

POLITECNICO DI TORINO



MASTER THESIS

Modelling, simulation and design of Electric Car Sharing systems with eC2S

Author:

Alessandro CIOCIOLA

Supervisors:

Marco Mellia
Michela Meo
Luca Vassio

*A thesis submitted in fulfillment of the requirements
for the Master degree in ICT for Smart Societies*

SmartData@Polito
Department of Electronics and Telecommunications

October 14, 2019

Declaration of Authorship

I, Alessandro CIOCIOLA, declare that this thesis titled, “Modelling, simulation and design of Electric Car Sharing systems with eC2S ” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“”

POLITECNICO DI TORINO

Abstract

Faculty of Electronics, Telecommunications and Physics Engineering
 Department of Electronics and Telecommunications

ICT for Smart Societies

Modelling, simulation and design of Electric Car Sharing systems with eC2S

by Alessandro CIOCIOLA

Transportation systems accounts for about a quarter of global energy-related CO₂ emissions, and the whole sector is considered as practically hard to decarbonise due to technological and socio-economic factors. Within the context of cities, shared electric mobility has the potential to play a decisive role for reducing emissions and make urban transportation more efficient. However, designing, planning and operating shared electric mobility systems present a number of challenges, often not present in traditional systems, which need proper identification, measurement, and analysis in order to be addressed. Among those challenges, one of the most critical is represented by charging management and infrastructure. The goal of this Master thesis is to investigate and compare different charging scenarios for Electric Free Floating Car Sharing (EFFCS) systems, with the aid of real world data coming from CS services. Data collection have been performed during 2017 thanks to the implementation of the software UMAP [25], which allowed to create a database of real rentals done by users in 25 cities spread across the EU and North America. These data are used as input to create a configurable demand model for EFFCS systems, composed by a parametric estimation of bookings' inter-arrival times based on Poisson processes, and a non-parametric estimation of OD (Origin-Destination) matrices based on Kernel Density Estimate. The demand model is used as the main input of eC2S [1], a data-driven, discrete-event simulation software for EFFCS systems in Smart Cities implementing specific fleet management and charging infrastructure deployment strategies, whose setting and tuning is critical for the correct implementation, deployment and functioning of EFFCS systems. Three main charging infrastructure scenarios have been considered: fully centralised, fully distributed, hybrid. First, we considered a case study about the city of Turin, and then we extended the analysis on other four cities: Milan, Berlin, New York City and Vancouver. Results show that the presence of a centralised hub in a highly dynamic hotspot, namely where many trips start or end, is beneficial from the mobility viewpoint, as more user requests tend to be satisfied. On the other side, the management of charging processes is expensive because each car needs to be driven to the hub in order to be charged, no matter its position. With a decentralised infrastructure, the increase in management complexity is traded with a much lower operational cost, as stations are widespread around the city. Furthermore, a distributed infrastructure allows for users contribution in charging processes, which can be primarily important for a further reduction of operational cost without hardly impacting users comfort.

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

Declaration of Authorship	iii
Abstract	vii
Acknowledgements	ix
1 Introduction	1
1.1 The energy transition	1
1.2 The role of transportation	2
1.3 Thesis presentation and motivation	5
2 Background	7
2.1 Transport modelling background	7
2.1.1 Introduction to transport modelling	7
2.2 Mathematical background	9
2.2.1 Stochastic processes	9
2.2.2 Poisson processes	9
2.2.3 Kernel Density Estimate	10
2.3 ICT background	12
2.3.1 MongoDB	13
2.3.2 Python	13
2.4 Related works	15
2.4.1 Poisson processes and KDE in geospatial analysis	15
2.4.2 EV, CS, ECS	16
3 Methodologies	23
3.1 Software description	23
3.1.1 Retrieval	24
3.1.2 Preprocessing	25
3.1.3 Loading	25
3.1.4 DataStructures	25
3.1.5 SimulationInput	26
3.1.6 Simulation	27
3.1.7 SimulationOutput	28
3.2 Demand modelling	30
3.2.1 Time estimation	30
3.2.2 Spatial estimation	31
3.3 Simulation assumptions	32
4 Model validation	33
4.1 Introduction	33
4.2 Time analysis	33
4.3 Spatial analysis	34

5	First case study: city of Turin	41
5.1	Single charging hub scenario	42
5.1.1	Parameters analysis	42
5.1.2	In-depth analysis	46
5.2	Distributed infrastructure scenario	54
5.2.1	Parameters analysis	54
5.2.2	In-depth analysis	59
5.3	Charging hub and distributed infrastructure scenario	64
5.3.1	Parameters analysis	64
6	Second case study: Cities comparison	69
6.1	Centralised infrastructure scenario	69
6.2	Distributed infrastructure scenario	72
6.3	Charging hub and distributed infrastructure scenario	81
7	Conclusions	89
A	Model validation for different cities	91
A.1	Milano	91
A.2	Berlin	93
A.3	New York City (Brooklyn)	95
A.4	Vancouver	97
	Bibliography	99

List of Figures

2.1	KDE kernels	11
2.2	KDE kernels	12
2.3	KDE kernels	13
2.4	KDE 2d 1	14
2.5	KDE 2d 2	15
2.6	Car-sharing taxonomy	17
2.7	Car-sharing service modes	18
2.8	engines and modes	19
2.9	optimisation goals and modes	19
2.10	time horizon and modes	20
2.11	optimisation goals and modes	21
4.1	Requests profile by hour, Turin	34
4.2	QQ plot	35
4.3	Events count error	35
4.4	Spatial error heatmap	37
4.5	Events percentages	39
4.6	Events percentages	39
5.1	Unsatisfied percentage β	43
5.2	Relocation cost β	43
5.3	Unsatisfied percentage n cars	44
5.4	Unsatisfied percentage n cars	44
5.5	Relocation cost n cars	45
5.6	Events profile, min unsatisfied	46
5.7	Events profile, min cost	47
5.8	Hub location	48
5.9	Events percentages	49
5.10	Hourly boxplot of number charging events managed by the system	49
5.11	Hourly boxplot for number of cars charging	50
5.12	Fleet status in time	50
5.13	Hourly boxplot for relocation cost expressed in hours	51
5.14	Histogram of charging events start timestamps	51
5.15	Clorophlet map of charging requests locations	52
5.16	Hourly boxplot for unsatisfied requests	52
5.17	Histogram of unsatisfied requests timestamps	53
5.18	Clorophlet map of unsatisfied requests locations	53
5.19	Events profile, no users, best centralised configuration	55
5.20	Events profile, no users, best centralised configuration	55
5.21	Percentage of unsatisfied events, best case configuration, varying willingness	56
5.22	Percentage of unsatisfied events, best case configuration, varying willingness	57

5.23	Percentage of unsatisfied events, best case configuration, varying willingness	57
5.24	Percentage of unsatisfied events, best case configuration, varying willingness	58
5.25	Events profile, best satisfied demand configuration	58
5.26	Cps location	60
5.27	Events percentages	61
5.28	Hourly boxplot of number charging events	61
5.29	Hourly boxplot for number of cars charging in events managed by the system	62
5.30	Hourly boxplot for number of cars charging in events managed by users	62
5.31	Fleet status in time	63
5.32	Hourly boxplot for relocation cost expressed in hours	63
5.33	Percentage of unsatisfied events, best case, varying willingness	65
5.34	Percentage of unsatisfied events, min cost, varying willingness	66
5.35	Relocation cost, min unsatisfied, varying willingness	66
5.36	Relocation cost, best case configuration, varying willingness	67
5.37	Events profile, min unsatisfied	67
6.1	Percentage of unsatisfied demand, varying β , cities comparison	70
6.2	Charging relocation cost, varying β , cities comparison	71
6.3	Percentage of unsatisfied demand, varying n_cars_factor, cities comparison	73
6.4	Charging relocation cost, varying n_cars_factor, cities comparison	74
6.5	Percentage of unsatisfied demand, varying willingness, cities comparison	75
6.6	Charging relocation cost, varying n_cars_factor, cities comparison	76
6.7	Percentage of unsatisfied demand, varying β , cities comparison	77
6.8	Charging relocation cost, varying β , cities comparison	78
6.9	Percentage of unsatisfied demand, varying n_cars_factor, cities comparison	79
6.10	Charging relocation cost, varying n_cars_factor, cities comparison	80
6.11	Percentage of unsatisfied demand, varying willingness, cities comparison	82
6.12	Charging relocation cost, varying willingness cities comparison	83
6.13	Percentage of unsatisfied demand, varying β , cities comparison	84
6.14	Charging relocation cost, varying β , cities comparison	85
6.15	Percentage of unsatisfied demand, varying n_cars_factor, cities comparison	86
6.16	Charging relocation cost, varying n_cars_factor, cities comparison	87
A.1	Requests profile by hour, Milano	91
A.2	QQ plot, Milano	92
A.3	Spatial error heatmap, Milano	92
A.4	Requests profile by hour, Berlin	93
A.5	QQ plot, Berlin	93
A.6	Spatial error heatmap, Berlin	94
A.7	Requests profile by hour, NYC	95
A.8	QQ plot, NYC	95
A.9	Spatial error heatmap, NYC	96
A.10	Requests profile by hour, Vancouver	97

A.11 QQ plot, Vancouver	97
A.12 Spatial error heatmap, Vancouver	98

List of Tables

List of Abbreviations

GHG	GreenHouse Gas
PV	PhotoVoltaic
LNG	Liquefied Natural Gas
LPG	Liquefied Petroleum Gas
RES	Renewable Energy Sources
DER	Distributed Energy Resources
EV	Electric Vehicles
CS	Car Sharing
FFCS	Free Floating Car Sharing
ECS	Electric Car Sharing
EFFCS	Electric Free Floating Car Sharing
MILP	Mixed Integer Linear Programming
SOC	Status Of Charge
iid	independent and identically distributed

List of Symbols

D	Transport demand
V	Traffic volumes
Q	Operating capacity
M	Management system
S	Level of service
A	Set of activities
I	Investments
α	Lower charging threshold.
β	Upper charging threshold.
$N_{i,t,\%}$	Percentage of trips at a given time frame.
$N_{i,t,\%}^o$	Percentage of trips at a given time frame for which the zone i is the origin.
$N_{i,t,\%}^d$	Percentage of trips at a given time frame for which the zone i is the destination.

For/Dedicated to/To my...

Chapter 1

Introduction

1.1 The energy transition

Climate change and the energy sector During the last decades, climate change issues have been intensively studied by scientific communities worldwide. More than 97% of related published works agrees that, since 1950s, Earth's climate is warming at an anomalously sharp rate, and that this is extremely likely due to human activities [47] [48] [57] [66]. Energy production is largely regarded as the major source of greenhouse gas (GHG) emissions [40] [41] [22] [24], and acting on it is thence a priority for the scope of a truly sustainable global development. Although many efforts are conducted by institutions and organisations worldwide, reshaping the energy system is a challenge with high degrees of complexity from every point of view: key conceptual pillars as affordability, reliability and sustainability are interlinked through highly dynamic tradeoffs requiring a deep and comprehensive approach in terms of policy, design, planning and deployment. For example, wind turbines and solar photovoltaic (PV) constitute a major source of affordable, low-emissions electricity into the picture, but create additional requirements for the reliable operation of power systems. Another example can be found in the movement towards a more interconnected global liquefied natural gas (LNG) market, which intensifies competition among suppliers, while changing the way countries need to think about managing potential shortfalls in supply [23]. Furthermore, the projected energy demand for next decades presents a trend of growth, mostly due to the demographic and economic booms of countries like India and China [58]. Although this rise might further complicate the scenario and will be likely accompanied by changes and shifting in global energy trade flow, the trend might be mitigated by advances in technology, energy efficiency and careful policies having real sustainability concerns. Also, new ways of sourcing energy are also visible at local level thanks to digitalisation and increasingly cost-effective renewable energy technologies (RES), which allow distributed and community-based energy provision models to be increasingly central to the prospects for meeting many of the world's sustainable development goals [24].

