Side-effects also defy the use of equational reasoning (Peyton Jones and Wadler [1993]), which is not only the basis of evaluation of pure functional programs, but can also be incorporated for theorem proving in such idioms.

(Plaisted [1993]) The power of the implementation of I/O while maintaining purity as done by Peyton Jones and Wadler [1993], as well as Launchbury and Peyton Jones [1995] is expressed by the possibility of equational reasoning even with the implication of monads. (Gibbons and Hinze [2011])

In Python however, one of the most drastic properties counteracting referential transparency is that there are no means of defining constant values. Thus, except for keywords every name in the namespace is ultimately (a) variable. (Beazley [2009]) This goes to the extent where in Python 2 the 'True' and 'False' values for Boolean expressions could be altered to hold any desired value, as they are implemented as traditional variables and not as primitives like in C-based languages or as data types like it is done in Haskell. Hence, counterintuitive things become possible as shown in listing 2.1.

Listing 2.1: Variability of Boolean primitives in Python 2.7.

```
>>>True == False
False
>>>True = False
>>>True == False
True
```

This has been remedied in Python 3, where the Boolean values have become keywords reminiscent of C-like languages:

Listing 2.2: Boolean primitives as keywords in Python 3.6.

```
>>>True = False
SyntaxError: can't assign to keyword
```

Although Python is inherently object oriented, no **private** or **protected** keywords are provided to keep variables inside classes from being manipulated by their environment. However, some conventional workarounds have been devised to counteract this shortcoming with the introduction of underscores into the naming scheme. Two of these at the beginning of a name initiate name mangling upon interpretation, where the final name consists of the class name lead by another underscore and followed by the initial name. The subsequently renamed variable or method hence becomes private. (Beazley [2009], Langtangen [2008]) A similar name convention is present in Python for constants which are to be written in capitalised letters, however, this is in no way similar to the actual constant values abound in languages with special keywords such as **const**. (Beazley [2009]) Read-only properties of classes can be achieved by exploiting the @property decorator. (Beazley [2009]) However, these still access mutable state and can therefore not be considered as referentially transparent.

2.3.2 Benefits for the impure

Some concepts like the vectorisation of the numerical python module 'numpy', enabling the Python programmer to operate on large data structures with high computational efficiency, are rather object oriented and less reminiscent of FP. (Langtangen [2008], Cai et al. [2005]) Nevertheless, also impure languages like Python could attract some love from the functional programming community and incorporate some of their advances themselves. Recently, more and more traditional languages have introduced features previously mostly known from Haskell and its relatives, as will be followed though hereinafter. (Hudak et al. [2007])

Lambda calculus

The first formalism to enable the problem description in a machine interpretable way was introduced in 1936 by Alonzo Church who supervised Alan Turing's Ph.D. (Enderton [1995]) in his publication on the foundations of logic (Church [1932]). The lambda calculus is a formalism for an abstraction of computational problems, which is accomplished by the definition of three different idioms: abstractions, applications and variables.

It can be shown that the lambda calculus is Turing complete and can hence denote any problem which is itself computable. Thus, it became an important basis in the development of the field of computer science and is often considered the first programming language. In fact, many of the principles found in modern functional programming languages are reminiscent of lambda calculus and even ones which are not primarily functional have introduced lambda abstraction as the advent of functional programming arose. (Hudak et al. [2007])

While the initial idea of the lambda calculus was the abstraction of mathematical problems in a way to make it machine interpretable, nowadays lambdas are predominantly used as a term for a formalism that facilitates the understandability of programming code for humans. (Hudak et al. [2007]) They are a means to create functions without the obligation and expense of having to name it and therefore often used if a function is needed just once, for instance when mapping such a function over a data structure. (Hudak et al. [2007], Beazley [2009]) For an example of a lambda expression in the Python programming language refer to listing 2.5.

Recursion

Functional programming favours the use of self-referencing functions over the use of stateful loop concepts. (Hughes [1989]) However, if such a function was to be implemented in Python a stack overflow would be immanent as soon as a mere 1000 references to the same function are reached. (Beazley [2009], Mertz [2015]) Nevertheless, Python features a handful of concepts for operating recursively, some of which emphasize side-effectful, imperative concepts and some of which are rather functional features reminiscent of Haskell semantics.

The traditional way of doing recursive computation can be traced back to the very first programming languages and comes in different flavours of loops. As the lowest in the order of abstraction resides the well known while-loop, followed by the for-loop which in Python is not merely a prettier way to denote the common while-loop, but has some more syntactic sugar added to ease its use. If a type is denoted as iterable, done by implementing the __iter__() and __next__() functions, a for-loop in Python can iterate over the items exposed by the iterable as shown in listing 2.3. (Mertz [2015])

Listing 2.3: A for-loop in Python

```
>>>for item in items:
>>>     #do stuff
```

Hence, the actual way Python implements for-loops forces the utilisation of the 'range'-function to enable use of such a loop in a traditional way and then even shunts the increment statement off the loop instantiation into the function defining the items iterated over. This is then reminiscent of the original while-construct, where the incrementation or similar concepts reside in the body of the loop as shown in listing 2.4. The for-loop thus discourages a traditional looping styles utilising iteration over a variable and encourages iteration over list-like items and function results.

Listing 2.4: A while-loop in Python

```
>>>list = range(3)
>>>while(list):
        print(list.pop())
```

A less imperative concept is denoted by list comprehensions, which stem from functional programming languages but have also been implemented for iteration over lists and other integrated data structures in Python. (Hudak et al. [2007]) It is important to note, that in Python these represent only an additional alternative to describe what is done in a for-loop. (Beazley [2009]) However, they are advised to be used when applying functional programming in Python by Mertz [2015] due to the perceptional shift from the imperative concept of how the computation is performed, to the more mathematical sense of what has to be done during computation.

Listing 2.5: Lambdas and some of their applications for recursion in the Python programming language.

```
>>>f = lambda x: #do alpha
>>>#List Comprehensions:
>>>[f(x) for x in range(10)]
#alpha happens for all x's in range(10)
>>>#Generators
>>>gen = (f(x) for x in range(exp(10)))
>>>next(gen)
#alpha happens lazily for the first x in range(exp(10))
>>>f = lambda x: #do other beta
>>>next(gen)
#beta happens lazily for the second x in range(exp(10))
```

Another concept brought forward from functional programming and introduced in Python as well, is laziness. Generators basically denote the lazy equivalent to list comprehensions and can thus be used on much larger iterables with high computational and memory efficiency. (Langtangen [2008], Mertz [2015]) Their syntax looks familiar once list comprehensions have been discussed, exchanging the square brackets for normal parentheses in an elsewise similar notation as shown in listing 2.5. Note however, that these break referential transparency. As shown in the last three lines of listing 2.5, changing the function f in the hindsight would change consecutive values given by the generator. Hence, additional care is advised to be taken when utilising generators for lazy evaluation in Python.

Higher order functions and currying

Functional programming emphasizes that data and functions can be used interchangeably. Thus, functions can be used as an argument for another function or even as its return value, they are citizens of first class. (Beazley [2009]) Functions taking other functions as arguments are characterised as higher order functions. (Hudak [1989]) Such functions can be utilised in multiple ways, such as defining a function that can apply another function to every element of a list (such a function is usually named 'map'), or to take another function with two arguments and apply it in between list elements, which is a technique known as 'folding'. (Hughes [1989])

Functions in Python form function objects and are in effect implemented as citizens of first class membership. (Beazley [2009]) Hence, these techniques are also present in Python and will be incorporated in the course of the implementation of the algorithms devised in this thesis. Some important higher order functions form in fact another pathway in alternatives to the imperative concept of loops. The 'map'-function for instance essentially denotes a synonymous means for the application of a function to all items of an iterator in the same way list comprehensions and for-loops do. However, being a function itself out of these options it emphasizes the functional principles the most. (Mertz [2015]) A similar concept is brought forward through the function 'filter', taking a function evaluating to a Boolean value, a so called predicate, to sieve out all items of a list which evaluate to

True via this predicate. (Hudak [1989], Mertz [2015]) Moreover, Python's function 'reduce' enables programmers to fold right like the 'foldr'-function in Haskell does, which enables a vast amount of applications denoted in a single line of code. (Hughes [1989], Beazley [2009]) Examples of these functions are shown in listing 2.6, note that the last argument in reduce can be omitted for convenience, it gives the starting point of the folding which has to be included in e.g. Haskell (see Hughes [1989]).

Listing 2.6: Some functions for application on iterables in Python.

```
>>>map(lambda x: x+1, range(3))
[1,2,3]
>>>filter(lambda x: x > 1, range(3))
[2]
>>>reduce(lambda x,y: x+y, range(3), 0)
3
>>>reduce(lambda x,y: x*y, range(1, 4), 1)
6
>>>#alternatively with the operator module
>>>import operator
>>>reduce(operator.add, range(3))
3
```

Currying is another useful application of functions as first class citizens. This term refers to the ability of applying an argument to a function f taking n arguments, yielding a function g with $n-1$ arguments where the one handed to the initial function has subsequently become a constant in g. (Hudak [1989]) This operation is made possible in Python in a simple way by the 'partial'-function of the 'iteritems'-module (Mertz [2015]), as demonstrated in listing 2.7. Additionally represented in the listing, as a more manual way to achieve currying in the Python programming language, closures are in effect nested functions, where the outer function returns its inner one in return to the initiating arguments. (Mertz [2015])

Note that the arguments handed to the outer function f have in effect become constants in g, referential transparency is now given. Furthermore, closures also enable lazy evaluation and are also exploited by decorators which wrap an existing function much in the way closures are defined. (Beazley [2009])

Listing 2.7: Currying as a function 'partial' and as done with closures.

```
>>>from iteritems import partial
>>>add_one = partial(map,lambda x: x+1)
>>>add_one(range(3))
[1,2,3]
>>>#Closure
>>>def f(x):
        def g(list):
            return [item+x for item in list]
        return g
>>>add_one = f(1)
>>>add_one(range(3))
[1,2,3]
```

2.4 Principles of the implementation

In effect, the implementation exploits functional paradigms in the various forms they are abundant in Python to in effect yield the exposure of a single function for the deduction of entry and exit nodes, as well as a third one to allow commuter agents to find their choice among the sets generated by the first two functions.

While some shortcuts had to be taken, accessing state of the underlying model's setup file, the functions do not alter any state and can therefore not interfere with the working principles of the initial code base. Moreover, for each function argument passed to them which is mutable, they immediately construct a copy in order to keep the code base safe in case mutation of state is performed.

Where inner state is changed extensively, the programming paradigm is consciously switched to yield an imperative style raising awareness for the impurity of the algorithmic expression in the area.

The code can be found in Appendix B.

Chapter 3

Realisation

Two steps have to be performed in order to model the commuter agents automatically, in contrast to the manual implementation of Hofer et al. [2017]. First all viable entries and exits from the network in scope have to be deduced, following this generation of a choice set, finally, the choice of the commuters itself can be modelled.

3.1 Finding ways in and out of a city

For the detection of the entry and exit nodes a set-theory based approach was applied. Let C_n be a set of the nodes of the network holding the city interior and let O_n be a set of the nodes in the OSMnx (Boeing [2017]) street network of the city area expanded by an factor x in all directions. Then the set of the nodes surrounding the initial city is denoted by:

$$S_n = O_n \setminus C_n \tag{3.1}$$

Now let C_e be the set containing all edges connecting the nodes of the city network's interior C_n and let S_e likewise be the set with the edges in between the nodes held by S_n, the intersected area. Therefore, the intersecting edges connecting nodes of C_n and S_n are given by:

$$I_e = O_e \setminus (C_e \cup S_e) \tag{3.2}$$

with O_e being the set with the edges of the expanded area. Subsequently, when I_n holds the nodes connected by the edges contained in I_e then the set of all the nodes applicable for entry and/or exit E_n can be found by intersection with the city interior nodes:

$$E_n = I_n \cap C_n \tag{3.3}$$

A graphical representation of this scheme is given in figure 3.1 and its results are depicted in comparison to the nodes initially contained in the model in figures 3.2 and 3.3.

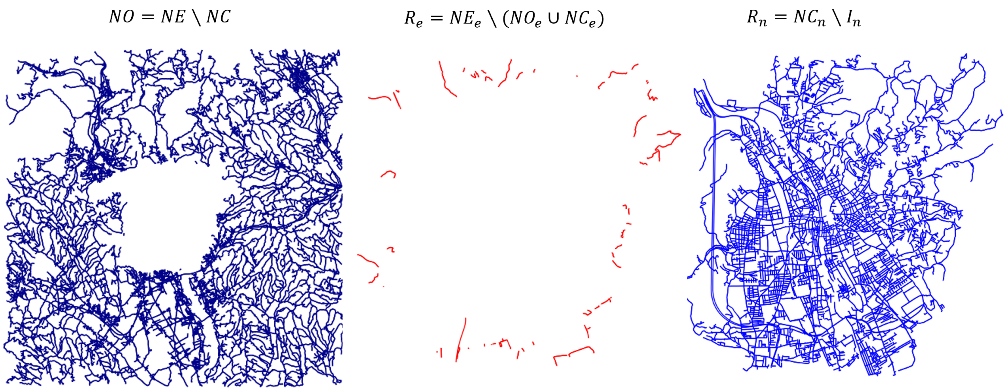


Figure 3.1: The definitions of the various sets with a graphical depiction of their constituents.

3.1.1 One-way streets

Nevertheless, this set of nodes has to be refined as there are still some nodes which are invalid for usage in certain scenarios. There are some 'one-way' nodes that can only act as entry or exit. Nodes only applicable as entry can be deduced by intersection with the set of nodes in C_n having $in\_degree = 0$ and exit nodes in an analogous way using $out\_degree = 0$.

