FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Microscopic Traffic Assignment under Dynamic Demand

**Daniel Pereira Machado**




Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rosaldo J. F. Rossetti

Second Supervisor: Sara Linhas Paiva

July 5, 2020

# Microscopic Traffic Assignment under Dynamic Demand

**Daniel Pereira Machado**

Mestrado Integrado em Engenharia Informática e Computação

July 5, 2020

# Abstract

In the last few decades, transportation systems have played a crucial part of human activities, increasing population dependence on them. Thus, Intelligent Transportation Systems (ITS) have emerged as an efficient way to deal with vehicular congestion, analysing and predicting the vehicles behaviour with traffic simulation models [ZWW+11].

Traffic simulation models aim to be a faithfully representation of the urban scenario they simulate. This approximation to a real scenario has to be carried out through calibration procedures, comparing simulated traffic values with sensors data. In fact, there is no such a generic method widely applicable to every single situation for traffic networks differ quite considerably, and as such, a specific calibration is required for each network simulation model.

Furthermore, calibration is a very laborious and timing consuming part of building valid network models, which requires vehicles readjustment to actual routes. Under dynamic demand, vehicle's flows are assigned between origin and destination pairs for a determined period of time. However, those metrics do not assure that route choice is always similar to reality. Frequently, the majority of simulated vehicles are assigned into the fastest route, which is not a credible representation of real-life traffic environments. Indeed, drivers will generally choose among a limited number of path options, which are not necessarily the quickest paths and will very much depend on drivers' knowledge of the network topology. Thus, this report proposes a calibration methodology able to improve route assignment under dynamic demand focusing on the network modelling parameters.

The proposed calibration method results from a case study involving the the set up of a traffic simulation model for the urban area of Porto City. Having SUMO selected to underlie the microscopic simulation of this case study, route assignment algorithms are chosen and tested allowing for a finer tuning of the devised network model. Among the algorithms available in SUMO, the Standard, Iterative, and Incremental are the chosen approaches, which are applied to the selected scenarios in this study. Results are then discussed and analyzed accordingly.

The main contributions of this research are the improvement of simulation calibration through route assignment as well as deadlocks discovery and correction. Through Iterative and Incremental approaches was possible to reduce simulation error measures, when compared to the default Standard assignment, previously used by Armis. Were further discovered focus of congested traffic, during calibration process. A reduction on simulation error measures was noticed by correcting some of them.

**Keywords**: Route Assignment , Traffic Simulation Models, Microscopic Simulation, Transportation Planning

# Resumo

Nas últimas décadas, os sistemas de transporte tornaram-se uma parte fundamental do quotidiano, aumentando assim, a sua dependência por parte da população. Deste modo, surgem os Sistemas Inteligentes de Transporte (ITS), relevando serem uma forma eficiente de lidar com engarrafamentos, uma vez que procederam à análise e previsão do comportamento de veículos em sistemas de simulação de tráfego [ZWW+11].

Os modelos de simulação de tráfego têm como objetivo ser uma fiel representação do cenário urbano que simulam. Esta aproximação a um cenário real tem que ser feita através de procedimentos de calibração, comparando os valores simulados com informação proveniente de sensores de trânsito. Acontece que não existe nenhum método de calibração que possa ser aplicado a qualquer situação, uma vez que as redes urbanas diferem bastante entre si, assim sendo, cada simulação necessita de um método específico de calibração.

Além disto, o processo de calibração é muito trabalhoso e consome bastante tempo até se conseguir um modelo válido, sendo necessário um reajuste dos veículos para as rotas correspondentes à realidade. Através de procura dinâmica, a circulação de veículos é definida em pares origem-destino, para um determinado intervalo de tempo. No entanto, estas métricas não asseguram que a escolha da rota seja sempre semelhante à realidade. Frequentemente, a maioria dos veículos simulados são afetados à rota mais rápida, o que não é uma representação credível da realidade. O que acontece de facto é que os condutores podem escolher entre um número limitado de caminhos possíveis, muitas vezes não sendo os caminhos mais rápidos, dependendo também do conhecimento do condutor relativamente à topologia da rede. Assim, este relatório propõe uma metodologia de calibração capaz de melhorar a afetação de rotas em ambientes de procura dinâmica, dando ênfase aos parâmetros de modelação da rede.

O método de calibração proposto resulta de um caso de estudo, que envolve a configuração de um modelo de simulação de tráfego para a zona urbana da cidade do Porto. Uma vez que o SUMO já havia sido escolhido como simulador de tráfego para o caso em estudo, os algoritmos de afetação de rotas são escolhidos e testados, de modo a permitir um melhor ajuste à referida rede. Entre os algoritmos disponíveis no SUMO, foram selecionadas as abordagens Padrão, Iterativa e Incremental, sendo aplicadas aos cenários selecionados. De seguida, os resultados são discutidos e analisados.

Este projeto tem como principais contribuições uma calibração melhorada através de afetação de rotas, assim como a descoberta e correção de bloqueios na rede. Através das abordagens Iterativa e Incremental foi possível reduzir as medidas de erro da simulação quando comparadas com a abordagem Padrão, utilizada previamente pela Armis. Durante o processo de calibração foram ainda descobertos focos de congestionamento de trânsito. Uma diminuição nas medidas de erro foi verificada aquando a correção de alguns dos mesmos.

**Keywords**: Afetação de Rotas, Simuladores de Tráfego, Simulação Microscópica, Planeamento de Transportes

# Acknowledgements

I would like to thank everyone that accompanied me throughout this journey, not just along this research project but also throughout these five amazing years. Those were, undoubtedly, the best years of my life until now, and I could not close this cycle without acknowledging everything they did for me.

First of all, I will mention my *parents*. When I first expressed my will of pursuing a computer engineering career, they were not supportive at all, but when they realised there was no going back, they conformed themselves and changed their mind, sponsoring all my adventures and supporting me in difficult times. I also have to thank them for giving me a fellow, who has become more of a friend than a *brother*. It was extraordinary to observe his growth along his university journey, I am proud to say that today I have a much mature housemate. To sum up, I would like to thank my entire *family*, particularly my *uncles* and *cousin* for putting up with me for a whole year, being extremely understanding with my daily routine.

Then, I would like to mention my *friends*. Along these five years I met the most extraordinary people ever. I have been involved in several organizations, connecting with lots of people with very different visions and background. I must thank all of them for my professional and personal growth, particularly two of them, which will always have a special place in my heart, *AEFEUP* and *Rotaract UP*. Furtherly, I must leave a special word to my *close friends*. I never though it was possible to meet so different people, but with so much chemistry. I strongly believe this was just the beginning of our journey. From this group of people I must emphasize my two *eternal project partners* for putting up with me during our long hours of late productivity. Finally, I would like to thank my *long-standing friends*, that despite studying in different cities across the country, continued reserving some time for a seldom weekend coffee.

In addition to this, I would like to mention my *girlfriend*. Thanks for all supportive words during this quarantine period, she knows better than anyone how difficult it was for me to be at home for so long. I must say, that she is, undoubtedly, one of the best rewards I could get along my academic journey.

Last but not least, I must mention my *supervisors*. I would be lying if I said this was a straightforward project. Indeed, it was a much harder process than I could imagine at first. Due to the lack of data, professor *Rossetti* offered me the opportunity of changing my research topic, introducing me to *Sara* and, consequently, this traffic simulator. I have to thank them for supporting my inner confidence and optimism, even when we were facing strongly discouraging situations.

I do not want to go on for very much long, so, I would like to end this section saying that writing this couple of paragraphs made me realise something: *I'm gonna miss this, a lot...*

Daniel Machado

*"Façam o favor de ser felizes!"*

Raúl Solnado

# Contents


**1 Introduction** **1**
- 1.1 Context . . . 1
- 1.2 Motivation and Goals . . . 2
- 1.3 Report Structure . . . 2

**2 Literature Review** **5**
- 2.1 Traffic Simulation Models . . . 5
  - 2.1.1 Macroscopic . . . 6
  - 2.1.2 Mesoscopic . . . 7
  - 2.1.3 Microscopic . . . 7
- 2.2 Calibration Models at Microscopic Simulation . . . 9
- 2.3 Route Assignment Algorithms . . . 11
  - 2.3.1 All or Nothing Assignment . . . 11
  - 2.3.2 Incremental Assignment . . . 12
  - 2.3.3 Equilibrium Assignment . . . 12
- 2.4 Route Assignment in SUMO . . . 16
  - 2.4.1 Overview . . . 16
  - 2.4.2 Dynamic User Assignment . . . 17
  - 2.4.3 Standard Assignment . . . 17
  - 2.4.4 Iterative Assignment . . . 19
  - 2.4.5 Incremental Assignment . . . 22
- 2.5 Summary . . . 23

**3 Methodological Approach** **25**
- 3.1 Problem Identification . . . 25
- 3.2 Proposed Method . . . 26
  - 3.2.1 SUMO Microscopic Simulation Structure . . . 26
  - 3.2.2 Running Route Assignment Models in SUMO . . . 29
  - 3.2.3 The Case-study Network . . . 31
- 3.3 Porto's network test bed . . . 32
- 3.4 Summary . . . 34

**4 Result Analysis and Discussion** **37**
- 4.1 Preliminary Results . . . 37
- 4.2 Experimental Results for the Study Network . . . 39
  - 4.2.1 Initial Network Results . . . 39
  - 4.2.2 Improved Network Results . . . 43
- 4.3 Discussion . . . 46

4.4 Summary . . . . . 47

**5 Conclusions and Future Work** **49**
5.1 Overview . . . . . 49
5.2 Main Contributions . . . . . 50
5.3 Further Developments . . . . . 50
5.4 Future Work . . . . . 50

**A Algorithms' detailed running options** **53**

**B Porto TAZ zones** **61**

**References** **63**

# List of Figures


2.1 SUMO's trips generation tools diagram . . . . . 17
2.2 Running options diagram for *duarouter* . . . . . 18
2.3 Running options diagram for duaIterate.py . . . . . 21
2.4 Running options diagram for one-shot.py . . . . . 23

3.1 Quickstart simulation network diagram . . . . . 27
3.2 Quickstart network - visual representation . . . . . 28
3.3 10x10 network - visual representation . . . . . 29
3.4 10x10 simulation network diagram . . . . . 30
3.5 Porto simulation network diagram . . . . . 33
3.6 Porto urban area - shapefiles representation . . . . . 34
3.7 Total number of trips in Porto urban area for 24 hours . . . . . 35

4.1 Possible routes for *quickstart* network . . . . . 37
4.2 Possible routes for *10x10* network . . . . . 38
4.3 Off-peak error measurement for initial network . . . . . 41
4.4 Peak hour error measurement for initial network . . . . . 43
4.5 Off-peak error measurement for improved network . . . . . 44
4.6 Peak hour error measurement for improved network . . . . . 46

# List of Tables


4.1 Number of possible routes for preliminary networks . . . 38
4.2 Real traffic values of Porto urban area from 4h00 to 5h00 . . . 39
4.3 Duarouter simulated traffic values of Porto urban area from 4h00 to 5h00 . . . 39
4.4 Dua Iterate simulated traffic values of Porto urban area from 4h00 to 5h00 . . . 40
4.5 One Shot simulated traffic values of Porto urban area from 4h00 to 5h00 . . . 40
4.6 Real traffic values of Porto urban area from 8h00 to 9h00 . . . 41
4.7 Dua Router simulated traffic values with initial network from 8h00 to 9h00 . . . 42
4.8 Dua Iterate simulated traffic values with initial network from 8h00 to 9h00 . . . 42
4.9 One Shot simulated traffic values with initial network from 8h00 to 9h00 . . . 42
4.10 Dua Router simulated traffic values with improved network from 4h00 to 5h00 . 43
4.11 Dua Iterate simulated traffic values with improved network from 4h00 to 5h00 . . 44
4.12 One Shot simulated traffic values with improved network from 4h00 to 5h00 . . . 44
4.13 Dua Router simulated traffic values with improved network from 8h00 to 9h00 . 45
4.14 Dua Iterate simulated traffic values with improved network from 8h00 to 9h00 . . 45
4.15 One Shot simulated traffic values with improved network from 8h00 to 9h00 . . . 45

A.1 Input options for *duarouter* . . . 53
A.2 Configuration options for *duarouter* . . . 53
A.3 Time options for *duarouter* . . . 54
A.4 Random number options for *duarouter* . . . 54
A.5 Output options for *duarouter* . . . 54
A.6 Default options for *duarouter* . . . 55
A.7 Processing options for *duarouter* - part 1 . . . 55
A.8 Processing options for *duarouter* - part 2 . . . 56
A.9 Report options for *duarouter* . . . 57
A.10 Ouput options for *duaIterate.py* . . . 57
A.11 Input options for *duaIterate.py* . . . 58
A.12 Configuration options for *duaIterate.py* . . . 58
A.13 Algorithm's options for *duaIterate.py* . . . 58
A.14 Incremental assignment options for *duaIterate.py* . . . 59
A.15 Mesoscopic simulation options for *duaIterate.py* . . . 59
A.16 Processing options for *duaIterate.py* . . . 59
A.17 Input options for *one-shot.py* . . . 60
A.18 Output options for *one-shot.py* . . . 60
A.19 Processing options for *one-shot.py* . . . 60

B.1 Porto Traffic Assignment Zones . . . 62

# Abbreviations

| | |
|---|---|
| AAE | Average Absolute Error |
| BPR | Bureau of Public Roads |
| CH | Contraction Hierarchies |
| C-ITS | Cooperative Intelligent Transport Systems |
| CMP | Câmara Municipal do Porto |
| DFROUTER | Detector Files ROUTER |
| DUA | Dynamic User Assignment |
| DUAROUTER | Dynamic User Assignment ROUTER |
| DUE | Dynamic User Equilibrium |
| FRESIM | FREeway Simulation Computer Model |
| GNL | Generalized Nested Log |
| ID | Identification Designation |
| IEEE | International Smart Cities Conference |
| INTRAS | INtegrated TRAffic Simulator |
| IP | Infraestruturas Portugal |
| ITS | Intelligent Transport Systems |
| JTRROUTER | Junctions Turning Ratios ROUTER |
| LKW | Last Kraft Wagen |
| LWR | Lighthill-Whitham-Richards |
| MAROUTER | Macroscopic ROUTER |
| MITSIM | Microscopic Traffic SIMulator |
| ML | Mixed Logit |
| NETCONVERT | NETwork CONVERTor |
| NETEDIT | NETwork EDIT |
| NETGENERATE | NETwork GENERATor |
| NL-ASC | Nested Logit with Alternative-Specific Constant |
| OD | Origin Destination |
| PARAMICS | PARAllel MICroscopic Simulation |
| PKW | Personen Kraft Wagen |
| RMS | Root Mean Square |
| RMSE | Root Mean Square Error |
| SO | System Optimal |
| STCP | Sociedade Transportes Coletivos do Porto |
| SUE | Stochastic User Equilibrium |
| SUMO | Simulation Urban MObility |
| TAZ | Traffic Assingnment Zones |
| TIS | Transportes Inovação e Sistemas |
| UE | User Equilibrium |
| VISUM | VISion Understanding and Machine intelligence |

# Chapter 1

# Introduction

This chapter will introduce the scope of the project, describing the problems that we will try to solve. In this first Section 1.1, the area where this project is inserted will be described, the second Section 1.2 reports the motivation and goals of this project, and at last, in Section 1.3 , how is this report structured and what can be found in each chapter.

## 1.1 Context

In the last few decades, transportation systems have played a crucial part of human activities, increasing population dependence on them. Nowadays, as the number of vehicles on the road increases and with the lack of development of roads infrastructure, transportation systems face some challenges, namely traffic congestion, which has become an increasingly important issue globally. This way, Intelligent Transportation Systems (ITS) have emerged as an efficient way to deal with vehicular congestion, analysing and predicting the vehicles behaviour with traffic simulation models [ZWW+11].

Traffic simulation models aim to be a faithfully representation of the urban scenario they simulate. The effectiveness of its analysis and predicting features depend on its proximity to reality.This approximation to a real scenario has to be carried out through calibration procedures, comparing simulated traffic values with sensors data. In fact, there is no method widely applicable to every single situation, as such, a specific calibration is required for each simulation.

Furthermore, proper calibration requires vehicles readjustment to actual routes. Under dynamic demand, vehicle's flows are assigned between origin and destination for a determined period of time. However, those metrics do not assure that route choice is similar to reality. Frequently, the majority of simulated vehicles are assigned into the fastest route, which is not a credible representation of real-life traffic environments. Thus, a calibration system able to improve route assignment under dynamic demand, would represent a major contribution to the ITS paradigm.

## 1.2 Motivation and Goals

The main goal of this project is to improve an existing traffic simulation model, bridging the gap between simulated and real traffic values. The improved simulation will be able to deal with vehicles' routes with better accuracy, modeling them more realistically.

This research will contribute for the improvement of *Porto* traffic simulation, but also for the calibration of other networks with similar characteristics. It is expected to develop a calibration methodology able to reduce error measurements by applying different route assignment algorithms. Under dynamic demand, vehicles tend to choose the fastest path between origin and destination, which is not a reliable representation of real scenarios. Proper route assignment techniques are expected to improve the simulation performance.

The final result will represent a major contribution to *Porto*'s decision-support systems. A more accurate representation of the given network will result in better predictions for traffic values in real time, but also to anticipate contingency measures. It will allow decision-makers to know in advance possible network constraints, acting according to them.

## 1.3 Report Structure

This report is divided in five different chapters, where each of them is divided into sections and subsections, so that the reader can easily explore distinct topics. This structure section is included on Chapter 1, an introductory part of this report. At this chapter is explained what is the scope and the context of the project, as well as which are the motivations and the goals. Secondly, Chapter 2 summarizes several traffic simulation models and techniques related to calibration, with major focus on route assignment. This is the State of the Art or the Literature Review as the Chapter is named. This Chapter is structured on four main sections: Traffic Simulation Models (2.1), Calibration Models at Microscopic Simulation (2.2), Route Assignment Algorithms (2.3) and Route Assignment in SUMO (2.4). On each of those sections several concepts and techniques are referenced. Finally, an overall summary is made, discussing what are the most promising techniques for the given case-study. Additionally, Chapter 3 is about the methodological approach. At Section 3.1 is possible to understand what is the real problem we are trying to solve (problem formalization). In Section 3.2, a methodological approach is presented, describing the numerous steps that should be followed in order to improve the simulation performance and then, in Section 3.3, all the materials and case-study data are presented. Moreover, Chapter 4 analyzes and discusses obtained results. Results are presented in two different sections: Section 4.1 is focused on initial experiments to gain sensitivity with algorithms requirements, whereas Section 4.2 analyzes the case-study results. Afterwards, both results are discussed on Section 4.3. Lastly, Chapter 5 is focused on the conclusions. A general overview (5.1) is followed by project main contributions (5.2). Then, Section 5.3 and Section 5.4 describe some of the work that may be developed in

the future. Two appendixes are also included on this report. Appendix A presents several complementary tables to Section 2.4 of Literature Review, providing detailed options for considered algorithms, whereas, Appendix B illustrates the studied network TAZ (Traffic Assignment Zones)

# Chapter 2

# Literature Review

This Chapter summarizes several traffic simulation models and techniques related to calibration, with major focus on route assignment, being structured on four main sections: Traffic Simulation Models (2.1), Calibration Models at Microscopic Simulation (2.2), Route Assignment Algorithms (2.3) and Route Assignment in SUMO (2.4). On each of those sections, several concepts and techniques are referenced. Finally, an overall resume is made (2.5), discussing what are the most promising techniques for the given case-study.

## 2.1 Traffic Simulation Models

Traffic volume has substantially grown during the last few decades, while the development of roads infrastructure has been insufficient for those traffic values [NH16]. Given that, Intelligent Transportation Systems (ITS) are being developed, providing solutions to vehicles congestion by analysing and predicting the vehicles behaviour with traffic simulation models.

In 1934, Greenshields' research, [GTDS34], defined the fundamental traffic flow description on roadways. For the first time, traffic flow, density and speed measurement tests were performed. Greenshields developed a mathematical model, establishing linear relationship between the flow, speed and density of vehicles on the roadways. Over the years, different traffic flow description models have evolved to give an in-depth detail of the interaction between vehicles and the roadway. Such models include the macroscopic, mesoscopic and microscopic models. [Ade17]

Microscopic models are models that continuously or discretely predict the state of individual vehicles, presenting individual measures of vehicle speeds and locations. On the other hand, macroscopic models aggregate the description of traffic flow, dealing with traffic stream flows, densities, and average speeds. Mesoscopic models are models that have aspects of both macro and microscopic models. Additionally, simulation models can also be classified by its functionality: signal, freeway, or integrated [BY00]. For further discussions on simulation models, methodologies and granularity approaches, the interested reader is referred to [PRK11, TARO10].

### 2.1.1 Macroscopic

Macroscopic models are concerned with massive flows of vehicles, working on the basis of statistics. A general description of traffic is presented as well as a continuum flow. The simulation in a macroscopic model takes place on a section-by-section basis rather than by tracking individual vehicles [Mey16]. Macroscopic traffic models were originally developed to model traffic in distinct transportation subnetworks, such as freeways, corridors (including freeways and parallel arterials), surface-street grid networks, and rural highways. Usually they convert single-entity units (from microscopic models) into global traffic flows, allowing to simulate traffic dynamics in several lanes by effective one-lane models considering a certain probability of overtaking.