Electricity and Smart Grids Among the global final consumption forms of energy, electricity is going to play a special role, as its share in global final consumption is approaching 20%, and is set to rise even further [23]: electricity is increasingly important in economies that are relying more on lighter industrial sectors, such as digital technologies and services, and cost reductions due to technology improvements are leading to rapid growth in RES. This puts the power sector in the vanguard of emissions reduction efforts but requiring the entire system to operate differently in order to ensure reliable supply and meet the increasing demand. Traditional power systems are characterised by:

- Carbon-intensive processes.
- Bulk generation and transmission.
- Centralised control.
- Unidirectional flows.

These systems are increasingly being considered outdated as they unfit to deal with growing demand, have slow response time in case of faults and network congestion, are not flexible enough for dynamic and local demand-supply matching, and present barriers to easily integrate RES.

On the contrary, desirable properties for next-future power systems are:

- Decarbonisation.
- Decentralisation and Distributed Energy Resources (DER) integration.
- Bidirectional flow of power and information.
- Security enhancement.
- Losses and faults reduction.

This transition brings a lot of added complexity and challenges at every level of the electricity life cycle (production, transmission, trading, distribution, storage and consumption), but has also concrete potential of creating significant ecological and economical value for the users and the whole society, providing more sustainable, usable, convenient, and efficient services without losing the benefits of existing infrastructures.

1.2 The role of transportation

Transport sector in the energy transition Transportation systems form a very complex network accounting for about a quarter of global energy-related CO₂ emissions but, in contrast to the electricity sector, transportation has not made significant progress in order to decrease these numbers [37]. A big percentage of mileage deriving from transport all over the world is also used inefficiently, resulting in needless energy consumption and emissions. Much of the transportation sector is considered as practically hard to decarbonise because of technological factors, such the high energy density of fuels required for many types of vehicles, which constrains low-carbon alternatives, and socio-demographic factors deriving from the highly direct impact of transport policies on end-users, resulting in higher controversies and more difficulties in deploying significant changes. Passenger and freight transportation are each responsible for about half of transport GHG emissions taking into account every transport mode (road, rail, water, air) but, at present, more than two-thirds of transportation emissions are from road travel [64].

Strategies to reduce GHG emissions from transportation may be grouped in 4 categories [64]:

- Reduce transport activity
 - **Improve transportation data**
 - **Modelling demand**

- **Shared mobility**
 - Freight routing and consolidation
 - Replace or reduce demand
- Increase vehicle efficiency
 - Improve design and vehicles engineering
 - Autonomous vehicles
- Shift to greener fuels
 - Alternative fuels
 - **Electric vehicles**
- Shift to lower-carbon modes
 - Improve understanding of users' modal choice
 - Improve low-carbon options

For the scope of this thesis, we focus on the strategies highlighted in bold.

Transportation data Many areas of transportation lack data, and decision-makers often plan infrastructure and policy based on uncertain information. In recent years, new types of technologies and data collection methods based on ICTs have become available, allowing for the development of more robust mathematical models, often based on statistical and machine learning techniques, and consequently more informed decision tools.

Mobility demand Modelling mobility demand, shaping trips characteristics and improving traditional transport models can significantly impact the planning phase of new infrastructure and systems, discourage sprawl and help to define more appropriate transportation links. It would be very useful for transport planners to access advanced information tools able to help in understanding whether a mobility system has the potential to take customers away from low-carbon transit modes, while keeping satisfied the mobility demand and increasing the efficiency in the usage of vehicles.

Electric Vehicles A particular role in this framework, which involves both the energy sector and the transport sector, is represented by Electric Vehicles (EVs). EVs might play an important role in the grid of the future, providing more flexibility to the power system and contributing to integrate renewable sources of energy. This last aspect is crucial, because even if replacing internal combustion engines would immediately bring cleaner air, especially in cities, the overall CO₂ emissions would not be significantly reduced without a greener energy mix in the grid [49]. A significant part of the challenges underpinning a wide EVs adoption is related with their charging networks (EVCNs), whose related infrastructures and market are still in active development, and with the range anxiety of users, which tends not to perceive it as accessible as traditional solutions. A lack of technologies and protocols for communication and payments is used nowadays in EVCNs, and interoperability is one of the major issues, together with security and user privacy. Furthermore, the fact the EVs are not yet widely spread forces research to work mostly on future projections and simulations, which in turn need reliable data to be built correctly and accurately.

Car sharing Modern mobility patterns of people and freights are characterized by a strong dominance of road transports. This phenomenon encases in itself many issues related to its negative environmental impacts, and to its social and economic cost. One of the most recent solution designed to face these problems in urban contexts is represented by car sharing systems. Their model of car rental proposes people to rent cars for short periods of time, often by the hour, park them and pay following provider-specific rules. The main operational features of car sharing can be summarized as follows:

- Short-term rentals: Car sharing charges by the hour, and usually by the mile as well, making short trips cost effective.
- Self-accessing: Car sharing allows members to reserve a car online or by telephone, open the doors with their own electronic key, and return the car without ever dealing with anyone else. This allows car-sharing to provide service more efficiently than rental car agencies, eliminating the time-consuming hassle of the check-in process.
- Full service: Car sharing services include fuel, maintenance and insurance, and some providers offer reserved parking in some spot. Avoiding the hassles of vehicle ownership is one of the key attractions of car sharing, as members "out-source" the chores that go along with ownership.
- Different vehicles for different uses: Most car sharing operators have a varied fleet (2-seats car, 4/5-seats car, scooter, ...).

In the passenger sector, shared mobility is certainly disrupting the way people travel and think about vehicle ownership, but it is largely unclear if its impact will be in fact positive or not. For example, shared cars can actually cause more people to travel by car, as opposed to using public transportation, similarly as on-demand taxi services add mileage when traveling without a customer. On the other hand, shared mobility can lead to higher utilization of each vehicle, which means a more efficient use of materials, and the use of newer and more efficient vehicles, especially electric ones, could increase with vehicle sharing concepts.

The biggest Car Sharing (CS) and Electric Car Sharing (ECS) provider in the world, the company car2go, which is also the source of the data used in this thesis, has publicly set itself the goal of making the future of its system electric [34]. The company claims that ECS has a decisive role in the development of electric mobility for five reasons [34]:

- ECS solves the chicken-and-egg problem regarding the development of a charging infrastructure.
- CS reduces people's reservations about using electric mobility.
- Everyday operation of car2go's ECS proves electric mobility is suitable for high intensity usage.
- ECS immediately improves the air quality in the cities
- ECS is the perfect testing ground and experimental field for electric mobility of the future

Although these statements might be considered biased due to the business goals of the company, the justifications behind them are shared by both academics, businesses and institutions [43] [55] [27] [26]. There is therefore a fairly spread agreement on the fact that the deployment and increased adoption of ECS services might bring advantages for a more effective deployment of electric mobility. As remarked previously, these advantages can be defined in terms of ICT integration, data collection and analysis, charging infrastructure deployment, immediate air-quality improvements in cities and easier connection with RES.

1.3 Thesis presentation and motivation

The goal of this Master thesis is the implementation of eC2S : a data-driven, discrete-event simulator for electric car sharing systems in Smart Cities. eC2S is able to model car sharing demand from data coming from real car sharing systems and run parametric simulation campaigns, providing also analysis and visualisation tools useful to compare different charging scenarios and fleet management strategies. Input data have been collected during 2017 using the software UMAP, whose main purpose is to retrieve data available on the web in order to extract insights about users mobility habits in different cities. UMAP is focused on car sharing systems, but it can also integrate other city-specific data sources. In particular, the database used for this thesis consist in a trace of real origin-destination bookings done by users in 25 cities spread across EU and North America. Those data are used to create a configurable demand model for ECS, composed by a parametric estimation of bookings' inter-arrival times based on Poisson processes, and a non-parametric estimation of OD (Origin-Destination) distribution based on Kernel Density Estimate. In chapter 4 we performed a quantitative validation of this approach for a more precise assessment about its pros and cons. The demand model is used as the main input of simulator, together with static parameters such as: fleet size, booking requests' arrival rate and number of charging poles. Other more complex parameters are meant to simulate different poles placement policies and different charging strategies. We ran a simulation campaign over 5 cities in order to get insights about charging from the points of view of system operators, users and city planners. From a functional point of view, this thesis tries to answer the following points:

- Are we able to build a robust demand model from available EFFCS data in different cities? If so, which are the pros and cons of the model?
- Can we use the demand model to build a configurable event-based simulator for EFFCS systems, able to implement different kind of charging and fleet management strategies?
- Can we fairly compare implemented charging and relocation strategies for EFFCS fleet management a given city?
- Can we fairly compare simulated EFFCS systems in different cities?

More specifically, the research questions we pose about EFFCS systems are the following:

- How do a centralised charging infrastructure compare to a distributed one in terms of service quality and operational cost, in a given city?

- How do system parameters such as fleet size and charging thresholds impact service quality and operational cost, in a given city?
- How do different cities respond to similar simulation settings?

The thesis is organised as follows. In chapter 2 we will introduce background information in transport modelling, mathematical tools and ICTs. In the same chapter, we will also review existing scientific literature covering similar topics. In chapter 3 we will introduce the implemented software by describing its architecture and the principles behind demand modelling and simulation. In chapter 4 we will illustrate the details of the implemented demand model, together with a quantitative assessment of its performances. In chapter 5 we will show the functionalities of eC2S by presenting and analysing results of a simulation campaign conducted for the city of Turin. In chapter 6 we will extend the analysis to other four cities, namely Milan, Berlin, Vancouver and New York City, in order to compare results from different spatial and socio-economic contexts. Finally, in chapter 7 we will discuss obtained results, underline the pros and cons of the implemented software and outline possible future research directions.

Chapter 2

Background

2.1 Transport modelling background

2.1.1 Introduction to transport modelling

This subsection is mainly based on the material provided by the comprehensive book on the topic [38]. The huge growth of transport activity during last century comes with a general increase of issues such as congestion, delays, accidents and environmental problems reaching a step well beyond sustainable limits [64]. In some areas, for example, economic growth seems to have generated levels of demand exceeding the capacity of many transport facilities. Contrarily, in other areas, under-investments in some transport modes resulted in a fragile supply system undergoing many critical service fault. An effort in improving transport systems planning and design is therefore more and more required in transport design and planning, as those issues tend to become more and more complex and not likely disappear in the next future. The characteristics of transport demand and supply depend on a large number of factors whose complexity in interactions is hardly describable, and for this reasons it is necessary to create mathematical representations of transport systems, in order to better study their properties and implement more intelligent management strategies. Transport modelling is a key element for improved decision making and planning in the transport field, although other factors such as administrative practises, institutional framework and businesses role play a different yet important role.

Transport demand is derived from activities taking place in space and time, it is highly qualitative and differentiated, and consequently needs a wide range of requirements and services in order to be matched. Explicit mathematical treatment of time and space is unavoidable and strongly desirable for more realistic and reliable models. The lack of coordination in one of the two dimensions may strongly affect the supply-demand equilibrium: for example, a taxi service may be demanded unsuccessfully in a part of a city while in other areas various taxis may be plying for passengers, as well as the concentration of demographic and economic activity in some areas may lead to the justification of a dedicated high-quality transit system. These strong dynamicity of elements composing the transport demand makes it difficult to model and forecast it, as approaches based on average demand might be not viable and too simplistic: peak and off-peak variations in demand are in fact a central problem in transport modelling and planning.

Transport supply is in broad terms the set of fixed assets, named infrastructures, and mobile units, named vehicles, whose combination makes the movement of people and goods possible. With the notable exception of many rail systems, supplier of infrastructures and vehicles are different actors, leading to increased complexity

of the interactions between government authorities, construction companies, developers, transport operators, travellers, shippers, and the final users. Provision of infrastructure and the relative economic investments are challenging to adjust to demand, being large and lumpy projects involving big amounts of resources, long deployment time, and eventually a good deal of disruption for users, non-users and the environment. As already said, an additional element of distortion is provided by the number of side-effects associated with the production of transport services: accidents, pollution and environmental degradation in general. These effects are seldom internalised; the user of the transport service rarely perceives nor pays for the costs of cleaning the environment or looking after the injured in transport related accidents. Internalising these costs could also help to make better decisions and to improve the allocation of demand to alternative modes.