3.1.2 Speed limit filtering

Furthermore, it might be of interest to ignore nodes connecting only streets of low attractivity. Therefore, we apply filtering functions. However, the information is not node-intrinsic but much rather contained in core properties of the streets depicted by edges. Thus the filtering function must work on a set of tuples, each holding an edge of I_e and its corresponding node from E_n.

Figure 3.2: The entry nodes of the filtered E_n in comparison to the nodes initially contained in the model of Hofer et al. [2017] on the right.

Multiple edges, respectively streets, holding the same node have to be considered as these can have different properties. However, duplicates of nodes in the result have to be ruled out in accordance to set theory.

Filtering by road quality can be useful when a lot of nodes were detected where the road use may be impractical due to low speed limits. However, if these data are contained in some of the edges, nodes might be ruled out incidentally. Hence, if such filtering is applied the addition of a verbose mode to keep the possibility of deactivating the sieve active is suggested.

3.2 Attractiveness of nodes: The commuter's choice model

With the set theory based approach in section 3.1 the commuter agents are successfully provided with a choice set of entry and exit ways for their journey into and out of the city. However, from this set the final choice has to be found for each of the agents individually. In the following description entry and exit nodes combined are referred to as the node set. As with most discrete choice models which stem from the field of econometrics,

Figure 3.3: The exit nodes of the filtered E_n in comparison to the nodes initially contained in the model of Hofer et al. [2017] on the right.

many route choice models incorporate the concept of utility. (Train [2009], Di and Liu [2016], Ben-Akiva and Bierlaire [1999]) For a closer look into this topic refer to section 1.3.2.

This formalism of defining utilities is closely followed in the assignment of nodes from the node set, nevertheless, since the node chosen does in effect not have any real utility to a traveller but just defines their starting point, the nomenclature is changed to instead refer to the attractiveness of nodes. Throughout the ways of the commuter agents, there is the city boundary which in the model in scope defines also a system boundary. Consequently, the attractiveness of the nodes connecting the inner microscopic system with the outer macroscopic one is composed out of two parts, one for each subsystem. Both of these can again be divided into two subroutines, first of which is the attractiveness of properties independent of the commuter and second the part which is different depending on origin, destination or other properties of the commuter.

The factors influencing the choice from the node set are derived from those found as most commonly used for modelling and stated as influential in the literature review on commuter behaviour (see section 1.3). Hence, time, distance and directness are the key attributes in the final algorithm. Nevertheless, because of the abstract way the system outside the city boundary is given, these have to be abstracted as well.

3.2.1 The time factors

As the most stated factor in route choice modelling, time is an attribute which should not be omitted in the modelling of the node choice by the commuter agents. For most commuter agents a valuable amount of their journey is situated outside of the city network, however, since the network outside the city boundaries is spared for computational efficiency, some abstractions have to be considered. The time t_n spent outside the city boundaries when travelling via node n is estimated using the ratio of the maximum speed allowed at the corresponding entry node $v_{max,n}$ and its distance d_n from the commuters origin:

$$t_n[h] = \frac{d_n[km]}{v_{max,n}[\frac{km}{h}]} \tag{3.4}$$

While for commuters starting from afar, there is little difference in the distances to the nodes depending on where they start in their county, for counties close to the city the assumption that all of its inhabitants start at its centre does not hold. To accomplish a better estimate for the starting points of the commuter agents' journeys a set of the nodes in the street network is randomly chosen from. Thus, more of the commuters start from areas where the density of street crossings, traffic lights and other road obstacles, respectively the density of nodes in the associated network is high, which gives a good estimate of the population density as found by Glover and Simon [1975] and more recently found to be true in Sweden by Jenelius [2009]. For the roads connecting the city network with its neighbouring ones, one of the edge's nodes is shared between the networks, therefore, the nodes from the node set are excluded from the nodes in the adjacent counties. Note, that while the absolute value for the travel time might be inaccurate, it is of no direct influence to the choice, only the differences in between the values are.

For the time advantage inside the city boundaries gained with the choice of a node the speed limits are taken as an estimate. Note that the further simulation of the path chosen inside the network is conducted in the same

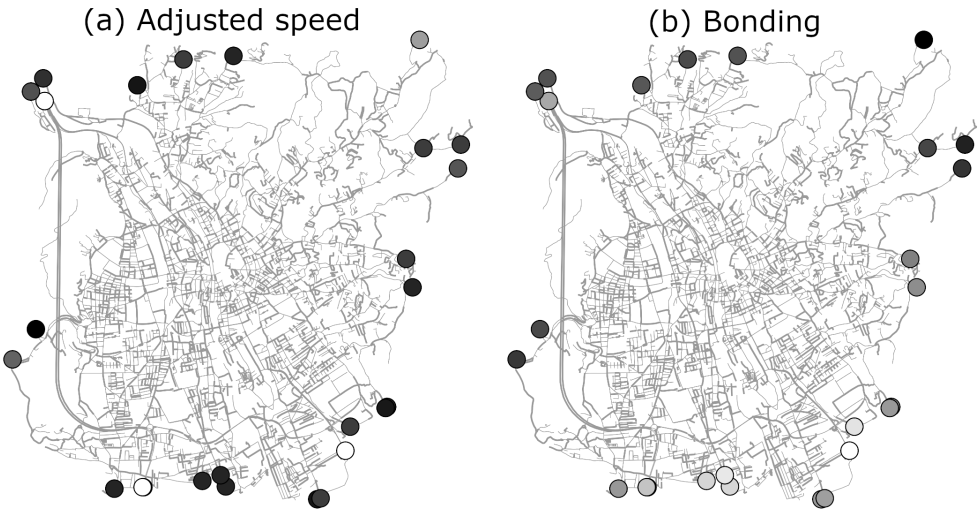


Figure 3.4: Attractiveness arising from the speed factor $a_{s,n}$ adjusted incorporating the 'highway'-tag (a) and from the bonding values given at instantiation (b).

way as for the agents depicting inhabitants of the city and hence, the scope of the advantage is predominantly focused on the different nodes in the node set. This speed-limit attractivity $a_{s,n}$ is given by the fraction of the maximum speed limit of the node's adjacent edges $v_{max,n}$ and the highest speed limit that can be found out of all the entry nodes respectively exit nodes in the set v_{max}:

$$a_{s,n} = \frac{v_{max,n}}{v_{max}} \tag{3.5}$$

However, since the maximum speed limit only gives a rough estimate of the actual speed travelled inside the urban area, the concept of free speed Bierlaire and Frejinger [2008] enables an enhancement upon it. Zilske et al. [2011] use the 'highway'-tag included in the metadata of the edges in the OSM network data to adjust the speed limit accordingly. Consequently, a_s is multiplied according to this tag for all entry nodes. Some exemplaric speed limits, free speed adjustment factors and the resulting values for attractivity can be found in table 3.1.

3.2.2 The distance factor

Furthermore, distance is an attribute often taken into consideration when choosing a route. However, with increasing distance from the city, the

Table 3.1: Some exemplaric nodes accompanied with their given speed limits, free speed adjustment factors and the resulting values for attractivity.

Node	Speed limit [kph]	Free speed adjustment factor	Attractivity $a_{s,n}$
333565415	100.0	1.2	1.2
21301848	100.0	0.8	0.8
33028761	50.0	0.8	0.4
269216372	50.0	0.5	0.25

difference in distance to the nodes entering the city becomes small compared to the total distance travelled. Ciscal-Terry et al. [2016] find that for a detour greater than the total distance taking it becomes increasingly unattractive and there appears to be an upper limit. Idealising the city in scope to be round, taking the mean distance between the nodes from the node set as its diameter ($d = 13.41km$), the added distance to be travelled to circumvent the city is found to be:

$$\frac{C}{2} = \pi\frac{d}{2} = 21.06km$$

which is more than twice of what Ciscal-Terry et al. [2016] found to be the upper limit for a feasible circumvention of the city centre in their study on drivers in Reggio Emilia. Nonetheless, an estimate can be derived for this upper limit, which consequently can be applied to find the distance from the city where all nodes from the node set appear equal in terms of total travelled distance. Consider a direct entry to the city to be accomplished in a straight line to the centre, thus with a length of the radius $r = \frac{d}{2} = 6.7km$ and a detour circumventing the city on a half circle drawn by this radius and then entering the city. The first route is accomplished on a road at the lower end, in respect to speed, of our sampling spectrum in the node set $v_1 = 50\frac{km}{h}$, while the detour can be driven with the maximum allowed speed found in the set $v_2 = 100\frac{km}{h}$. The excess distance travelled while circumventing the city is found with the least angle where this becomes attractive in the idealized round city found by Haight [1964] and closer inspected in section 1.3.4 and thus given as:

$$d_e = 2.4 * r = 16.09km$$

Finally, adding the findings of Srinivasan and Mahmassani [2002] that with a time saving of 15 minutes a great share of the commuters consider the alternative route, these two routes can be compared to find the threshold distance d_{max} at which the nodes can be assumed to become equally attractive.

$$\frac{r + d_{max}[km]}{50[km/h]} = \frac{1[h]}{4} + \frac{d_e + r + d_{max}[km]}{100[km/h]}$$

$$d_{max} = 34.385km$$

With the introduction of a threshold the commuter agents share a bounded rationale considering their perception of distance to the nodes in the choice set. (Mahmassani [2001])

Consequently, the attractivity of a node induced by its distance to the commuter is calculated as follows. Let D be the set of all the Euclidean distances then m denotes the minimum of these values and Δ the difference between the maximum and the minimum. As mentioned above we assume, that at some distance d_{max} the difference Δ has no more influence on the attractivity an hence define a threshold t:

$$t(d_{max}) = m + (m/d_{max}) * \Delta \tag{3.6}$$

This threshold can then be used to get the attractivity $a_{dis,n}$ of the nodes depending on their distance d_n to the commuter's starting point:

$$a_{dis,n}(d_n, d_{max}) = \begin{cases} 1.0 & : t(d_{max})/d_n > 1.0 \\ max(0, t(d_{max})/d_n - \delta) & : t(d_{max})/d_n \leq 1.0 \end{cases} \tag{3.7}$$

Where $\delta \in [0, 1]$ is the "out-of-scope"-factor which reduces the attractivity if a node lies outside the threshold. The attractivities derived from this part of the algorithm for different starting points are depicted in figure 3.5.

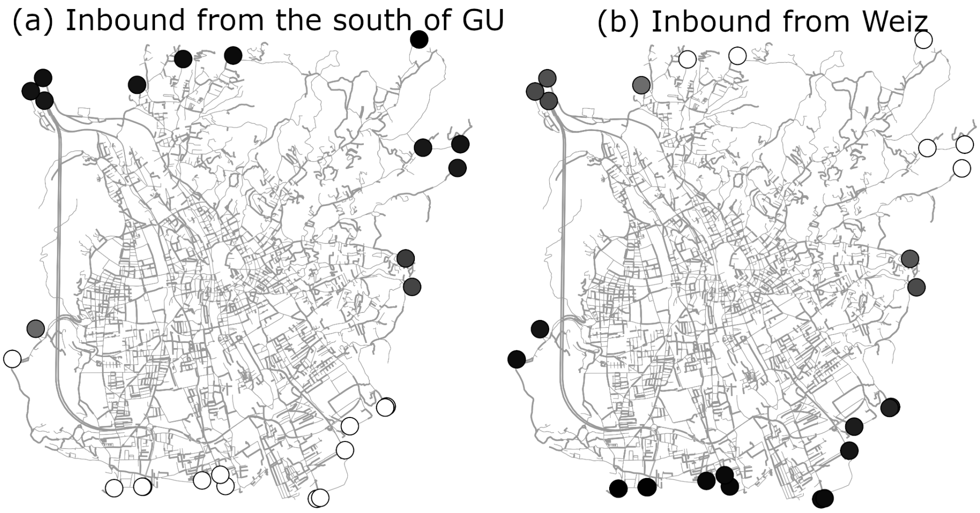


Figure 3.5: Attractiveness arising from the distance factor ($a_{dis,n}$) given for a commuter originating from the south of the county encircling the City of Graz (a) and for one inbound from the County of Weiz (b), where a brighter depiction of a node denotes a higher attractiveness value.

3.2.3 Directness

Directness is a means which enables the destination choice and subsequently origin within the network to draw an impact on the attractiveness of the nodes. With the route directness index (RDI) (Ciscal-Terry et al. [2016]) and the minimum angular path assumption (MAP) (Turner [2000]) two concurrent gauges were found to yield promising results. In order to overcome the limitations set by the lack of a road network and computational effort a repeated shortest path search would induce, these measures are abstracted and successively combined. Disregarding the cumulative aspect of turns taken successively, commuters who want to enter the city from a direction deviating from the direct connection between their origin and destination are assumed to take at least a total turn with the angle enclosed by the triangle origin - entry/exit node - destination, where a turn of 180 degrees is defined as being the least attractive option. Thus, the simplified least angular path is defined through mapping of the dot product onto the interval $[0, 1]$:

$$sMAP = \frac{\overline{ON} \cdot \overline{ND} + 1}{2} \tag{3.8}$$

where O denotes the origin, N the node of the node set and D the destination. The special case of an destination which coincides with a node, sMAP is set to yield 1 for this particular node. Likewise, while the RDI is defined as the actual route length divided by the direct distance of the OD pair, in the simplified route directness index (sRDI) the actual route length is substituted by the sum of the lengths of $\overline{ON}$ and $\overline{ND}$. Consequently, its inverse is taken into account to yield the final value for the commuter agents' modelled perception of directness a_{dir}:

$$a_{dir} = \frac{sMAP}{sRDI} \tag{3.9}$$

3.2.4 Bonding

The choice as seen from the macro scale already makes for a result, which is close to the simulation conducted with the implication of expert knowledge on the one hand and additionally, to the depiction of Google Maps [2019] on the other hand. However, the road network of the urban area so far has no capability of influencing the attractiveness of the nodes. Nevertheless, to assess the impact of policies or infrastructural changes on the commuter impact on congestion, this influence has to be modelled as well. It seems trivial to evaluate the shortest path for each node of the node set to the predefined destination. However, this approach is computationally expensive since the evaluation of the node choice takes on average about 3 seconds per agent in this case.