Those models describe vehicle dynamics considering spatial vehicle density$\rho(x,t)$ and average velocity $V(x,t)$ as a function of location ($x$) and time ($t$), [HHST02]. Considering the fact that there is a conservation of the number of vehicles, every macroscopic model is based on this continuity equation 2.1 for vehicles density. Vehicles rate entering or leaving is given by $v(x,t)$, considering a sections of length $\partial x$.

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho V)}{\partial x} = v(x,t) \tag{2.1}$$

The origins of macroscopic approaches stems from the assumption that traffic streams are comparable to fluid streams [Ade17]. In 1955, Lighthill and Whitham presented the first major research[LW55] following this approach. Their model compared 'traffic flow on long crowded roads' with 'flood movements in long rivers'. One year later Richards [Ric56], went further, presenting the idea of 'shock waves on highway' completing LWR (Lighthill-Whitham-Richards) model. Several models appear after their kick-off, with different approaches on how to calculate velocity $V(x,t)$ equation. A generic formulation of this equation may be found here, 2.2

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} + \frac{1}{\rho}\frac{\partial P}{\partial x} = \frac{1}{\tau}(V_e - V) \tag{2.2}$$

$V\frac{\partial V}{\partial x}$ = Transport term $\quad \frac{1}{\rho}\frac{\partial P}{\partial x}$ = Pressure term $\quad$ s $\frac{1}{\tau}(V_e - V)$ = Relaxation term

Lighthill-Whitham model describes the equilibrium relation as $V(x,t) = V_e(\rho(x,t))$, being followed by further articles with the same assumption: [New93], [Dag94], [Dag95], [Leb97] and [HW95]. On the other hand, Payne [PAY71] and Papageorgiou [Pap83] have a different pressure definition, $P(\rho) = [V_0 - V_e(\rho)]/(2\tau)$, assuming average velocity as $V_0 = V_e(0)$. Cremer goes even further on his model [Cre79], defining it as $\frac{\partial V}{\partial x} = -\frac{\rho}{2\tau(\rho+k)}\frac{\partial V_e}{\partial \rho}$. Phillips [Phi79] introduces the concept of velocity variance, $\theta$, where $P = \rho\theta$. Kühne [Küh84] [Kuh87], Kerner and Konhäuser [KK94], and Lee [LLK98] use this variance paradigm on their KKKL model, defining pressure as $P = \rho\theta_0 - \eta\frac{\partial V}{\partial x}$, considering $\theta_0$ a positive constant and $\eta$ a viscosity coefficient. In Whittham [Whi99] model the fraction $-\eta\frac{\partial V}{\partial}x$ needs an additional term for viscosity $(\eta/\rho)\frac{\partial^2 V}{\partial x^2}$. The goal of this is reducing shock fronts (empirical and numerical desirable).

### 2.1.2 Mesoscopic

Mesoscopic models combine the properties of both microscopic (discussed in Section 2.1.3) and macroscopic simulation models, fitting the gap between the aggregation level approaches (macroscopic) and individual interactions. Those intermediate models use high level detail to describe traffic entities, but individual behaviour and interactions are described with a lower level of detail, [Bur]. Given this, mesoscopic models provide less fidelity than microsimulation tools, but are superior to the typical planning analysis techniques (similar to macroscopic simulations). It is a good approach for situations where the detail of microscopic simulations is desirable, but unachievable because of the network large size or limited resources to code and debug the simulation.

There are two main approaches for this kind of simulation: vehicles are grouped in platoons and move along the link or individual vehicle's dynamics are simplified. The first approach is based on queuing theory [Erl]. Each network link (edge) is splint into a queuing part and a moving part. Vehicles in the queuing part are delayed due to capacity constraints at the link downstream, whereas the moving part computes link density from real data. The second approach uses vehicle loading model and route choice model to maximize predefined goals, but, to simplify the process, the simulation does not describe the interaction between vehicles, such as lane changing and car following models of microscopic simulation.

Another aspect of these simulations is how time is handled, it is possible to use either time-step or event-based model. Time-step model requires more computational resources than event-based, and it offers better support for real-time traffic management. On the other hand, despite being faster, event-based does not work efficiently for complex systems.

There are several paradigms in mesoscopic simulations. CONTRAM model [LGT89], groups vehicles into packets, acting as one single entity. This model is based on speed-density modelling vehicles behaviour according to that function. Another approach consists on grouping vehicles into cells, and those cells control their behaviour. This paradigm was used by Khattak [KPBA96] to develop DynaMIT model. Jayakrishnan [JMH94], Gawron [Gaw98] and Florian [FMT01] have pursued another path, a queue-server approach. Those models configure the roadway as a queuing and a running part, combining the advantages of dynamic disaggregated traffic stream modelling (since the vehicles are modelled individually), with the ease of calibration and use of macroscopic speed/density relationships. Cellular automaton is another option for mesoscopic simulations, where the road is divided into cells that may either be empty or occupied by one vehicle. Nagel [NS92a] and Barret [BBB+] are the most notorious applications of cellular automaton.

### 2.1.3 Microscopic

Microscopic traffic flow models are focused on individual vehicles, predicting speed and location of single vehicle units. Those measures are mathematically expressed as positions, $x_\alpha(t)$ and velocities, $x_\alpha(t)$ of interacting vehicles - $\alpha$. This kind of simulation may follow two different approaches: car-following or cellular automaton models.

Also commonly known as time-continuous models, car-following models are based on ordinary differential equations, describing vehicles' positions and velocities. Besides velocity, net distance is part of the general equation. This component represents bumper-to-bumper distance and can be expressed through Equation 2.3, where $\ell_{\alpha-1}$ defines the length of the leading vehicle, $\alpha-1$. The motion Equation 2.4 of each vehicle, can be depicted as an acceleration function, depending on the inputs above explained. A more generalized equation can be found in 2.5, considering not just the leader, $\alpha-1$, but also the $n_a$ vehicles in front.

$$s_\alpha = x_{\alpha-1} - x_\alpha - \ell_{\alpha-1} \tag{2.3}$$

$$\ddot{x}_\alpha(t) = \dot{v}_\alpha(t) = F(v_\alpha(t), s_\alpha(t), v_{\alpha-1}(t)) \tag{2.4}$$

$$\dot{v}_\alpha(t) = f(x_\alpha(t), v_\alpha(t), x_{\alpha-1}(t), v_{\alpha-1}(t), \ldots, x_{\alpha-n_a}(t), v_{\alpha-n_a}(t)) \tag{2.5}$$

One of the most notorious models was presented in 1961, by Newell [New61], with the main idea of keeping a minimum space and time gap between two vehicles. In 1995, another remarkable approach was introduced by Bando [BHN+95], representing an optimal-velocity model. Another important goal was to anticipate the driver behaviour, so that collisions could be avoided. This way, velocity-dependent acceleration models were proposed. Gipps' [Gip81] and Helbing's [HT98] models are an example of this kind of solution, but also Krauß [Kra], whose 1998's dissertation on Collision Free Vehicle Dynamics would be the basis of SUMO, a very popular traffic simulator. There are even more complex models, named "high-fidelity models", such as Wiedemann's [Wie74] and MITSIM, [Mic], making the simulation as realistically as possible, but recurring to several highly complex parameters, very difficult to calibrate.

Cellular automaton models is the other main approach when it comes to microscopic simulation. System dynamical properties are described by integer variables. The road is divided into sections of a certain length $\Delta x$ whereas time is assigned to steps of $\Delta t$. Given that, each road section maybe occupied by one vehicle or no vehicle at all. Dynamics are given by equations 2.6 and 2.7.

$$v_\alpha^{t+1} = f(s_\alpha^t, v_\alpha^t, v_{\alpha-1}^t, \ldots) \tag{2.6}$$

$$x_\alpha^{t+1} = x_\alpha^t + v_\alpha^{t+1} \tag{2.7}$$

Simulation time $t$ is measured in $\Delta t$, whereas $\Delta x$ units are used for vehicle positions $x$. With $\Delta t$ fixed, the length of the road sections determines the granularity of the model. Is defined that, at complete standstill, one vehicle occupies approximately 7.5 meters of road length.

Rule 184 is a big reference in cellular automaton paradigm. This particle-hopping model resemble to Li's [Li87] and Krug's [KS88] research, as a one-dimensional model. Despite being very primitive, its traffic flow application already predicts the basic behaviour on a real road, simulating free-flow vehicles for light traffic and stop-and-go ghost jams for heavy traffic. This

basic approach inspired the work of many authors, such as Nagel [NS92b] and Biham [BML92]. Kai Nagel and Michael Schreckenberg have developed a notorious cellular automaton model to reproduce traffic jams. Nagel–Schreckenberg model [NS92b], shows how vehicles interact on crowded amazingly closely to a real car density situation. Biham–Middleton–Levine traffic model [BML92], was first formulated by Ofer Biham, A. Alan Middleton, and Dov Levine in 1992, and evolves Rule 184 into a two-dimensional model.

Those models are less accurate when compared to car-following models but are pretty effective when it comes to large roads networks in real time or even faster. This can be explained by its simplicity, allowing them to be numerically strongly efficient.

## 2.2 Calibration Models at Microscopic Simulation

Through out the years, calibration procedures have been discussed. However, there is no methodology that would work on any simulator. Developed strategies are very specific, working well for a particular simulation, but failing among the others.

INTRAS [WL80], MITSIM [BADT+], PARAMICS [RRR15] and FRESIM [CJN+98] are the most widely known examples of calibration. However, a different process is used on each one of them. INTRAS performed a sequential parameter calibration, considering that while a specific parameter was calibrated, the remaining ones were considered constants. The main goal was to reduce the regression coefficient and the RMS (Root Mean Square) percent error. In MITSIM both driver and travel behaviour were calibrated. Travel behaviour calibration followed a similar method when comparing to INTRAS: the goal was to minimize the sum of squares of errors (RMS) between the simulated and actual sensor speeds. PARAMICS performed several simulations with multiple combinations of parameters (mean headway and mean reaction time) using average network speed and maximum vehicle throughput as performance indicators. Lastly, FRESIM used Goldberg's genetic algorithm [Gol89] optimization technique to find the best parameters combination, using Average Absolute Error (AAE) as objective function.

Despite the different approaches, there are two aspects present in the majority of models: the election of calibration parameters (traffic measurements) as well as the error measurements. A widely used error measure that can provide a fairly good initial estimate of the degree of fit between the simulated and the actual traffic measurements is the Root Mean Square Error (RMSE) [HMK03]. This error measure, which gives an estimate of the total error may be found in equation 2.8.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \tag{2.8}$$

$x_i$ = simulated traffic measurement value at time $i$

$y_i$ = actual traffic measurement value at time $i$

$n$ = the number of traffic measurement observations

Another popular measure among traffic simulations calibration is the correlation coefficient ($r$), computing the strength of the linear association between the simulated and the actual traffic measurements. Equation 2.9 gives a definition of this measure.

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(x_i - \hat{x})(y_i - \hat{y})}{\sigma_x \sigma_y} \tag{2.9}$$

$\hat{x}$ = mean of simulated traffic measurement values

$\hat{y}$ = mean of actual traffic measurement values

$\sigma_x$ = standard deviation of simulated traffic measurement values

$\sigma_y$ = standard deviation of actual traffic measurement values

$n$ = number of traffic measurement observations

Despite being effective measures, RMSE and $r$ do not provide information about the nature of the error. During an economic forecasting [The61], Theil developed a more accurate measure, that can be divided into three other measures able to determine the error origin. This measure was named Theil's Inequality Coefficient and can be found on equation 2.10.

$$U = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} \tag{2.10}$$

The numerator of equation 2.10 may be decomposed into three different components as it is shown on equation 2.11

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 = (\hat{y} - \hat{x})^2 + (\sigma_x - \sigma_y)^2 + 2(1-r)\sigma_y\sigma_x \tag{2.11}$$

The first component is $U_m$, the bias proportion 2.12, which is a measure of systematic error that can be used to determine consistent over-counting or undercounting caused by an excess/loss of vehicles.

$$U_m = \frac{n(\hat{y} - \hat{x})^2}{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{2.12}$$

$U_s$ is the variance proportion 2.13, which can be used to measure the simulated measurements' ability to replicate the degree of variability (fluctuations) in the actual measurements.

$$U_s = \frac{n(\sigma_y - \sigma x)^2}{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{2.13}$$

Lastly, $U_c$ is the covariance proportion 2.14, which is a measure of unsystematic error.

$$U_c = \frac{2(1-r)n\sigma y\sigma x}{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{2.14}$$

## 2.3 Route Assignment Algorithms

Route assignment focuses on the drivers' choice from among several route alternatives (alternative paths) between a given origin and destination pair on the transportation network. This assignment is the fourth and final step in the conventional transportation forecasting model. This model was presented in the 1954 by Fagin [Fag54], being the first model to establish a link of travel and activities (or land use) and also offering a modern framework to deal with travel behaviour. Expanded during the 1970s with Manheim [Man79] and 1980s with Sosslau [SHCW78], the first comprehensive application of four-step model just appeared in 1997, in a study of Chicago transportation area, conducted by Weiner [Wei97].

As the name suggests, the referenced model has four steps: trip generation, trip distribution, mode choice and, finally, route assignment. The first step consists on defining the frequency of origins or destinations of trips in each zone by trip purpose, considering the household demographics but also other socioeconomic factors. Then, it is necessary to match origins with destination, which is computed at the second step, trip distribution. Mode choice is the third step, matching each trip with a transportation mode (car, bus, metro or other). Finally, it necessary to allocate a specific trip, by a given transportation mode into a particular route, giving the basic formulation of route assignment problem.

### 2.3.1 All or Nothing Assignment

Since the early studies on transportation networks, route assignment techniques were required. It was necessary to estimate how many vehicles travel on each route. Lacking tools to study complex local streets traffic, the first approach was to divert traffic to freeways, having more freedom to manage vehicle's flow. A shortest-path algorithm is computed, assigning all the traffic to shortest-path, originating all or nothing assignment, either all the traffic moves from A to B or it does not.

All or nothing assignment can be defined as the simplest route choice and assignment method. This approach considers that there are no congestion effects, route attributes and weights are the same for all drivers, moving between two points as a whole. Once link costs are fixed, every vehicle going from $i$ to $j$ must choose the same route. So, all drivers are assigned to a single route from $i$ to $j$, ignoring all other possible, less attracting routes. This assumptions do not produce a faithful representation of all real scenarios, but can work as an approximate scenario for sparse and uncongested networks with few route possibilities and very distinctive costs. This method is also useful to predict the desired path with the absence of congestion, oftenly working as basis for other, more complex, models.

### 2.3.2 Incremental Assignment

To address the restriction of diversion based methods (all or nothing assignment), heuristic procedures were developed. The main goal of this new approach is to consider traffic loading effect as well as travel times. This incremental approach divides traffic volumes into several steps. Each step assigns a fraction of total demand, based on all or nothing assignment. This method follows the given steps:

1. Assign first part of traffic
2. Compute new travel times
3. Assign the next part of the traffic
4. Repeat steps 2 and 3 until all traffic is assigned

Step 2 uses a volume delay function to recalculate link-volume-based travel times. In 1968, Dafermos [Daf68], was the first to apply the Frank-Wolfe algorithm, [FW56], to deal with the traffic equilibrium problem, leading to the development of a link performance function by the Bureau of Public Roads (BPR). The basic structure of this volume delay function can be found in 2.15.

$$S_a(v_a) = t_a\left(1+0.15\left(\frac{v_a}{c_a}\right)^4\right) \tag{2.15}$$

$c_a$ = Capacity of link a per unit of time
$v_a$ = Volume of traffic on link per unit of time
$t_a$ = Free flow travel time on link a per unit of time
$S_a(v_a)$ = Average travel time for a vehicle on link $a$

If a big amount of increments is used, traffic flows may present an equilibrium assignment output. However, this method does not ensure an equilibrium solution, it is just a possibility. So this kind of approach must not be chosen for equilibrium purposes. Incremental assignment may lead to inconsistencies between link volumes and travel times, once the method is influenced by the order in which OD volume pairs are assigned. This can raise errors in evaluation measures an additional bias in results.

### 2.3.3 Equilibrium Assignment

In 1924, Frank Knight was the first to propose the idea of using network equilibrium models to predict traffic flows in congested transportation networks [Kni24]. The proposal was very similar to Nash equilibrium [Nas50] for game theory. However, traffic assignment would offer a more complex situation, due to the several players involved. So, in 1952, Wardrop proposed two different notions of equilibrium, Wardrop Equilibrium Conditions [WAR52]:

1. The journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route

2. At equilibrium, the average journey time is minimum.

The first principle states that each individual vehicle attempted to minimize his transportation costs. When this principal if fulfilled, User Equilibrium (UE) is achieve, since each user has chosen the best route. A more specific equilibrium situation is when no single user can lower his cost with a unilateral action. Stochastic User Equilibrium (SUE) is the name of this stationary state.

The second principal requires every users cooperation to ensure the most efficient use of the global system. To perform this, a central mechanism is required, so that the vehicles receive its route information. Another possibility is to achieve the equilibrium through marginal cost road pricing. This principal is commonly designated as System Optimal (SO).

#### 2.3.3.1 User Equilibrium

To compute User Equilibrium (UE), there are some assumptions that should be considered:

1. The user has perfect knowledge of the path cost.

2. Travel time on a given link is a function of the flow on that link only.

3. Travel time functions are positive and increasing.

Those assumptions will validate the equilibrium equations. BPR curve (Equation 2.15) is modelled in a nonlinear programming problem. The equilibrium may be found by solving Equation 2.16 and considering the restrictions of 2.17, 2.18 and 2.19.

$$\min \sum_{a} \int_{0}^{v_a} S_a(x)dx \tag{2.16}$$

Considering that,

$$v_a = \sum_{i}\sum_{j}\sum_{r} \alpha_{ij}^{ar} x_{ij}^{r} \tag{2.17}$$

$$\sum_{r} x_{ij}^{r} = T_{ij} \tag{2.18}$$

$$v_a \geq 0,\ x_{ij}^{r} \geq 0 \tag{2.19}$$

$S_a(x)$ is the result of Frank-Wolfe algorithm, representing the average travel time for a vehicle on link $a$, while $v_a$ assumes the same representation, the volume of traffic on link a per unit of time. On the other hand, $x_{ij}^{r}$ is the number of vehicles on path $r$ from origin $i$ to destination $j$. Both $i$ and $j$ are positive integers: $1...n$. The coefficient $\alpha$ is oftenly set to 0.15 as the case of Equation 2.15. If $a$ is a link on path $r$ from $i$ to $j$, $\alpha_{ij}^{ar} = 1$, otherwise it assumes zero value.

The problem formulated in Equation 2.16 can be solved by following several step, until the equilibrium is found. Those steps are the following:

1. Find shortest routes
2. Assign trips to shortest routes
3. Update link travel times using capacity restraint function
4. Compute now shortest routes using new travel times
5. Assign trips to shortest route
6. Average trips for last two assignments
7. Update link travel times using average trips assigned and capacity restraint function
8. Go back to step 1
9. Continue until volumes and travel times on links are in equilibrium

#### 2.3.3.2 Stochastic User Equilibrium

On the other hand, Stochastic User Equilibrium (SUE) has a different approach. Once UE do not fully predict behavioural dynamics, a more accurate method was proposed, SUE. Stochastic User Equilibrium performs a more correctly approximation of decision making process to the driver making a route choice [GP10]. If UE methods have a good performance on macroscopic assignment, SUE goes even further, having good solutions for microscopic simulations. SUE models instead of using random utility discrete choice theory, which assumes that the utility users derive from a given route has a stochastic part, which cannot be explained by an observer, but follows a known distribution. [WB13].

$$U_{ir} = V(-c_r) + \varepsilon_{ir} \tag{2.20}$$

This utility function can be found on Equation 2.20, giving the utility ($U_{ir}$) of a user $i$, deriving from an alternative $r$. The deterministic part is represented by the value function $V$, while the $\varepsilon_{ir}$ is the stochastic term. For simplicity reasons, the value is represented by a generic cost $c_r$. Stochastic part ($\varepsilon_{ir}$) is what differs between models.

There are two main algorithmic approaches to SUE: Probit and Logit model. Both models are regurlarly used in equilibrium studies. Probit are able to deal with overlapping routes, but also routes with very different lengths. Uchida [USWC07], Connors [CSW07] and Meng [MLW12] research's use this model to compute route assignment. On the other side, Logit models assume the same level of stochasticity for all routes costs, being unable to deal with overlaped routes. This model was used by Meng [MLCY04] and Yang [Yan99] [YMB01]. However, Logit does not require as many computational resources as numerical integration and samples techniques used on

Probit models. Multinomial Probit is one of the most flexible models, appearing as the reference of Probit models. Logit has some variations widely used: Generalized Nested Logit (GNL), Nested Logit with Alternative-Specific Constants (NL-ASC) and Mixed Logit (ML). The user equilibrium has also been studied from a microcopic perspective resorting to the metaphor of agents, with earlier practical examples being reported in the literature elsewhere [RBB+02, RLCB02].

## 2.4 Route Assignment in SUMO

SUMO has four different route generation tools. Those tools generate traffic for a given map, and each of them as a specific purpose, according to the given input. Those inputs can be measures of detectors (DFROUTER), junction turning ratios (JTRROUTER), demand definitions (DUAROUTER) or even use demand between zones to compute a macroscopic simulation (MAROUTER).