In general terms, the role of transport planning is to ensure the satisfaction of a demand D for people and goods movement, with different space/time constraints and trip purposes, given a transport system. The main element of a transport system are therefore:

- Infrastructure
- Vehicles
- Management system
- Transport modes
- Operators

Let's now formulate a simple, high-level mathematical description of the dependencies between the element of demand-supply equilibrium in transport as in [38]. Let's consider the following variables:

- Set of traffic volumes on a network V .
- Operating transport capacity Q .
- Management system M .
- Level of service S , namely measurable quality indicators of the transport system.
- Set of human activities A , such as work, leisure, necessities.
- Investment programs I .

Starting from those definition we can write the following high-level relations:

$$S = f\{Q, V, M\} \quad (2.1)$$

$$Q = f\{I, M\} \quad (2.2)$$

$$D = f\{S, A\} \quad (2.3)$$

Combining equations 2.1 and 2.3 for a fixed activity system one would find the set of equilibrium points between supply and demand for transport. But then again,

the activity system itself would probably change as levels of service change over space and time. Therefore one would have for example two different sets of equilibrium points: short-term and long-term ones. The task of transport planning is to forecast and manage the evolution of these equilibrium points over time so that social welfare is maximised. This is, of course, not a simple task: modelling these equilibrium points should help to understand this evolution better and assist in the development and implementation of management strategies M and investment programs I .

2.2 Mathematical background

2.2.1 Stochastic processes

A stochastic process can be defined as a collection of Random Variables (RV) indexed by a mathematical set, called index set, which is often interpreted as time. Each RV composing the process takes values from the same mathematical space known as the state space. An increment is the amount that a stochastic process changes between two consecutive index values. A stochastic process can clearly have many possible outcomes, and a single outcome of a stochastic process is called a sample function or realisation. Common examples of stochastic processes include:

- **Bernoulli process:** discrete sequence of independent and identically distributed (iid) RVs, where each RV takes two possible values, one with probability p and the other with probability $1 - p$.
- **Wiener process:** continuous stochastic process with stationary and independent increments that are distributed as zero mean, unitary variance Gaussian RVs.
- **Poisson process:** stochastic process representing intuitively the number of events happening in a certain time frame. These events are often referred to as points, and the number of points of the process located in the interval from zero to some given time is a Poisson RV. A Poisson RV is defined by the following probability density function, representing the probability of k events in a given time interval:

$$P(k) = e^{-\lambda} \frac{\lambda^k}{k!}$$

2.2.2 Poisson processes

For a more complete treatment of Poisson processes we refer to [2] and in particular [63].

In probability, statistics and related fields, a Poisson process is a type of stochastic process, widely used for its simplicity and nice mathematical properties. The Poisson process depends on a single parameter, which, depending on the context, may be a constant, a locally integrable function (or a Radon measure in a more general setting). In the first case, the constant, known as the rate or intensity, the resulting stochastic process is called a homogeneous or stationary Poisson process. In the second case, the process is called an inhomogeneous or nonhomogeneous Poisson process.

Homogeneous Poisson process In the particular case for which the parameter is a constant, the whole process can be interpreted as a counting process $N(t)$, representing the total number of occurrences or events that have happened up to and including time t .

A counting process is a homogeneous Poisson counting process with rate $\lambda > 0$ if it has the following three properties:

- $N(0) = 0$
- follows the independent increments property: given a stochastic process X_t , it is said to have independent increments if for every $m \in \mathbb{N}$ and $t_0 < t_1 < \dots < t_m$, the RVs $(X_1 - X_0), (X_2 - X_1), \dots, (X_m - X_{m-1})$ are statistically independent.
- the number of events in any interval of length t is a Poisson random variable with parameter, equal to the mean, λt

The Poisson counting process can also be defined by stating that the time differences between events of the counting process are exponential RVs with mean $1/\lambda$. The time differences between the events or arrivals are known as interarrival times. The previous definition has many important mathematical consequences, among which:

- the number of arrivals in each finite interval has a Poisson distribution;
- the number of arrivals in disjoint intervals are independent random variables.
- the Poisson distribution of the number of arrivals in each interval $(a + t, b + t]$ only depends on the interval's length $b - a$.
- the existence of one point existing in a finite interval does not affect the probability (distribution) of other points existing
- for any finite $t > 0$, the random variable $N(a + t, b + t]$ is independent of t , so the process is also stationary.

Inhomogeneous Poisson process The inhomogeneous or nonhomogeneous Poisson process differs from the homogeneous counterpart for the form of its parameter. In this case, the Poisson parameter is function $\lambda(x)$ defined in the underlying space on which the Poisson process itself is defined. Of our particular interest is the case in which $\lambda(x)$ is a piecewise constant function having a finite number of pieces.

2.2.3 Kernel Density Estimate

Kernel Density Estimate (KDE) is an unsupervised, non-parametric statistical estimator for the Probability Density Function (PDF) of a Random Variable (RV). KDE can be seen as an extension of the concept of histograms, meant to have a better model and representation for the distribution of a data sample.

Histogram A histogram is a function that counts the number of observations that fall into disjoint subsets called bins. Let n be the total number of observations and k be the total number of bins, the histogram h_i meets the following property:

$$n = \sum_{i=1}^k h_i \quad (2.4)$$

However, a major problem with histograms is that the choice of binning might have a disproportionate impact on the resulting visualization. Let's therefore define the univariate KDE and then represent the differences between the two approaches.

Univariate KDE Let (x_1, x_2, \dots, x_n) be a univariate independent and identically distributed (iid) sample of N elements drawn from some distribution with an unknown density f . Its kernel density estimator \hat{f} for f is:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right) \quad (2.5)$$

where K is a non-negative function called kernel, and $h > 0$ is a smoothing parameter called the bandwidth. Equation 2.5 can be also seen as a convolution of the data sample with the kernel function K , smoothed by the bandwidth h . A range of kernel functions are commonly used (uniform, triangular, biweight, triweight, Epanechnikov, normal, and others), while the choice of the bandwidth depends strictly from data. A visual representation of different kernels and bandwidths choices is shown respectively in figures 2.1 and 2.2.

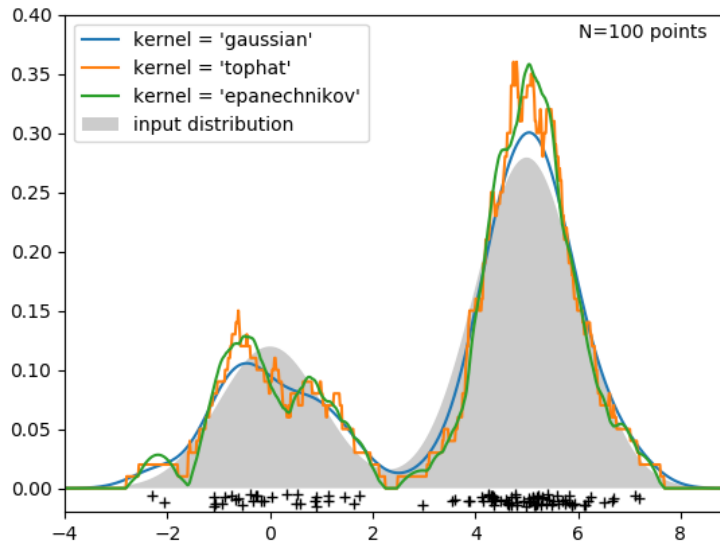


FIGURE 2.1: KDE for a univariate normal distribution using different kernels [3]

In figure 2.3 it is shown a comparison between histograms and a KDE estimation for a bimodal Gaussian distribution. The upper-left chart shows an histogram of the data, while the upper-right chart shows a histogram over the same data, with the bins shifted right. The results of the two visualizations look entirely different,

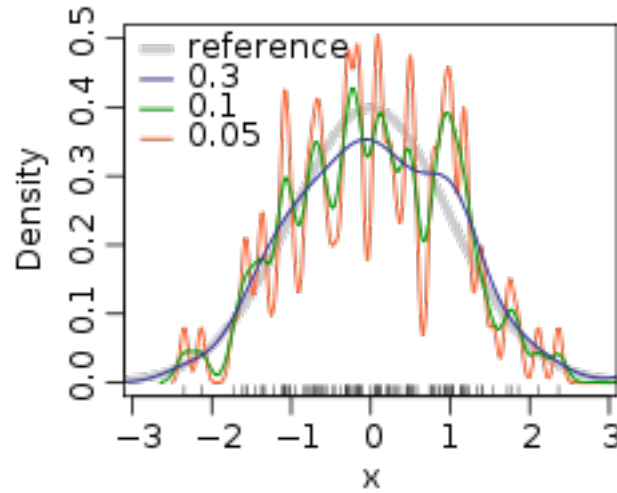


FIGURE 2.2: KDE for a univariate normal distribution using different bandwidths [4]

and might lead to different interpretations of the data. The lower-left chart shows a KDE estimation with a rectangular kernel for the same data. The intuition for understanding the differences between the two approaches is the following: while in histograms "blocks" are stacked in a regular grid centered in a single point, in KDE each block is centered on the point it represents, and the final result is the sum the total height at each location. The bottom-right plot shows a KDE on the same data but using a Gaussian kernel, meaning that each point contributes as a Gaussian curve, instead of a counting function.

Multivariate KDE A development of research about the topic during the 1990's, and the increase in available computing power, have been possible to define and implement multivariate version of KDE, like the one we use in eC2S [3]. The formulation for a d-dimensional KDE can be expressed as:

$$\hat{f}_{\mathbf{H}}(x) = \frac{1}{n} \sum_{i=1}^N K_{\mathbf{H}}(x - x_i) \quad (2.6)$$

Where x and x_i are d-dimensional vectors, \mathbf{H} is the bandwidth (or smoothing) $d * d$ matrix which is symmetric and positive definite, and K is the kernel function which is a symmetric multivariate density. Example of visualisation for a 2-dimensional KDE for a multimodal 2-dimensional Gaussian distribution are shown in figure 2.4 and 2.5.

2.3 ICT background

In this section we will introduce the programming and software tools which has been useful to implement UMAP and eC2S . First, we will present the DataBase Management Systems (DBMS) used to manage the big amount of data involved, namely MongoDB. Then, we will present the Python programming language and the related modules used to implement the logic of data-related operations, mathematical models and simulation.

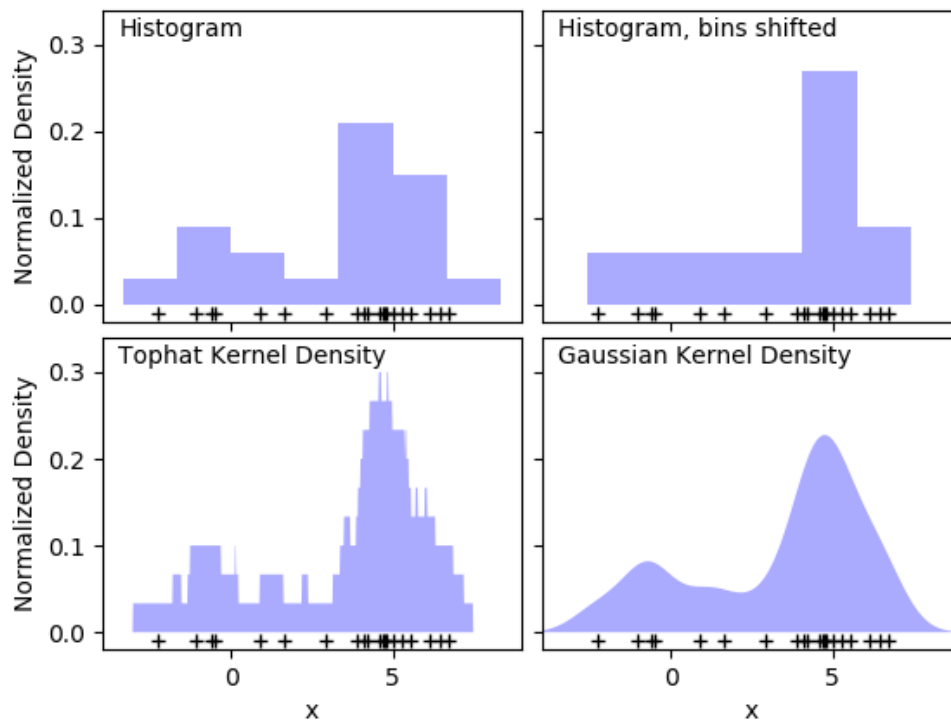


FIGURE 2.3: KDE vs histograms [3]

2.3.1 MongoDB

MongoDB is a free and open-source, cross-platform, document-oriented NoSQL DBMS, whose main implementation is written mostly in **C++**, but with an extensive support for **Javascript**, **Python** and **Go** languages applications [6]. MongoDB takes its name from the word *humongous*, making clear that it has been designed to work with big amount of data. **JSON**-like *documents* are the fundamental unit of Mongo's data organization. They can come without predefined schemas, which is an essential functionality when dealing with unstructured and heterogeneous information. In the semantics of MongoDB, documents are organized in *collections*, and a *database* is a container for many collections. The closeness of MongoDB, Javascript and JSON leads to a very good support for web applications, which culminates for example in projects like the MEAN stack [7], a free and open-source JavaScript software stack for building dynamic web sites and web applications composed by MongoDB, Express.js, AngularJS (or Angular), and Node.js. Like most DBMS, MongoDB provides many tools for data indexing and aggregation, query optimization, chronological data support, and it is particularly optimised for distributed and cloud-based applications.