Therefore, a general factor is introduced which can be calculated ab initio for each node and yields an estimate for the quality of its connection to the rest of the network. This bonding attractivity a_b is gained by looking for the duration of travelling on the shortest path to the other nodes in the set, respectively from the nodes for outputs and calculating the mean out of these values, thus crossing the city network in various directions emphasizing the area of the network surrounding the node in scope as depicted in figure 3.6. Subsequently, the ratio of the least duration with the corresponding bonding value is used for each node such that the best connection results in a value for a_b of 1 and the rest resides below that in the interval $[0, 1]$ where no connection is penalized with a value of 0.

3.2.5 The total attractiveness

Ultimately, these influencing factors combined yield a heuristic choice algorithm where the total attractiveness a_n of a node n is calculated as:

$$a_n = a_{b,n} * (t_n^{-1} + \alpha * a_{dis,n} + \beta * a_{s,n} + \gamma * a_{dir,n}) \tag{3.10}$$

where α, β and γ are parameters governing the influence of the individual attractiveness factors. Note that the bonding factor is positioned such that a total loss of connection would yield a total attractiveness a_n of zero. Finally, the values for the parameters where found through a parameter sweep, fitting to the results of the model devised by Hofer et al. [2017] and are listed in table 3.2. Figure 3.7 depicts the attractiveness values gained from this formulation of directness for different agents in an artificial scenario.

Figure 3.6: The shortest paths from node 318441610 in the road network of the City of Graz to the other entry nodes.

Table 3.2: Parameters found to be best fitting for the model of entry and exit nodes attractivity.

$\alpha = 0.4$	$\beta = 0.2$	$\gamma = 0.1$	$\delta = 0.5$

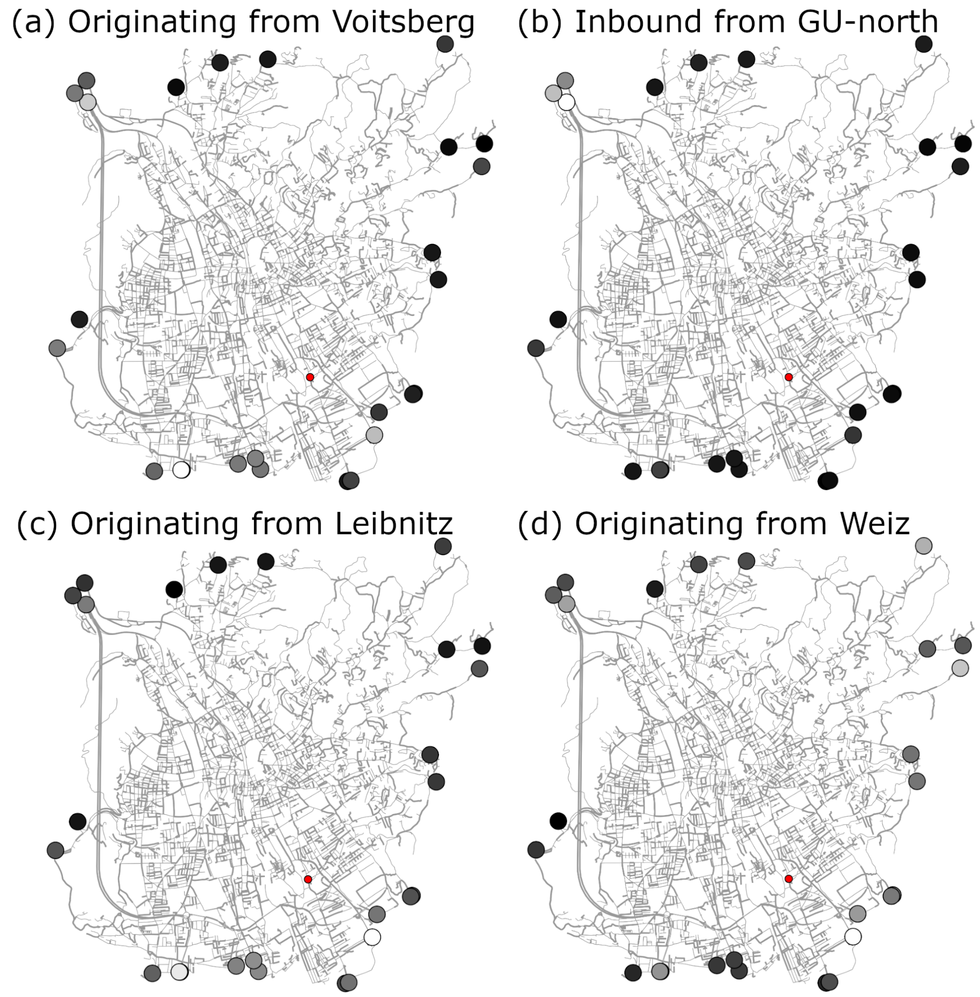


Figure 3.7: The total attractiveness a_n for different commuter origins, where the red dot represents their target and light nodes have higher attractiveness-values than dark ones.

Chapter 4

Evaluation

The evaluation of the model is performed twofold. First, a qualitative assessment of the new results in comparison to the old results given by Hofer et al. [2017]'s model and the depiction of congestion arising in reality by Google Maps [2019] is provided. The extended model is subsequently put to use to forecast CO_2-emissions for different scenarios and an assessment of the impact of commuters on emissions in a city environment is conducted. In order to evaluate the model to estimate commuter movements presented in chapter 3 the commuters integrated in the model of Hofer et al. [2017] were substituted to be calculated with the algorithm presented.

4.1 Evaluation of the simulated commuter flows

The prediction of the congestion arising in a city is an important feature of the initially proposed model by Hofer et al. [2017] and the extension to it devised in 3 should thus not corrupt these results. Therefore, a qualitative comparison of these results has been conducted.

4.1.1 Congestion in the City of Graz

The initial model had been designed to simulate the congestion arising in the City of Graz, it thus stood to reason for the evaluation to also be performed on this city, with the same underlying data that had been used in Hofer et al. [2017]. However, due to the need for the surrounding road network for the deduction of the node set (see section 3.1) the network of the city itself had to be renewed as well. Nevertheless, as for the original simulations the inhabitants making up a total of 280232 agents, were divided into age groups and distributed across the city's districts according to the same arrangement. Furthermore, the amounts of commuters allocated to the counties they originate from were kept as well. Their quantities are listed in table A.1. Additionally, since found to yield more accurate results in Hofer et al. [2018a] and suggested by various studies on the behaviour of drivers (Srinivasan and Mahmassani [2002], Palma and Rochat [1999]) congestion prevention was included and conducted with the 10% share detected to be most fitting by Hofer et al. [2018a].

Comparison of the cumulative results

A comparison of the congestion arising from the initial hard coded commuters to the now automated procedure of commuter simulation is depicted in figure 4.1 and shows the combined result of inhabitants and commuters for both models. Furthermore, it also depicts the average traffic situation at 7:30 AM as shown by Google Maps [2019], while both simulations were set for the time period of 7:00 to 8:00, all of which is in the realm of rush hour. This qualitative analysis shows a good consesus of the commuter model with the model conducted utilising expert knowledge, thus giving a good estimate of the congestion airsing in reality.

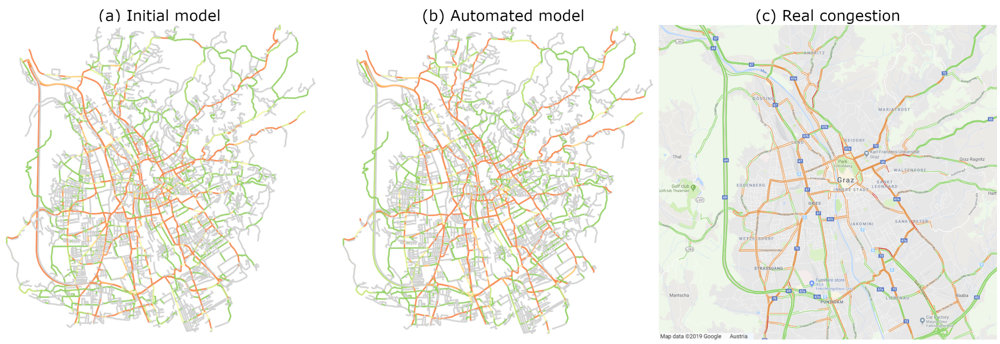


Figure 4.1: Comparison of the emerging congestion in the initial simulation (a), the same simulation with the commuter model provided in section 3.2 (b) and the actual congestion as depicted by Google Maps [2019] for an average tuesday at 7:30 AM (c).

To enhance the evaluation of the automated commuter flows the congestion arising from only these drivers set in the same time period is shown in figure 4.2, where in contrast the figure also shows a simulation result with commuter influx disabled and goes to express the need for these drivers in order to yield realistic results in this traffic model. It can thus be assumed, that no traffic model with a spatial scope similar to the one of Hofer et al. [2017] can be conducted disregarding commuter flows.

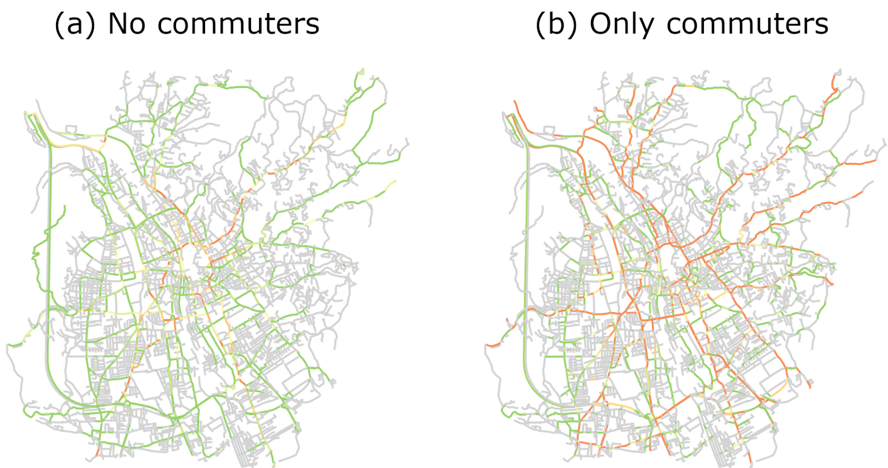


Figure 4.2: The congestion arising in a simulation of only the commuters (b) and without them (a) for the City of Graz.

Comparison of commuter groupings

As another measure for qualitative validation a comparison of the path choice for the different commuter groups as given in A.1 was devised both for the old and the new model. The graphs gained from these simulations are shown in comparison in figures A.1, for commuters from the County of Graz-Umgebung encircling the City of Graz, A.2, for those counties sharing a border with Graz-Umgebung and A.3 for the rest of the commuter origins.

Note that only the latter shows remarkable deviation from the original in exhibiting less congestion in the areas the original commuters of the 'far away'-group experienced constrained (Höfler [2004]) traffic flow (orange).

However, this group makes for under a fifth of the total quantity of commuters and thus has less effect on the overall picture of congestion when combined, which could be the reason for the worse fit to the original. During the fitting of the parameters, attention was payed to the total outcome and thus, the importance of high speed entries, namely freeways for larger distancances may have been underestimated.

Nevertheless, some areas seem to be more evenly congested with the new model and since there is no data available on the concrete route choice of this group of commuters, it cannot be concluded which model fits reality better.

4.1.2 Congestion in the City of Salzburg

One of the main reasons longing for an automation of the simulation of commuters in Hofer et al. [2017]'s model was the enabling of adabtability to other cities without the need for further expert knowledge. Hence, taking into account the model extension presented in chapter 3 this should now be possible. To show the newly gained ability of the model to simulate the congestion in other cities of comparable size to that of Graz, a case study was performed on the Austrian city of Salzburg, where severe congestion problems are abound, as it is the most congested city in Austria in 2016 according to TomTom International BV [2016]. A total of 136056 agents depict the inhabitants of Salzburg while the 33496 commuters were allocated to counties as given in table A.1 according to their distribution as stated by Statistik Austria [2011].

However, Salzburg is situated close to the border of Germany and there was no data abundant for commuters travelling across the border. This deficiency can be spotted in figure 4.3 where on the north-western boundary the main road shows no congestion in contrast to the very high congestion which is shown by Google Maps [2019].

Another protruding deviation is visible in the eastern part of the network in an area called "Gaisberg", where a long stretch of road is shown highly congested although according to Google Maps [2019] no such phenomenon should occur at this time of day.