### 2.4.1 Overview

DFROUTER creates vehicles' routes based on the measures of the detectors located on network streets. It uses induction values to compute vehicle routes that will further be used on the simulation. Induction loop data comes is collected from *on-* and *off-ramps* covered with detectors, inductive loop or other cross-section detectors. This approach is quite successful when applied to highway scenarios, but fails to inner-city simulations, where inductive-loop traffic detectors coverage is low. Another limitation of this approach is that only considers two types of vehicles: PKW and LKW (abbreviation for passenger cars and trucks in German). In order to run this algorithm, the user should gather the following inputs: road network, induction loop definition and induction loop measures. After processing this information, it will be possible to look at the outputted routes, as well as the file emitters (generated vehicle associated to each route).

JTRROUTER approach generates vehicle's routes based on demand for flows and turning ratios at street intersections. It uses definitions of turn percentages at intersection for computing routes through the network. Such an approach can be used to set up the demand within a part of a city's road network consisting of few nodes. As input, the description of the turning ratios has to be provided in addition to the road network and the demand for traffic as vehicles' flows. The output consists in a route file, describing each vehicle route. Additionally, it is possible to segment the simulation in several time intervals, and for each one, the user can configure different turn probabilities to intersection streets.

MAROUTER is a macroscopic approach that allows to compute an iterative assignment without the need for time-consuming microscopic simulation. This tool converts the OD (Origin-Destination) matrices information into vehicle flows as macroscopic assignment. Route assignment is obtained thought several inputs. The user should specify the road network, the OD matrices and the districts definition. As output this algorithm produces a file with vehicle flows between different TAZ (Traffic Assignment Zones). This tools allows the user to choose between different assignment methods. Route assignment can be achieved incrementally, using User Equilibrium (UE) or through Stochastic User Equilibrium (SUE).

DUAROUTER allows importing different demand definitions and retrieves vehicular routes, using the shortest path algorithm. Another goal of this tool is to repair connectivity problems in existing route files. Its name is due to the fact of performing Dynamic User Assignment (DUA) if DUAROUTER is iteratively called. This is smoothed by using *duaIterate.py* tool, which converges to Dynamic User Equilibrium (DUE).

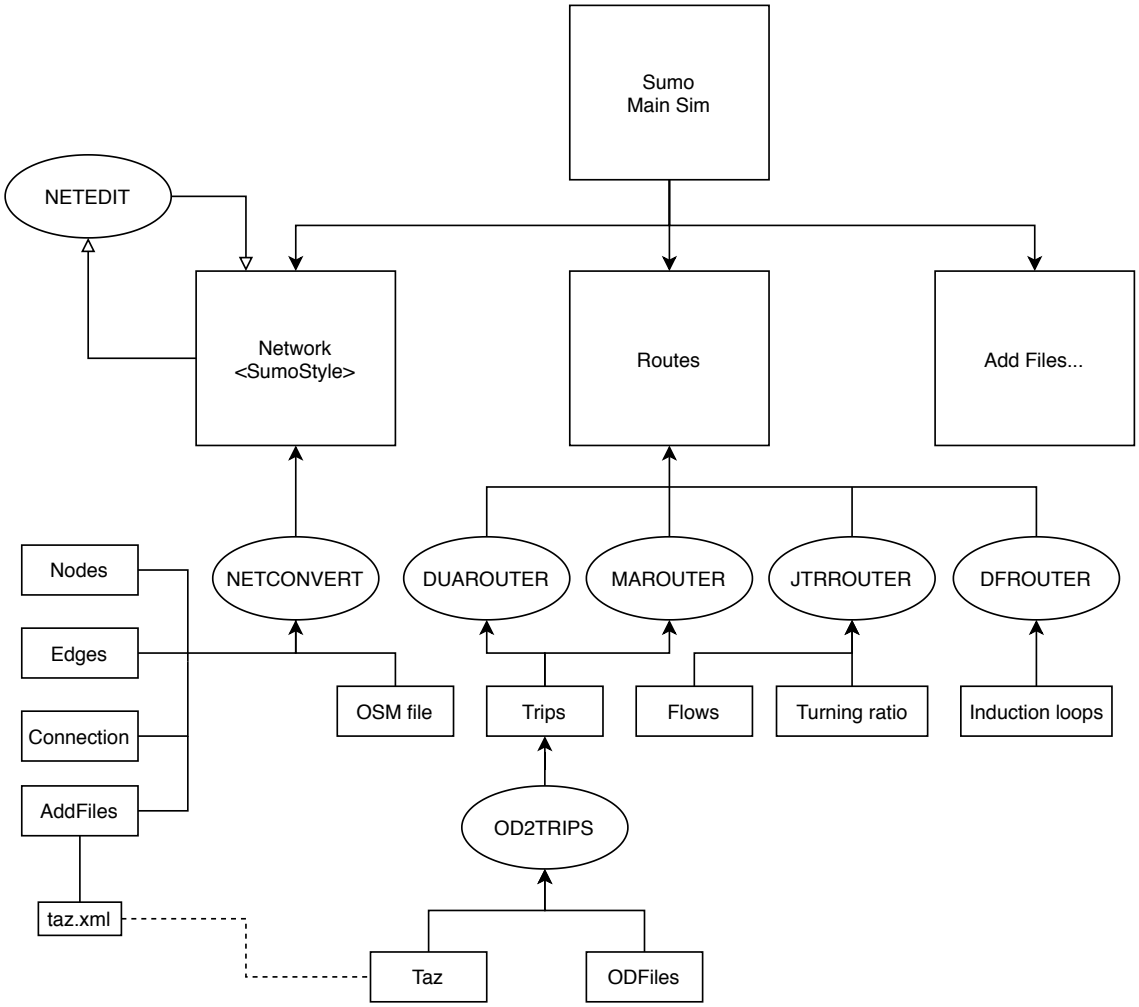



Figure 2.1: SUMO's trips generation tools diagram

### 2.4.2 Dynamic User Assignment

When an origin-destination relation (trips) is established, SUMO decides which route each vehicle is going to perform, assigning several edges from origin to destination. The easiest way to find those routes is by performing an algorithm that computes the fastest/shortest path, for example Dijkstra or A*. When those algorithms are used in this kind of simulation, they use previous knowledge of network distances and travel times, discarding the number of vehicles in the network. A naive approach, considering that each vehicle is alone in the network, puts all vehicles on the fastest path and leads to heavy traffic congestion and bottlenecks due to the large amount of traffic. This problem can be solved by following an User Assignment approach. SUMO already has some tools that allow to perform this kind of assignment in its simulation.

### 2.4.3 Standard Assignment

DUAROUTER performs the most basic form of User Assignment by calculating the shortest path. This tool allows the choice between several routing algorithms. By default, Dijkstra is the selected

one, but there are other: A*, CH (Contraction Hierarchies), CH Wrapper. They all work for the same goal - finding the shortest path, but they might present different performances, according to the given network.

In its "plain" form *duarouter* does nothing but calculate the shortest path using Dijkstra's algorithm and the given edge weights (via the *–weights* option). If no weight is given it uses the minimum travel time (edge length divided by edge max speed). Its full power is unleashed when the weights are adapted iteratively (using *duaIterate.py*) to calculate a user equilibrium.

#### 2.4.3.1 Usage Description

There are several running options for *duarouter*, however, only three parameters are required. It is necessary to define the simulation network, demand file (trips) and the file where computed routes will be written. An example of basic usage can be found in Listing 2.1

```
/* Run command for Linux */
.duarouter --net-file net.net.xml  --route-files trips.xml  -o routes.xml
```

Listing 2.1: Running command for *duarouter*

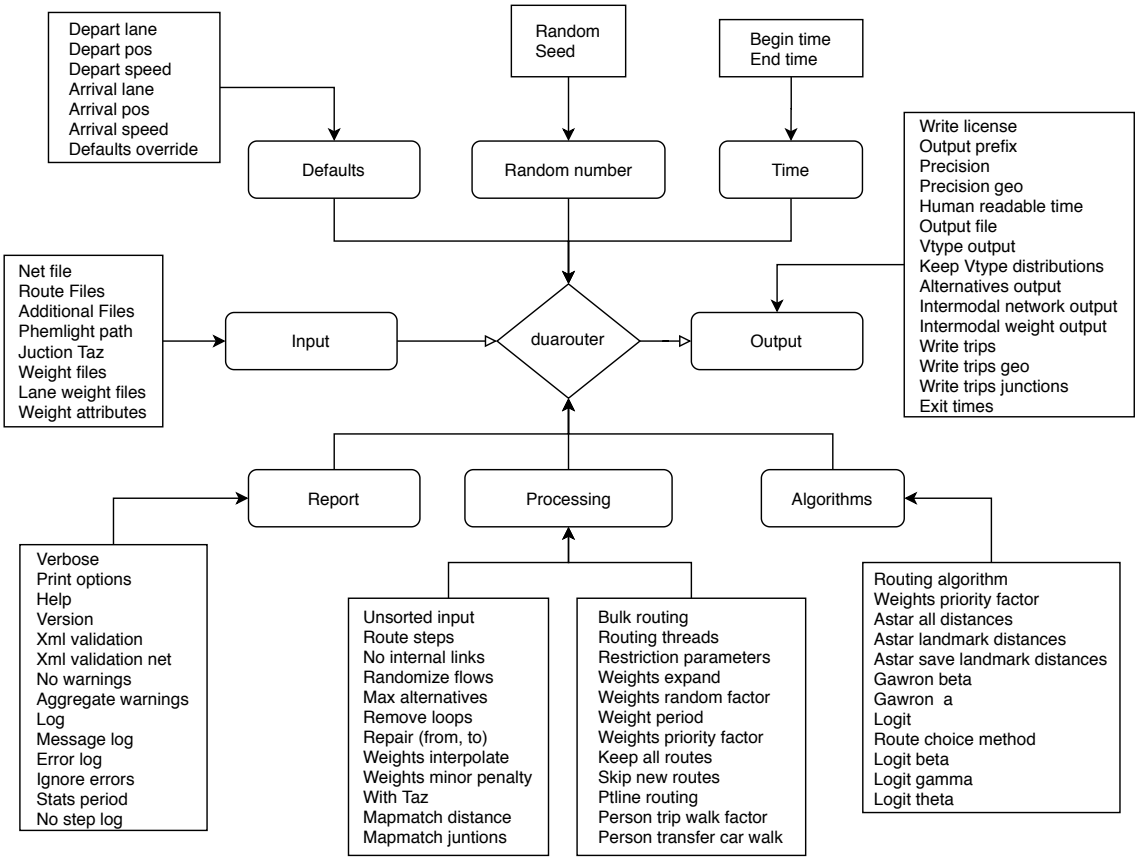



Figure 2.2: Running options diagram for *duarouter*

This algorithm input consists of the road network as well as vehicular demand. The network file can be generated by NETCONVERT (OpenStreetMap, VISUM, and Shapefiles are the most popular formats) or via NETGENERATE, where the user draws his network. Vehicular demand as a more flexible format, the choice is between predefined routes, alternative routes, vehicle flow, or individual trips. Here is defined the types of vehicles to be simulated as well as each vehicle flow according to its type and for each interval. The output majorly consists of a route file, where each vehicle has the shortest path for the given trip, but more output options can be configured (Table A.5). As it is shown in Figure 2.2 there are many other options to configure *duarouter* algorithm. A detailed description can be found in Appendix A, from Table A.1 to A.9.

### 2.4.4 Iterative Assignment

To perform Dynamic User Assignment, SUMO has released an iterative script, that (approximately) achieves Dynamic User Equilibrium - *duaIterate.py*. This algorithm goal is to find the best route for each vehicle, reducing its travel cost, that is, the travel time. It uses an iterative approach, using different routes until User Equilibrium is achieved. Each iteration has two major steps:

1. Calling DUAROUTER - to perform the re-routing
2. Calling SUMO - to simulate travel times

The first step is responsible to assign a route to each vehicle, according to the last known edges. *Iteration_0* starts with empty travel times. By considering weighted edges, obtained by previous iterations simulated times, DUAROUTER will reach its full potential, performing a Dynamic User Assignment. Secondly, it is necessary to call a simulation configuration file, in order to perform a simulated environment, with new routes. In this step, new travel times will be calculated, resulting in new edge costs that will be used for the next iteration. The algorithm may run with a fixed number of iterations, defined as a parameter, but they also can be dynamically determined, depending on the convergence criteria. To do so, the user must choose the option *--max-convergence-deviation*. This convergence is ensured by different methods, calculating route-choice probability from route cost, so that, the cheapest route is always chosen. In each iteration, an auxiliary file, *.rou.alt.xml* records the current best route, but also all routes previously computed, so that they can be used when deciding next iteration best route.

#### 2.4.4.1 Route-choice algorithms

There are two methods that can be used when it comes to route choice: Gawron and Logit. Both algorithms have the weight / cost function ($w$) as an input. This parameter represents network edge costs and comes from the simulation in each iteration (except for iteration 0, where this cost has not been calculated yet). A set of routes $R$ is also required, where each route $r$ has an old cost $c_r$ and an old probability $p_r$ (from last iteration). These algorithms calculate a new cost $c_r^{'}$, as well as a new probability $p_r^{'}$

Gawron is the default method used by *duaIterate.py* to determine Dynamic User Equilibrium. Each driver's route choice is modeled by a discrete probability distribution which is used to select a route in the simulation. To do this, the algorithm takes in consideration the following parameters:

- the travel time along the used route in the previous simulation step
- the sum of edge travel times for a set of alternative routes
- the previous probability of choosing a route

Logit method has a fixed formula on probability calculation for each route. It ignores old costs and old probabilities and takes the route cost directly as the sum of the edge costs from the last simulation.

$$c_r^{'} = \sum_{e \in r} w(e) \tag{2.21}$$

On the other hand, probabilities are calculated from an exponential function with parameter $\theta$ scaled by the sum over all route values:

$$p_r^{'} = \frac{\exp(\theta c_r^{'})}{\sum_{s \in R} \exp(c_s^{'})} \tag{2.22}$$

#### 2.4.4.2 Usage description

This algorithm (*duaIterate.py*) has several running options and combinations, but on its basic usage only two parameters are required. The user should define the road network and also the demand. This last parameter may have several formats: trips, routes or even flows. An example of a basic command to run the algorithm may be found on Listing 3.1.

```
/* Run command for Linux */
./duaIterate.py -n net.net.xml -t trips.xml
```

Listing 2.2: Running command for *duaIterate.py*

Besides network file, a demand file is the only mandatory parameter to run this algorithm successfully. But there are other files that can be included to the simulation, gathering additional information that may reduce inconsistencies from the real scenario. It is possible to include a file that has defined districts, that is, Traffic Assignment Zones (TAZ). It is also common to include file that specifies vehicle types. Further details on input configuration parameters for *duaIterate.py* may be found in Table A.11.

When it comes to the algorithm outputs there are several options that can be configured when the user is running it on the command line and can be found on Table A.10. It is possible to disable some outputs (warnings, summary and tripinfo), but also define some additional measures (eco-measure) or file formats.

Table A.12 presents different options to configure the simulation. The user can define when the simulation begins and ends, but also, how does it evolves. It possible to configure the number of steps, the weights associated to each edge, and also, the convergence of the assignment algorithm.

As referenced in Section 2.4.4.1, this iterative assignment can choose between two different algorithms, when it comes to computing route choice probabilities. Gawron's algorithms has both *alfa* and *beta* variables with a 0.5 of default value. But the user can change those parameters if he specifies different running options - Table A.13. The other algorithm is logit, but it is also possible to use its variant c-logit. To do so, beta value should be zero. For this *model*, *beta*, *gama* and theta value can be defined. It is also possible to define the routing algorithm (different from route choice probabilities), that computes the fastest path for an origin-destination pair. Dijkstra is used by default, but it is also possible to choose A*, CH or CH Wrapper.

There is also the possibility of achieving DUE through incremental assignment. SUMO provides a specific script for that (*one-shot.py*), that will be described in Section 2.4.5. But, apart from that script, it is also possible to implement an incremental assignment by selecting some options in *duaIterate.py*. Table A.14 presents those options. In order to have an incremental assignment, an incrementing scale should be defined. By default, incrementation is not considered, so, it correspond to a negative value. Then, there are other parameters to be defined, such as, incrementation start, maximum value or period.

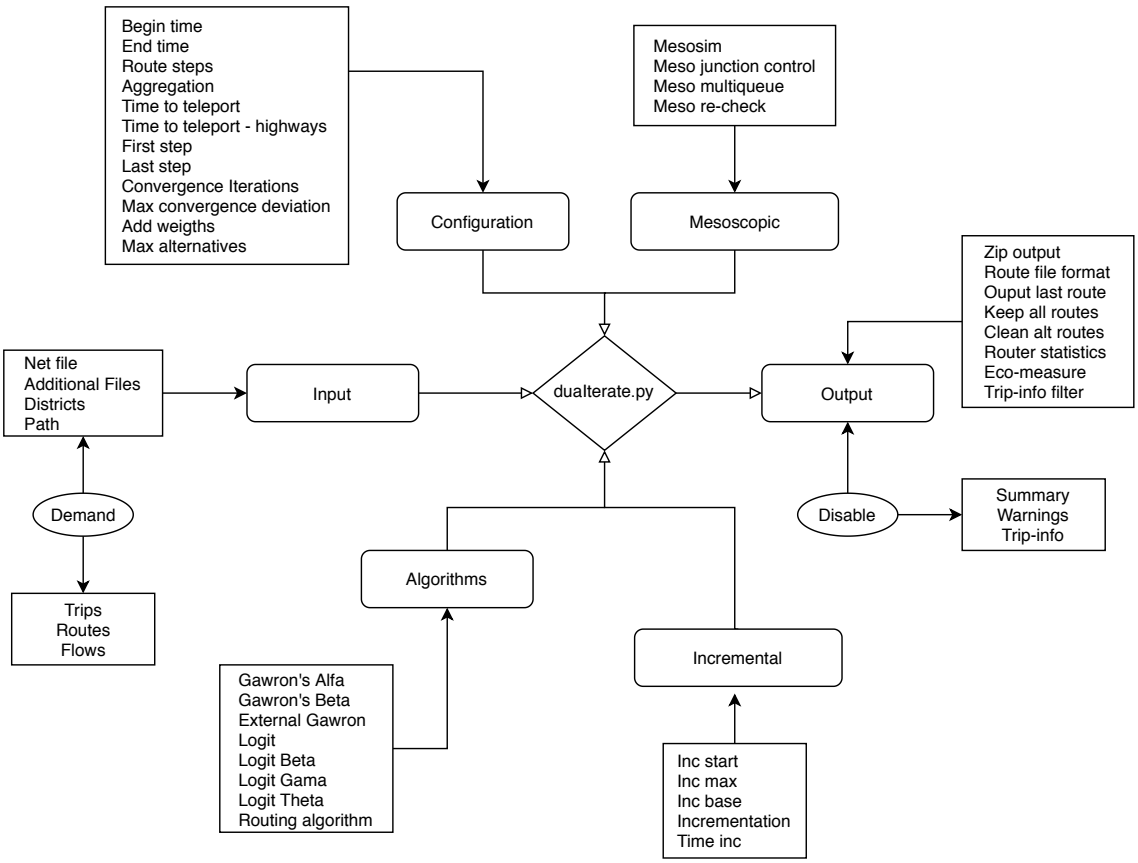



Figure 2.3: Running options diagram for duaIterate.py

Despite being focused on microscopic simulation, SUMO also provides a mesoscopic module for some algorithms, and *duaIterate.py* is one of them. The difference between this two types of simulation is not visible on isolated vehicles. Once there are queues of vehicles on the road, that difference becomes more relevant, due to the spacing between vehicles. In Table A.15 some configuration options for mesoscopic simulation can be found.

Table A.16 is focused on processing options for *duaIterate.py*. It is possible to define a cost modifier for the given routes; grohnde and isar are available values. Not simulate internal links and and smooth weights across iterations are other processing options provided by *duaIterate.py*.

### 2.4.5 Incremental Assignment

Another approach, implemented by SUMO, to perform Dynamic User Assignment is incremental assignment. This variant should be used if there is not enough time or computing power to wait for the Dynamic User Equilibrium, if net changes occur during the simulation or if vehicles need to adapt their route while running. With this approach, vehicles can periodically change their route, avoiding bottlenecks and traffic jams, once they take in account network current status.

The assignment method very similar to Standard (Section 2.4.3) and Iterative (Section 2.4.4) assignment. This model uses directly <*trips*> instead of routes of <*vehicles*>. Those trips will be assigned to their respective fastest routes, considering their departure time, but also a new parameter used by this algorithm - *travel-time updating interval*. This interval is determined by the user, for example, if the parameter value is 300, link travel times will be updated every 300 seconds. On the other hand, if a negative value is assigned (for example -1), travel times will not be updated, being used free-flow speed for every trips. This trip-based approach will compute the fastest path at the time of departure which prevents all vehicles from driving blindly into the same jam and works pretty well empirically (for larger scenarios). However contrary to *duaIterate.py*, Stochastic User Equilibrium will not be achieved when using this algorithm.

#### 2.4.5.1 Usage Description

Similarly to the algorithms presented in previous sections, *one-shot.py* has several running options and combinations. There are three mandatory parameters to run this algorithm successfully: travel-time updating interval, road network and trips. An example of a basic command to run the algorithm may be found on Listing 3.2.

```
1 /* Run command for Linux */
2 ./one-shot.py -f 900 -n net.xml -t trips.xml
```

Listing 2.3: Running command for *one-shot.py*

Table A.17 has a detailed description of those mandatory inputs referenced above, but also for some additional inputs. It is possible to load edges or lane weights from a file or even specify

the path to SUMO binaries. Similarly to other SUMO algorithms, it is possible to configure the simulation with additional files, usually containing information on vehicle types.