2.3.2 Python

The **Python** [8] programming language has been chosen mainly because it allows to easily write codes integrating multiple and heterogeneous software systems, also

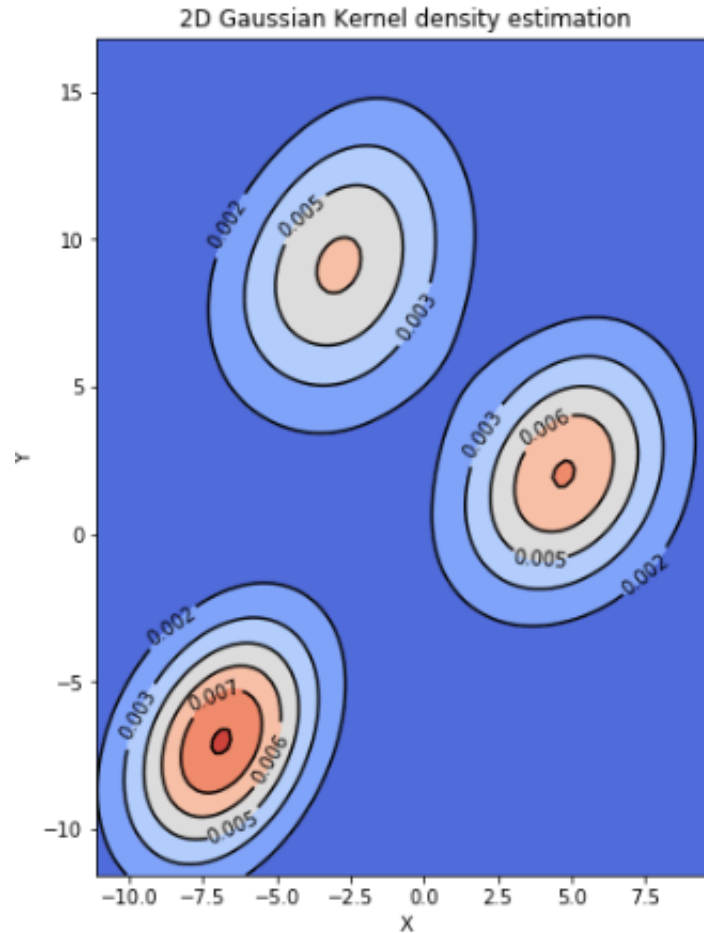


FIGURE 2.4: Contour plot of a 2-dimensional KDE for three-modal, Gaussian distributed data [5]

thanks to an impressive set of available open-source libraries covering very different tasks and ecosystems. Furthermore, Python libraries for scientific development and data science have been becoming increasingly popular, due to their readability, flexibility and very powerful features [9] [10]. We present now all the used modules:

- During the data collection phase, developed in [25], the most useful library has been **Requests** [11], which is a powerful and human-readable HTTP library. Using Requests together with network inspection on a web console, it has been written the code to easily scrape Enjoy's website and get a JSON feed, minute by minute, about parked cars in Turin. Furthermore, Requests has been used to retrieve information from the Car2Go Open API.
- **PyMongo** [12] is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python.
- The need for a powerful tool for data analysis has turned into the choice of using **Pandas** [13], an open source library providing high-performance, easy-to-use data structures and data analysis tools. Furthermore, Pandas can be extended to work specifically with geospatial data through the library *GeoPandas* [14].

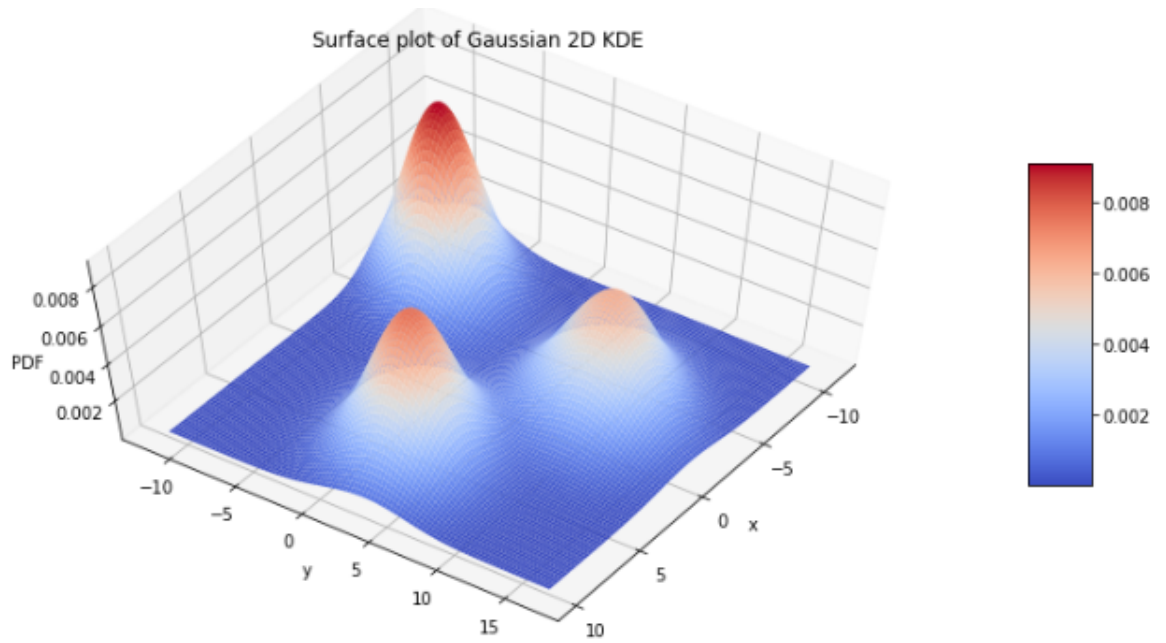


FIGURE 2.5: Surface plot of a 2-dimensional KDE for three-modal, Gaussian distributed data [5]

- Pandas is built on top of **NumPy** [15] library, which is the fundamental package for scientific computing with Python, providing among other things a powerful and efficient N-dimensional array object.
- Pandas and Numpy take advantage of an excellent support on visualization thanks to the strong connection with **Matplotlib** [16] package.
- NumPy, Matplotlib and Pandas are part of the **SciPy** [17] scientific stack, which is a Python-based ecosystem of open-source software for mathematics, science, and engineering that allow to turn Python into a mature functional programming environment.
- For which concerns the simulation, we entirely relied on **SimPy** [18], a process-based discrete-event simulation framework based on standard Python.
- Other useful tools to work with Python include the IDEs **Spyder** [19](Scientific PYthon Development EnviRonment) and **PyCharm** [20], and the **Conda** [21] package manager, an open source command-line utility to manage packages, apps and virtual environments managed and maintained by Continuum Analytics.

2.4 Related works

2.4.1 Poisson processes and KDE in geospatial analysis

The mathematical tools we propose in this thesis in order to model mobility demand have been already used and validated in other works for similar purposes, although to our knowledge their combined usage is still unexplored.

Poisson processes Authors in [54] proposed the usage of inhomogeneous Poisson processes to forecast short-term public transport demand. The focus of this work is given to the better suitability of inhomogeneous Poisson processes in comparison with the homogeneous ones, due to the intrinsic non-stationary nature of mobility demand. Their results show a clear improvement of forecasting using this tool, in particular for mobility hotspots where the overall demand is lower, and consequently more sparse in time. Inhomogeneous Poisson processes are also successfully used in [50] in order to predict taxi demand in the area of Munich. In this work, authors performed also a spatial analysis using a one-dimensional linear regression having as independent variable the density of Point Of Interest (POI, namely work, living, and leisure spots), and as dependent variable the number of pickup events. Their results show that this approach performs significantly better than the one considering as independent variable only the density of population. Finally, they also underline the strong impact of special events (in their case the Munich Oktoberfest) in the composition of mobility demand.

KDE Authors in [67] underlines the advantages of KDE for the analysis and visualisation of POIs density in urban studies. In particular, they compare three methods of density analysis which have been widely used in geographical analysis domain: Quadrat analysis, Voronoi-based analysis and KDE. While both of them require a division of the area in subregions, the former two methods have some disadvantages in practical applications. First, the definition of zones size is very critical for the first two methods, and might lead to significant errors. Second, those methods do not take into account the distribution in the neighborhood of grid cell. The authors claim that the use of KDE can improve the above situations: although KDE still requires the use of a grid of square superimposed cells, a smooth estimate of a density can be obtained for minimizing the above losses of information as long as the mesh size is small enough. Due to the above advantages, KDE is largely used as a general tool in spatial analysis. Authors in [39] use it for a general analysis of points moving in the space in a given time frame, and acknowledge the usage of KDE for identifying hotspots of crime, diseases and traffic accidents, as well as spatial distributions of plants, animals and people. Authors in [33] use KDE for estimation of road density and its impact on landscape fragmentation, while authors in [30] analyse the density of building renovation in the city of Lisbon. Of our particular interests are the results shown in [44], in which it is shown that the choice of the kernel function for spatial KDE in a continuous space setting has a very little impact for the results, especially when compared with the tuning of the bandwidth.

2.4.2 EV, CS, ECS

Literature about EV and ECS charging and relocation is wide and detailed, since EVs are increasingly being studied during recent years. We will therefore illustrate in this review only works which have been directly useful and/or inspiring for this thesis. Furthermore, in order to depict the general characteristics of CS-related problems, we will rely on existing reviews, such the ones in [42] and [29].

EV charging/discharging profiles A data-driven model for charging profiles of electric vehicles, using in particular a database containing 18,300 journeys, is provided by [46]. The model was developed using GPS data collected from 15 iMiev EVs (14 kWh battery capacity) operating nationwide in Ireland for 9 months as part of an EV demonstration project. The model developed in [46] combines Monte Carlo

simulation techniques and Bayesian inference. In a preliminary phase, not shown in this thesis, we used the results provided by this work to validate qualitatively our dataset, comparing in particular the charging/discharging behaviour of vehicles in our dataset with the one obtained in [46]. We also used in the simulation the same formulas describing the theoretical charging/discharging behaviour of vehicles, which we will report in Chapter 3.

EV charging optimisation with activity-based models data [56] proposes a centralised, agent-based mechanisms to optimise charge of EVs following a trip agenda generated using the activity-based software in [28], referring to an area in Flandre, Belgium. The optimisation is made to fulfill trip schedules while sustaining grid capacity constraints and minimising electricity bills. Using data provided again by [28], [60] proposes a Vehicle-to-Vehicle (V2V) energy trading mechanisms, having as optimisation goal a market equilibrium with lower prices for users and lower load at business hours for grid operator. The same authors refined their research by introducing time varying electricity prices in [59]. One of the purpose of eC2S is to provide a framework to apply these kind of optimisation techniques using a more general setting, especially for which concerns the parametrisation of input data.

CS problems taxonomy An extensive review of car sharing scientific literature is provided by [42], which is our main reference for data and figures in this subsection. This work analyses 137 papers appeared from 2002 to 2017 in order to classify them under a certain number of taxonomy axes. The taxonomy provided is meant to better focus the area of interests of this thesis work.