Consequently, as was done for the City of Graz, figure 4.4 shows a graph of the results once without commuter influx and once only for these and thus it can be shown, that this incident also occurs when the commuters are spared, leading to the conclusion, that a deeper look into the issue is

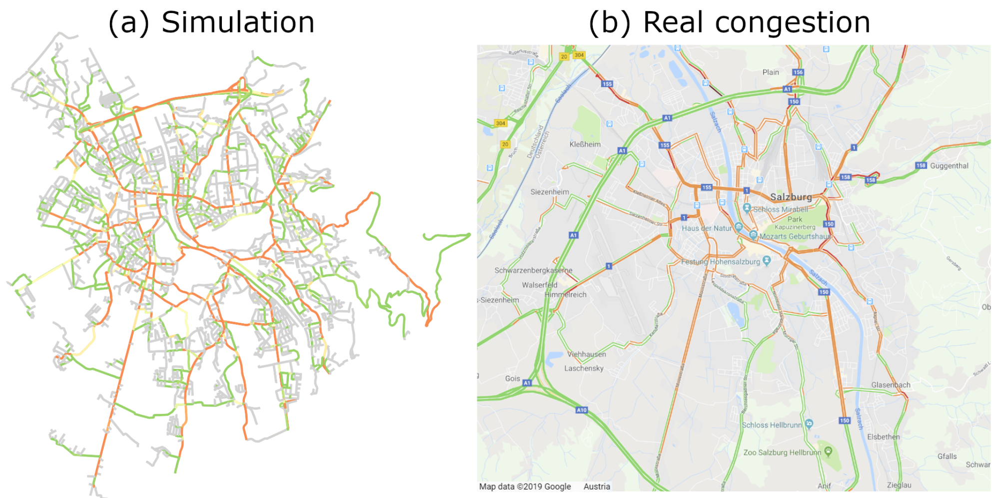


Figure 4.3: Congestion arising in a simulation of the traffic in Salzburg (a) compared to the actual congestion as given by Google Maps [2019] for a typical Tuesday morning at $7:30AM$ (b).

required.

Since for the City of Salzburg no accurate data on inhabitants per district could be found, initially the simulation fell back to a simulation of the whole city, where the fractions of inhabitants were divided according to the fraction of nodes for each district in the network. This division had to be applied in order to enable multitasking and thus shortening the computation time. Nonetheless, with this approximation the area of "Gaisberg", which coincides with one of the statistical counting districts of the City of Salzburg (see Stadt Salzburg / Statistik [2019]), held 1379 residents, which is about the fifteen-fold of the total resident count given in Stadt Salzburg / Statistik [2018]. Therefore, for all districts where a statistical district with matching boundary was abundant, the actual value of inhabitants was used and the rest of the districts were adjusted accordingly to yield their fraction given by their proportional size in nodes. However, note that the data given in Stadt Salzburg / Statistik [2018] was a year older than the population count of Stadt Salzburg / Statistik [2019].

Both nodes and population shared by the remaining districts were the total amounts substracted by the amounts taken by those districts for which more accurate data was given. Despite these changes in the underlying

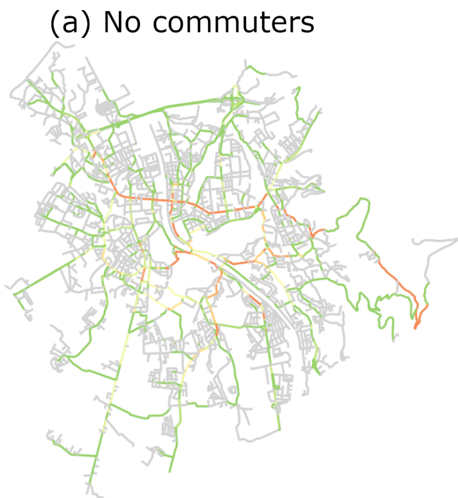


Figure 4.4: A simulation run for the City of Salzburg once considering only agents depicting inhabitants of the city (a) and once only including the commuters themselves (b).

data, the simulation results only changed marginally and while this is an indicator for a more profound imperfection in the model it also indicates that it is robust against poor data which further emphasizes its advantages. Finally, all selfloops of nodes were removed from Salzburg's network, leading to some improvement in the 'Gaisberg' area and this network subsequently became the one used for the results depicted in the graphs.

4.2 Commuter's impact on the city pollution

For the assessment of the commuter impact on emissions within the city limits the average speed based emissions model by Hofer et al. [2018b] was appended to the traffic model of Hofer et al. [2018a] extended with the commuter model devised in this thesis (see chapter 3). Thus, the total CO_2-output inside the city limits including the increase in emissions due to congestion could be calculated and the influence of commuters, both on the total emissions and the emissions increase caused by their contribution to congestion could be discovered. Analogous to the evaluation of the commuter model 4.1 this was undertaken as case studies for the City of Graz for which the model was initially devised and additionally for the smaller City of Salzburg to test and eventually prove the versatility of the now extended model.

4.2.1 Scenarios

Consequently, in order to understand the impact of the commuters on a city traffic's CO_2-emissions, various scenarios were simulated, starting with a base line simulation of all traffic participants combined and subsequently removing commuters from the roads according to different groupings. These groupings were based on the administrative division of the areas the commuters originate from, to yield results more viable to policy makers and to practical application. From the cities in scope, both are encircled by a county from which of course the majority of commuters travel into the city, however, in Salzburg another county is not separated by much from the city boundaries and thus also taken into the first grouping of close distance commuters. The consecutive grouping is the middle distance grouping, which is constituted for Graz of all the counties sharing a border with the county of Graz-Umgebung which is itself the one encircling the city. For Salzburg this scheme is continued, nevertheless the two remaining counties from the federal state of Salzburg, namely Tamsweg and Zell Am See are also included since they are within comparable distance to those from the second set for Graz. Finally, the third grouping is considered to include those commuters travelling from a farther distance to the city and thus consists of the rest of the commuters' origins. The concrete origins and their associated amounts of commuters as well as the groupings are shown in A.1. The values for the commuter quantities in Graz were taken over from Hofer et al. [2018b] in order to yield comparability and therefore gathered for Salzburg from the same dataset (Statistik Austria [2011]) and subsequently corrected in the same way as was done in the previous study by Hofer et al. [2018b].

The results were generated applying congestion avoidance corresponding to its implementation in Hofer et al. [2018a] and thus 10% of the agents were simulated to decide for looking into quicker routing options. Additionally, like for the simulations done in section 4.1 in the City of Graz 280232 agents depicted the city's inhabitants and for Salzburg 136056 agents were going about their day.

The following scenarios were carried out:

- Scenario 1: Omitting the commuters from close distance to the city. (group 0 and group 3 respectively)
- Scenario 2: Omitting commuters of the middle group. (group 1 or 4)
- Scenario 3: Omitting the group of commuters from a farther distance. (groups 2 and 5)
- Scenario 4: Omitting both the middle distance group and the commuters from far away. (thus only group 0 or 3 additionally to the city's inhabitants)

The cumulative emissions as given by the model for the City of Graz are $577.58 \pm 11.55 \frac{tCO_2}{day}$ and are depicted for the different scenarios in figure 4.5a while for Salzburg 4.5b shows the total emissions of $237.53 \pm 4.75 \frac{tCO_2}{day}$ including the scenarios. The relative error for these estimates due to the random nature of the OD generation is of appriximately 2% as the total figures stem from the mean of two simulations. (Hofer et al. [2018b]) For all other results given the error is within the low single digit realm at about five percent, as all simulations were carried out once. A lower relative error can be reached by repeating the simulations with variable seeds feeding the random number generator and values below one percent are yielded after approximately six simulation runs. (Hofer et al. [2018b])

As expected, the results of the scenarios (given in figure 4.5) show a decrease in emissions with a decreasing quantity of commuters entering the city. However, the CO_2 per agent taking part and especially the CO_2-reduction per agent taken out of the bulk (as stated in table 4.1) differs between the scenarios. These phenomenons and especially the latter effect were considered worth to be taken a closer look at. Therefore, four additional scenarios were devised increasing the quantity of commuters in group 0 or respectively group 3 in increments of 20%.

However, as depicted in figure 4.6 no sizeable differences could be discovered in these runs with incremental reductions of the fraction of commuters travelling into the city from the closest counties. Note especially, that the

(a) Graz (b) Salzburg

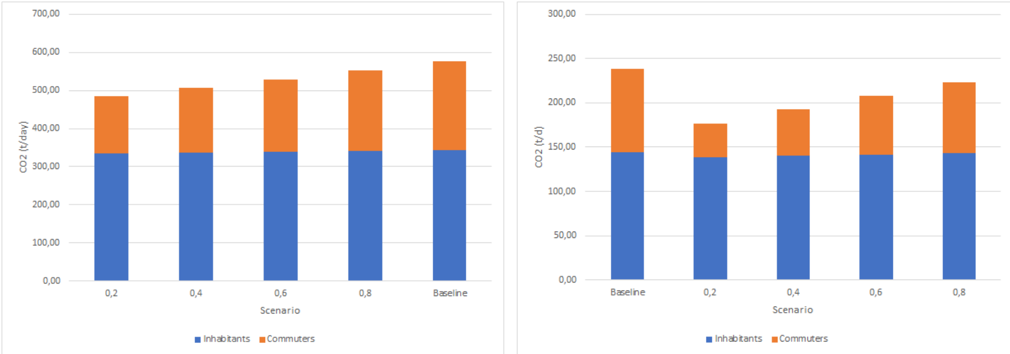
Figure 4.5: Total CO_2-emissions for the four initial scenarios, as well as the baseline scenario in tons of CO_2 per day for each city.

Table 4.1: Reduction of CO_2-emissions per commuter spared depending on their origin (grouping) in kg of CO_2 per day per commuter agent.

Model	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Salzburg	3.57	3.36	3.17	3.22
Graz New	5.37	5.22	4.92	5.14
Graz Old	5.36	4.95	4.21	-

scenario including 40% of group 0 making a total of 8575 commuters has a similar size to group 2, which includes 8565 agents, yet shows a reduction of $5.32\frac{kg}{day}$ for each commuter left out which is close to the value of $5.37\frac{kg}{day}$ for the whole of this group and far from the result of scenario 3 of $4.92\frac{kg}{day}$.

Additionally, the scenarios 1 to 3 were conducted utilising the old commuter model used by Hofer et al. [2018b]. Since the lowest reduction of CO_2 per commuter left out was explored in the groups which had the largest distance to the city, also representing the group with the least fit as depicted in A.3 and previously discussed in section 4.1 the results for the scenarios with incorporation of the old model also served as another point of validation for the automated, novel commuter model. These results are also given in table 4.1 and show results qualitatively comparable to that of the new model, however more spread out quantitatively.



(a) Graz

(b) Salzburg

Figure 4.6: Total CO_2-emissions for the four consecutive scenarios incorporating commuter fractions in increments of 20% of the closest districts, as well as the baseline scenario in tons of CO_2 per day for each city.

Compared to the cumulative results of Hofer et al. [2018b] one can ascertain that less emissions were expelled by the city inhabitants within the city limits. However, this phenomenon is also observed when the original implementation of the commuters is used. As a validation of the commuters' influence on emissions these two models can thus also be compared by their total results which are given in table 4.2 with and without congestion effects and lie within their margin of error.

Table 4.2: Emissions as given by the old implementation of the commuter agents and the new, algorithmic one in tons of CO_2 per day. The relative error is about 2%.

Model	Inhabitants	Inhabitants with congestion	Commuters	Commuters with congestion	Total with congestion
Old	295.53	343.82	151.21	245.05	588.87
New	295.50	343.45	147.11	234.13	577.58

Ultimately, the figures of the total emissions in comparison to values stated by authorities serve as another point of validation for the model. A model of the City of Salzburg yielded a yearly emission of $103604t$ of CO_2 from cars inside the city limits (Huemer et al. [2015]) which calculates to $283.85\frac{tCO_2}{day}$. Considering that the simulation devised here does not include tourists, cold starts, parking lots and moreover no commuters from germany as insufficient data was available for these, our result is within reasonable reach of this figure.

For Graz data was only available with values given for individual car traffic for the year of 1995 (Pischinger [1995]), where the estimate was of $179\frac{g}{km}$ of CO_2 emitted from cars while driving. With a total mileage of $1152 * 10^6\frac{km}{a}$ this gives $206208\frac{t}{a}$ of yearly CO_2-emissions and hence per day $564.95t$ which is below our estimate and makes for 77.27% of the total $266860\frac{t}{a}$ emitted due traffic disregarding cold starts and parking lots. (Pischinger [1995])

However, according to a similar report (Heiden et al. [2008]), in between 1995 and 2001 the CO_2-emissions in Graz increased by 29% due to an increase in traffic. The same value for CO_2-emissions due to only traffic (excluding cold starts or parking lots) is given to be $280.550,4\frac{t}{a}$ in the year of 2003. (Heiden et al. [2008]) Thus, even when the increase in this figure is only due to individual car traffic, on an average day $602t$ of CO_2 were emitted by car traffic at this point.

While this estimate lies above the value calculated by the model, it is within a reasonable margin of error of 4.18%. However, this data can be considered too old to be drawn any conclusions from, no more recent estimates were available.

Chapter 5

Discussion

With the extension of Hofer et al. [2017]'s model at hand, the effect of policies can now be evaluated incorporating a meaningful reaction of agents not depicting inhabitants of the city in scope, but commuters, as well.

Furthermore, the model now is applicable to basically every Austrian city with comparable size to Graz and Salzburg, and it could be shown that the commuters themselves make a significant portion of congestion in the Austrian mid-sized cities. Realistic results could not be calculated disregarding the influx from areas surrounding the network and moreover, a simulation of just the commuter flows showed more realistic results than one only calculating agent movements of city inhabitants.

This leaves the question if commuter activities should be spread out over a wider range of daytime, as of now, these still do not actively vary their departure time, or respectively time arriving at the network boundary, but are still distributed over the hours from 7 AM to 9 AM in a Gaussian distribution. While this is a viable solution in modelling human behaviour (Manley et al. [2015]), an analysis of travel counts at the city boundary could yield some insight into how a better distribution for the commuter agents would be constituted.

Moreover, a deeper understanding of the model itself could be gained by applying it to another city. While the mismatch of congestion in the eastern area of the City of Salzburg (see 4.3) enlightens where the model needs to be revisited and probably adjusted, the otherwise good match to the congestion in reality shows, that lt is resilient against poor data for the population of a city.