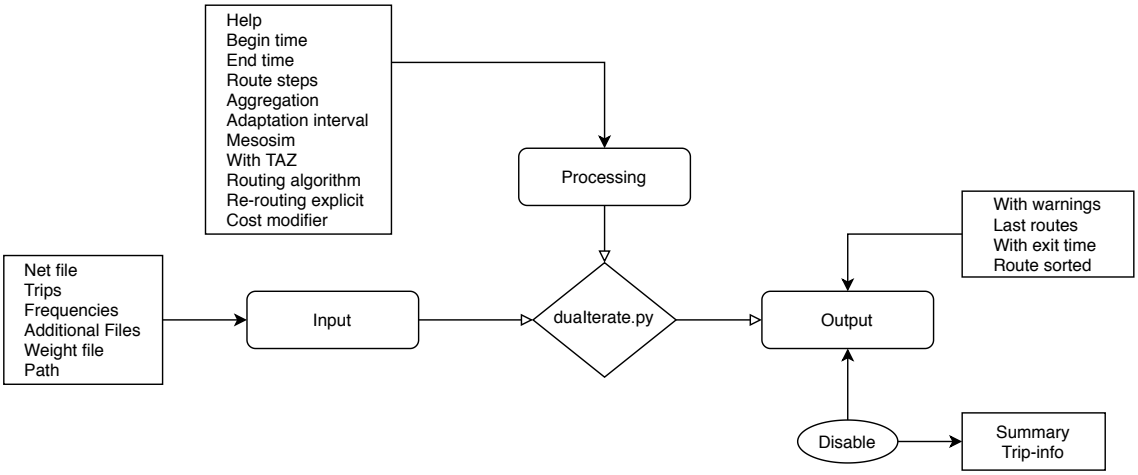



Figure 2.4: Running options diagram for one-shot.py

Thus, output information is summarized on Table A.18. It is possible to enable or disable output warnings, summaries and tripinfos. Additionally, it is possible to only save last computed routes, to sort the output by departure time and to write exit times for all edges.

Further options can be found on Table A.19. The user is able to define the routing begin and end, as well as the aggregation period and the route steps. It is also possible to define which routing algorithm should be used (*one-shot.py* uses A* as default) and weights adaption interval. Mesoscopic simulation is also available, despite not being recommended for this kind of approach. Other options are related to re-routing and modify link travel costs.

## 2.5 Summary

This Chapter provides a theoretical introduction to microscopic simulation calibration through route assignment. To begin with, traffic simulation models are classified according their detail level: Macroscopic, Microscopic and Mesoscopic. This analysis concludes that Microscopic are able to provide the most approximate traffic simulation values. Its mathematical expressions can precisely define vehicles' position and velocity, allowing a better insight to decision support systems. Calibration strategies for this kind of simulation models are also explored, presenting a methodology inspired by the most remarkable examples of simulation calibration. It is crucial to elect traffic measurements that will work as calibration parameters as well as what error measurements will be applied. Root Mean Square Error (RMSE) and Correlation coefficient are considered essential, whereas Theil's Inequality Coefficient may be used as an additional measure.

Furthermore, route assignments algorithms are explored, revising the most notable literature, but also its practical application to an open-source simulation software, SUMO. All or nothing assignment is the most rudimentary technique, once it only works with maximum edge capacity, it is unable to reproduce majority of real traffic scenarios. Despite this, it can be a good option

for sparse and uncongested networks and has inspired more complex models, such as Incremental assignment. An Incremental approach is useful to produce acceptable results with few time consumption. Once a volume delay function is used to recalculate link-volume based travel times, big amounts of increments may lead to an equilibrium scenario. Equilibrium assignment is the last presented technique, but is also the most effective one. An Iterative approach is used to achieve User Equilibrium (UE), where all vehicles are assigned to a route with equal or less cost than any unused one, this equilibrium state is achieved once each user has chosen the best route. A more specific equilibrium situation, Stochastic User Equilibrium (SUE), is when no single user can lower his cost with a unilateral action. Both equilibrium situations are very time and resource consuming, however, the results are very feasible and promising. Afterwards, SUMO's approach to route assignment is explored. There is a module prepared to deal with Dynamic User Assignment, providing three different algorithms: Standard, Iterative and Incremental. Standard assignment should be used as a comparative measure between other more advanced, since it basically computes the shortest path for dynamic demand. Iterative assignment is a variant of equilibrium assignment, pursuing SUE goal. On the other side, Incremental approach use the assignment techniques described above, using a volume delay function.

# Chapter 3

# Methodological Approach

Chapter 3 is focused on the methodological approach. In Section 3.1, it is possible to understand what real problem we are trying to solve (problem formalization). In Section 3.2, the methodological approach is presented, describing the numerous steps that are followed in order to improve the simulation performance and then, in Section 3.3, all the materials and case-study data are presented, ending with a summary (3.4) of this Chapter contents.

## 3.1 Problem Identification

C-ROADS is an European joint project [Pla], and its main goal consists in harmonising the deployment activities of Cooperative Intelligent Transport Systems (C-ITS) across Europe. There are several working groups, deploying C-ITS software across 43 European cities. Porto is one of those cities, involving several partners. Porto City Hall (CMP) is a central stakeholder of this project, resorting to the services and expertise of other partners in the project. Data is collected from STCP (Porto Collective Transports Society), IP (Portugal Infrastructures) and CMP. Then, this information is gathered by Armis under their ITS solution, NEXT [NEX], providing the city hall with a decision-making support system. NEXT is a traffic analysis, simulation and prediction solution that makes use of a simulation model of the real network upon which it operates. It follows the concept of Artificial Transportation Systems (ATS) [RL14] and its implementation is based on the MAS-Ter Lab agent-based reference architecture [ROB07], recurring to prediction algorithms and simulation models, such as those described in [BAR15, SRM+16, AAA+17, AFR19]. This project also involves other partners such as TIS.pt (Consultores em Transportes Inovação e Sistemas, S.A.) who has produced the origin-destination matrices of the city network, and FEUP, contributing with RD on network modelling and assessment.

Considering that Armis has been developing a microscopic traffic simulation model for Porto urban area, the simulation module of NEXT is being developed with an open-source traffic simulator, SUMO. Using TIS's Origin-Destination (OD) Matrices, demand is being generated, allowing

simulated vehicles to commute from one zone to another. However, those vehicles are not behaving as it would be expected on a real scenario. Despite travelling from one origin into the specific destination, almost all of the commuters in the expected trips choose the same path, leaving adjacent streets with no traffic. This causes excessive over-saturation on certain paths of the network yielding deadlocks and compromising the analysis of the network performance. So, it is imperative to understand the reason why such behaviour emerges and to suggest possible approaches to fix it.

## 3.2 Proposed Method

In order to improve the simulation behaviour, several assignment algorithms were applied. Configuring their parameters and adapting their restrictions to the simulation over study was crucial for this research. On a first analysis it was necessary to understand what route assignment algorithms were available on SUMO and how they could be used to improve the simulation of Porto urban area. Only after this approach using SUMO's material, other route assignment algorithms should be implemented on the given network. SUMO's assignment tools module has four route assignment algorithms, but after a more detailed research we concluded that *Assignment.py* script is unavailable, leaving us three algorithms to test:

- **Dua Router** - Standard assignment
- **Dua Iterate** - Iterative assignment
- **One Shot** - Incremental assignment

Each of these algorithms has a different approach, presenting different strategies and configuration parameters. At a first stage, experimental networks were used to test each algorithm performance. It was necessary to understand how the same network would give feedback on different route assignment approaches.

### 3.2.1 SUMO Microscopic Simulation Structure

For testing purposes, a simple network was chosen, so that we could better understand SUMO microscopic simulation structure as well as the required arguments for route assignment scripts. SUMO tutorials module had a network that fitted perfectly on these requirements. Given that, **quickstart**[1] was the elected network to begin with some experiments.

This is a very simple simulation network, having only four origins, four destinations and two unsignalized intersections. The network layout can be found at Figure 3.2. Each road has three lanes for outbound traffic, restricting some movements on the traffic lane, for example, u-turns are not allowed at intersections. When it comes to priority, an higher priority is given to eastbound and westbound traffic.

[1] https://sumo.dlr.de/docs/Tutorials/quick_start.html

This network also has a very simple file structure. NETCONVERT is used to summarize network properties (edges, nodes and connections) into a uniform network respecting SUMO's style - ***quickstart.net.xml***. Routes are given by ***quickstart.rou.xml*** file, whereas ***quickstart.sumocfg*** controls the simulation files.

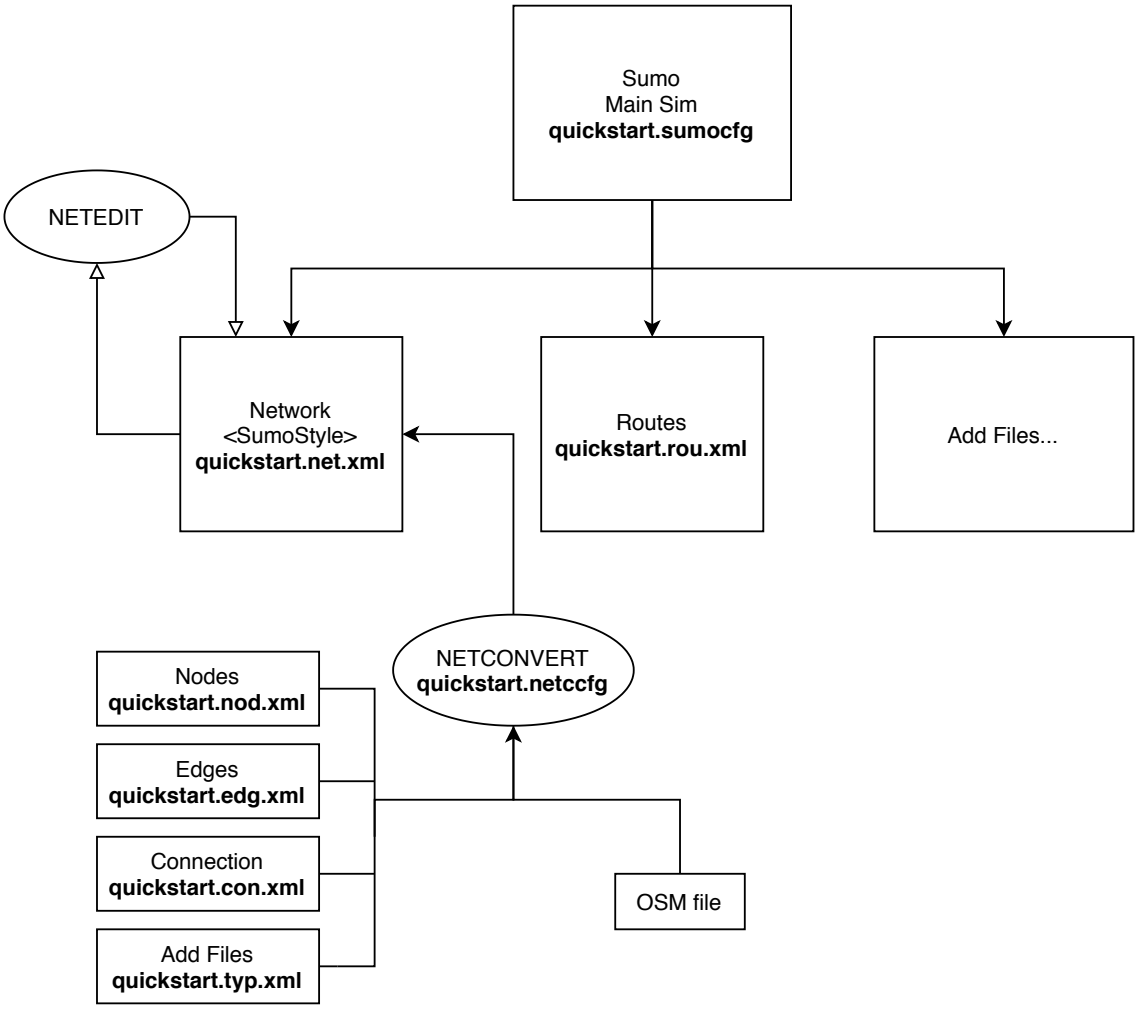



Figure 3.1: Quickstart simulation network diagram

Considering this sample network, efforts were made to run and understand the behaviour of Iterative Assignment referenced on Section 2.4.4. This script requires both network and trips files. Once quickstart refers to vehicles just as routes, it was necessary to use another SUMO script, *route2trips.py*, converting those routes into trips. The command used to run *duaIterate.py* for this network can be found on Listing 3.1.

```
/* Run command for Linux */
./duaIterate.py -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/quick
start/quickstart.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/armis/S
umo/quickstart/quickstart.trips.xml
```

Listing 3.1: Running *duaIterate.py* script for quickstart simulation

This iterative algorithm, *duaIterate* produces a microsimulation for each iteration. Configuration file is given by ***iteration_007.sumocfg***, compiling different route (***quickstart_007.rou.xml***) and additional files (***dua_dump_007.add.xml***) for each iteration, but keeping the same network file (***quickstart.net.xml***). This configuration file results on iteration summary (***summary_007.xml***) and trip information (***tripinfo_007.xml***). Route files are produces by computing weight files (***dua_dump_006.add.xml***) of former iteration using a specific configuration file (***iteration_007_quickstart_006.duarcfg***)

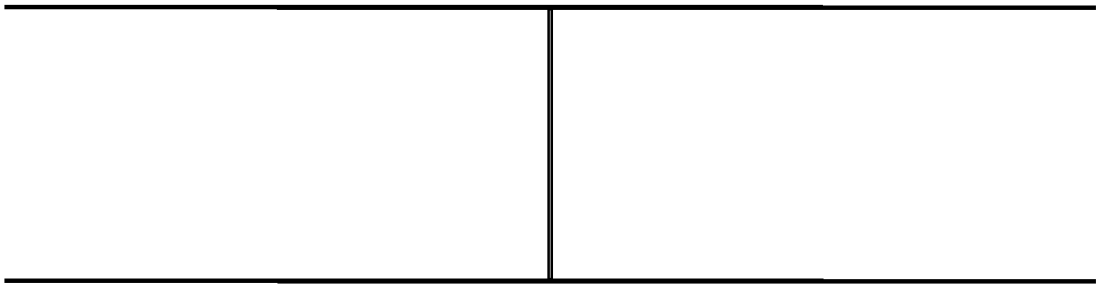


Figure 3.2: Quickstart network - visual representation

This network was also used to test another algorithm, *one-shot.py*, the incremental approach of Section 2.4.5. To perform this it a converted trip file was, similarly to *duaIterate.py*, with the same network file, ***quickstart.net.xml***. Listing 3.2 describes the usage of the algorithm. It was also necessary to define the frequency of the updating period (900). Additionally, only the last routes are saved on output file (**-L**) and routes are sorted according to their departure time (**-s**), for more output information, Table A.18 should be consulted.

```
/* Run command for Linux */
./one-shot.py -f 900 -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/
quickstart/quickstart.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/ar
mis/Sumo/quickstart/quickstart.trips.xml -L -s
```

Listing 3.2: Running *one-shot.py* script for quickstart simulation

This incremental test for 3.2 has produced several outputs. A configuration file, ***one_shot_900.sumocfg***, is created to manage the algorithm performance. Additionally, some files are generated, giving information about algorithms' steps and general behaviour: ***dump_900.add.xml***, ***dump_900_900.xml***, ***one_shot-log.txt*** and ***tripinfo_900.xml***. The most important output are the generated routes, where the re-routing results can be found and evaluated. With this configuration, only one route file is created, ***vehroutes_900.xml***.

### 3.2.2 Running Route Assignment Models in SUMO

Although the acquired knowledge with *quickstart* experiments, that was not enough to perform route assignment for a complex real network. The experiments conducted on Section 3.2.1 produced good results on understanding the required file formats for input as well as the generated output. But those experiments revealed some restrictions. It was not possible to test algorithms with dynamic demand (OD matrices) and the assignment results were very similar to the initial routes/trips. So, we decided to build a more complex artificial network using dynamic demand.

This artificial network was created from scratch on NETEDIT, SUMO's visual network editor. There are 20 edges in total, distributed by 4 traffic assignment zones (TAZ). This network definition would allow us to make experiments using OD matrices, simulating dynamic demand between different TAZ instead of specific pairs of nodes (origin-destination). A visual representation of this network may be found in Figure 3.3.

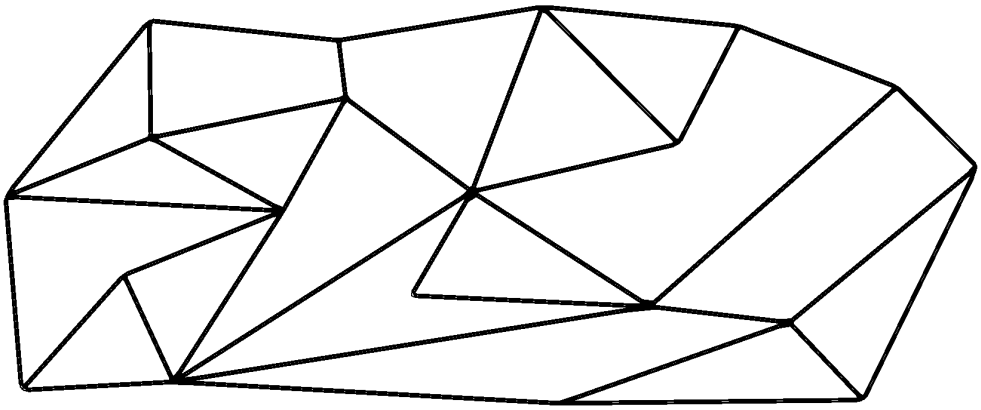


Figure 3.3: 10x10 network - visual representation

When it comes to network architecture (Figure 3.4), this simulation has some different aspects when comparing to *quickstart* simulation. The major difference is a direct consequence of dynamic demand. Once, OD matrices are used to estimate demand, ***matrix.0.0.1.0.1.xml*** file is the initial file, from where ***routes.xml*** will be obtained. This routing process is illustrated on Listing 3.3. Additionally, this network was automatically converted into a net file, ***net.net.xml***, avoiding intermediate files (nodes, edges and connections). The configuration file is given by ***speed.sumo.cfg***.

In order to run *duarouter* a trip file must be defined. Dynamic demand should be converted through **OD2TRIPS** tool. As the name refers, it converts OD matrices into a SUMO trips file. For this procedure, TAZ and OD matrices files should be given as input. The file where the trips will be written must also be specified. Then, this trip file might be used as input for *duarouter*. Besides trips, it is also required to include a specification of the network (net file) and of its TAZ. Generated routes will be written on the specified output file. Running instructions for *od2trips* and *duarouter* may be found on Listing 3.3.