The taxonomy axes proposed by [42] are shown in figure 2.6.

Service specifications		
Mode	Engine	
Two-way (station based)	Fully thermic	
One-way (station based)	Green	
Free floating		
Not Applicable		
Research problem		
Optimization objectives	Time horizon	Methodologies
Business and Service	Design	Simulation
Infrastructure	Planning	Stochastic optimization
Fleet management	Operational and real time	Combinatorial optimization
		Statistical analysis

FIGURE 2.6: Taxonomy of service mode and research problems

Mode of service supply The first meaningful classification of car sharing systems can be done considering the mode of service supply: the most obvious choice is a 'one-way' policy, in which the user starts from an 'origin' and has to leave the car in some 'destination'. Some providers let the user pick and leave the car choosing among some fixed placement/spots (station-based), while others let the user freely ride and park within a specified operational area (free-floating). While on a first sight the first approach may create advantages on the provider side concerning the ease of monitoring and control over the whole system, the free-floating approach is likely preferred by users, as it gives the theoretical possibility of reaching directly every destination. For this reason, many modern car sharing providers adopt the free-floating strategies, with the notable exception, for example, of hybrid and electric car

sharing providers, for which the presence of stations is a technical constraint. Within the free-floating approach, the main issue is the demand-supply imbalance created by service flexibility, resulting in additional costs for the providers to develop relocation strategies. Furthermore, the free-floating approach implies by definition that the user has to find a parking, while station-based strategies offer the possibility of booking in advance a place in dedicated parking spots. Research focus across the different modes are shown in figure 2.7. The tag "Not applicable" considers the papers analyzing car-sharing services without reference to any specific mode.

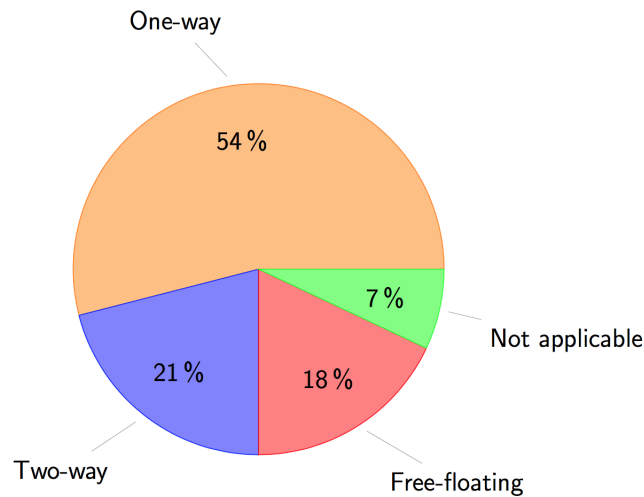


FIGURE 2.7: Pie chart of service modes

Type of engine This dimension is used to classify the engine type of the cars involved in the car-sharing service. The category "Fully thermic" is composed by systems having vehicles powered by traditional (i.e., fossil derived) fuels such as gasoline or diesel, while the category "Green" comprises fleets of vehicles with less-polluting engines, as electrical, hybrid, plug-in hybrid, LNG or LPG. Figure 2.8 shows a pie chart of the distribution of papers in [42] along this taxonomy axis.

Optimisation objectives This dimension classifies the analyzed papers according to the high-level component of the car-sharing service that is subjected to optimization. Those component are:

- Business and service: study of business models and the definition of the car-sharing service, including the identification of the user behaviors and the demand estimation.
- Infrastructure: optimal design and location of car-sharing facilities, as for example parking and charging stations.
- Fleet management: operations focused on fleet characteristics and status, such as determining the fleet size or defining relocation strategies.

Figure 2.9 shows a bar plot of the distribution of papers in [42] across this taxonomy axis and service modes.

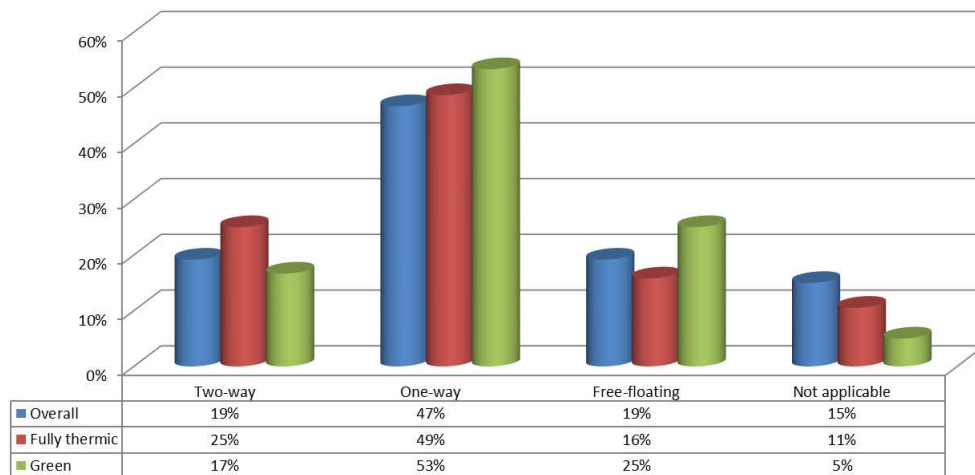


FIGURE 2.8: Bar chart of service engines and modes

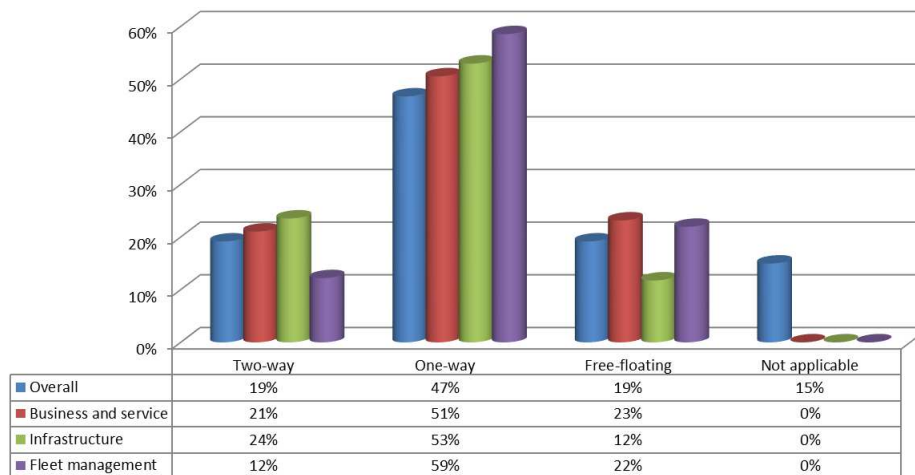


FIGURE 2.9: Bar chart of service optimisation goals and modes

Time horizon This axis considers the interval of time for which the decisions remain valid.

- Design and strategic decisions are the ones that car-sharing organisations must keep into account in the designing of the service, including fleet type definition, user behavior, pricing policies, market place and demand identification.
- Planning and tactical decisions deal with a better definition of the specific service, and include fleet size definition, location of facilities (parking stations, e-charging stations, car maintenance facilities), urban areas boundaries, management of uncertainty of local demand.
- Operational and real time decisions are taken day-by-day related to the service provided: operative car maintenance, refueling, car washing, relocation strategies to balance the system, avoiding stations with an excess of vehicles, or empty stations.

Figure 2.10 shows a bar plot of the distribution of papers in [42] across time horizon dimension and service modes.

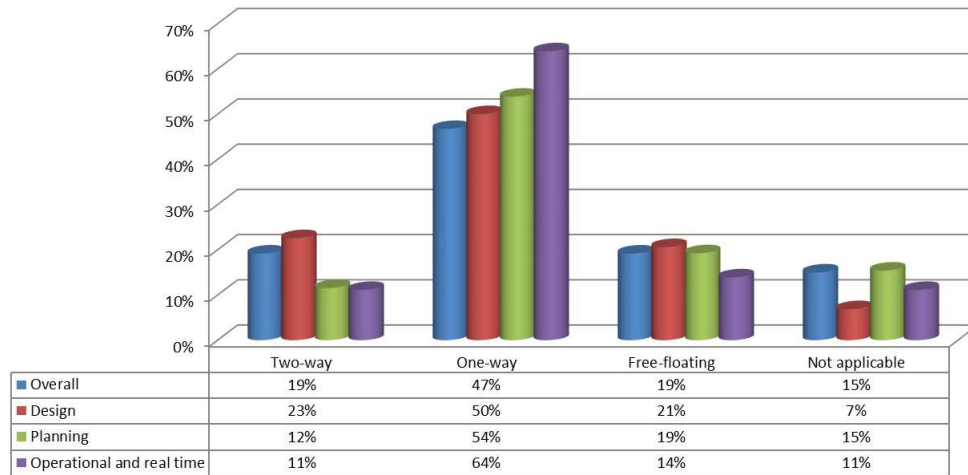


FIGURE 2.10: Bar chart of time horizon of the study and service modes

Mathematical methodologies This axis groups the papers according to the scientific approach used by the authors.

- **Simulation:** simulation aims to imitate the operation of the real-world processes in order to help the decision making for new operators. Simulations are often based on real data referring the transport behavior of the dwellers of a specific area.
- **Combinatorial optimization:** combinatorial optimization consists in modeling, analyzing and solving a decision problem finding the optimal objective function according to a set of constraints when the data involved in the problem under study can be considered as deterministic.
- **Stochastic optimization:** stochastic optimization methods are used to solve decision problems where the data are affected by uncertainty. In the car-sharing services case, uncertainty often affects the service demand or the flows of vehicles between different parking stations or within the service area.
- **Statistical analysis:** statistical analysis methods are mainly adopted for analyzing data sets deriving from real observation (such as data sets provided by operators) or from surveys and focus groups.

Figure 2.11 shows a bar plot of the distribution of papers in [42] across mathematical methodology dimension and service modes.

Following this taxonomy, we set this thesis work in the dimensions of Electric Free Floating Car Sharing (EFFCS), flexible time horizons, and simulation as main mathematical methodology. However, eC2S is designed with the goal of being highly modular, parametric and extensible in terms of input, data processing techniques and output.

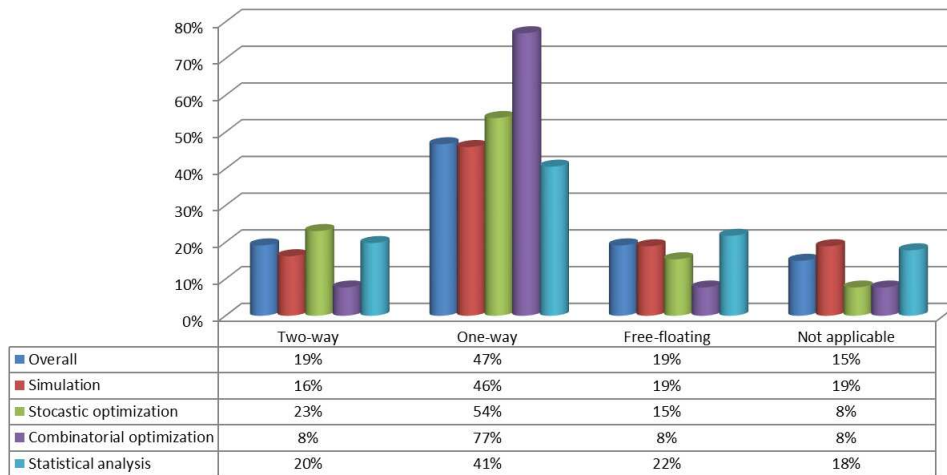


FIGURE 2.11: Bar chart of service optimisation goals and modes

ECS optimisation problems For which concerns typical optimisation problems in ECS, [29] provides a summary of the main contributions as of September 2016. The analysis of this document allows to set a frame for typical operational problems in ECS, and highlights that the study of EFFCS systems is still young and not well explored. In the next paragraphs we will show instead more recent contributions. [29] organises ECS-related research problems as follow:

- Strategic and tactical problems
 - Location of charging stations
 - Allocation of vehicles to existing stations
- Operational problems
 - **Relocation of vehicles**
 - Battery swap
 - Routing problems

In this thesis, we will focus on charging relocation problems, although eC2S might represent a framework for studying all categories of problems identified by [29]. With the expression "charging relocation" we mean the operations needed to ensure that vehicles spread in the city have enough energy to serve the users. In other words, those operations are the ones strictly needed from the operational point of view in running an ECS fleet. The term "relocation" is also used very often in literature to describe operations of spatial balancing of vehicles aimed to ensure availability of cars rather than energy, namely to minimise the probability that users do not find a car when they ask for it. In this thesis, the latter form of relocation is considered only marginally, and it represents one of the possible future work path.