These deficiencies in the representation of congestion in the east of the Salzburg could not be traced back to the commuter's influence. Even after the temporary removal of the entry node close to the road in scope, no major improvement could be found. Thus, the problematics may be situated in the generation of the OD pairs, where a farther distance given by the dataset of Österreich Unterwegs (Tomschy et al. [2016]) can only be covered by reaching out to these relatively distant nodes. Further research is required in this area to enhance upon the algorithm.

Some deficiencies of the commuter model are immanent through the data generation incorporating OSMnx (Boeing [2017]) and OSM (OpenStreetMap contributors [2017]), where inaccurate or missing data can lead to wrong estimates about whether and where commuters can enter, or exit the city's road network and furthermore, about the attractivities calculated for the nodes of the generated choice set.

However, at least for the entry and exit node detection a graphical analysis is convenient and artefacts of these data vacancies can be spotted without the need for extensive knowledge about the model itself. In effect, the false classified nodes can be removed in a matter of minutes.

Furthermore, although the model on attractiveness of nodes yields good results for the cities studied, the bonding value is also highly dependent on the road network in scope. If there are parts disconnected from the rest of the network, although an entry node might be abundant and the only viable option to enter this part of a city for commuters, the corresponding node would be assigned with a bonding value of 0, due to having no connection to the rest of the nodes in the node set. This is a misbehaviour which has to be revisited in the future if other cities are to be assessed

While in the scope of the model and especially with the commuter model being of vast abstraction, far from the detail of a road network, the assumption that the nodes in the networks of the surrounding counties correlate with the population density, appears to yield results good enough for the simulation, this assumption additionally appears on the opposite end of the commuter paths and is much more problematic in that realm. The targets of the commuters are, as described in section 1.2, determined randomly, with an even distribution across all nodes in the street network of the city in scope. However, this leads to a distribution which is likely to be rather the opposite from where commuters might have their place of employment, since bigger enterprises in Europe usually do not contain greater areas of publicly accessible roads and thus, little nodes in their area in the network respectively.

For validation, a comparison to the model devised utilising expert knowledge, to which the new commuter model's parameters also had been fitted to, as well as a comparison to congestion as shown by Google Maps [2019], have been conducted. However, this is only a qualitative validation, while the discrete values of cars travelling on sections of road would indeed be convenient for quantitative analysis.

Such quantitative validation in comparison to reality failed when applying the data gained from the model to an emissions model. However, with the high level of aggregation of the data underlying these results, little can be learned about the deficiencies leading to the underestimation of the total CO_2-emissions. Furthermore, the model is limited to the city area, where no estimates for CO_2-emissions could be found for the City of Graz. While for the City of Salzburg there were such estimates, which are themselves results of an emissions model incorporated by the city itself (see Huemer et al. [2015]), there the underestimation was immanent as well, leading to the conclusion, that some effects or drivers might be missing from the model. Cold motor starts, mentioned in Huemer et al. [2015] and Heiden et al. [2008] could be one such effect, tourists and commuters reaching the City of Salzburg from across the border to Germany and therefore with missing accurate data on their amounts, are likely to drive the result closer to that of Huemer et al. [2015] once included.

Quantitative validation by comparison of the results for different scenarios, once simulated with the old and once with the new model showed, that the deviation from the original model increases with increasing distance. In the new model the commuter agents do not choose the freeways as frequently to enter the city when they originate from a large distance, hence it is likely, that in the entry node choice formalisation for the speed factor, the dependency on the distance has to be increased.

Furthermore, incorporating parameters based on quantitative validation would also yield a meaningful introduction of a reaction of the commuter agents to congestion. The cornerstone for this expansion of the model has already been implemented with the attractiveness score on bonding described in section 3.2.4.

However, it became evident, that one reason why many traffic and emissions models are not or only partially validated as stated by Borge et al. [2012] and Smit et al. [2010] could be the lack of accurate data and thus, the extensive efforts that have to be taken in order to accomplish a validation.

Subsequently, results could be gained for the impact of commuters on the CO_2-emissions arising from the traffic in the cities in scope and their assessment showed, that the model predicts a higher decrease of emissions for commuters from close distance than for those inbound from afar. This is due to the combined effect of the commuters from close distance having a higher per driver emissions impact themselves and a higher effect on congestion as well. Thus, it is likely that these commuters have to undertake longer journeys through the city interior in comparison to the commuters coming from afar, who approach on freeways heading straight into the centre of the city.

However, these results only comprise the area of the city itself and, in effect, also only the emissions expelled in this region. Of course, due to the fact that commuters coming from counties or other cities in a greater distance have to travel farther, the total emissions of these are likely higher compared to those of commuters living close to the city in scope. Furthermore, consider that the results are given by a model and thus, do not perfectly resemble reality.

Nevertheless, while in the process of this thesis an emissions model was applied which currently only focuses on CO_2, conversion factors are abundant (INFRAS [2010]) and with these, air quality measures can be assessed for good measure. Emissions depleting air quality often are more locally bound than CO_2 (Fontes et al. [2015]) and may thus be more interesting to a city, subsequently yielding a higher incentive to assess results for just a city's area. Therefore, one of the next steps in the further development of this model is, to implement other pollutants exploiting the factors given by INFRAS [2010].

Finally, regarding the research question if the model's computational efficiency can be kept in the process of automating the commuter flows it can be concluded that this goal has been reached. The initial setup time increases depending on the network due to the detection of entry and exit nodes and is done in 100 to 300 seconds, yet nevertheless, the threads running the calculation of the commuters are less time consuming than the threads for the inhabitants. Moreover, due to the functional programming approach the commuter model is well prepared for future advances to enable concurrent computation of small packets of agents, which eventually might be applicable to GPU acceleration. Furthermore, while the detection is done on every instantiation of the setup process, it actually only has to be performed once for every update of the road network for the city in scope and could subsequently also be stored, reducing the setup time somehow, yet not leading to a reduction of overall computation time to a great extent, as most of it is taken up by the simulation of the agents themselves.

Chapter 6

Conclusion

The model devised in this thesis shows, that the choice of commuters which entry to and exit from a city to take can be modelled, with a result which fits well enough with expert knowledge and actual map data, to yield an aggregate congestion effect that resembles the average scenario in a city closely. While the parameters for the commuter model were fitted to expert knowledge, a possibly better result could be found by incorporation of more accurate data like traffic counts. However, such traffic counts might not be useful without the addition of areas of attraction to the network, since close to a supermarket or parking lot more traffic might occur and these non-linear effects are not yet included in the model. Thus, further research is required to improve upon the network representation of the model.

Furthermore, an approach combining set theory with network theoretical principles has been provided, in order to find choice sets for the commuters to choose from for their entry and exit purposes. Moreover, such an algorithm could be used in other fields like emergency planning too, when quick results for possible exit-ways out of a network are in need. (Cova and Johnson [2002]) It could also be used to improve path prediction algorithms, where so far exits of a transportation network have been found by intersection with a circle or similar regular shapes (Jeung et al. [2010]) and now, with this novel approach, the scope can be set on irregular shaped networks as well.

However, while the effects of commuters on congestion could be evaluated and effectively the results for the emissions show correspondence to the expected behaviour, the evaluation of the absolute values for the calculated CO_2-emissions failed. Hence, the emissions model needs to be revisited and tested more thoroughly in order to eradicate its deficiencies. A comparison with other emissions models combined with different input from various traffic models could be of considerable effect. Additionally, a comparison of the traffic model's DTA with the data generated by the dynamic traffic assignment of other models might enhance the depth of knowledge about the model and its weak and strong points.

Herein, the commuter model serves in another way, as it bridges the gap to enable the traffic model of Hofer et al. [2017] to be used on other cities. It can thus now be tested against models which utilize OD-data and can therefore not trivially be evaluated for the City of Graz, which previously was the only city Hofer et al. [2017]'s model had been tested on. Furthermore, since it could be shown in the case study on Salzburg, that the model behaves well despite poor data on the city's inhabitants, research is required to ascertain if the data of "Österreich Unterwegs" (Tomschy et al. [2016]) can be generalised for Europe in some way such that the model can yield results with good match to reality for cities in other countries.

Nonetheless, the relative emissions-results are likely to be indicative for such scenarios and can serve as a first estimate for policy makers in the process of finding attractive projects to take a closer look into. The model goes to show, that the effect of commuters not only on congestion, but additionally on the emissions of the traffic in a city, is significant. While a reduction in commuters travelling by car trivially leads to the reduction of the total emissions due to less cars driving, their absence shows an effect exceeding this linear one, with a non-linear decrease in total CO_2 caused by the decline in congestion on the city's roads. This is shown by the model to be predominantly influenced by the distance from which the commuters originate from, with a higher impact per agent for those starting closer to the city in scope. It can thus be concluded, that for the emissions in a city itself it could be more effective to try driving suburban commuters from using their own vehicles, to more sustainable alternatives.

However, it should be stressed once again, that a model is only a simplified description of reality, and herein the devised commuter model especially so, as it is optimised for quick computation. Therefore, results found with the traffic model should be additionally tested with other models at least and in addition proven against reality if such results are abundant. Never trust a single model.

List of Figures

List of Tables

Listings

Appendices

Appendix A

Figures and tables.

Figure A.1: Congestion arising from only the commuter group 0 as given in A.1 (groups 0-2) as calculated by the old model (a) and the new model (b).

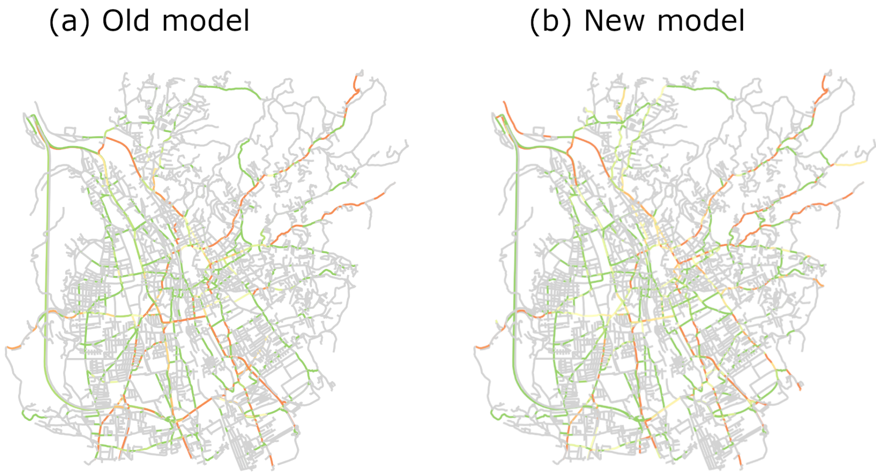


Figure A.2: Congestion arising from only the commuter group 1 as given in A.1 (groups 0-2) as calculated by the old model (a) and the new model (b).

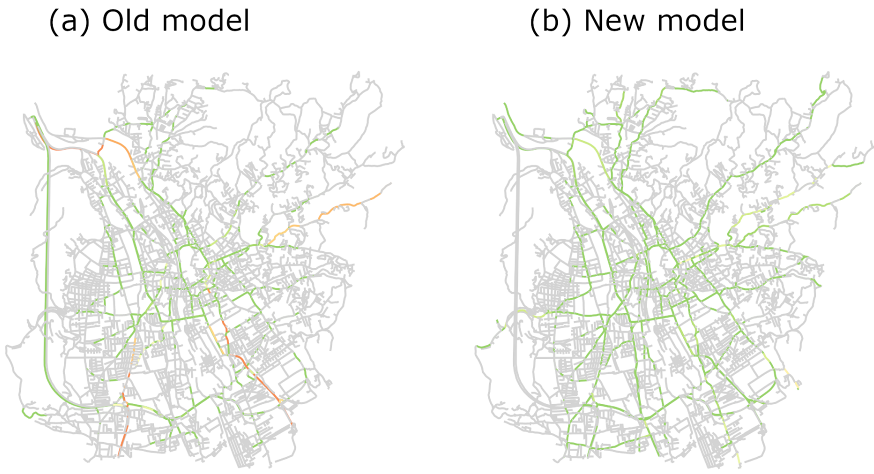


Figure A.3: Congestion arising from only the commuter group 2 as given in A.1 (groups 0-2) as calculated by the old model (a) and the new model (b).

Table A.1: Groupings of commuters and the quantities of commuters from the counties and federal states included in these groups

Group	Number	Constituents and their amounts of commuter agents	Total quantitiy of agents
Graz - close	0	Graz-Umgebung: 21438	21438
Graz - middle	1	Deutschlandsberg: 3457, Leibnitz: 5291, Leoben: 873, Weiz: 4582, Murtal: 1036, Bruck-Mürzzuschlag: 1637, Voitsberg: 3437 & Südoststeiermak: 4419	24712
Graz - great	2	Hartberg-Fürstenfeld: 2346, Liezen: 655, Murau: 436, Carinthia: 1364 & Vienna: 3764	8565
Salzburg - close	3	Salzburg-Umgebung: 16980 & Hallein 4608	21588
Salzburg - middle	4	St. Johann Im Pongau: 1823, Tamsweg: 510, Zell Am See: 1171, Vöcklabruck: 1463, Braunau: 1886 & Gmunden: 410	7263
Salzburg - great	5	Linz: 195, Wels: 53, Grieskirchen: 56, Kirchdorf: 33, Linz-Land: 119, Perg: 33, Ried im Innkreis: 155, Schärding: 73, Urfahr-Umgebung: 64, Wels-Land: 82, Styria: 914, Carinthia: 512, Tyrol: 580, Lower Austria: 855 & Vienna: 921	4645

Appendix B

Code listings of the implementation.