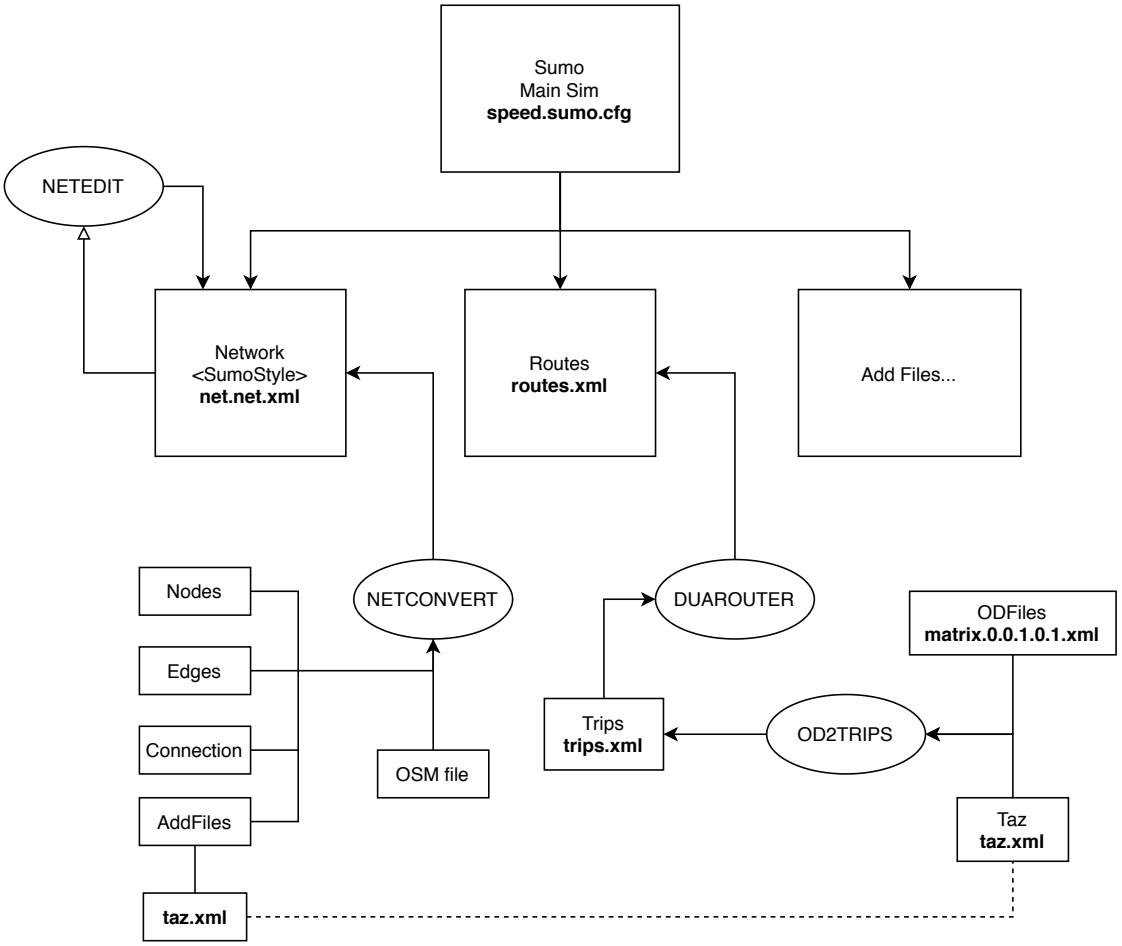



Figure 3.4: 10x10 simulation network diagram

```
/* Run command for Linux */
od2trips -n taz.xml -d matrix.0.0.1.0.1.xml -o trips.xml
duarouter --net-file net.net.xml --additional-files taz.xml --route-files trips.xml
-o routes.xml
```

Listing 3.3: Running *duarouter* for 10x10 simulation

Similarly to *quickstart* experiment, *duaIterate.py* script was ran on this 10x10 network. The inputs are the same, network and trips files are required. When it comes to output, the structure is similar, each iteration producing the following files: ***dua_dump_002.add.xml***, ***iteration_002.sumo.log***, ***iteration_002.sumocfg***, ***iteration_002_routes_001.duarcfg***, ***routes_002.rou.xml***, ***summary_002.xml***, ***tripinfo_002.xml*** (for iteration 2).

```
/* Run command for Linux */
./duaIterate.py -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/10x10
/net.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/10x10/tr
ips.xml
```

Listing 3.4: Running *duaIterate.py* script for 10x10 simulation

Contrary to *quickstart* simulation, *duaIterate.py* generated other routes, allowing the vehicles with the same origin and destination to choose between paths. So, it was necessary to repeat the experience for *one-shot.py*. The same frequency and output options were used, but this time with our 10x10 network and generated trips. The full running command can be found on Listing 3.5.

```
/* Run command for Linux */
./one-shot.py -f 900 -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/
10x10/net.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/10x
10/trips.xml -L -s
```

Listing 3.5: Running *one-shot.py* script for 10x10 simulation

### 3.2.3 The Case-study Network

For the network under study, the same method of Section 3.2.2 should be followed. At a first stage, Standard assignment should be simulated, using OD matrices as dynamic demand (Listing 3.6). Then the same proceeding should be applied using Iterative (Listing 3.7) and Incremental (Listing 3.8) algorithms.

```
/* Run command for Linux */
od2trips -n taz.xml -d matrix.8.0.9.0.2.xml -o trips.xml
duarouter --net-file net.net.xml --additional-files taz.xml --route-files trips.xml
-o routes.xml
```

Listing 3.6: Running *duarouter* for Porto simulation

```
/* Run command for Linux */
./duaIterate.py -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/Porto
/net.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/Porto/ro
utes.xml
```

Listing 3.7: Running *duaIterate.py* script for Porto simulation

```
/* Run command for Linux */
./one-shot.py -f 900 -n ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/
```

```
Porto/net.net.xml -t ../../../../../home/daniel/Documentos/tese-feup/armis/Sumo/Por
to/trips.xml -L -s
```

Listing 3.8: Running *one-shot.py* script for Porto simulation

This approach must be repeated for each different scenario. Scenarios may differ on induced demand (OD Matrices) but also on network file, whether some corrections are introduced on it. Thus, different scenarios must be compared, discussing the best approach to bridge the gap between real and simulated values.

## 3.3 Porto's network test bed

Our case-study is focus on a simulation of Porto urban area, characterized by its modeling under dynamic demand. Similarly to 10x10 network example, **OD2TRIPS** tool processes OD matrices, according network TAZ, resulting in trip files which are converted into routes through **DUAROUTER**. This simulation has an additional file, ***sumoSaveData.xml***, with a specific configuration for both edge and lane output information. A complete diagram with this network architecture can be found in Figure 3.5.

It is also possible to characterize the network through a group of Shapefiles, geospatial vectors with information associated. Each shapefile consists one three diferent files, so that, the geospatial information is complete. The shape format ***.shp*** is main file, containing the feature geometry itself, shape index format ***.shx*** is a positional index of the feature geometry (allowing forward and backward quick seek), and attribute format ***.dbf*** with colunar attributes for each shape. Porto urban area is divided into 119 different zones (Figure 3.6), each zone colored in red, delimited by a black line, marking the borders between adjacent zones. This characterization is represented on ***Zones_Porto*** layer with its respective Shapefiles - ***Zones_Porto.shp***, ***Zones_Porto.shx*** and ***Zones_Porto.dbf***

Besides this, it is necessary include an accurate representation of the urban network, representing the group of roads and freeways inside the city. This step is facilitated by ***arm_ways_porto*** layer. In Figure 3.6 this layer is represented by continuous water-blue lines. Apart from this geographical representation it is also necessary to include virtual connectors between different zones, so that, dynamic demand flows can be operated. ***vert_with_connect*** are the main network edges, usually each zone has one vertex, but wider zones can have more. They are represented by water-blue circles, and are linked to each other through yellow connectors, ***Connectors_Porto***.

Vehicular movements are given by OD matrices. Those matrices are contingency tables in which each row represents an origin and each column a destination, with each cell indicating the flow between a distinct OD pair during a given time period, therefore they describe people movement in a certain area [GKW18].

In this research, OD pairs are distributed over 119 different zones, which id-zone correspondence can be found in Table B.1. According to the existing flow between origin and destination the standard OD matrix table is exported into an ***.xml***, where each line contains the number of

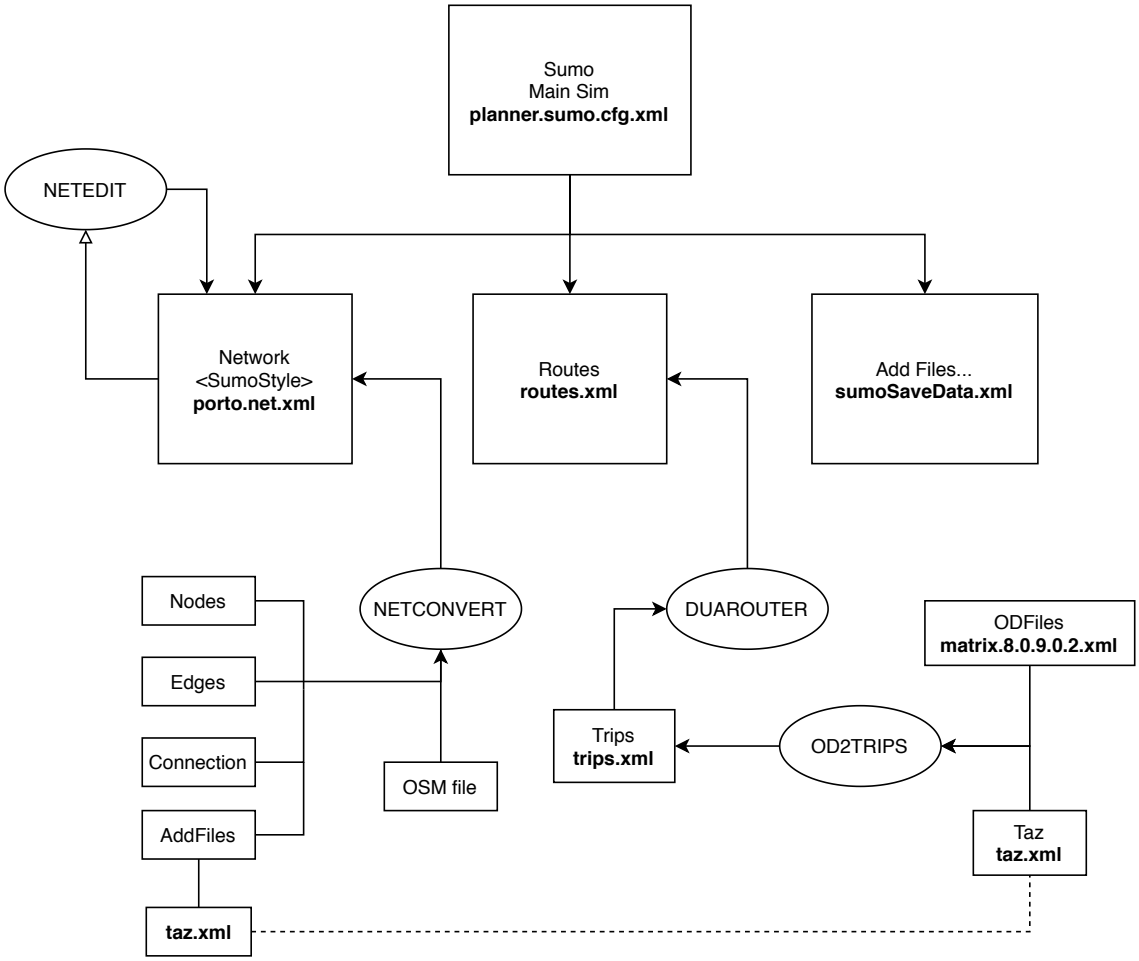



Figure 3.5: Porto simulation network diagram

trips between a specific origin and destination. A 24-hour profile of those flows may be found on Figure 3.7.

There are two main curves worth mentioning. Weekend evolution is given by the brown curve, whereas the blue curve represents the average weekday profile. A simplistic analysis may point that weekdays have more traffic demand when comparing to weekends, with the exception of the first hours of the day, when people are more willing to go out than during a weekday. By analysis the weekday curve evolution, it is possible to find one global minimum from 4h00 to 5h00, a good point to work with and analyse the simulation behaviour, implementing some changes.Weekday analysis allows the finding of two local maximums, one from 8h00 to 9h00 and another from 18h00 to 19h00. Despite global maximum is situated on the afternoon period (18h00-19h00), we believe it is more interesting to work with the other maximum (8h00-9h00), which total number of trips is very close to global maximum, but has exponential growth when comparing to the hours immediately before, including the global minimum. This approach will make us understand how the simulation model reacts to almost instantaneous network saturation.

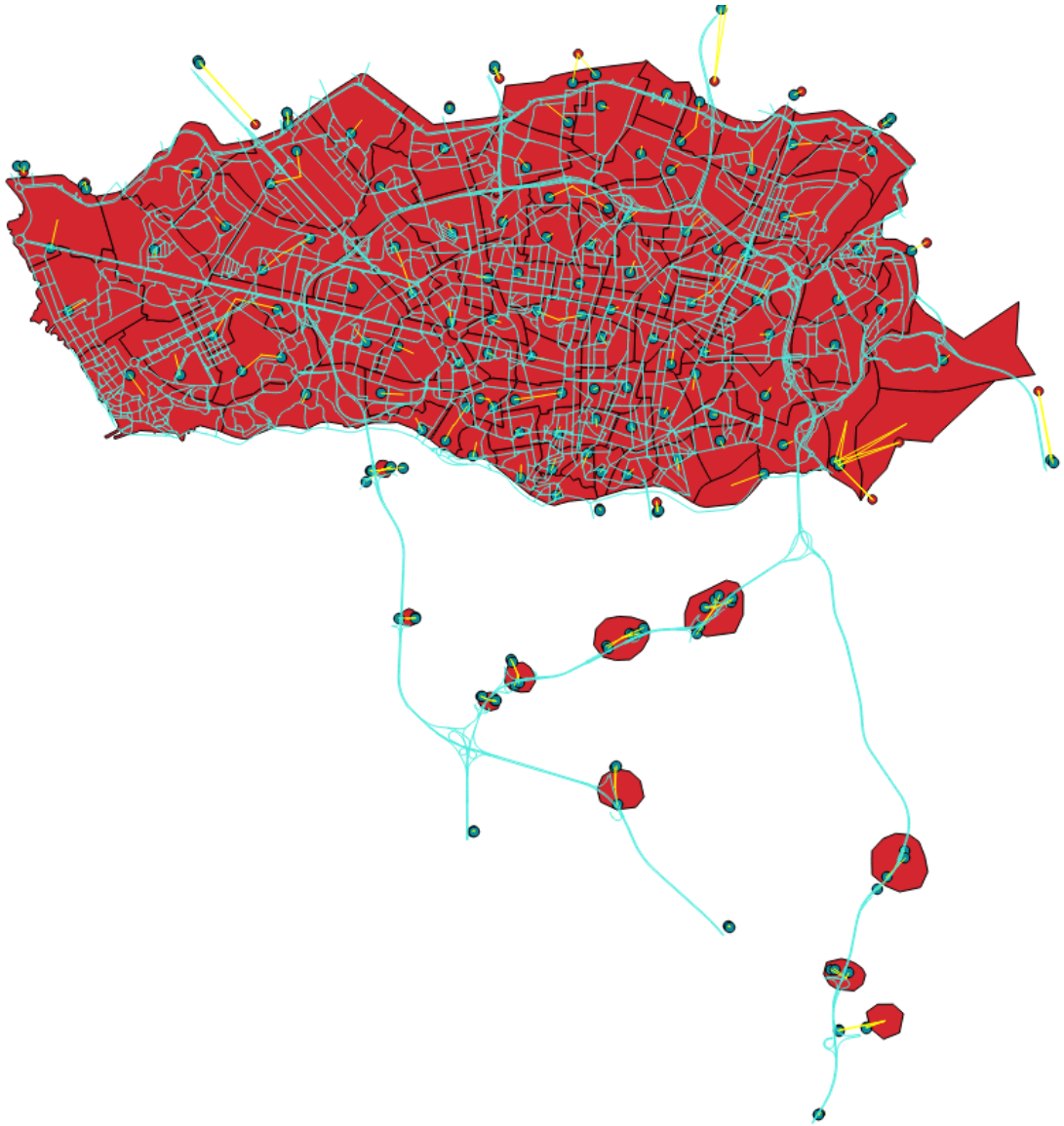


Figure 3.6: Porto urban area - shapefiles representation

## 3.4 Summary

This Chapter presents an approach to calibrate the given simulation thought route assignment, considering its dynamic demand characteristics. First of all, it was necessary to study SUMO's algorithms behaviour using example networks. This preliminary study provided useful information about algorithm's requirements, usage and application. Thus, it was possible to define a strategy considering three progressive types of assignment: Standard, Iterative and Incremental. For each scenario, these three algorithms would be applied, analysing how they improve simulation performance. Furthermore, case-study network was fully described. This exhaustive description included simulation architecture, shapefiles representation, TAZ distribution but also trips evolution for a 24-hour period. The evolution of that 24-hour profile justifies the chosen time-slots

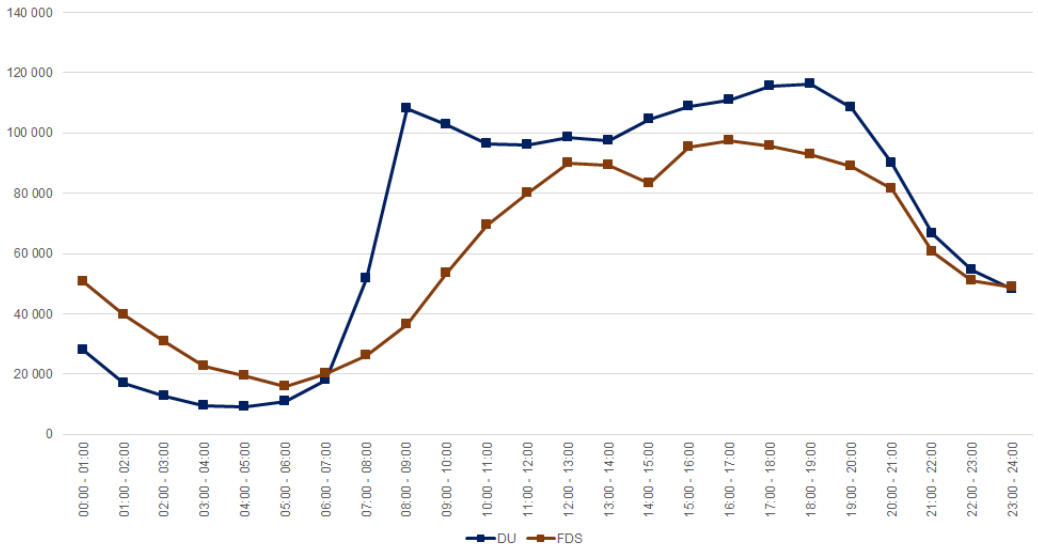



Figure 3.7: Total number of trips in Porto urban area for 24 hours

for assignment experiments. Off-peak period, from 4h00 to 5h00 presents the lowest number of trips, whereas peak hour period, from 8h00 to 9h00, is a local maximum, with an exponential trip increase.

# Chapter 4

# Result Analysis and Discussion

This chapter presents the results of several experiments conducted during the implementation, and also discusses results in detail. Section 4.1, explores a preliminary approach, testing the different algorithms with simple networks. Section 4.2 consider the same algorithms, but this time for our case-study. And, finally, Section 4.3 discusses all the results presented on this chapter.

## 4.1 Preliminary Results

Before working on our case-study it was necessary to conduct some preliminary experiences. This approach had the goal of acquiring more knowledge on how route assignment algorithms work. So, it was necessary to test them with more simple networks, evaluating their performance and potential.

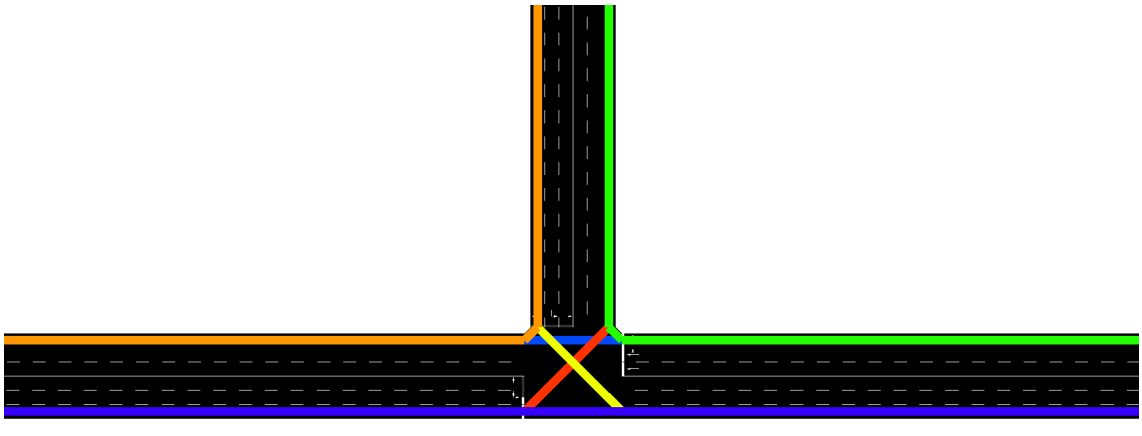


Figure 4.1: Possible routes for *quickstart* network

The first experiment was conducted with *quickstart* network. This testing network has 4 edges that are both used as origins and destinations. It also has two traffic junctions as Figure 3.2 shows.

For this analysis, the focus should be the number of different routes that each algorithm generates. Once, there is a fixed number of vehicles driving in the simulation, other parameters would not have the expected result. This simulation routes are already defined in ***quickstart.rou.xml*** file, allowing a direct comparison with algorithm's results. In fact, all algorithms outputed the same number of possible routes (12). This can be explained considering that the basic scenario, with predefined routes, already considers all those routes. If each origin has three possible destinations, 12 (3x4) would be the maximum number of different routes. Figure 4.1 has a visual representation of some routes for *quickstart* network.