ECS simulation [35] proposed a framework for modelling and simulating the evolution of an ECS system. Despite being fairly general and not computationally expensive, this model is fully parametric, makes strong assumptions about the probability distributions of all processes involved (not only users interarrival times) and

does not include geographical aspects and constraints. A more complete model is provided by [62], who developed a simulator written in the C# language able to represent reservations and different relocation strategies. This simulator allows partial floating (a vehicle can be dropped-off outside a station in a close range if the station is already full) and is validated on a real-world case study about the city of Nice, France. Authors in [45] simulated the operation of electric vehicles in urban car sharing networks with a focus on economical aspects. Their key findings underline how technical vehicle range limits might be overcome without serious financial drawbacks, provided an adequate charging infrastructure is provided. They also support the idea that a bigger fleet with shorter range vehicles might be more convenient than a smaller fleet with long range vehicles. This work also compares the potential profitability of ECS services in Europe and North America, stating that the former might be a more profitable area to serve due to higher cost of traditional fuel. [31] addresses the problem of insufficient vehicle utilization in ECS systems by developing a framework which increases utilization, improves charging schedules, increases battery life and consequently mitigates range anxiety of users. Authors in [36] developed a trace-based simulator to face the problem of charging stations placement in the city of Turin. Their framework includes different high-level placement strategies, as well as more complex heuristics optimising the placement with regard to user discomfort. Their placement strategy called "Num_parkings", which ended up performing best, has been implemented in eC2S.

Relocation of vehicles The PhD thesis [68] represents a deep and complete study of relocation policies in station-based one-way CS, as it analyses a range of different exact and heuristic algorithms to tackle the problem. Although the study does not explicitly refer to the electric case, the presence of given parking stations makes it quite similar for certain aspects, and it is useful to depict general characteristics of the relocation problem when the optimisation goal is minimising rejected user demand. An intuitive yet important general result of this work is that approaches not considering future user demand are not effective in reducing the number of rejected demands. The input data used to validate the proposed algorithms are based on survey and socio-demographical information about the area of Paris. [53] is one of the first works proposing a relocation strategy for one-way station-based ECS. Their approach consists of a modified stable marriage problem solver which minimises relocation distance and time, and it is validated on real trips data taking place in Jeju City, Republic of Korea. Similarly, [32] proposes a Mixed Integer Linear Programming (MILP) approach for solving the relocation problem with a case study in the area of Milan, introducing also the realistical presence of workers having the task of moving vehicles. Authors in [52] consider the FFCS case for the city of Rome and Florence, and propose an algorithm to minimise the walking distance that workers might have to travel when they have to reach a car to relocate.

Chapter 3

Methodologies

3.1 Software description

eC2S is a software written from scratch in Python which allows to simulate EFFCS systems. eC2S is composed by the following folders and Python modules:

- **Retrieval:** contains the instructions for querying the MongoDB database and creating files containing desired input data.
- **Preprocessing:** contains functions for transforming and integrating original data with other useful ones, in particular for geospatial processing.
- **Loading:** contains functions for loading data into RAM with efficient data structures.
- **DataStructures:** contains classes representing real world abstractions such as cities and cars.
- **SimulationInput:** contains classes implementing the logic for managing the input of the simulation. This includes configuration, statistical models and shared data structures.
- **Simulation:** contains classes implementing the simulation logic. This includes the abstraction for user requests generation, mobility and charging.
- **SimulationOutput:** contains classes for results collection, aggregation and visualisation.
- **SingleRun:** contains functions for running a single simulation with a specified configuration.
- **ModelValidation:** contains procedures to run model validation in time and space.
- **MultipleRun:** contains functions for running a set of simulations following a grid of configuration parameters. It is possible to run a set of simulations on many cores in parallel.
- **utils:** contains utility functions used across many modules.
- **Data:** contains raw data queried from the database in form of .xlsx files, and processed data needed for simulation input in form of pickles (binary files).
- **Figures:** contains charts produced in the simulation output phase organised by simulation scenario and city.

- **Results:** contains simulation results in form of pickles organised by simulation scenario and city.
- **venv:** contains the Python virtual environment having only strictly necessary libraries for eC2S .

Let's now describe each module one by one.

3.1.1 Retrieval

The Retrieval module is meant to query the remote MongoDB database in order to fetch data of our interest. This module is also responsible of standardising format of input data and make them available locally and speed up the upcoming operations. A MongoDB query performed through the Pymongo library returns an object of type cursor, which is essential an iterator of MongoDB documents, represented as Python dictionaries. When performing intensive data operations, such as for example KDE fitting, it is more handy to convert these collections of dictionaries into pandas DataFrame. For this reason, after querying, bookings are aggregated by month and saved as .xlsx files, directly convertible into pandas DataFrame. The Retrieval module is composed by the following scripts:

- **BigDataDB_Proxy.py:** main class for interaction with a MongoDB instance, containing the functions for authenticating and querying the database.
- **DB_to_df.py:** utility functions for conversion into pandas DataFrame.
- **get_data_from_DB.py:** script used to run queries, taking as parameters the desired city and month

Before going into the details of software description, it is appropriate to introduce the structure of our input data. The database is composed by real rentals performed by Car2go users during 2017 in 25 cities, and the procedure of data collection is described in [25] . For each booking, the following features are available:

- **plate:** Identifier of a car.
- **start_time:** datetime object identifying the time instant in which our system detects the moving of a car from the set of available cars to the set of the busy ones.
- **end_time:** datetime object identifying the time instant in which our system detects the moving of a car from the set of the busy cars to the set of the available ones.
- **start_latitude:** latitude of the origin point estimated by car GNSS receiver corresponding to the last moment in which the system sees the car in the set of available ones.
- **start_longitude:** longitude of the origin point estimated by car GNSS receiver corresponding to the last moment in which the system sees the car in the set of available ones.
- **end_longitude:** longitude of the destination point estimated by car GNSS receiver corresponding to the first moment in which the system sees the car appearing again in the set of available ones.

- **end_latitude**: latitude of the destination point estimated by car GNSS receiver corresponding to the first moment in which the system sees the car appearing again in the set of available ones.
- **start_soc**: integer between 0 and 100 referring to the percentage of available battery at start time.
- **end_soc**: integer between 0 and 100 referring to the percentage of available fuel at end time.
- **duration**: difference between end and start in minutes.
- **euclidean_distance**: beeline distance between origin and destination point in kilometers:

3.1.2 Preprocessing

This module is mainly responsible for the discretisation of a city surface into zones. Zoning systems are a complex and important topic in transport modelling, and our approach has to be considered preliminary as it has the main goal of reducing computational time for spatial processing. This is achieved by organising the city in 500mX500m squared zones, each one identified by an index and therefore accessible in a $O(1)$ time. Every distance in the simulation is computed from a zone centroid to another zone centroid. Another important aspect is that the choice of 500m bins has an impact on how we model user behavior: in fact, when simulating user requests, we allocate them only if a car with enough energy is available in the zone where the request came from, or in its 1-hop neighboring zones. This is equivalent to say that the maximum walking distance a user is willing to cross in order to take a car in the simulation is 700m. The Python class responsible for city binning is contained in the file **CityGeoProcessor.py**. This class uses efficient spatial query using a specific data structure called RTree, both provided by the Python library GeoPandas within the class GeoDataFrame. The output of the preprocessing phase is a file containing the spatial description of the city grid.

3.1.3 Loading

The Loading module can be seen as a local version of the query engine implemented in Retrieval, aimed to load into RAM a subset of available data and run the upcoming part of the pipeline to them.

3.1.4 DataStructures

In this module, an abstraction for real world concept such as city and car is provided. For the scope of this thesis, only the City class has been implemented, because all the simulation have been conducted with the same model of car (Smart ForTwo electric) and charging pole (2 KWh, 0.92 charging efficiency), and also users are supposed to behave deterministically in the same way. Despite this, the code is already prepared for introducing different models of cars and charging poles, as well as introducing more detailed description of cities and user behaviors. The class City implements the following methods:

- **get_neighbors_dict** Initialise a dictionary in which each key is an identifier of a city zone and each value is a dictionary of 1-hop neighbors indices and

distances. This data structures represent the topology of neighboring zones and is useful to guarantee $O(1)$ access to neighbors information.

- **get_input_bookings_filtered** Initialise a pandas DataFrame containing a subset of the available booking for a given city. The subset is created following a certain configurable filtering criterion. In our case, we retained only bookings having duration between 3 and 60 minutes and euclidean distance between origin and destination greater than 500m. This filtering criterion is though to remove from the dataset reservations cancelled by users, and to consider for simplicity only trips in which the user went straight to a destination. Clearly, this criterion cannot guarantee that all of those trips are actually removed, and a more precise trips characterisation might lead to consider different kind of trips, such as for example round trips.
- **get_requests_rates** Initialise a dictionary containing average request rates measured per each couple (hour, daytype) where hour is in the range [0, 23] and daytype can be "weekday" or "weekend".
- **get_trip_kdes** Initialise a dictionary containing KDE fitted per each couple (hour, daytype) where hour is in the range [0, 23] and daytype can be "weekday" or "weekend".
- **get_valid_zones** Initialise a set of zones corresponding to the operational area of car2go in a given city. This corresponds to retain only zones which have been origin and destination of at least one trip.

3.1.5 SimulationInput

The input to the simulator is implemented in two main classes:

- **EFFCS_SimConfGrid.py**: class managing a configuration grid i.e. a set of different configurations for running simulation campaigns
- **EFFCS_SimInput.py**: class managing the initial conditions of the simulation and the data structures used during it.

A single configuration is composed by the following parameters:

- **requests_rate_factor**: multiplicative factor for the original request rate of a given city as measured from data. If this is set to 1, the original request rate will be used in the simulation.
- **n_cars_factor**: multiplicative factor for the observed number of cars present in the fleet of a given city. If this is set to 1, the original number of cars will be used in the simulation.
- **time_estimation**: flag indicating whether to perform time and SOC consumption estimation during charging relocation.
- **queuing**: Flag indicating whether cars are queued at stations when they need charging or they are left available.
- **alpha**: Lower charging threshold. A car below this threshold after a booking will request charging.

- **beta**: Upper charging threshold. A car will be always charged up to this level.
- **hub**: Flag indicating the presence of a centralised charging hub.
- **hub_zone_policy**: String identifying a strategy for hub placement in the city. Currently supported strategies are "num_parkings" and "manual", in which the zone id of the hub is set manually.
- **hub_n_charging_poles**: Number of charging poles in the hub.
- **relocation**: Flag indicating whether a car is moved to the original destination of the last booking after charging.
- **distributed_cps**: Flag indicating the presence of a decentralised charging infrastructure.
- **cps_placement_policy**: String identifying a strategy for charging poles placement in the city. The only currently supported strategy is "num_parkings".
- **n_charging_poles**: Number of distributed charging poles in the city.
- **user_contribution**: Flag indicating the possibility for users to contribute to charging, with a probability of contribution modelled as a Bernoulli RV with p = willingness.
- **willingness**: Probability of a single user contribution.

Let's give a deeper look to the methods and data structures implemented in `EFFCS_SimInput.py`:

- **init_cars** Initialise a dictionary containing SOC status for each car, and another dictionary containing the zone in which a particular car is. The former dictionary is initialised using a random uniform distribution of SOC between 25 and 100, while the latter dictionary is initialised picking uniformly random zones among the valid ones.
- **init_cars_dict** Initialise a dictionary containing the list of available cars for each zone, and another dictionary containing the list of available cars for each neighbor of each zone. This redundancy is needed to guarantee $O(1)$ access to cars' location information.
- **init_hub** Initialise a charging hub following the strategy and the parameters specified in the configuration.
- **init_charging_poles** Initialise charging poles distributed in the city following the strategy and the parameters specified in the configuration.