Listing B.1: Lines of code added to the setup.py file of the original model in order to enable ad hoc calculation of some attractivness values with lower computational cost.

```
def calc_attrs(G_in,nodes,out=False):
    bonding = get_bonding(G_in, nodes, out)
    min_bonding = min(bonding.values())
    dict_bonding = {k:min_bonding/v if v is not np.NaN else 0.0
                for k,v in bonding.iteritems()}
    dict_speed   = {k:v for k,v in zip(nodes,normalize([speed_limit(G_in,node)
                for node in nodes]))}
    dict_realsp  = {n:speed_limit(G_in,n) for n in nodes}
    dict_highway = {k:v for k,v in zip(nodes,[highway(G_in,node) for node in nodes])}

    return [(n,{
        'speed':dict_speed[n]
    ,   'realspeed':dict_realsp[n]
    ,   'bonding':dict_bonding[n]
    ,   'highway':dict_highway[n]
    }) for n in nodes]

def normalize(data,ref=None):
    if ref == None:
        ref = max(data)
    return [d/ref for d in data]

def speed_limit(G_in, node):
    edges = get_edges(G_in,node,data='maxspeed')
    return max([float(read_num(sp)) if read_num(sp) is not None else 30.0
                for _,_,sp in edges])

def highway(G_in, node):
    es = get_edges(G_in, node, data=True)
    hw = []

    for u,v,d in es:
        try:
            hw.append((read_num(d['maxspeed']),read_highway(d['highspeed'])))
        except KeyError, e:
            hw.append((30,read_highway(d['highway'])))

    _,res = max([(speed,free_speed) for speed,free_speed in hw])
    return res

def read_highway(tag):
```

```python
    free_speeds = {'motorway':1.2
                  , 'motorway_link':1.2
                  , 'trunk': 0.5
                  , 'trunk_link': 0.5
                  , 'primary': 0.5
                  , 'primary_link': 0.5
                  , 'secondary': 0.5
                  , 'tertiary': 0.8
                  , 'minor': 0.8
                  , 'unclassified': 0.8
                  , 'residential': 0.6
                  , 'living_street': 1.0
                  }
    if type(tag) == list:
        try:
            sp = [free_speeds[t] for t in tag if t is not 'unclassified']
        except KeyError, e:
            sp = [free_speeds['unclassified']]
        return max(sp)
    else:
        try:
            return free_speeds[tag]
        except KeyError, e:
            return free_speeds['unclassified']

def lanes(G_in, node):
    es = get_edges(G_in, node, data=True)
    lan = []

    for u,v,d in es:
        try:
            lanes = int(read_num(d['lanes']))
        except KeyError, e:
            lanes = 1

        try:
            oneway = d['oneway']
        except KeyError, e:
            oneway = False

        if lanes > 20 or lanes is None:
            lanes = 1
        if not oneway:
            lanes = lanes/2.0

        lan.append(lanes)

    return max(lan)

def get_edges(G_in, node, data=None):
    return list(G_in.in_edges(node, data=data))
                + list(G_in.out_edges(node, data=data))

def get_bonding(G_in,nodes,out=False):
    def of_node(node):
        if out:
            return map(lambda n: path_length(G_in,n)(node),
                [n for n in nodes if nx.has_path(G_in,n,node) and n != node])
        else:
            return map(path_length(G_in,node),
                [n for n in nodes if nx.has_path(G_in,node,n) and n != node])

    return {n:np.mean(of_node(n)) if of_node(n) != [] else np.NaN for n in nodes}

def path_length(G_in,start):
    def inner(end):
        path = nx.shortest_path(G_in, source=start, target=end, weight='time')
        return sum(ox.get_route_edge_attributes(G_in,path,'time'))
    return inner

def get_paths(G_in,nodes,out=False):
```

```
    def of_node(node):
        if out:
            return map(lambda n: path_path(G_in,n)(node),
                [n for n in nodes if nx.has_path(G_in,n,node) and n != node])
        else:
            return map(path_path(G_in,node),
                [n for n in nodes if nx.has_path(G_in,node,n) and n != node])

    return {n:of_node(n) for n in nodes}


def path_path(G_in, start):
    def inner(end):
        return nx.shortest_path(G_in, source=start, target=end, weight='time')
    return inner
```

Listing B.2: commuters2.py file containing the algorithm for attractivenes as discussed in section 3.2 on attractiveness of nodes.

```
# coding: utf8
import osmnx as ox
import networkx as nx
import numpy as np
import simplejson as json
from math import tanh, exp, sqrt

import setup
import helper
from entrypoints import read_num
from car import Car

def get_commuter_ways(G_in, district_code, entry_nodes, exit_nodes):

    target_node = np.random.choice(G_in.nodes())

    entry = select(G_in, district_code, entry_nodes,target_node)
    exit = select(G_in, district_code, exit_nodes, target_node)

    if not nx.has_path(G_in, entry, target_node)
            or not nx.has_path(G_in, target_node, exit):
        return get_commuter_ways(G_in, district_code, entry_nodes, exit_nodes)

    return target_node, entry, exit


def _find_nodes(G_in, district_code, base_selection, f):
    return select(G_in, district_code, [e for e in base_selection if f(e)])

def find_entry_for(G_in, district_code, base_selection, target):
    return _find_nodes(G_in, district_code, base_selection,
                lambda e: nx.has_path(G_in, e, target))

def find_exit_for(G_in, district_code, base_selection, source):
    return _find_nodes(G_in, district_code, base_selection,
                lambda e: nx.has_path(G_in, source, e))

## ALGORITHM

def select(G_in, district_code, base_selection, target_node):
    attract = attractiveness(G_in, district_code, base_selection, target_node)
    return np.random.choice(attract.keys(),
                p=normalize(attract.values(),sum(attract.values())))

def attractiveness(G_in, district_code, base_selection, target_node):
    dict_coord,dict_cos,dict_rdi,dict_dist =
        coord_attr(G_in, district_code,[n for n,_ in base_selection],
                get_geo(G_in)(target_node))

    return {n:v['bonding']*(
        v['realspeed'] / dict_dist[n] # TIME
        +0.4*dict_coord[n] # DISTANCE
        +0.2*v['speed']*v['highway'] # TIME in City speed is normalized
        +0.1*dict_rdi[n]*dict_cos[n] # DIRECTNESS
        )
        #dict_coord[n]+0.3*v['speed']
        for n,v in base_selection}


## NORMALIZATION

def normalize(data,ref=None):
    if ref == None:
        ref = max(data)
    return [d/ref for d in data]

## FEATURES
```

```python
def lanes(G_in, node):
    edges = get_edges(G_in, node, data='lanes')

    return max([int(read_num(lanes))
        if int(read_num(lanes)) < 30 and read_num(lanes) is not None
        else 1
        for _,_,lanes in edges])

def speed_limit(G_in, node):
    edges = get_edges(G_in,node,data='maxspeed')
    return max([float(read_num(sp)) if read_num(sp) is not None else 50
            for _,_,sp in edges])

def path_length(G_in, path):
    return sum(ox.get_route_edge_attributes(G_in,path,'time'))

def get_edges(G_in, node, data=None):
    return list(G_in.in_edges(node, data=data))
            + list(G_in.out_edges(node, data=data))

## SELECTION BASED ON COORDINATES
def coord_attr(G_in, district_code, base_selection,target_pos):
    g = helper.get_distance_from_geo

    lat_pos, lon_pos = get_district_geo(district_code)
    geos = map(get_geo(G_in),base_selection)

    distances = [g(lat_pos, lon_pos, lat_node, lon_node)/1000.0
          for lat_node, lon_node in geos]

    m = min(distances)
    delta = max(distances)-m
    threshold = m + (m/34.385) * delta

    lat_target, lon_target = target_pos

    return ({node:(1.0 if threshold/dist > 1 else max(0,threshold/dist -0.5))
                    for node,dist in zip(base_selection,distances)}

        , {node:(get_cos_phi((lat_pos,lon_pos),geo,target_pos)+1.0)/2.0
                    for node,geo in zip(base_selection,geos)}

        , {node:g(lat_pos,lon_pos,*target_pos) /
            (dist+g(lat_target,lon_target,*get_geo(G_in)(node)))
                for node,dist in zip(base_selection,distances)}

        , {n:d for n,d in zip(base_selection,distances)}
    )

def get_district_geo(district_code):
    geo = setup.district_geos[district_code]
    if type(geo) is not nx.MultiDiGraph:
        return geo
    else:
        return get_geo(geo)(np.random.choice(geo))

def get_geo(G_in):
    def of_node(n):
        return G_in.nodes('y')[n] , G_in.nodes('x')[n]
    return of_node

def get_cos_phi(pos_a,pos_b,pos_c):
    (ab_x,ab_y) = tuple_diff(pos_b,pos_a) #from a to b
    (bc_x,bc_y) = tuple_diff(pos_c,pos_b) #from b to c
    if bc_x == 0.0 and bc_y == 0.0:
        return 1.0 #can but shall not reach zero
    return (ab_x*bc_x+ab_y*bc_y)/(dist(ab_x,ab_y)*dist(bc_x,bc_y))

def tuple_diff(t1,t2):
    a,b = t1
```

```
    c,d = t2
    return (a-c,b-d)

def dist(a,b):
    return sqrt(a*a+b*b)
```

Listing B.3: entrypoints.py file containing the algorithm for the detection of entry and exit nodes as discussed in section 3.1 on finding ways in and out of a city.

```
# coding: utf8
import osmnx as ox
import networkx as nx

def find_entries(G_inner,G_outer=None):
    return _filter_candidates(G_inner.copy(),G_outer,
                G_inner.in_degree,G_inner.out_degree)

def find_exits(G_inner,G_outer=None):
    return _filter_candidates(G_inner.copy(),G_outer,
                G_inner.out_degree,G_inner.in_degree)

#make this the main export function in end version discarding f_fix and f_no
def _filter_candidates(G_in,G_out,f_fix,f_no):

    # including fix points only leads to errors
    # when applied to simultaniously generated
    # base and outer networks
    fix  = []# [n for n in G_in if f_fix(n) == 0 and G_in.degree(n) > 0]
    no   = [n for n in G_in if f_no(n)  == 0]
    n, s, e, w = get_boundary(G_in)

    if G_out == None:
        margin_ew = e-w
        margin_ns = n-s
        G_out = ox.graph_from_bbox(n+margin_ns,s-margin_ns,e+margin_ew,w-margin_ew
                ,network_type='drive',simplify=True)

    G_diff = G_out.copy()
    G_diff.remove_nodes_from( G_in )

    #remove edges, left over edges are those connecting G and G_diff
    G_out.remove_edges_from( G_diff.edges() )
    G_out.remove_edges_from( G_in.edges() )

    G_out.remove_nodes_from( no )

    #remove nodes connected with too little capacity roads
    G_cap = filter_by_speed( G_out )
    G_out.remove_nodes_from(
        [n for n in G_cap if G_cap.degree(n) < 1] )
    G_out.remove_nodes_from( G_diff )

    return list(set(list(G_out.nodes())+ fix))


def filter_by_speed(G_in):
    G_out = G_in.copy()

    fedges = [(u,v) for u, v, s in G_out.edges(data='maxspeed')
                if read_num(s) < 50]

    G_out.remove_edges_from( fedges )

    return G_out

#some elements in maxspeeds are lists of speeds
def read_num(elem):
    if type(elem) == list:
        elems = [read_speed(e) for e in elem]
        return max(elems)
    else:
        return read_speed(elem)
```

```
def read_speed(spdstr):
    try:
        return int(spdstr)
    except ValueError, e:
        return 30 #forbidden
    except TypeError, e:
        return 30 #allowed if strict mode disabled

def get_boundary(G_in):
    x = nx.get_node_attributes(G_in,'x').values()
    y = nx.get_node_attributes(G_in,'y').values()

    w = min(x)
    e = max(x)
    n = max(y)
    s = min(y)
    return n, s, e, w
```

Bibliography

Mohamed A. Abdel-aty, Kenneth M. Vaughn, Ryuichi Kitamura, Paul Jovanis, and Fred Mannering. Models of commuters' information use and route choice: Initial results based on a southern california commuter route choice survey. *Institute of Transportation Studies, UC Berkeley, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings*, 01 1993.

Sameera Abar, Georgios K. Theodoropoulos, Pierre Lemarinier, and Gregory M.P. O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, may 2017. doi: 10.1016/j.cosrev.2017.03.001.

Harold Abelson and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs*. MIT Press Ltd, 1996. ISBN 0262510871.

Kay W. Axhausen and. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, aug 2016. doi: 10.5334/baw.

Richard Arnott, Andre de Palma, and Robin Lindsey. A temporal and spatial equilibrium analysis of commuter parking. *Journal of Public Economics*, 45(3):301–335, aug 1991. doi: 10.1016/0047-2727(91)90030-6.

Javier Asensio. Transport mode choice by commuters to barcelona's CBD. *Urban Studies*, 39(10):1881–1895, sep 2002. doi: 10.1080/0042098022000003000.

Javier Asensio and Anna Matas. Commuters' valuation of travel time variability. *Transportation Research Part E: Logistics and Transportation Review*, 44(6):1074–1085, nov 2008. doi: 10.1016/j.tre.2007.12.002.

John Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, aug 1978. doi: 10.1145/359576.359579.

Michael Balmer, Kay Axhausen, and Kai Nagel. Agent-based demand-modeling framework for large-scale microsimulations. *Transportation Research Record: Journal of the Transportation Research Board*, 1985: 125–134, jan 2006. doi: 10.3141/1985-14.

S. C. Bankes. Agent-based modeling: A revolution? *Proceedings of the National Academy of Sciences*, 99(Supplement 3):7199–7200, may 2002. doi: 10.1073/pnas.072081299.