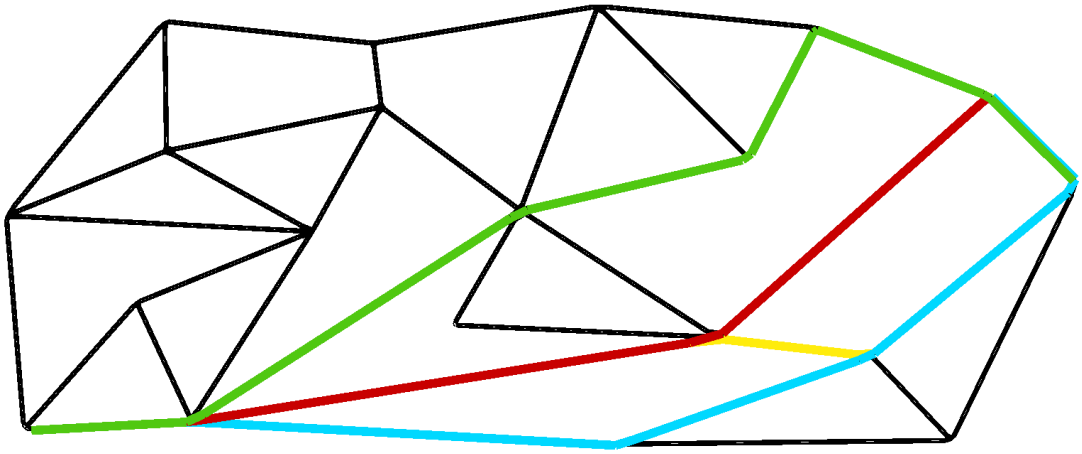


Figure 4.2: Possible routes for *10x10* network

Once, this initial network (*quickstart*) does not evidence algorithm's performance and behaviour, a more complex network was required. So, the algorithms were tested under dynamic demand with a 10x10 network. The experience consisted in inducing a flow of 100 vehicles from Taz 1 to Taz 4. Using the Standard algorithm vehicles are assigned between two possible routes. Considering Figure 4.2, blue and red routes are the only two options for this kind of assignment. On the other hand, both Iterative and Incremental algorithms increase the available options to 4, adding the green and the yellow routes. This simple test suggests that Iterative and Incremental assignment actually have a better performance when applied to networks with dynamic demand. The results of these two experiments are summarized on Table 4.1.

| **Network** | **Standard** | **Iterative** | **Incremental** |
|---|---|---|---|
| quickstart | 12 | 12 | 12 |
| 10x10 | 2 | 4 | 4 |

Table 4.1: Number of possible routes for preliminary networks

## 4.2 Experimental Results for the Study Network

### 4.2.1 Initial Network Results

To evaluate the network behaviour as well as route assignment algorithms performance, a collection of real data was gathered. Six representative edges were chosen, being divided into three different types: motorway, primary and secondary. *A20* and *N14* edges describe a motorway scenario. where big amounts of traffic flow, with a considerable high speed. Those edges are associated with Drive sensors, making reads of all vehicles driving those edges. These sensors are the ones with more technological attributes, collecting three different types of information: volume, occupancy and speed. Volume can be defined as the total amount of vehicles breaking through a specific edge, occupancy is the rate of vehicles on the edge, when compared to its full capacity (percentage expressed), finally, speed information is the average speed (*km/h*) of all the vehicles in the edge. *Avenida dos Aliados* is the only secondary edge, similarly to *Avenida Fernão Magalhães*, its data come from CMP (volume) and STCP (speed). The edges corresponding to *Estrada da Circunvalação*, 1051017 and 1954763, are primary edges, however, there is only one data source, STCP, providing only speed information. Table 4.2 shows real data information for the elected edges.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 213 | 0.432 | 71.777 |
| N14 | 1062796 | 192 | 0.816 | 64.034 |
| Avenida Fernão Magalhães | 1523231 | 65 | ——— | 21.5 |
| Avenida dos Aliados | 1706060 | 71 | ——— | 6.75 |
| Estrada da Circunvalação | 1051017 | —— | ——— | 30.667 |
| Estrada da Circunvalação | 1954763 | —— | ——— | 7 |

Table 4.2: Real traffic values of Porto urban area from 4h00 to 5h00

A different approach is used to treat each traffic value. Volume data is presented with an numerical value, corresponding to vehicle units (#). Each sensor volume read is summed into a total volume for a specific edge. Occupancy is presented in percentage (%), where 1 corresponds to 100% value. It was calculated as an average of all occupancy reads. Lastly, speed approach is very similar to occupancy, an average is calculated, presenting the result in *km/h*.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 263 | 0.39 | 67.068 |
| N14 | 1062796 | 90 | 0.22 | 66.888 |
| Avenida Fernão Magalhães | 1523231 | 163 | 1.8 | 22.968 |
| Avenida dos Aliados | 1706060 | 80 | 0.56 | 24.228 |
| Estrada da Circunvalação | 1051017 | 27 | 0.16 | 40.536 |
| Estrada da Circunvalação | 1954763 | 48 | 0.47 | 25.704 |

Table 4.3: Duarouter simulated traffic values of Porto urban area from 4h00 to 5h00

When it comes to simulated traffic values, another approach has been followed. Simulation configuration file includes an additional file, *SumoSaveData.xml*, which <*edgeData*> property determines that edge's information would be written into *simData.xml*. On this file each <*edge*> node as the same descriptive ID of real data Table 4.2. Then, the required attributes are determined as *occupancy*, *speed* and *entered* (volume). All of them respect the units previously determined with the exception of speed. It is necessary to convert *m/s* value into *km/h*.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 325 | 0.48 | 67.032 |
| N14 | 1062796 | 87 | 0.21 | 67.14 |
| Avenida Fernão Magalhães | 1523231 | 115 | 1.09 | 26.64 |
| Avenida dos Aliados | 1706060 | 57 | 0.4 | 24.3 |
| Estrada da Circunvalação | 1051017 | 31 | 0.19 | 40.464 |
| Estrada da Circunvalação | 1954763 | 33 | 0.32 | 25.848 |

Table 4.4: Dua Iterate simulated traffic values of Porto urban area from 4h00 to 5h00

Tables 4.3, 4.4 and 4.5 gather the simulated traffic information, so that is possible to compare them with real data of Table 4.2. A quick overview enhances the similar values of *A20* and *N14* highways to a real scenario, particularly on *A20*. However, this analysis does not provide enough information on each algorithm predictive performance, being required more robust error measures.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 219 | 0.32 | 66.924 |
| N14 | 1062796 | 85 | 0.21 | 66.888 |
| Avenida Fernão Magalhães | 1523231 | 99 | 1.12 | 22.428 |
| Avenida dos Aliados | 1706060 | 62 | 0.43 | 24.516 |
| Estrada da Circunvalação | 1051017 | 33 | 0.2 | 40.86 |
| Estrada da Circunvalação | 1954763 | 55 | 0.53 | 26.028 |

Table 4.5: One Shot simulated traffic values of Porto urban area from 4h00 to 5h00

Root Mean Square Error (RMSE) was chosen to make a global analysis of simulated values deviation when compared to real values. The different types and scale of the considered values requires an error measure that does not discriminate between them. RMSE considers the individual deviations (simulated vs real) and gathering them in a single value. To calculation purposes some simulated values had to be ignored, once there are no equivalent real values. As Table 4.2 shows, sensors could not obtain Volume data for edges 1051017 and 1954763 (*Estrada da Circunvalação*). Occupancy values are also not available for these two edges, but also for 1523231 (*Avenida Fernão Magalhães*) and 1706060 (*Avenida dos Aliados*). So, RMSE only considers the 12 remaining values, assigning the same weight to all of them.

Another measure that was also considered is the correlation coefficient between real and simulated values. The correlation value gives the strength and direction of this linear relationship. Values near 1 theoretically mean that the that is an almost perfect fit of the simulation model when

compared to urban traffic reality. In terms of calculations, the same restrictions of RMSE were applied. The edges without real values were excluded, giving the same weight to all remaining individual values.

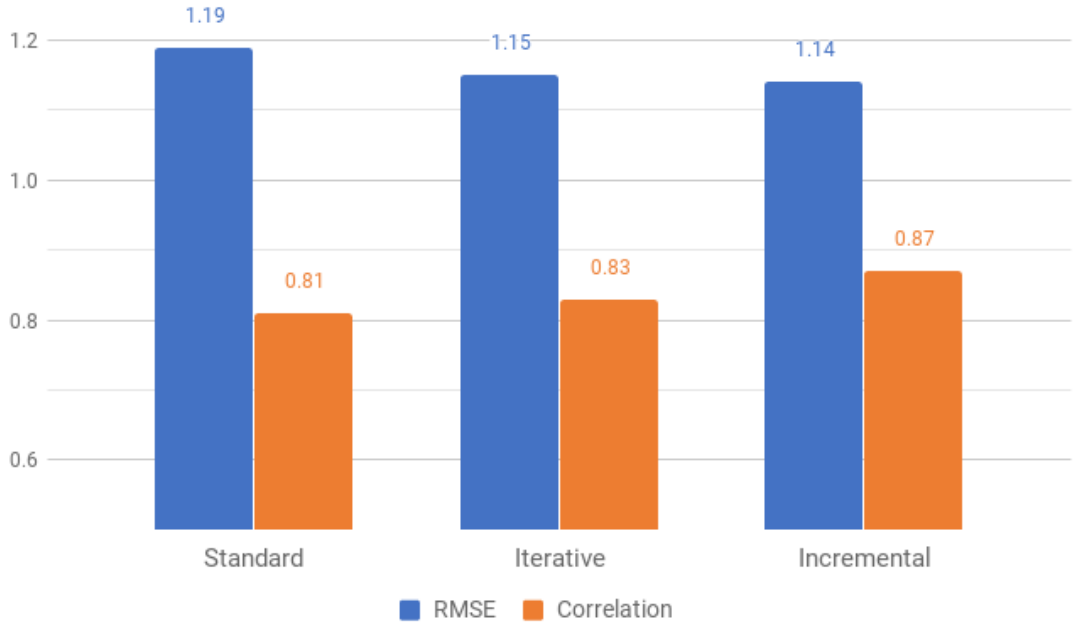



Figure 4.3: Off-peak error measurement for initial network

The graphics of Figure 4.3 illustrate the calculated error values for each route assignment algorithm. This off-peak, from 4h00 to 5h00, draws two straight lines, one for RMSE decrease and another depicting the correlation increase. When comparing to Standard assignment, the Iterative algorithm improves both measured values, increasing the correlation to 0.83 and decreasing the RMSE to 1.15. But, if we consider an Incremental approach, the results are even better, reducing the interval to 1.14 of RMSE and 0.87 when it comes to correlation coefficient. From the three, Incremental assignment is the most appropriate for this particular case, producing the most approximate results to real scenario.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 5707 | 10,82 | 80,486 |
| N14 | 1062796 | 3500 | 0.816 | 80,101 |
| Avenida Fernão Magalhães | 1523231 | 635 | ——— | 11,024 |
| Avenida dos Aliados | 1706060 | 306 | ——— | 4,926 |
| Estrada da Circunvalação | 1051017 | —— | ——— | 4,893 |
| Estrada da Circunvalação | 1954763 | —— | ——— | 12,923 |

Table 4.6: Real traffic values of Porto urban area from 8h00 to 9h00

A second test was performed regarding a peak hour simulation. Sensors data were collected, gathering information from 8h00 to 9h00. This hour is time interval is particularly interesting,

once the network is quickly populated with working trips. Table 4.6 summarizes the same data types used in 4h00 to 5h00 period: volumes, occupancy and speed.

Similarly to off-peak period, three types of assignment were performed: Standard, Iterative and Incremental. Each one of them resulted in different simulated values when it comes to the considered variables (volume, occupancy and speed). Tables 4.7, 4.8 and 4.9 summarize this information for the previously selected edges.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 719 | 4.2 | 16.164 |
| N14 | 1062796 | 406 | 1 | 66.492 |
| Avenida Fernão Magalhães | 1523231 | 338 | 52.89 | 1.296 |
| Avenida dos Aliados | 1706060 | 105 | 19.14 | 1.188 |
| Estrada da Circunvalação | 1051017 | 305 | 1.86 | 40.5 |
| Estrada da Circunvalação | 1954763 | 255 | 11.22 | 5.616 |

Table 4.7: Dua Router simulated traffic values with initial network from 8h00 to 9h00

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 894 | 7.27 | 12.24 |
| N14 | 1062796 | 315 | 0.77 | 66.852 |
| Avenida Fernão Magalhães | 1523231 | 98 | 14.91 | 3.6 |
| Avenida dos Aliados | 1706060 | 91 | 23.89 | 0.828 |
| Estrada da Circunvalação | 1051017 | 568 | 39.55 | 3.384 |
| Estrada da Circunvalação | 1954763 | 299 | 3.26 | 23.076 |

Table 4.8: Dua Iterate simulated traffic values with initial network from 8h00 to 9h00

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 1185 | 17.55 | 6.66 |
| N14 | 1062796 | 281 | 0.69 | 66.78 |
| Avenida Fernão Magalhães | 1523231 | 281 | 7.93 | 9.036 |
| Avenida dos Aliados | 1706060 | 107 | 17.17 | 1.368 |
| Estrada da Circunvalação | 1051017 | 1004 | 20.13 | 12.528 |
| Estrada da Circunvalação | 1954763 | 258 | 3.62 | 17.82 |

Table 4.9: One Shot simulated traffic values with initial network from 8h00 to 9h00

To compare the obtained results with real values of Table 4.6, the same approach of off-peak experiment was used. Residual errors were grouped into RMSE and correlation coefficient measures, showing how each algorithm behaves in a more stressed environment. The graphics of Figure 4.4 summarize these information. A notable higher RMSE error is noticed in Standard assignment when compared to off-peak period. But is also noticeable that both Iterative and Incremental assignment reduce that error, while increasing the correlation coefficient. For this scenario, Iterative assignment has the best performance, presenting the lowest RMSE value (0.73) and the

highest correlation (0.97). This is contrasting to the better performance of Incremental within off-peak period. Another relevant aspect is that all algorithms increase correlation coefficient to values really close to the perfect fit (1).

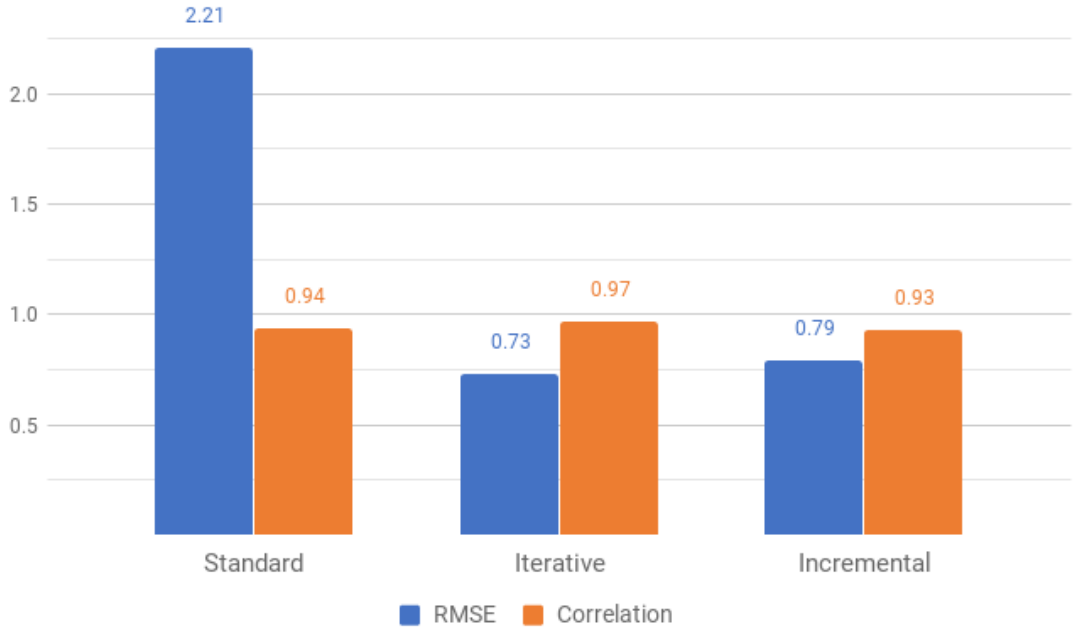



Figure 4.4: Peak hour error measurement for initial network

### 4.2.2 Improved Network Results

After solving some priority issues, another battery of tests was conducted. This improved network would be able to deal with junctions deadlocks, allowing some vehicles ta have better flows. It is expected that those corrections represent an approximation to real values, allowing some vehicles to increase their speed but also the edge volume. To validate the corrections of this improved network it is necessary to run those three algorithms (Standard, Iterative and Incremental) for both peak and off-peak periods.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 272 | 0.41 | 65.664 |
| N14 | 1062796 | 90 | 0.22 | 66.6 |
| Avenida Fernão Magalhães | 1523231 | 165 | 1.79 | 23.364 |
| Avenida dos Aliados | 1706060 | 80 | 0.53 | 25.776 |
| Estrada da Circunvalação | 1051017 | 27 | 0.16 | 40.428 |
| Estrada da Circunvalação | 1954763 | 48 | 0.46 | 25.92 |

Table 4.10: Dua Router simulated traffic values with improved network from 4h00 to 5h00

Considering the off-peak period, the extracted results were grouped in three tables. Table 4.10 is relative to Standard assignment, Table 4.11 is focused on the Iterative approach, and finally,

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 337 | 0.51 | 65.556 |
| N14 | 1062796 | 90 | 0.22 | 66.42 |
| Avenida Fernão Magalhães | 1523231 | 130 | 1.14 | 28.764 |
| Avenida dos Aliados | 1706060 | 56 | 0.36 | 26.64 |
| Estrada da Circunvalação | 1051017 | 32 | 0.19 | 40.932 |
| Estrada da Circunvalação | 1954763 | 31 | 0.27 | 28.224 |

Table 4.11: Dua Iterate simulated traffic values with improved network from 4h00 to 5h00

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 218 | 0.33 | 65.736 |
| N14 | 1062796 | 88 | 0.22 | 66.384 |
| Avenida Fernão Magalhães | 1523231 | 90 | 1.08 | 21.06 |
| Avenida dos Aliados | 1706060 | 63 | 0.41 | 25.776 |
| Estrada da Circunvalação | 1051017 | 36 | 0.22 | 39.6 |
| Estrada da Circunvalação | 1954763 | 54 | 0.53 | 25.596 |

Table 4.12: One Shot simulated traffic values with improved network from 4h00 to 5h00

Table 4.12 sums up Incremental results. Regarding this, those results only have a proper meaning when directly compared with error measures of the initial network. This comparison can be found with the graphics of Figure 4.5.

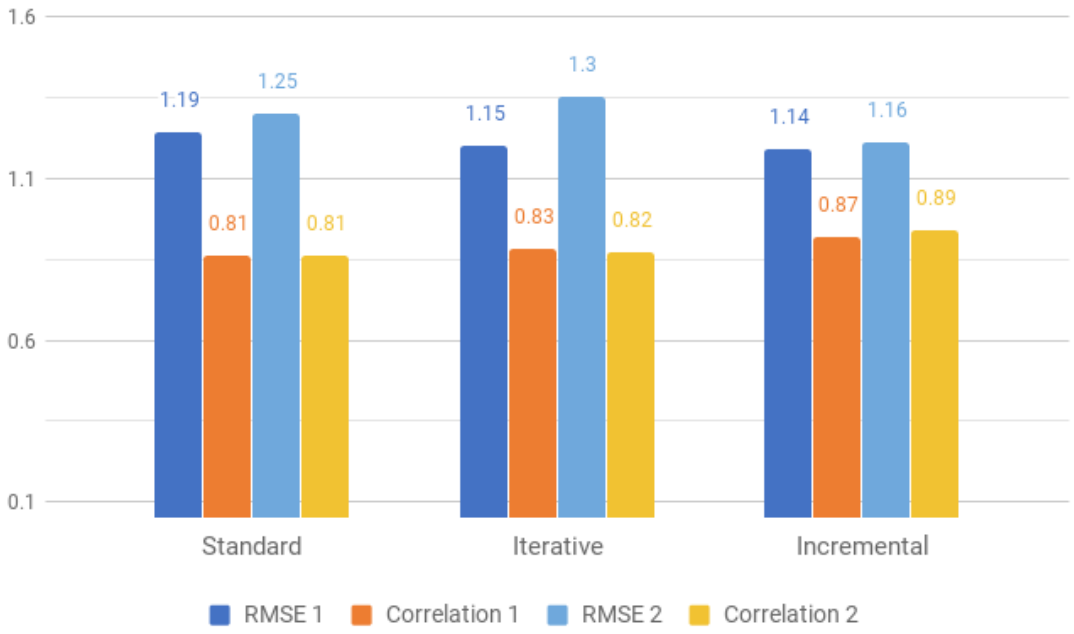



Figure 4.5: Off-peak error measurement for improved network

In this graphical representation, the initial network error measures are given by RMSE 1

and Correlation 1, whereas RMSE 2 and Correlation 2 refer to the improved network. A quick overview suggests a that the supposedly improved network does not improve nothing at all, by the contrary. But this in not true, despite higher RMSE values, the correlation values are approximately the same, increasing for Incremental assignment. Another point is that we should also compare a peak hour simulation with this improved network before general conclusions.

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 878 | 10.77 | 7.704 |
| N14 | 1062796 | 400 | 1 | 65.628 |
| Avenida Fernão Magalhães | 1523231 | 389 | 46.79 | 1.836 |
| Avenida dos Aliados | 1706060 | 114 | 19.02 | 0.936 |
| Estrada da Circunvalação | 1051017 | 311 | 1.92 | 39.924 |
| Estrada da Circunvalação | 1954763 | 218 | 7.68 | 7.02 |

Table 4.13: Dua Router simulated traffic values with improved network from 8h00 to 9h00

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 1098 | 8.41 | 12.996 |
| N14 | 1062796 | 271 | 0.67 | 65.736 |
| Avenida Fernão Magalhães | 1523231 | 177 | 3.79 | 11.988 |
| Avenida dos Aliados | 1706060 | 75 | 14.66 | 0.792 |
| Estrada da Circunvalação | 1051017 | 807 | 25.05 | 8.136 |
| Estrada da Circunvalação | 1954763 | 395 | 6.14 | 16.128 |

Table 4.14: Dua Iterate simulated traffic values with improved network from 8h00 to 9h00

| Description | Edge | Volume | Occupancy | Speed |
|---|---|---|---|---|
| A20 | 1016654 | 1270 | 19.59 | 6.156 |
| N14 | 1062796 | 335 | 0.84 | 65.772 |
| Avenida Fernão Magalhães | 1523231 | 336 | 22.01 | 3.924 |
| Avenida dos Aliados | 1706060 | 144 | 10.48 | 2.16 |
| Estrada da Circunvalação | 1051017 | 1136 | 15.89 | 17.892 |
| Estrada da Circunvalação | 1954763 | 213 | 3.19 | 16.128 |

Table 4.15: One Shot simulated traffic values with improved network from 8h00 to 9h00

When it comes to peak hour period, the same method was followed. Each algorithm produced volume, occupancy and speed information. Table 4.7 is focused on the Standard assignment, Table 4.8 represents the simulated values with an Iterative approach, and lastly, Incremental assignment results may be found on Table 4.9. Then, this information was computed into the chosen error measures and graphically compared to initial network values (Figure 4.6). A general analysis suggests that this improved network has a better performance when compared to the initial network. There are lower RMSE values in both Standard (2.17 instead of 2.21) and Iterative (0.68 instead of 0.73) algorithms. On the other hand, Incremental assignment presents an higher RMSE value

(1 instead of 0.79). When it comes to correlation coefficient the values are very similar in both experiments, with a slight decrease (from 0.97 to 0.94) on Iterative approach.