3.1.6 Simulation

The core simulation module is composed by classes managing the processes of user requests allocation, mobility and charging. Those classes are:

- **EFFCS_Sim.py**: main simulation module implementing mobility requests.
- **EFFCS_TraceB_Sim.py**: subclass of `EFFCS_Sim.py` for trace based simulation.

- **EFFCS_EventG_Sim.py**: subclass of **EFFCS_Sim.py** for event generation based simulation.
- **EFFCS_ChargingPrimitives.py**: class managing the scheduling and the evolution of charging processes.
- **EFFCS_ChargingStrategy.py**: class implementing different charging policies.

We will discuss in details the internal assumptions of the simulation in section 3.3.

3.1.7 SimulationOutput

This module is responsible of simulation statistics collection and plot. It is composed by the following classes:

- **EFFCS_SimOutput.py**: class creating an aggregated simulation statistics object.
- **EFFCS_SimOutputPlotter.py**: class containing the code for different plots.

The metrics contained in the output object managed by **EFFCS_SimOutput.py** are:

- **n_booking_reqs**: Number of user requests.
- **n_bookings**: Number of satisfied user requests.
- **n_unsatisfied**: Number of unsatisfied user requests.
- **n_same_zone_trips**: Number of user requests resulting in a successful booking for which a car was found in the same zone as the request itself.
- **n_not_same_zone_trips**: Number of user requests resulting in a successful booking for which a car was found in a zone adjacent to the one of the request.
- **n_no_close_cars**: Number of user requests for which no cars were found, neither in the requests zone nor in its neighboring zones.
- **n_deaths**: Number of user requests for which no cars with enough energy in the battery were found, neither in the requests zone nor in its neighboring zones.
- **percentage_booking_reqs**: Percentage of user requests.
- **percentage_bookings**: Percentage of satisfied user requests.
- **percentage_unsatisfied**: Percentage of unsatisfied user requests.
- **percentage_same_zone_trips**: Percentage of user requests for which a car was found in the same zone as the request itself.
- **percentage_not_same_zone_trips**: Percentage of user requests for which a car was found in a zone adjacent to the one of the request.
- **percentage_no_close_cars**: Percentage of user requests for which no cars were found, neither in the requests zone nor in its neighboring zones.

- **percentage_deaths**: Percentage of user requests for which no cars with enough energy in the battery were found, neither in the requests zone nor in its neighboring zones.
- **percentage_same_zone_trips_satisfied**: Percentage of satisfied requests for which a car was found in the same zone as the request itself.
- **percentage_not_same_zone_trips_satisfied**: Percentage of satisfied requests for which a car was found in a zone adjacent to the one of the request.
- **percentage_no_close_cars**: Percentage of unsatisfied requests for which no cars were found, neither in the requests zone nor in its neighboring zones.
- **percentage_deaths**: Percentage of unsatisfied requests for which no cars with enough energy in the battery were found, neither in the requests zone nor in its neighboring zones.
- **n_charges** Total number of charging events.
- **n_charging_requests_system** Total number of bookings resulting in a charging request for the system operator.
- **n_charges_system** Total number of charging events managed by system operator.
- **n_charges_deaths** Total number of charging requests for which it is not possible to bring the car to the closest charging station, due to the insufficient level of battery of the car.
- **percentage_charge_deaths_system** Percentage of charging requests to the system resulting in a charge death.
- **soc_avg** Average SOC of the fleet.
- **soc_med** Median SOC of the fleet.
- **charging_time_avg** Average charging time for all charging events.
- **charging_time_med** Median charging time for all charging events.
- **n_charges_by_car_avg** Average number of charge for a vehicle.
- **n_charges_by_car_system_avg** Average number of charge for a vehicle considering only system-managed charging events.
- **n_charges_by_car_users_avg** Average number of charge for a vehicle considering only user-managed charging events.
- **tot_energy** Total charging energy requested by the fleet during simulation time.
- **percentage_charges_system** Percentage of charging events managed by the system.
- **percentage_charges_users** Percentage of charging events managed by the users.
- **percentage_energy_system** Percentage of charging energy requested to the grid during system charging events.

- **percentage_energy_users** Percentage of charging energy requested to the grid during users' charging events.
- **percentage_duration_system** Percentage of charging duration of system charging events.
- **percentage_duration_users** Percentage of charging duration of users charging events.
- **cum_relo_out_t** Total time spent for outward charging relocation, i.e. to bring car to charge.
- **cum_relo_ret_t** Total time spent for return charging relocation, i.e. to bring car from charging station to a relocation zone.
- **cum_relo_t** Total time spent for relocation.
- **cum_relo_kwh** Total energy consumed for relocation.

3.2 Demand modelling

The general question we try to answer is “When and where booking requests might occur in a given city and in a certain timeframe?”. In order to model this mobility demand, we use as input real bookings performed by users collected through the software UMAP. Therefore, we are practically estimating the whole demand (which may be satisfied or unsatisfied) using only the satisfied demand: for this reason, it is impossible using only these input data to detect or estimate when and where users' demand was really unsatisfied. As a consequence of the introductory hypothesis, we are in this way comparing the number of booking requests in the simulation with the number of real bookings in the trace. Therefore, we can assess the statistical properties of the model, but we cannot use the concept of “booking” in the same way between the real trace and the simulated one. Other data source might be needed to better model more components of mobility demand.

3.2.1 Time estimation

Our assumption for time estimation is that inter-arrival times between two consequent bookings follow an exponential distribution, namely that the arrival stochastic process is a Poisson process. We further suppose that the average rate of arrivals is not constant with time, but varies with the type of day and the hour. We suppose that the type of day can be only “work day” or “weekend”, namely that there are no differences on average between a Monday and a Wednesday or a Saturday and a Sunday. Therefore, we will obtain 48 (2×24) different rates and a de-facto inhomogeneous Poisson process. During simulation, the arrival of the next booking request is generated as a random sample of an exponential distribution having as rate the one corresponding to the hour and day type associated with the current simulation time. The procedure for rate estimation uses a frequentist approach: for each couple (daytype, hour) in the input trace, the average number of bookings for a certain couple (daytype, hour) is computed as number of bookings divided by the number of instances for that couple (daytype, hour). The rate is then obtained as the inverse of the computed average. This procedure is implemented in the following method of the class **City**:

```

def get_requests_rates (self):

    self.request_rates = {}

    for daytype, daytype_bookings_gdf \
in self.input_bookings.groupby("daytype"):
        self.request_rates[daytype] = {}
        for hour, hour_df \
in daytype_bookings_gdf.groupby("hour"):
            self.request_rates[daytype][hour] = \
                hour_df.city.count() \
                / (len(hour_df.day.unique())) \
                / 3600

    self.sim_general_conf["avg_request_rate"] = \
        pd.DataFrame(self.request_rates.values()).mean().mean()

    return self.request_rates

```

3.2.2 Spatial estimation

Spatial estimation is performed by fitting the probability density function of OD couples using Kernel Density Estimate. As for time estimation, we suppose that the probability density function is not constant with time, but varies with the type of day and the hour. Therefore, we will obtain also in this case 48 (2×24) different distributions. Following the conclusions of [51] we used KDE in conjunction with hot spots analysis, namely we fit KDE on a discretised space composed by 500m x 500m zones. In this way, the input OD couples are identified by sequential integer IDs and not by points coordinates, and the spatial resolution is given by zones size, and not by KDE bandwidth, given that it is set lower or equal than 1. A bandwidth equal to 1 would in fact mean that the resolution of the spatial KDE is set as the resolution of city binning, namely again the 500m x 500m zones. A smaller bandwidth does not bring significant advantages for estimation, as a finer-grained KDE would anyway be mapped on spatial bins. A bigger bandwidth would instead lose the granularity of city binning, and lead to a reduced precision in detecting spatial patterns, as shown in [67]. Binning the search space of a spatial KDE brings also significant advantages in terms of computational performances, although more precise and efficient yet complex techniques have been developed, such as in [61] and [65]. The Python code used for KDE computation is shown below. Note that default kernel for scikit-learn implementation of KDE is Gaussian [3].

```

def get_trip_kdes (self):

    self.trip_kdes = {}
    self.kde_columns = [
        "origin_id",
        "destination_id",
    ]

    for daytype, daytype_bookings_gdf \
in self.input_bookings.groupby("daytype"):

```

```

self.trip_kdes[daytype] = {}
for hour, hour_df\
in daytype_bookings_gdf.groupby("hour"):
    self.trip_kdes[daytype][hour] = \
        KernelDensity(
            bandwidth=1
        ).fit(\
            hour_df[self.kde_columns].dropna())

return self.trip_kdes

```

3.3 Simulation assumptions

The simulation relies on a set of assumptions implemented in its input and in its internal logic. These assumptions are not static, and can be changed by properly setting some parameters, or by explicitly implementing rules in the simulator. In this section, we list the assumptions made for the specific results shown in this thesis work. We identified three categories for the assumptions:

- **Zoning assumptions**
 - **a.0** City area is divided into 500mX500m zones
 - **a.1** Trip distance is computed as euclidean distance from the origin centroid to the destination centroid and multiplied by a correction factor representing the average driving distance.
- **Cars and poles assumptions**
 - **a.2** Poles are supposed to have the same characteristics, namely 2kW nominal power and 92% charging efficiency.
 - **a.3** Cars are supposed to have the same characteristics, namely 16.7kWh capacity and 15.9 kWh / 100 km energy efficiency
- **Charging process assumptions**
 - **a.4** Charge and discharge are assumed to evolve linearly and without losses.
 - **a.5** A user takes the most charged car among the ones with enough battery in the same zone or in a 1-hop neighboring zone.
 - **a.6** At the end of each rental, a car is released only if there is no need of charging. Otherwise, the car will remain unavailable until the desired charging process is performed. A car is considered to be in need of charging when its SOC reaches a value of 25 or below, and it is not released until its SOC reaches the value of β set in the particular simulation.
 - **a.7** At the end of each charging process, a car is considered to have been unplugged and parked in the same zone, or relocated and parked somewhere else.
 - **a.8** Charging relocation times are computed using 15 km/h as average speed, therefore including parking and traffic times.
 - **a.9** In case of a distributed infrastructure, cars are always relocated to the closest charging point.

Chapter 4

Model validation

4.1 Introduction

In this chapter we will validate the assumptions made for model estimation, namely the Poisson process hypothesis for inter-arrival times of bookings and the usage of KDE for spatial estimation, for the city of Turin. In order to perform this validation, we will run a simulation using the input trace and a simulation using the implemented model, and then we will compare the temporal aspects and the spatial aspects. Model validation results for other cities is shown in [appendix A](#).

Before starting, let's make an important remark. In order to model mobility demand, we use as input real bookings performed by users collected through the software UMAP. Therefore, we are practically estimating the whole demand (which may be satisfied or unsatisfied) using only the satisfied demand: for this reason, it is impossible using only these input data to detect or estimate when and where users' demand was really unsatisfied. As a consequence of this introductory hypothesis, we are in this way comparing the number of booking requests in the simulation with the number of real bookings in the trace. In such setting, we can assess the statistical properties of the model, but we do not use in fact the concept of "booking" in the same way between the real trace and the simulated one. Other data source might be therefore needed to better model more components of mobility demand. For our scopes this is not critical, as we are not actually interested in reproducing the trace as similarly as possible, but rather in detecting temporal and spatial patterns in mobility demand. For which concerns the former, we expect it to be quite dependent on typical human activities patterns, while the latter clearly presents in the trace a strong correlation with actual availability of cars in a precise zone and time frame. This means that in the case of spatial modelling, we are interested in uncorrelating trace samples from simulated ones, yet maintaining spatial patterns reasonably compatible with the city under study, in order that locations of user requests and locations of cars does not evolve synchronously.

4.2 Time analysis

Before quantifying the errors, it is useful to visualise the distribution of trips along the hour of the day, as shown in [figure 4.1](#). On the x-axis is represented the hour of the day, while on the y-axis is represented the fraction of trips. We can easily observe that the profile of simulated bookings is very similar to the one of real bookings.