J. Barceló, L. Montero, M. Bullejos, O. Serch, and C. Carmona. Dynamic od matrix estimation exploiting bluetooth data in urban networks. In *Proceedings of the 14th International Conference on Automatic Control, Modelling & Simulation, and Proceedings of the 11th International Conference on Microelectronics, Nanoelectronics, Optoelectronics*, ACMIN'12, pages 116–121, Stevens Point, Wisconsin, USA, 2012. World Scientific and Engineering Academy and Society (WSEAS). ISBN 978-1-61804-080-0.

Ana Lécia C. Bazzan, Joachim Wahle, and Franziska Klügl. Agents in traffic modelling — from reactive to social behaviour. In Wolfram Burgard, Armin B. Cremers, and Thomas Cristaller, editors, *KI-99: Advances in Artificial Intelligence*, pages 303–306, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-48238-3.

David Beazley. *Python Essential Reference (4th Edition)*. Addison-Wesley Professional, 2009. ISBN 978-0-672-32978-4.

Moshe Ben-Akiva and Michel Bierlaire. *Discrete Choice Methods and their Applications to Short Term Travel Decisions*, pages 5–33. Springer US, Boston, MA, 1999. ISBN 978-1-4615-5203-1. doi: 10.1007/978-1-4615-5203-1_2.

Moshe Ben-Akiva and Michel Bierlaire. *Discrete Choice Models with Ap-*

plications to Departure Time and Route Choice, pages 7–37. Springer US, Boston, MA, 2003. ISBN 978-0-306-48058-4.

Emmanuel Bert, Edward Chung, and André-Gilles Dumont. Exploring the use of dta for origin-destination matrix estimation. In *Proceedings of the 6th STRC Swiss Transport Research*, 01 2006.

Chandra R. Bhat. Work travel mode choice and number of non-work commute stops. *Transportation Research Part B: Methodological*, 31(1):41–54, feb 1997. doi: 10.1016/s0191-2615(96)00016-1.

Chandra R. Bhat and Rupali Sardesai. The impact of stop-making and travel time reliability on commute mode choice. *Transportation Research Part B: Methodological*, 40(9):709–730, nov 2006. doi: 10.1016/j.trb.2005.09.008.

Michel Bierlaire and Emma Frejinger. Route choice modeling with network-free data. *Transportation Research Part C: Emerging Technologies*, 16 (2):187–198, apr 2008. doi: 10.1016/j.trc.2007.07.007.

Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002. ISSN 0027-8424. doi: 10.1073/pnas.082080899.

Rafael Borge, Isabel de Miguel, David de la Paz, Julio Lumbreras, Javier Pérez, and Encarnación Rodríguez. Comparison of road traffic emission models in madrid (spain). *Atmospheric Environment*, 62:461–471, dec 2012. doi: 10.1016/j.atmosenv.2012.08.073.

Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling : Reasons , techniques , tools. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, 2004.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

N. Caceres, J.P. Wideberg, and F.G. Benitez. Deriving origin–destination data from a mobile phone network. *IET Intelligent Transport Systems*, 1(1):15, 2007. doi: 10.1049/iet-its:20060020.

Xing Cai, Hans Petter Langtangen, and Halvard Moe. On the performance of the python programming language for serial and parallel scientific computations. *Scientific Programming*, 13(1):31–56, 2005. doi: 10.1155/2005/619804.

Charles Carpenter, Mark Fowler, and Thomas J. Adler. Generating route-specific origin–destination tables using bluetooth technology. *Transportation Research Record: Journal of the Transportation Research Board*, 2308(1):96–102, jan 2012. doi: 10.3141/2308-10.

Liang Chen. Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research*, 1(2):166–177, jun 2012. doi: 10.1016/j.foar.2012.03.003.

P.S. Chen and Hani Mahmassani. Dynamic interactive simulator for studying commuter behavior under real-time traffic information supply strategies. *Transportation Research Record Journal of the Transportation Research Board*, 1413:12–21, 11 1993.

Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2):346–366, 1932. ISSN 0003486X. URL `http://www.jstor.org/stable/1968337`.

Wilner Ciscal-Terry, Mauro Dell'Amico, Natalia Selini Hadjidimitriou, and Manuel Iori. An analysis of drivers route choice behaviour using GPS data and optimal alternatives. *Journal of Transport Geography*, 51:119–129, feb 2016. doi: 10.1016/j.jtrangeo.2015.12.003.

Christy M. Collins and Susan M. Chambers. Psychological and situational influences on commuter-transport-mode choice. *Environment and Behavior*, 37(5):640–661, sep 2005. doi: 10.1177/0013916504265440.

Loveday Conquest, Jan Spyridakis, Mark Haselkorn, and Woodrow Barfield. The effect of motorist information on commuter behavior: Classification of drivers into commuter groups. *Transportation Research Part C: Emerging Technologies*, 1(2):183–201, jun 1993. doi: 10.1016/0968-090x(93)90013-6.

Thomas J Cova and Justin P Johnson. Microsimulation of neighborhood evacuations in the urban–wildland interface. *Environment and Planning A: Economy and Space*, 34(12):2211–2229, dec 2002. doi: 10.1068/a34251.

Department For Transport. National travel survey, 2002-2016, 2017.

Xuan Di and Henry Liu. Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Research Part B: Methodological*, 85, 02 2016. doi: 10.1016/j.trb.2016.01.002.

H. B. Enderton. In memoriam: Alonzo church, 1903-1995. *Bull. Symbolic Logic*, 1(4):486–488, 12 1995.

Kutluhan Erol, Renato Levy, and J Wentworth. Application of agent technology to traffic simulation. Technical report, US Department of Transportation, 01 2000.

European Environment Agency. Greenhouse gas emissions from transport, November 2018. URL `https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases/transport-emissions-of-greenhouse-gases-11`. Access: 05.03.2019; 21:23.

Martin Fellendorf and Peter Vortisch. *Microscopic traffic flow simulator VISSIM*, chapter 2, pages 63–93. Springer, 06 2011. doi: 10.1007/978-1-4419-6142-6_2.

T. Fontes, S.R. Pereira, P. Fernandes, J.M. Bandeira, and M.C. Coelho. How to combine different microsimulation tools to assess the environmental impacts of road traffic? lessons and directions. *Transportation Research Part D: Transport and Environment*, 34:293–306, jan 2015. doi: 10.1016/j.trd.2014.11.012.

Lawrence Frank, Mark Bradley, Sarah Kavage, James Chapman, and T. Keith Lawton. Urban form, travel time, and cost relationships with tour complexity and mode choice. *Transportation*, 35(1):37–54, jul 2007. doi: 10.1007/s11116-007-9136-6.

E. Frejinger, M. Bierlaire, and M. Ben-Akiva. Sampling of alternatives for route choice modeling. *Transportation Research Part B: Methodological*, 43(10):984–994, dec 2009. doi: 10.1016/j.trb.2009.03.001.

Jose Manuel Galan, Luis R. Izquierdo, Segismundo S. Izquierdo, Jose Ignacio Santos, Ricardo del Olmo Martinez, Adolfo Lopez-Paredes, and Bruce Edmonds. Errors and artefacts in agent-based modelling. *J. Artificial Societies and Social Simulation*, 12, 2009.

Conor K. Gately, Lucy R. Hutyra, and Ian Sue Wing. Cities, traffic, and CO2: A multidecadal assessment of trends, drivers, and scaling relationships. In *Proceedings of the National Academy of Sciences*, volume 112, pages 4999–5004, apr 2015. doi: 10.1073/pnas.1421723112.

Jeremy Gibbons and Ralf Hinze. Just do it: simple monadic equational reasoning. In *Proceeding of the 16th ACM SIGPLAN international conference on Functional programming - ICFP '11*. ACM Press, 2011. doi: 10.1145/2034773.2034777.

P.G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2):105 – 111, 1981. ISSN 0191-2615. doi: 10.1016/0191-2615(81)90037-0.

Donald R. Glover and Julian L. Simon. The effect of population density on infrastructure: The case of road building. *Economic Development and Cultural Change*, 23(3):453–468, 1975. ISSN 00130079, 15392988.

Google Maps. Google maps, 2019. URL `https://www.google.com/maps`. Access: 25.02.2019; 14 : 35.

V. Grimm. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science*, 310(5750):987–991, nov 2005. doi: 10.1126/science.1116681.

Xiaolei Guo and Henry X. Liu. Bounded rationality and irreversible network change. *Transportation Research Part B: Methodological*, 45(10): 1606–1618, dec 2011. doi: 10.1016/j.trb.2011.05.026.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. In G Varoquaux, T Vaught, and J Millman, editors, *Proceedings of the 7th Python in Science conference (SciPy 2008)*, pages 11–15, 01 2008.

Frank A. Haight. Some probability distributions associated with commuter travel in a homogeneous circular city. *Operations Research*, 12(6):964–975, 1964. ISSN 0030364X, 15265463.

B. Heiden, M. Henn, M. Hinterhofer, O. Schechtner, and K. Zelle. Endbericht emissionskataster graz 2001. Technical Report FVT-08/08/Hin V&U 03/16/6100, Amt der Steiermärkischen Landesregierung Fachabteilung 17 c, Landhausgasse 7, 8010 Graz, February 2008. URL `http://www.umweltservice.graz.at/infos/luft/EmiKat_Gesamtbericht.pdf`. Access: 21.02.2019; 11 : 20.

Dirk Helbing, Ansgar Hennecke, Vladimir Shvetsov, and Martin Treiber. MASTER: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model. *Transportation Research Part B: Methodological*, 35 (2):183–211, feb 2001. doi: 10.1016/s0191-2615(99)00047-8.

David A. Hensher. The valuation of commuter travel time savings for car drivers: evaluating alternative model specifications. *Transportation*, 28 (2):101–118, 2001. doi: 10.1023/a:1010302117979.

Daniel Baldwin Hess. Effect of free parking on commuter mode choice: Evidence from travel diary data. *Transportation Research Record: Journal of the Transportation Research Board*, 1753(1):35–42, jan 2001. doi: 10.3141/1753-05.

Frank Höfler. *Verkehrswesen-Praxis*. Beuth Verlag, 2004. ISBN 3410216138.

Christian Hofer, Georg Jäger, and Manfred Füllsack. Generating realistic road usage information and origin-destination data for traffic simulations: Augmenting agent-based models with network techniques. In *Studies in Computational Intelligence*, pages 1223–1233. Springer International Publishing, nov 2017. doi: 10.1007/978-3-319-72150-7_99.

Christian Hofer, Georg Jäger, and Manfred Füllsack. Including traffic jam avoidance in an agent-based network model. *Computational Social Networks*, 5(1), may 2018a. doi: 10.1186/s40649-018-0053-y.

Christian Hofer, Georg Jäger, and Manfred Füllsack. Large scale simulation of CO2 emissions caused by urban car traffic: An agent-based network approach. *Journal of Cleaner Production*, 183:1–10, may 2018b. doi: 10.1016/j.jclepro.2018.02.113.

E.M. Holroyd. Theoretical average journey lengths in circular towns with various routeing systems. Technical Report 43, Crowthorne Road Research Laboratory, Ministry of Transport, 1966.

Paul Hudak. Conception, evolution, and application of functional programming languages. *ACM Comput. Surv.*, 21:359–411, 1989.

Paul Hudak, John Hughes, Simon Peyton Jones, and Philip Wadler. A history of haskell: Being lazy with class. In *In Proceedings of the 3rd ACM SIGPLAN Conference on History of Programming Languages (HOPL-III*, pages 1–55. ACM Press, 2007.

F. Huemer, J. Reithofer, P. Lüftenegger, G Strobl-Schilcher, H. Strasser, G. Baumgartner, N. Dorfinger, and D. Reiter. Energiebericht 2013 - smart city salzburg grundlagen für eine zukunftsfähige energie- und klimaschutzpolitik. Technical report, Stadt

Salzburg Magistrat, MA 6/00 Baudirektion, Smart City Koordination, 2015. URL https://www.stadt-salzburg.at/pdf/energiebericht_2013_der_stadt_salzburg.pdf. Access: 21.02.2019; 14 : 25.

John Hughes. Why functional programming matters. *Comput. J.*, 32:98–107, 1989.

INFRAS. *Handbuch Emissionsfaktoren des Straßenverkehrs*. UBA Berlin, BUWAL Bern, UBA Wien, Zürich, 3.1 edition, 2010.

Anahita Jamshidnejad, Ioannis Papamichail, Markos Papageorgiou, and Bart De Schutter. A mesoscopic integrated urban traffic flow-emission model. *Transportation Research Part C: Emerging Technologies*, 75:45–83, 02 2017. doi: 10.1016/j.trc.2016.11.024.

Erik Jenelius. Network structure and travel patterns: explaining the geographical disparities of road network vulnerability. *Journal of Transport Geography*, 17(3):234–244, may 2009. doi: 10.1016/j.jtrangeo.2008.06.002.

Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, and Christian S. Jensen. Path prediction and predictive range querying in road network databases. *The VLDB Journal*, 19(4):585–602, may 2010. doi: 10.1007/s00778-010-0181-y.

Yan-Qun Jiang, Pei-Jie Ma, and Shu-Guang Zhou. Macroscopic modeling approach to estimate traffic-related emissions in urban areas. *Transportation Research Part D: Transport and Environment*, 60:41–55, may 2018. doi: 10.1016/j.trd.2015.10.022.

H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling monte carlo method. National Bureau of Standards Applied Mathematics Series No. 12, 1951.

Evanthia Kazagli, Michel Bierlaire, and Gunnar Flötteröd. Revisiting the route choice problem: A modeling framework based on mental representations. *Journal of Choice Modelling*, 19:1–23, jun 2016. doi: 10.1016/j.jocm.2016.06.001.

Asad J. Khattak and André De Palma. The impact of adverse weather conditions on the propensity to change travel decisions: A survey of brussels commuters. *Transportation Research Part A: Policy and Practice*, 31(3): 181–203, may 1997. doi: 10.1016/s0965-8564(96)00025-0.