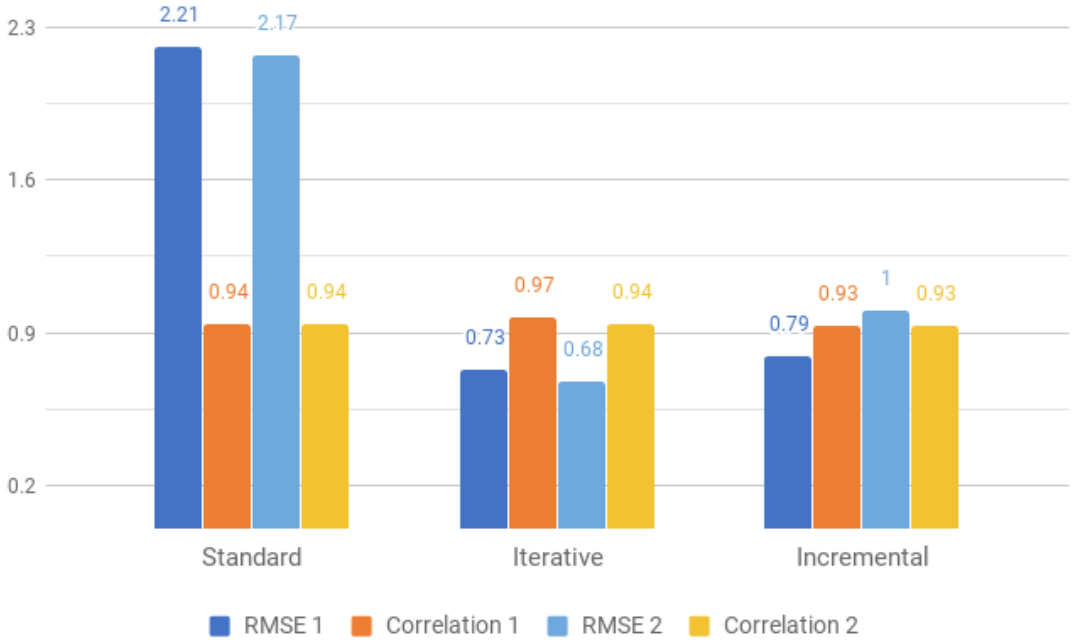



Figure 4.6: Peak hour error measurement for improved network

## 4.3 Discussion

There are some common aspects to all experiments. Standard and Iterative algorithms are strongly correlated, their error measurements increase and decrease under the same stimulus. Figure 4.5 and Figure 4.6 provide a clear representation of this dependence. When Standard RMSE and Correlation values decrease (Figure 4.6), considering the improved network, they also decrease with Iterative assignment. Figure 4.5 represents the complementary situation, once Standard error values increase, Iterative values also increase. A possible explanation for this relationship is the fact that Iterative assignment uses the same algorithm of Standard assignment, but in its full power, conducting to network equilibrium (SUE). Contrasting this direct relationship, Incremental assignment reveals to be quite unpredictable. Its error values do not even follow the expected behaviour. It would be expected a Correlation decrease when RMSE increases, similarly to Standard and Iterative approaches, but this is not the case. Another point as it can evolve from a very good prediction to a very doubtful one, or vice-versa. A lucky application of this algorithm may reveal very helpful, otherwise it is just a waste of resources. Once it consumes approximately the same amount of time when comparing to the Iterative approach, it is recommended to avoid it, preferring the a more reliable and secure iterative assignment.

Relatively to the studied network there are some points that should be discussed. The proposed route assignment approaches approximated the studied simulation to a real traffic scenario. There

is a clear advantage on Iterative/Incremental approaches when compared to the usage of *duarouter*, the Standard algorithm, chosen by ARMIS for vehicle routing. A first experiment, using the initial network revealed that both Iterative and Incremental algorithms reduced the error (RMSE), while increasing correlation. This positive behaviour was verified for both off-peak (Figure 4.3) and peak hour (Figure 4.4) periods. For the second experiment, with an improved network, those algorithms had a slightly different behaviour. Peak hour period followed the tendency of reducing the error, while increasing the correlation coefficient. However, for off-peak period, worse results were presented. This can be explained by two factors: the lack of sensors data for volume & occupancy and the low rate of vehicles driving during this period. Once the chosen error measure gives the same importance to all values, an edge lacking volume and occupancy information is subject to only speed analysis. If the simulated speed parameter presents a big difference from sensors information, the edge is not able to compensate with other error measures, increasing the error when it could be a situation more close to reality. The fact of being a period with few vehicles may also influence this error increase. Once, there are few vehicles on the network, a small differences in speed and route choice have greater influence on error parameters. Considering this, improved network results for off-peak period should be overlooked. Those improvements had a very positive effect with an highly stressed scenario (peak hour period), reducing simulation error and the amount of teleported vehicles. It is expected that further changes within priority values conduct to even better results.

Finally, we must focus on what is causing the mismatch between reality and the studied simulation, proposing a method to reduce this mismatch. After a first experiment, where different route assignment algorithms were used, it was necessary to discover what was the cause of such high error values. Considering this, experiments with reduced traffic flows were conducted. In fact, it was possible to observe several deadlocks, vehicles were getting stuck at junctions, originating long traffic queues. We concluded that these congestion is originated by erratic priority values, so we fixed some of them, expecting to reduce error measurement values within a second experiment. As it is discussed on the previous paragraph those improvements produced a positive impact on the network. We consider that solving the remaining priority issues will produce a very close approximation of this network to Porto urban scenario, supported by the Iterative approach to perform route assignment.

## 4.4 Summary

Along with this Chapter, outcomes from the proposed methodology have been presented, analyzed, and discussed. Preliminary results, testing those three chosen algorithms with simple networks, achieved the pursued goal of acquiring more knowledge of their behavior for an effective scenario. Afterwards, case-study results were analyzed and discussed considering the initially given network, and then, an improved network, solving some priority issues in order to reduce deadlocks impact. Finally, discussion section focused the performance of each algorithm and its impact on the considered networks. I was also explored the applicability of the chosen error measures and

how they could lead to misinterpretations, once there is lack of sensors information, as well as that further deadlocks elimination would result on reducing the gap between simulated and real traffic values.

# Chapter 5

# Conclusions and Future Work

This Chapter is focused on conclusions. A general overview (5.1) of carried out work is followed by project main contributions (5.2). Then, Section 5.3 and Section 5.4 describe some of the work that may be developed in the future.

## 5.1 Overview

This reports starts by describing traffic simulation models, presenting some calibration strategies, namely route assignment methods. Given simulation was developed on SUMO, an open-source microscopic traffic simulator, so it was crucial to be familiarized with its architecture, but also with its traffic assignment tools. This way, it was possible calibrate this simulation recurring to route assignment algorithms.

After some preliminary tests, it was possible to define a methodology, using three different types of assignment: Standard, Iterative and Incremental. The application of those algorithms conducted to deadlocks discovery across the network, raising other problem: what strategy should be followed to eliminate those deadlocks. Thus, an hypothesis related to junctions priorities was formulated. By solving some priority issues there were less teleported vehicles, as well as more satisfying results among chosen error measures. It is expected that eliminating those deadlocks reveals the true potential of applied route assignment algorithms, producing a simulation extremely close to reality.

Overall, the proposed solution address raised issues, pursuing the goal of improving given simulation calibration. Despite the lack of real data from traffic sensors, it was possible to validate this approach, providing a positive contribution to C-ROADS project and every involved entities, through calibration of Porto urban area simulation network.

## 5.2 Main Contributions

This dissertation is focused on simulation calibration in order to bridge the gap between real and simulated data. Considering this, the main contributions of this project are the following:

- **Simulation calibration through route assignment** - through Iterative and Incremental approach was possible to reduce simulation error measures, when compared to the default Standard assignment, previously used by ARMIS
- **Deadlocks discovery and correction** - during calibration process focus of congested traffic were discovered and then corrected. This reduced simulation error measures.

This methodology may be used to calibrate other microscopic simulations. Thus, a scientific article is being written, reporting how network priorities may influence the calibration of medium scale urban networks considering a microscopic simulation. This paper will be submitted at IEEE International Smart Cities Conference (ISC2 2020).

## 5.3 Further Developments

A detailed analysis after a first experiment with the studied network revealed several deadlocks along Porto urban area. Those deadlocks induce long traffic queues surrounding adjacent areas, blocking almost every vehicle which route includes those affected edges. In terms of traffic values, vehicles' speed exponentially decreases for values of approximate 1km/h - 2km/h, on the other hand, affected edges increase their number of vehicles, but volume and occupancy do not increase, once vehicles get stuck in those edges, not increasing flow rate. It is crucial to eliminate all deadlocks, so that the induced number vehicles are able to flow with reasonable speed. Our strategy was focused on priority issues, resulting in error reduction, however, there was not enough time to validate all problematic junctions. Thus, it is necessary to continue this task, fixing junctions priorities. Additionally, semaphoric information of Porto's network can be used to solve more complex situations, increasing simulation accuracy level. Those improvements can be validated by considerable decrease on the number of teleported vehicles, but also by observing deadlocks absence in progressively denser networks. Then, having a free-deadlock network, the same approach of this research should be used, performing route assignment with those three algorithms, for both peak and off-peak periods.

## 5.4 Future Work

This research was conducted under dynamic demand. At a first stage, OD matrices were collected, retrieving information about city's mobility patterns. Those matrices define that a given number of vehicles commute from one TAZ to another during a determined time period of the day. Trips and routes are generated for this dynamic demand, resulting in big amounts of vehicles populating

the network. Then, it is required to calibrate this network according data collected from sensors. However, this pipeline could be inverted, resulting in a new research project. Sensors information could be first simulation input, assigning specific traffic values for each edge. This information would include volume, occupancy and speed values, but also information that artificial detectors generate. SUMO already has a module to develop this kind of simulation from observation points, DFROUTER. The calibration process would be focused on simulation detectors and its properties. As a final output, it would be possible to generate OD matrices, representing dynamic demand that would validate the calibration process of this second simulation.

# Appendix A

# Algorithms' detailed running options

| Option | Description |
|---|---|
| --net-file <FILE> **or** -n <FILE> | Sumo network (**mandatory**) |
| --route-files <FILE> **or** -t | Read sumo routes, alternatives, flows, and trips from FILE(s) |
| --additional files <FILE> **or** -d | Read additional network data (districts, bus stops) from FILE(s) |
| --phemlight-path <FILE> | Determines where to load PHEMlight definitions from |
| --junction-taz <BOOL> | Initialize a TAZ for every junction to use attributes toJunction and fromJunction |
| --weight-files <FILE> **or** -w <FILE> | Read network weights from FILE(s) |
| --lane-weight-files <FILE> | Read lane-based network weights from FILE(s) |
| --weight-attribute <W> **or** -x <W> | Name of the xml attribute which gives the edge weight (default: traveltime) |

Table A.1: Input options for *duarouter*

| Option | Description |
|---|---|
| --configuration-file <FILE> **or** -c <FILE> | Loads the named config on startup |
| --save-configuration <FILE> **or** -C <FILE> | Saves current configuration into FILE |
| --save-template <FILE> | Saves a configuration template (empty) into FILE |
| --save-schema <FILE> | Saves the configuration schema into FILE |
| --save-commented <FILE> | Adds comments to saved template, configuration, or schema (default: false) |

Table A.2: Configuration options for *duarouter*

| **Option** | **Description** |
|---|---|
| --begin <TIME> **or** -b <TIME> | Defines the begin time; Previous trips will be discarded (default :0) |
| --end <TIME> **or** -e <TIME> | Defines the end time; Later trips will be discarded; Defaults to the maximum time that SUMO can represent (default : 9223372036854774) |

Table A.3: Time options for *duarouter*

| **Option** | **Description** |
|---|---|
| --random <BOOL> | Initialises the random number generator with the current system time (default: false) |
| --seed <INT> | Initialises the random number generator with the given value (default : 23423) |

Table A.4: Random number options for *duarouter*

| **Option** | **Description** |
|---|---|
| --write-license <BOOL> | Include license info into every output file (default: false) |
| --output-prefix <FILE> | Prefix which is applied to all output files. The special string 'TIME' is replaced by the current time. |
| --precision <INT> | Defines the number of digits after the comma for floating point output (default: 2) |
| --precision.geo <INT> | Defines the number of digits after the comma for lon,lat output (default: 6) |
| --human-readable-time <BOOL> **or** -H <BOOL> | Write time values as hour:minute:second or day:hour:minute:second rather than seconds (default: false) |
| --output-file <FILE> **or** -o <FILE> | Write generated routes to FILE |
| --keep-type-distributions <BOOL> | Keep vTypeDistribution ids when writing vehicles and their types (default: false) |
| --alternatives-output <FILE> | Write generated route alternatives to FILE |
| --intermodal-network-output <FILE> | Write edge splits and connectivity to FILE |
| --write-trips <BOOL> | Write trips instead of vehicles (for validating trip input) (default: false) |
| --write-trips.geo <BOOL> | Write trips with geo-coordinates (default: false) |
| --write-trips.junctions <BOOL> | Write trips with fromJunction and toJunction (default: false) |
| --exit-times <FILE> | Write exit times (weights) for each edge (default: false) |

Table A.5: Output options for *duarouter*

| **Option** | **Description** |
|---|---|
| --departlane <STRING> | Assigns a default depart lane |
| --departpos <STRING> | Assigns a default depart position |
| --departspeed <STRING> | Assigns a default depart speed |
| --arrivallane <STRING> | Assigns a default arrival lane |
| --arrivalpos <STRING> | Assigns a default arrival position |
| --arrivalspeed <STRING> | Assigns a default arrival speed |
| --defaults-override <FLOAT> | Defaults will override given values (default: false) |

Table A.6: Default options for *duarouter*

| **Option** | **Description** |
|---|---|
| --astar.save-landmark-distances <FILE> | Save lookup table for astar ALT-variant to the given file |
| --gawron.beta <FLOAT> | Use FLOAT as Gawron's beta (default: 0.3) |
| --gawron.a <FLOAT> | Use FLOAT as Gawron's a (default: 0.5) |
| --keep-all-routes <BOOL> | Save routes with near zero probability (default: false) |
| --skip-new-routes <BOOL> | Only reuse routes from input, do not calculate new ones (default: false) |
| --ptline-routing <BOOL> | Route all public transport input (default: false) |
| --logit <BOOL> | Use c-logit model (deprecated in favor of –route-choice-method logit) (default: false) |
| --route-choice-methos <STRING> | Choose a route choice method: gawron, logit, or lohse (default: gawron) |
| --logit.beta <FLOAT> | Use FLOAT as logit's beta (default: -1) |
| --logit.gamma <FLOAT> | Use FLOAT as logit's gamma (default: -1) |
| --logit.theta <FLOAT> | Use FLOAT as logit's theta (negative values mean auto-estimation) (default: -1) |
| --persontrip.walkfactor <FLOAT> | Use FLOAT as a factor on pedestrian maximum speed during intermodal routing (default: 0.75) |
| --persontrip.transfer.car-walk | Where are mode changes from car to walking allowed (possible values: 'parkingAreas', 'ptStops', 'allJunctions', 'taxi' and combinations) (default: parkingAreas) |
| --weights.priority-factor <FLOAt> | Consider edge priorities in addition to travel times, weighted by factor; (default: 0) |
| --astar.all-distances <FILE> | Initialize lookup table for astar from the given file (generated by marouter –all-pairs-output) |
| --weight-period <TIME> | Aggregation period for the given weight files; triggers rebuilding of Contraction Hierarchy (default: 3600) |

Table A.7: Processing options for *duarouter* - part 1

| **Option** | **Description** |
|---|---|
| --unsorted-input <BOOL> **or** -s <BOOL> | Assume input is unsorted (default: false) |
| --route-steps <TIME> | Load routes for the next number of seconds ahead (default: 200) |
| --no-internal-links <BOOL> | Disable (junction) internal links (default: false) |
| --randomize-flows <BOOL> | generate random departure times for flow input (default: false) |
| --max-alternatives <BOOL> | Prune the number of alternatives to INT (default: 5) |
| --remove-loops <BOOL> | Remove loops within the route; Remove turnarounds at start and end of the route (default: false) |
| --repair <BOOL> | Tries to correct a false route (default: false) |
| --repair.to <BOOL> | Tries to correct an invalid starting edge by using the first usable edge instead; (default: false) |
| --repair.from <BOOL> | Tries to correct an invalid destination edge by using the last usable edge instead (default: false) |
| --weights.interpolate <BOOL> | Interpolate edge weights at interval boundaries (default: false) |
| --weights.minor-penalty <FLOAT> | Apply the given time penalty when computing routing costs for minor-link internal lanes (default: 1.5) |
| --with-taz <BOOL> | Use origin and destination zones (districts) for in- and output (default: false) |
| --mapmatch.distance <FLOAT> | Maximum distance when mapping input coordinates (fromXY etc.) to the road network (default: 100) |
| --mapmatch.junctions <BOOL> | Match postions to junctions instead of edges (default: false) |
| --bulk-routing <BOOL> | Aggregate routing queries with the same origin (default: false) |
| --routing-threads <INT> | The number of parallel execution threads used for routing (default: false) |
| --restriction-params | Comma separated list of param keys to compare for additional restrictions |
| --weights.expand <BOOL> | Expand weights behind the simulation's end (default: false) |
| --weights.random-factor <FLOAT> | Edge weights for routing are dynamically disturbed by a random factor drawn uniformly from [1,FLOAT) (default: 1) |
| --routing-algorithm <ALGH> | Select among routing algorithms ['dijkstra', 'astar', 'CH', 'CHWrapper']; (default: dijkstra) |

Table A.8: Processing options for *duarouter* - part 2

| **Option** | **Description** |
|---|---|
| --verbose <BOOL> **or** -v <BOOL> | Switches to verbose output (default: false) |
| --print-options <BOOL> | Prints option values before processing (default : false) |
| --help <BOOL> **or** -? <BOOL> | Prints help screen (default: false) |
| --version <BOOL> **or** -V <BOOL> | Prints the current version (default: false) |
| --xml-validation <STRING> **or** -X <STRING> | Set schema validation scheme of XML inputs ("never", "auto" or "always") (default: auto) |
| --xml-validation.net <STRING> | Set schema validation scheme of SUMO network inputs ("never", "auto" or "always") (default: never) |
| --no-warnings <BOOL> **or** -W <BOOL> | Disables output of warnings (default: false) |
| --aggregate-warnings <INT> | Aggregate warnings of the same type whenever more than INT occur (default : -1) |
| --log <FILE> **or** -l <FILE> | Writes all messages to FILE (implies verbose) |
| --message-log <FILE> | Writes all non-error messages to FILE (implies verbose) |
| --error-log <BOOL> | Writes all warnings and errors to FILE |
| --ignore-errors <BOOL> | Continue if a route could not be build (default : false) |
| --stats-period <INT> | Defines how often statistics shall be printed (default : false) |
| --no-ste-log <BOOL> | Disable console output of route parsing step (default : false) |

Table A.9: Report options for *duarouter*

| **Option** | **Description** |
|---|---|
| --zip **or** -7 | Zip old iterations using 7zip (default: false) |
| --disable-warnings | Disable warnings (default: false) |
| --disable-summary **or** -E | No summaries are written by the simulation |
| --disable-tripinfo **or** -T | No tripinfos are written by the simualtion |
| --vehroute-file <F> **or** -x <F> | Choose the format F (none, routesonly, detailed) of the route file |
| --output-lastRoute **or** -z | Output the last routes (default: false) |
| --keep-allroutes **or** -K | Save routes with near zero probability (default: false) |
| --clean-alt | Whether old rou.alt.xml files shall be removed |
| --router-verbose | Let duarouter print some statistics |
| --eco-measure <M> **or** -Q <M> | Define eco-measure (M) - CO, CO2, PMx, HC, NOx, fuel,noise |
| --tripinfo-filter <FILTER> | Filter tripsinfo attributes (default: none) |

Table A.10: Ouput options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --net-file <FILE> **or** -n <FILE> | Sumo network (**mandatory**) |
| --trips <FILE> **or** -t | Trips in step 0 |
| --routes <FILE> **or** -r | Routes in step 0 |
| --flows <FILE> **or** -f | Flows in step 0 |
| --additional <FILE> **or** -+ <FILE> | Additional files (default:) |
| --districts <FILE> **or** -D <FILE> | Use districts as sources and targets (default: None) |
| --path <PATH> **or** -p <PATH> | Path to binaries |

Table A.11: Input options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --begin <TIME> **or** -b <TIME> | Set simulation/routing begin (default:0) |
| --end <TIME> **or** -e <TIME> | Set simulation/routing end (default:None) |
| --route-steps <STEPS> **or** -R <STEPS> | Sets simulation route steps (default:200) |
| --aggregation <PERIOD> **or** -a <PERIOD> | Sets main weights aggregation period (default:200) |
| --time-to-teleport <TIME> | Delay before blocked vehicles are teleported (default:300) |
| --time-to-teleport.highways <TIME> | Teleport delay on wrong highway lanes (default:0) |
| --first-step <STEP> **or** -f <STEP> | First DUA step (default: 0) |
| --last-step <STEP> **or** -l <STEP> | Last DUA step (default: 50) |
| --convergence-iterations <CONVIT> | Number of iterations to use for convergence calculation (default: 10) |
| --max-convergence-deviation <CONDEV> | Maximum relative standard deviation in travel times (default: none) |
| --addweights <WEIGHTS> **or** -J <WEIGHTS> | Additional weights for duarouter |
| --max-alternatives <MAX_ALT> | Prune the number of alternatives to INT (default: 5) |

Table A.12: Configuration options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --ga <ALFA> **or** -A <ALFA> | Sets Gawron's Alpha (default: 0.5) |
| --gBeta <BETA> **or** -B <BETA> | Sets Gawron's Beta (default: 0.5) |
| --logit | Use the logit model for route choice (default: false) |
| --logitbeta <BETA> **or** -g <BETA> | Use the c-logit model for route choice. logit model when beta = 0 (default: 0.15) |
| --logitgama <GAMA> **or** -g <GAMA> | Use the c-logit model for route choice (default: 0.1) |
| --logittheta <THETA> **or** -g <THETA> | Parameter to adapt the cost unit(default: none) |
| --external-gawron **or** -M | Use the external gawron calculation (default: none) |
| --routing-algorithm <ALGTHM> | Select the routing algorithm (default: dijkstra) |

Table A.13: Algorithm's options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --inc-start <INCSTART> | Start for incrementing scale |
| --inc-max <INCMAX> | Maximum for incrementing scale |
| --inc-base <INCBASE> | Give the incrementing scale. (default: -1) |
| --incrementation <INC> | Give the incrementation (default: 1) |
| --time-inc <TIME> | Give the time incrementation (default: 0) |

Table A.14: Incremental assignment options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --mesosim **or** -m | Whether mesosim shall be used (default:false) |
| --meso-junctioncontrol **or** -j | Enable mesoscopic traffic light and priority junction hanldle |
| --meso-multiqueue **or** -q | Enable multiple queues at edge ends |
| --meso-recheck <DELAY> | Delay before checking whether a jam is gone. |

Table A.15: Mesoscopic simulation options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --help **or** -h | Show help message |
| --absrand **or** -y | Use current time to generate random number (default: false) |
| --no-internal-link **or** -I | Not simulate internal link (default: false) |
| --eager-insert | Eager insertion tests on simulation |
| --cost-modifier {grohnde,isar,None} | Whether to modify link travel costs of the given routes |
| --continue-on-unbuild **or** -C | Continues on unbuild routes (default: false) |
| --skip-first-routing | Run simulation with demands before first routing (default: false) |
| --calculate-oldprob **or** -N | calculate the old route probabilities with the free-flow travel times when using the external gawron |
| --weight-memory | Smooth edge weights across iterations |
| --pessimism <PESS> | give traffic jams a higher weight when using option |
| --binary | Use binary format for intermediate and resulting route |

Table A.16: Processing options for *duaIterate.py*

| **Option** | **Description** |
|---|---|
| --net-file <FILE>**or** -n <FILE> | Sumo network (**mandatory**) |
| --trips <FILE> **or** -t <FILE> | Trips in step 0 (**mandatory**) |
| --frequencies <TIME> **or** -f <TIME> | Set the frequencies to iterate over (**mandatory**) |
| --additional <FILE> **or** +- <FILE> | Additional files |
| --weight-file <FILE> **or** -F <FILE> | Load edge/lane weights from file |
| --path <PATH> **or** -p <PATH> | Path to binaries |

Table A.17: Input options for *one-shot.py*

| **Option** | **Description** |
|---|---|
| --with-warnings **or** -W | Enables warnings (default: false) |
| --disable-summary **or** -E | No summaries are written by the simulation |
| --disable-tripinfo **or** -T | No tripinfos are written by the simulation |
| --last-routes **or** -L | Only save the last routes in the vehroute-output |
| --with-exittime **or** -x | Writes exits times for all edges |
| --route-sorted **or** -s | Sorts the output by departure time |

Table A.18: Output options for *one-shot.py*

| **Option** | **Description** |
|---|---|
| --help **or** -h | Show help message |
| --begin <TIME> **or** -b <TIME> | Sets routing begin |
| --end <TIME> **or** -e <TIME> | Sets routing end |
| --route-steps <TIME> **or** -e <TIME> | Set simulation route steps |
| --aggregation <TIME> **or** -e <TIME> | Set main weights aggregation period |
| --adaptation-interval <I> **or** -i <I> | Set edge weight adaptation interval |
| --mesosim **or** -m | Whether mesosim shall be used |
| --with-taz **or** -w | Whether districts shall be used |
| --routing-algorithm <A> **or** -A <A> | Type of routing algorithm |
| --rerouting-explicit <R> **or** -r <R> | Define the ids of the vehicles that should be re-routed |
| --cost-modifier <MODIFIER> | Whether to modify link travel costs of the given routes |

Table A.19: Processing options for *one-shot.py*

# Appendix B

# Porto TAZ zones

| ID | Zone | ID | Zone | ID | Zone | ID | Zone |
|---|---|---|---|---|---|---|---|
| 1 | Aliados | 31 | Monte Burgos | 61 | Vilarinha | 104 | Marginal |
| 2 | Batalha | 32 | Carvalhido | 62 | Antunes Guimarães | 107 | Via Rápida |
| 3 | Guindais | 33 | 9 de Abril | 63 | Foco | 108 | Sra. Hora Oeste |
| 4 | Sé | 34 | Campo Lindo | 64 | Gomes da Costa | 111 | Padrão |
| 5 | Mouzinho SIlveira | 35 | Salgueiros | 65 | Pasteleira | 112 | Via Norte |
| 6 | Ribeira | 36 | Covelo | 66 | Império | 113 | S. Mamede |
| 7 | Cordoaria | 37 | João Pedro Ribeiro | 67 | Boavista Oeste | 114 | A3/A4 |
| 8 | Alfândega | 38 | Bonjardim | 68 | Requesende | 115 | EN 115 |
| 9 | H.S. António | 39 | Monte Cativo | 69 | Zona Industrial | 116 | Rebordões |
| 10 | Carlos Alberto | 40 | Constituição | 70 | Viso | 117 | EN 15 |
| 11 | Cedofeita | 41 | Vale Formoso | 71 | Santa Luzia | 118 | IC 29 |
| 12 | Trindade | 42 | Oliveira Monteiro | 72 | Miosótis | 119 | EN 209 |
| 13 | Bolhão | 43 | Mouzinho Albuquerque | 73 | Ameal | 120 | EN108 |
| 14 | Alvares Cabral | 44 | Bessa | 74 | Alameda 25 Abril | 126 | A44 |
| 15 | Campo 24 Agosto | 45 | Sidónio Pais | 75 | Paranhos | 129 | A1 |
| 16 | Bonfim | 46 | Prelada | 76 | Pólo Universitário | 149 | A20 |
| 17 | Bonjóia | 47 | Av. França | 77 | Areosa | 155 | Paranhos |
| 18 | Noeda | 48 | D. Pedro V | 78 | Fernão Magalhães | 158 | Polo Universitário |
| 19 | Colégio Órfãos | 49 | Arrábida | 79 | S. João Deus | 159 | Bonfim |
| 20 | Soares Reis | 50 | CDUP | 80 | Contumil | 160 | Afurada |
| 21 | Fontainhas | 51 | Venezuela | 81 | S. Roque | 162 | Canidelo |
| 22 | Santos Pousada | 52 | Agramonte | 82 | Maceda | 162 | Continente |
| 23 | Doze Casas | 53 | Campo Alegre | 83 | Cerco do Porto | 163 | Barrosa |
| 24 | Marquês | 54 | Rodrigues Freitas | 84 | Corujeira | 164 | Av. República |
| 25 | Costa Cabral | 55 | Júlio Dinis | 85 | Freixo | 165 | Gervide |
| 26 | Velásquez | 56 | Restauração | 86 | Azevedo | 166 | Santo Ovídio |
| 27 | Antas | 57 | Fluvial | 87 | Lagarteiro | 167 | N222 |
| 28 | Flores | 58 | Foz Velha | 100 | Ponte do Infante | 168 | S. Lourenço |
| 29 | Joaquim Urbano | 59 | Foz Nova | 101 | Ponte Luís I | 169 | A20/A32 |
| 30 | Ramada Alta | 60 | Parque Cidade | 103 | Matosinhos Marginal | | |

Table B.1: Porto Traffic Assignment Zones

# References


[AAA+17] I. Alam, M. F. Ahmed, M. Alam, J. Ulisses, D. M. Farid, S. Shatabda, and R. J. F. Rossetti. Pattern mining from historical traffic big data. In *2017 IEEE Region 10 Symposium (TENSYMP)*, pages 1–5, 2017.

[Ade17] Adekunle Adebisi. A Review of The Difference Among Macroscopic, Microscopic And Mesoscopic Traffic Models. (December), 2017.

[AFR19] Ishteaque Alam, Dewan Md. Farid, and Rosaldo J. F. Rossetti. The prediction of traffic flow with regression analysis. In Ajith Abraham, Paramartha Dutta, Jyotsna Kumar Mandal, Abhishek Bhattacharya, and Soumi Dutta, editors, *Emerging Technologies in Data Mining and Information Security*, pages 661–671, Singapore, 2019. Springer Singapore.

[BADT+] Moshe E Ben-Akiva, Angus Davol, Tomer Toledo, Haris N Koutsopoulos, Wilco Burghout, Ingmar Andréasson, Tobias Johansson, and Christer Lundin. Calibration and Evaluation of MITSIMLab in Stockholm. Technical report.

[BAR15] J. Barros, M. Araujo, and R. J. F. Rossetti. Short-term real-time traffic prediction methods: A survey. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 132–139, 2015.

[BBB+] Christopher Barrett, Richard Beckman, K Berkbigler, Keith Bisset, Bryan Bush, K Campbell, S Eubank, K Henson, J Hurford, D Kubicek, et al. Transims: Transportation analysis simulation system.

[BHN+95] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035–1042, 2 1995.

[BML92] Ofer Biham, A. Alan Middleton, and Dov Levine. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46(10):R6124–R6127, 6 1992.

[Bur] Wilco Burghout. Hybrid microscopic-mesoscopic traffic simulation. Technical report.

[BY00] Sharon Adams Boxill and Lei Yu. An Evaluation of Traffic Simulation Models for by. 7(2):114, 2000.

[CJN+98] Ruey Long Cheu, Xin Jin, Kim Chwee Ng, yuk. leung. ng, and Dipti Srinivasan. GENETIC ALGORITHM APPROACH IN FRESIM CALIBRATION. 1998.

[Cre79] Michael. Cremer. *Der Verkehrsfluss auf Schnellstrassen : Modelle, Uberwachung, Regelung*. Springer, 1979.

[CSW07] Richard D. Connors, Agachai Sumalee, and David P. Watling. Sensitivity analysis of the variable demand probit stochastic user equilibrium with multiple user-classes. *Transportation Research Part B: Methodological*, 41(6):593–615, 7 2007.

[Daf68] Constantine M. Dafermos. On the existence and the asymptotic stability of solutions to the equations of linear thermoelasticity, 1 1968.

[Dag94] Carlos F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B*, 28(4):269–287, 8 1994.

[Dag95] Carlos F. Daganzo. The cell transmission model, part II: Network traffic. *Transportation Research Part B*, 29(2):79–93, 4 1995.

[Erl] Agner Krarup Erlang. The theory of probabilities and telephone conversations. *undefined*.

[Fag54] Henry Fagin. Urban Traffic—A Function of Land Use. By Robert B. Mitchell and Chester Rapkin. New York 27, Columbia University Press, 1954. xviii, 226 pp. $5.00. *National Municipal Review*, 43(11):608–609, 12 1954.

[FMT01] M. Florian, M. Mahut, and N. Tremblay. A hybrid optimization-mesoscopic simulation dynamic traffic assignment model. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 118–121, 2001.

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 3 1956.

[Gaw98] Christian Gawron. Simulation-Based Traffic Assignment Computing User Equilibria in Large Street Networks. Technical report, 1998.

[Gip81] P. G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B*, 15(2):105–111, 4 1981.

[GKW18] Jason B. Gordon, Haris N. Koutsopoulos, and Nigel H.M. Wilson. Estimation of population origin–interchange–destination flows on multimodal transit networks. *Transportation Research Part C: Emerging Technologies*, 90:350–365, 5 2018.

[Gol89] David Goldberg. *Genetic algorithms in search optimization and machine learning*. 1989.

[GP10] Shivam Gupta and Transportation Planning. Stochastic User Equilibrium Shivam Gupta CE682, Infrastructure and Transportation Planning. pages 1–6, 2010.

[GTDS34] B D Greenshields, J T Thompson, H C Dickinson, and R S Swinton. THE PHOTOGRAPHIC METHOD OF STUDYING TRAFFIC BEHAVIOR. *Highway Research Board Proceedings*, 13, 1934.

[HHST02] D. Helbing, A. Hennecke, V. Shvetsov, and M. Treiber. Micro- and macro-simulation of freeway traffic. *Mathematical and Computer Modelling*, 35(5-6):517–547, 2002.

[HMK03] John Hourdakis, Panos G Michalopoulos, and Jiji Kottommannil. A PRACTICAL PROCEDURE FOR CALIBRATING MICROSCOPIC By. *Transportation Research Board*, 1852(January):130–139, 2003.

[HT98] Dirk Helbing and Benno Tilch. Generalized force model of traffic dynamics. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 58(1):133–138, 6 1998.

[HW95] Martin Hilliges and Wolfgang Weidlich. A phenomenological model for dynamic traffic flow in networks. *Transportation Research Part B: Methodological*, 29(6):407–431, 12 1995.

[JMH94] R. Jayakrishnan, Hani S. Mahmassani, and Ta Yin Hu. An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research Part C*, 2(3):129–147, 9 1994.

[KK94] B. S. Kerner and P. Konhäuser. Structure and parameters of clusters in traffic flow. *Physical Review E*, 50(1):54–83, 7 1994.

[Kni24] F. H. Knight. Some Fallacies in the Interpretation of Social Cost. *The Quarterly Journal of Economics*, 38(4):582, 8 1924.

[KPBA96] Asad Khattak, Amalia Polydoropoulou, and Moshe Ben-Akiva. Modeling Revealed and Stated Pretrip Travel Response to Advanced Traveler Information Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1537(1):46–54, 1 1996.

[Kra] Stefan Krauß. Deutsches Zentrum f ur Luft-und Raumfahrt. Technical report.

[KS88] J. Krug and H. Spohn. Universality classes for deterministic surface growth. *Physical Review A*, 38(8):4271–4283, 1988.

[Küh84] Reinhart D. Kühne. MACROSCOPIC FREEWAY MODEL FOR DENSE TRAFFIC - STOP-START WAVES AND INCIDENT DETECTION, 1984.

[Kuh87] R D Kuhne. FREEWAY SPEED DISTRIBUTION AND ACCELERATION NOISE : CALCULATIONS FROM A STOCHASTIC CONTINUUM THEORY AND COMPARISION WITH MEASUREMENTS. *TRANSPORTATION AND TRAFFIC THEORY*, 1987.

[Leb97] J.P. Lebacque. A Finite Acceleration Scheme for First Order Macroscopic Traffic Flow Models. *IFAC Proceedings Volumes*, 30(8):787–792, 6 1997.

[LGT89] D R Leonard, P Gower, and N B Taylor. CONTRAM: STRUCTURE OF THE MODEL. 1989.

[Li87] Wentian Li. Power spectra of regular languages and cellular automata. *Complex Systems*, 1:107–130, 1987.

[LLK98] H. Y. Lee, H. W. Lee, and D. Kim. Origin of synchronized traffic flow on highways and its dynamic phase transitions. *Physical Review Letters*, 81(5):1130–1133, 8 1998.

[LW55] Michael Lightthill and Gerald Whitham. On kinematic waves I. Flood movement in long rivers. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):281–316, 5 1955.

[Man79] Marvin L. Manheim. Fundamentals of Transportation Systems Analysis, Volume 1: Basic Concepts. 1979.

[Mey16] MD Meyer. Transportation planning handbook. 2016.

[Mic] Microsimulation of Traffic Flow: Onramp.

[MLCY04] Qiang Meng, Der-Horng Lee, Ruey Long Cheu, and Hai Yang. Logit-Based Stochastic User Equilibrium Problem for Entry-Exit Toll Schemes. *Journal of Transportation Engineering*, 130(6):805–813, 11 2004.

[MLW12] Qiang Meng, Zhiyuan Liu, and Shuaian Wang. Optimal distance tolls under congestion pricing and continuously distributed value of time. *Transportation Research Part E: Logistics and Transportation Review*, 48(5):937–957, 9 2012.

[Nas50] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1 1950.

[New61] G. F. Newell. Nonlinear Effects in the Dynamics of Car Following. *Operations Research*, 9(2):209–229, 4 1961.

[New93] G. F. Newell. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B*, 27(4):281–287, 8 1993.

[NEX] NEXT | Armis ITS.

[NH16] Kapileswar Nellore and Gerhard Hancke. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. *Sensors*, 16(2):157, 1 2016.

[NS92a] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12), 1992.

[NS92b] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12), 1992.

[Pap83] M. (Markos) Papageorgiou. *Applications of automatic control concepts to traffic flow modeling and control*. Springer-Verlag, 1983.

[PAY71] HAROLD J PAYNE. MODELS OF FREEWAY TRAFFIC AND CONTROL. *MATHEMATICAL MODELS OF PUBLIC SYSTEMS*, 1971.

[Phi79] Warren F. Phillips. A kinetic model for traffic flow with continuum implications. *Transportation Planning and Technology*, 5(3):131–138, 1 1979.

[Pla] Platform: C-Roads.

[PRK11] L. S. Passos, R. J. F. Rossetti, and Z. Kokkinogenis. Towards the next-generation traffic simulation tools: a first appraisal. In *6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*, pages 1–6, 2011.

[RBB+02] Rosaldo JF Rossetti, Rafael H Bordini, Ana LC 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, 2002.

[Ric56] Paul I. Richards. Shock Waves on the Highway. *Operations Research*, 4(1):42–51, 2 1956.

[RL14] Rosaldo JF Rossetti and Ronghui Liu. *Advances in artificial transportation systems and simulation*. Academic Press, Cambridge, MA, 2014.

[RLCB02] Rosaldo JF Rossetti, Ronghui Liu, Helena BB Cybis, and Sergio Bampi. A multi-agent demand model. In *Proceedings of the 13th Mini-Euro Conference and The 9th Meeting of the Euro Working Group Transportation*, pages 193–198, 2002.

[ROB07] Rosaldo JF Rossetti, Eugénio C Oliveira, and Ana LC Bazzan. Towards a specification of a framework for sustainable transportation analysis. In *13th Portuguese Conference on Artificial Intelligence, EPIA, Guimarães, Portugal*, pages 179–190. APPIA, 2007.

[RRR15] Nedal T. Ratrout, Syed M. Rahman, and Imran Reza. Calibration of PARAMICS Model: Application of Artificial Intelligence-Based Approach. *Arabian Journal for Science and Engineering*, 40(12):3459–3468, 12 2015.

[SHCW78] Arthur B Sosslau, Amin B Hassam, Maurice M Carter, and George Wickstrom. QUICK-RESPONSE URBAN TRAVEL ESTIMATION TECHNIQUES AND TRANSFERABLE PARAMETERS. USER'S GUIDE. 1978.

[SRM+16] M. Sandim, R. J. F. Rossetti, D. C. Moura, Z. Kokkinogenis, and T. R. P. M. Rúbio. Using gps-based avl data to calculate and predict traffic network performance metrics: A systematic review. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1692–1699, 2016.

[TARO10] I. J. P. M. Timóteo, M. R. Araújo, R. J. F. Rossetti, and E. C. Oliveira. Trasmapi: An api oriented towards multi-agent systems real-time interaction with multiple traffic simulators. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1183–1188, 2010.

[The61] Henri Theil. *Economic forecasts and policy,*. North-Holland Pub. Co., Amsterdam, 2d rev. ed. edition, 1961.

[USWC07] Kenetsu Uchida, Agachai Sumalee, David Watling, and Richard Connors. A study on network design problems for multi-modal networks by probit-based stochastic user equilibrium. *Networks and Spatial Economics*, 7(3):213–240, 9 2007.

[WAR52] J G WARDROP. ROAD PAPER. SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 5 1952.

[WB13] Adriaan Hendrik van der Weijde and Vincent A.C. van den Berg. Stochastic User Equilibrium Traffic Assignment with Price-sensitive Demand: Do Methods matter (much)? *Tinbergen Institute Discussion Papers*, 2013.

[Wei97] Edward Weiner. URBAN TRANSPORTATION PLANNING IN THE UNITED STATES: AN HISTORICAL OVERVIEW FIFTH EDITION. Technical report, 1997.

[Whi99] G. B. Whitham. *Linear and Nonlinear Waves*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 6 1999.

[Wie74] R Wiedemann. *Simulation des Straßenverkehrsflusses*. Karlsruhe: Univ., Inst. fur Verkehrswesen., Fourth edition, 1974.

[WL80] D A WICKS and E B Lieberman. DEVELOPMENT AND TESTING OF INTRAS, A MICROSCOPIC FREEWAY SIMULATION MODEL, VOLUME 1: PROGRAM DESIGN, PARAMETER CALIBRATION AND FREEWAY DYNAMICS COMPONENT DEVELOPMENT. 10 1980.

[Yan99] Hai Yang. System optimum, stochastic user equilibrium, and optimal link tolls. *Transportation Science*, 33(4):354–360, 11 1999.

[YMB01] Hai Yang, Qiang Meng, and Michael G.H. Bell. Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. *Transportation Science*, 35(2):107–123, 5 2001.

[ZWW+11] Junping Zhang, Fei Yue Wang, Kunfeng Wang, Wei Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey, 12 2011.