Franziska Klügl and Ana L. C. Bazzan. Simulation of adaptive agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02*. ACM Press, 2002. doi: 10.1145/544741.544795.

Franziska Kluegl and Ana L. C. Bazzan. Route decision behaviour in a commuting scenario: Simple heuristics adaptation and effect of traffic forecast. *Journal of Artificial Societies and Social Simulation*, 7(1):1–1, 2004.

G Kotusevski and Ken Hawick. A review of traffic simulation software. *Res. Lett. Inf. Math. Sci.*, 13, 08 2009.

Hans Petter Langtangen, editor. *Python Scripting for Computational Science*. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-73916-6.

Anahid Nabavi Larijani, Ana-Maria Olteanu-Raimond, Julien Perret, Mathieu Brédif, and Cezary Ziemlicki. Investigating the mobile phone data to estimate the origin destination flow and analysis; case study: Paris region. *Transportation Research Procedia*, 6:64–78, 2015. doi: 10.1016/j.trpro.2015.03.006.

John Launchbury and Simon L. Peyton Jones. State in haskell. *LISP and Symbolic Computation*, 8(4):293–341, Dec 1995. ISSN 1573-0557. doi: 10.1007/BF01018827.

Hainan Li, Randall Guensler, and Jennifer Ogle. Analysis of morning commute route choice patterns using global positioning system–based vehicle activity data. *Transportation Research Record: Journal of the Transportation Research Board*, 1926(1):162–170, jan 2005. doi: 10.1177/0361198105192600119.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes

Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

Charles Macal and Michael North. Tutorial on agent-based modeling and simulation PART 2: How to model with agents. In *Proceedings of the 2006 Winter Simulation Conference*. IEEE, dec 2006. doi: 10.1109/wsc.2006.323040.

Charles M. Macal and Michael J. North. Agent-based modeling and simulation: Desktop ABMS. In *2007 Winter Simulation Conference*. IEEE, dec 2007. doi: 10.1109/wsc.2007.4419592.

Hani Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1:267–292, 09 2001. doi: 10.1023/A:1012831808926.

Hani S. Mahmassani. Dynamic models of commuter behavior: Experimental investigation and application to the analysis of planned traffic disruptions. *Transportation Research Part A: General*, 24(6):465 – 484, 1990. ISSN 0191-2607. doi: 10.1016/0191-2607(90)90036-6.

E.J. Manley, S.W. Orr, and T. Cheng. A heuristic model of bounded route choice in urban areas. *Transportation Research Part C: Emerging Technologies*, 56:195–209, jul 2015. doi: 10.1016/j.trc.2015.03.020.

Simon Marlow. Fighting spam with haskell, June 2015. URL `https://code.fb.com/security/fighting-spam-with-haskell/`. Access: 23.02.2019; $3:06$.

Ellen Matthies, Silke Kuhn, and Christian A. Klöckner. Travel mode choice of women. *Environment and Behavior*, 34(2):163–177, mar 2002. doi: 10.1177/0013916502034002001.

David Mertz. *Functional Programming in Python*. O'Reilly Media, Inc., 2015.

Patricia Mokhtarian, Liang Ye, and Meiping Yun. The effects of gender on commuter behavior changes in the context of a major freeway reconstruction. In *Women's Issues in Transportation: Summary of the 4th International Conference, Volume 2: Technical Papers*, volume 2. Transportation Research Board, 05 2011. doi: 10.17226/22887.

E Morello, S Toffolo, and G Magra. Traffic models enhancements for properly assess environmental impacts of its/ict systems: generalities and eco-driving example. In *Transport Research Arena (TRA) 2014 Proceedings*, Paris, France, 2014.

M. Niazi and A. Hussain. Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks. *IEEE Communications Magazine*, 47(3):166–173, mar 2009. doi: 10.1109/mcom.2009.4804403.

Muaz Niazi and Amir Hussain. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics*, 89(2): 479–499, aug 2011. doi: 10.1007/s11192-011-0468-9.

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org` , 2017.

André De Palma and Denis Rochat. Understanding individual travel decisions: results from a commuters survey in geneva. *Transportation*, 26 (3):263–281, 1999. doi: 10.1023/a:1005121605832.

Dominik Papinski, Darren M. Scott, and Sean T. Doherty. Exploring the route choice decision-making process: A comparison of planned and observed routes obtained using person-based GPS. *Transportation Research Part F: Traffic Psychology and Behaviour*, 12(4):347–358, jul 2009. doi: 10.1016/j.trf.2009.04.001.

Simon L. Peyton Jones and Philip Wadler. Imperative functional programming. In *Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '93*. ACM Press, 1993. doi: 10.1145/158511.158524.

R. Pischinger. Emissionskataster graz 1995. Technical report, Amt der steiermärkischen Landesregierung, 1995. URL `http://www.umweltservice.graz.at/infos/luft/EMIKAT95.pdf`. Access: 21.02.2019; 11 : 20.

David A. Plaisted. *Handbook of Logic in Artificial Intelligence and Logic Programming: Volume 1: Logical Foundations*, chapter Equational reasoning and term rewriting systems, pages 274–364. OXFORD UNIV PR, 1993. ISBN 019853745X.

JW Polak, PM Jones, PC Vythoulkas, S Meland, and T Tretvik. The trondheim toll ring: Results of a stated preference study of travellers' responses. Technical report, Transport Studies Unit, University of Oxford, 1991.

Carlo Giacomo Prato and Shlomo Bekhor. Modeling route choice behavior. *Transportation Research Record: Journal of the Transportation Research Board*, 2003(1):64–73, jan 2007. doi: 10.3141/2003-09.

D. A. Quarmby. Choice of travel mode for the journey to work: Some findings. *Journal of Transport Economics and Policy*, 1(3):273–314, 1967. ISSN 00225258.

W. V. O. Quine. *Word & Object*. MIT Press, 1960.

Yihui Ren, Mária Ercsey-Ravasz, Pu Wang, Marta C. González, and Zoltán Toroczkai. Predicting commuter flows in spatial networks using a radiation model based on temporal ranges. *Nature Communications*, 5(1), nov 2014. doi: 10.1038/ncomms6347.

Paul Richmond and Daniela Romano. Agent based gpu, a real-time 3d simulation and interactive visualisation framework for massive agent based modelling on the gpu. In *In Proceedings International Workshop on Super Visualisation (IWSV08)*. Press, 2008.

Philipp Ringler, Dogan Keles, and Wolf Fichtner. Agent-based modelling and simulation of smart electricity grids and markets – a literature review. *Renewable and Sustainable Energy Reviews*, 57:205–215, may 2016. doi: 10.1016/j.rser.2015.12.169.

Cynthia Rosenzweig, William Solecki, Stephen A. Hammer, and Shagun Mehrotra. Cities lead the way in climate–change action. *Nature*, 467 (7318):909–911, oct 2010. doi: 10.1038/467909a.

Rosaldo J.F Rossetti, Rafael H Bordini, Ana L.C Bazzan, Sergio Bampi, Ronghui Liu, and Dirck Van Vliet. Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5-6):373–398, oct 2002. doi: 10.1016/s0968-090x(02)00027-x.

Christos Samaras, Dimitris Tsokolis, Silvana Toffolo, Giorgio Magra, Leonidas Ntziachristos, and Zissis Samaras. Improving fuel consumption and co2 emissions calculations in urban areas by coupling a dynamic micro traffic model with an instantaneous emissions model. *Transportation Research Part D: Transport and Environment*, 65:772 – 783, 2018. ISSN 1361-9209. doi: 10.1016/j.trd.2017.10.016.

Michel F. Sanner. Python: A programming language for software integration and development. *Journal of Molecular Graphics and Modelling*, 17 (1):55–84, feb 1999. doi: 10.1016/s1093-3263(99)99999-0.

Patrick M. Sansom and Simon L. Peyton Jones. Generational garbage collection for haskell. In *Proceedings of the conference on Functional programming languages and computer architecture - FPCA '93*. ACM Press, 1993. doi: 10.1145/165180.165195.

R. J. Smeed. The traffic problem in towns: A review of possible long term solutions. *The Town Planning Review*, 35(2):133–158, 1964. ISSN 00410020, 1478341X.

Robin Smit, Leonidas Ntziachristos, and Paul Boulter. Validation of road vehicle and traffic emission models – a review and meta-analysis. *Atmospheric Environment*, 44(25):2943–2953, aug 2010. doi: 10.1016/j.atmosenv.2010.05.022.

L. Smith, R. Beckman, and K. Baggerly. TRANSIMS: Transportation analysis and simulation system. Technical report, Los Alamos National Laboratory (LANL), jul 1995.

Jan Spyridakis, Woodrow Barfield, Loveday Conquest, Mark Haselkorn, and Carol Isakson. Surveying commuter behavior: Designing motorist information systems. *Transportation Research Part A: General*, 25(1): 17–30, jan 1991. doi: 10.1016/0191-2607(91)90152-g.

Karthik Srinivasan and Hani Mahmassani. *Dynamic Decision and Adjustment Processes In Commuter Behavior Under Real-Time Information*. Center for Transportation Research, University of Texas at Austin, 3208 Red River, Suite 200 Austin, TX, 01 2002.

Karthik K. Srinivasan and Hani S. Mahmassani. Modeling inertia and compliance mechanisms in route choice behavior under real-time information. *Transportation Research Record: Journal of the Transportation Research Board*, 1725(1):45–53, jan 2000. doi: 10.3141/1725-07.

Stadt Salzburg / Statistik. Bevölkerungsstatistik - statistische zählbezirke und zählsprengel, November 2018. URL `https://www.stadt-salzburg.at/internet/bildung_kultur/salzburg_in_zahlen/bevoelkerungsstatist_321402/statistische_zaehlbezirke_und_zaehlspren_251948.htm`. Access: 21.02.2019, 14:25.

Stadt Salzburg / Statistik. Bevölkerungsstatistik - bevölkerungszahlen 1.1.2019, January 2019. URL `https://www.stadt-salzburg.at/internet/bildung_kultur/salzburg_in_zahlen/bevoelkerungsstatist_321402/bevoelkerungszahlen_1_1_2019_476598.htm`. Access: 21.02.2019, 14:25.

Statistik Austria. Registerzählung 2011: Statistik austria pendlerstatistik, 2011.

Marius Thériault, Marie-Hélène Vandersmissen, Martin Lee-Gosselin, and Denis F Leroux. Modelling commuter trip length and duration within gis : Application to an od survey. *Journal of Geographic Information and Decision Analysis*, 3(1):41–55, 1999.

Tom Thomas and Bas Tutert. Route choice behavior in a radial structured urban network: Do people choose the orbital or the route through the city center? *Journal of Transport Geography*, 48:85–95, oct 2015. doi: 10.1016/j.jtrangeo.2015.08.013.

Elvira Thonhofer, Toni Palau, Andreas Kuhn, S Jakubek, and Martin Kozek. Macroscopic traffic model for large scale urban traffic network design. *Simulation Modelling Practice and Theory*, 80:32–49, 01 2018. doi: 10.1016/j.simpat.2017.09.007.

R. Tomschy, M. Herry, G. Sammer, R. Klementschitz, S. Riegler, R. Follmer, D. Gruschwitz, F. Josef, S Gensasz, R. Kirnbauer, and T. Spiegel. Österreich unterwegs 2013/2014. ergebnisbericht zur österreichweiten mobilitätserhebung ,,Österreich unterwegs 2013/2014". Herausgeber: Bundesministerium für Verkehr, Innovation und Technologie, 2016.

TomTom International BV. Tomtom traffic index, February 2016. URL `https://www.tomtom.com/en_gb/trafficindex/list?citySize=ALL&continent=EU&country=ALL`. Access: 21.02.2019; 11:25.

Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009. ISBN 0521747384.

A Turner. Angular analysis: a method for the quantification of space. *Turner, A. (2000) Angular analysis: a method for the quantification of space. Working paper. CASA Working Papers (23). Centre for Advanced Spatial Analysis (UCL), London, UK.*, 01 2000.

Philip Wadler. How enterprises use functional languages, and why they don't. In *The Logic Programming Paradigm*, pages 209–227. Springer Berlin Heidelberg, 1999. doi: 10.1007/978-3-642-60085-2_9.

Lihong Wang, Steven L. Jacques, and Liqiong Zheng. MCML—monte carlo modeling of light transport in multi-layered tissues. *Computer Methods and Programs in Biomedicine*, 47(2):131–146, jul 1995. doi: 10.1016/0169-2607(95)01640-f.

Kevin Washbrook, Wolfgang Haider, and Mark Jaccard. Estimating commuter mode choice: A discrete choice analysis of the impact of road pricing and parking charges. *Transportation*, 33(6):621–639, sep 2006. doi: 10.1007/s11116-005-5711-x.

www.toibe.com. Tiobe index for february 2019, February 2019. URL `https://www.tiobe.com/`. Access: 02.02.2019; $13:10$.

Hongli Xu, Jing Zhou, and Wei Xu. A decision-making rule for modeling travelers' route choice behavior based on cumulative prospect theory. *Transportation Research Part C: Emerging Technologies*, 19(2):218–228, apr 2011. doi: 10.1016/j.trc.2010.05.009.

S.K. Zegeye, Bart De Schutter, J Hellendoorn, Ewald Breunesse, and A Hegyi. Integrated macroscopic traffic flow, emission, and fuel consumption model for control purposes. *Transportation Research Part C: Emerging Technologies*, 31:158–171, 06 2013. doi: 10.1016/j.trc.2013.01.002.

Michael Zilske, Andreas Neumann, and Kai Nagel. Openstreetmap for traffic simulation. In *Proceedings of the 1st European State of the Map – OpenStreetMap conference*, pages 126–134, 07 2011.