In [4.2](#) we can observe the quantile-quantile (Q-Q) plot of inter-arrival times. A Q-Q plot is a graphical method for comparing two probability distributions by plotting their quantiles against each other. The closer points are to the bisector line, the more similar are the two probability distributions. From the plot, we can observe that

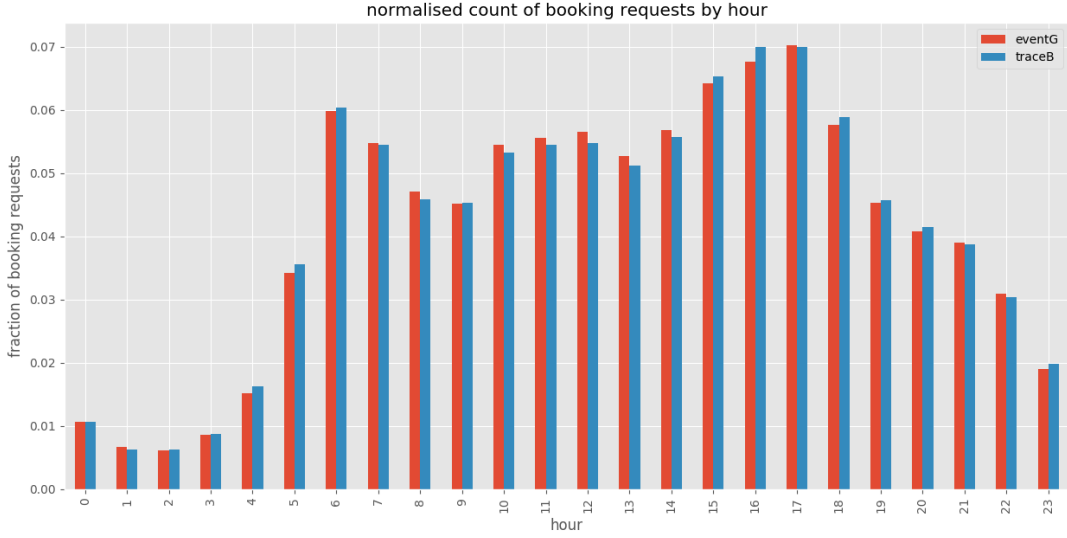


FIGURE 4.1: Percentage of bookings by hour for the city of Turin, simulation vs trace

The simulated hourly bookings profile for the city of Turin is very close to the one deduced from input data. It is also possible to notice that there is not a constant underestimation or overestimation.

points are tightly close to the bisector line, meaning that the Poisson process may well represent the occurrences of bookings in time.

For which concerns the error in the time distribution of events, we define the following relationship:

$$\epsilon_{t,\%}^N = |N_{t,\%}^{\hat{}} - N_{t,\%}| \quad (4.1)$$

Where:

- t : selected time frame.
- $N_{i,t,\%}$: Percentage of trips in input data at a given time frame.
- $N_{i,t,\%}^{\hat{}}$: Percentage of trips in simulated data at a given time frame.

Figure 4.3 shows the error resulting from equation 4.1 for each couple (hour, day-type). From the figure, it is possible to notice that errors are constantly lower than 1% and that there is not a very large variation between weekends and weekdays.

4.3 Spatial analysis

We define the following spatial error metric:

$$\epsilon_{i,t,\%}^o = |N_{i,t,\%}^{\hat{o}} - N_{i,t,\%}^o| \quad (4.2)$$

$$\epsilon_{i,t,\%}^d = |N_{i,t,\%}^{\hat{d}} - N_{i,t,\%}^d| \quad (4.3)$$

$$\epsilon_{i,t,\%}^{od} = \epsilon_{i,t,\%}^o + \epsilon_{i,t,\%}^d \quad (4.4)$$

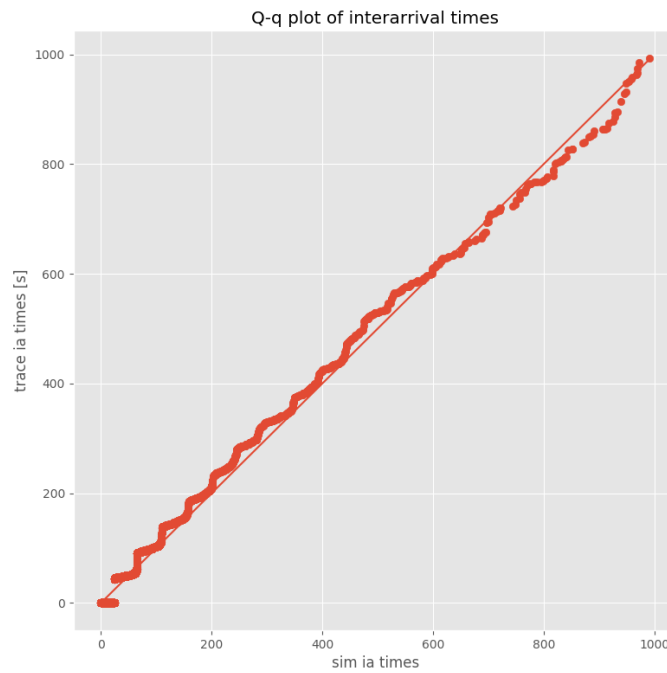


FIGURE 4.2: Q-Q plot of inter-arrival times for the city of Turin

Scattered points are very close to the bisector line, meaning that an inhomogeneous Poisson process is a good fit for input data.

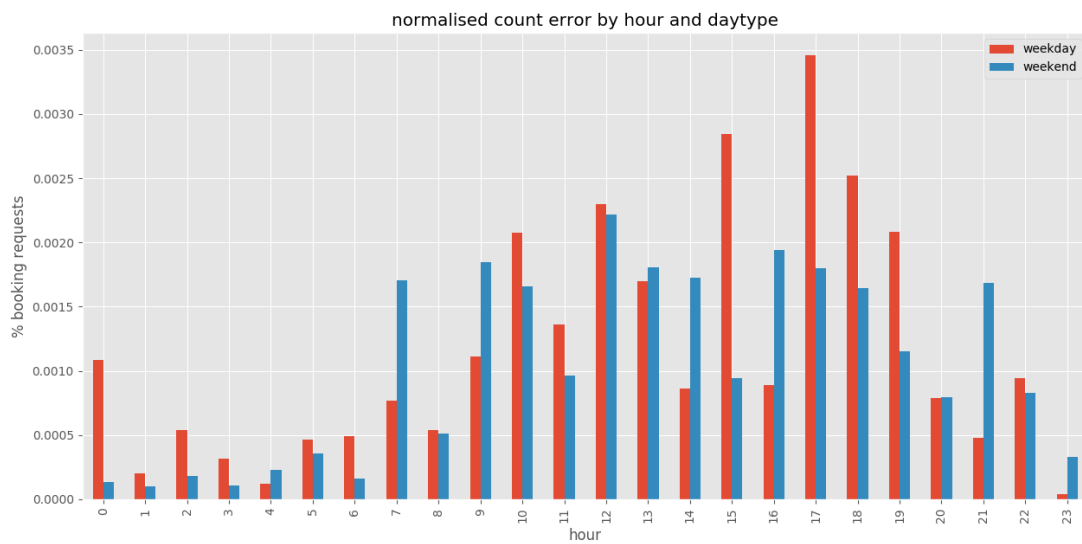


FIGURE 4.3: $\epsilon_{t,\%}^N$ for the city of Turin

The relative hourly error is always under 1%, both in case of working days and weekend days.

Where:

- i : zone id.
- t : selected time frame.
- $N_{i,t,\%}^o$: Percentage of trips in input data at a given time frame for which the zone i is the origin zone.
- $N_{i,t,\%}^{\hat{o}}$: Percentage of trips in simulated data at a given time frame for which the zone i is the origin zone.
- $N_{i,t,\%}^d$: Percentage of trips in input data at a given time frame for which the zone i is the origin zone.
- $N_{i,t,\%}^{\hat{d}}$: Percentage of trips in simulated data at a given time frame for which the zone i is the destination zone.

Equation 4.2 represent the error in the estimate of the percentage of trips for which a zone i is origin. Equation 4.3 represent the error in the estimate of the percentage of trips for which a zone i is destination. Equation 4.4 represents the sum of the two components. In figure 4.4 it is shown $N_{i,t,\%}^{\hat{d}}$ at four different moments of the day (night, morning, afternoon, evening). We can observe that the error never exceeds 3% for a single zone, and that it is more concentrated on central zones, presenting in general more trips.

As we were previously stating in the introductory subsection, we are not properly interested in reproducing the same spatial patterns of the trace: on the contrary, we are interested in partially uncorrelating simulated spatial patterns and original ones, yet maintaining spatial patterns reasonably compatible with the city under study and related human activities. The reason behind it, as we were introducing, is that we only have data about successful bookings, and we are not consequently able to model unsatisfied demand. Therefore, a model resembling closely input data may be too optimistic, as it would be based only on successful bookings.

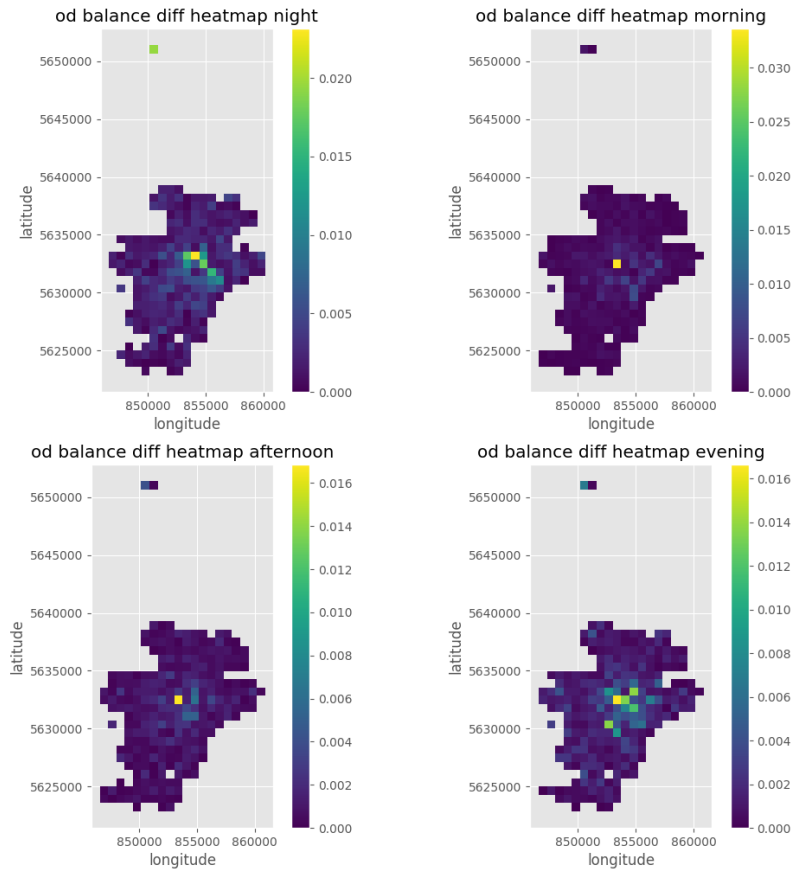


FIGURE 4.4: $\epsilon_{i,t,\%}^{od}$ for the city of Turin during night, morning, afternoon and evening

The maximum relative error for a single zone never exceeds 3%, and it is mostly around 1%.

In order to test how uncorrelated the simulated trace can be, we now run two simulations, one using the real trace and one using a trace coming from our demand model, in order to compare the percentage of satisfied demand. The simulations are tuned on an ideal (without cost estimation) scenario characterised by the following configuration:

- **requests_rate_factor**: 1
- **n_cars_factor**: 1
- **time_estimation**: False
- **queuing**: True
- **alpha**: 25
- **beta**: 100
- **hub**: True
- **hub_n_charging_poles**: 60
- **relocation**: False
- **distributed_cps**: False
- **user_contribution**: False

Figure 4.5 shows aggregated statistics of satisfied/unsatisfied requests for the simulation using the original trace, while in 4.6 are shown the same statistics for the simulation using the implemented demand model. We can notice how the percentage of satisfied demand drops significantly in the second case, as the input users requests lose their correlation with the position of cars.

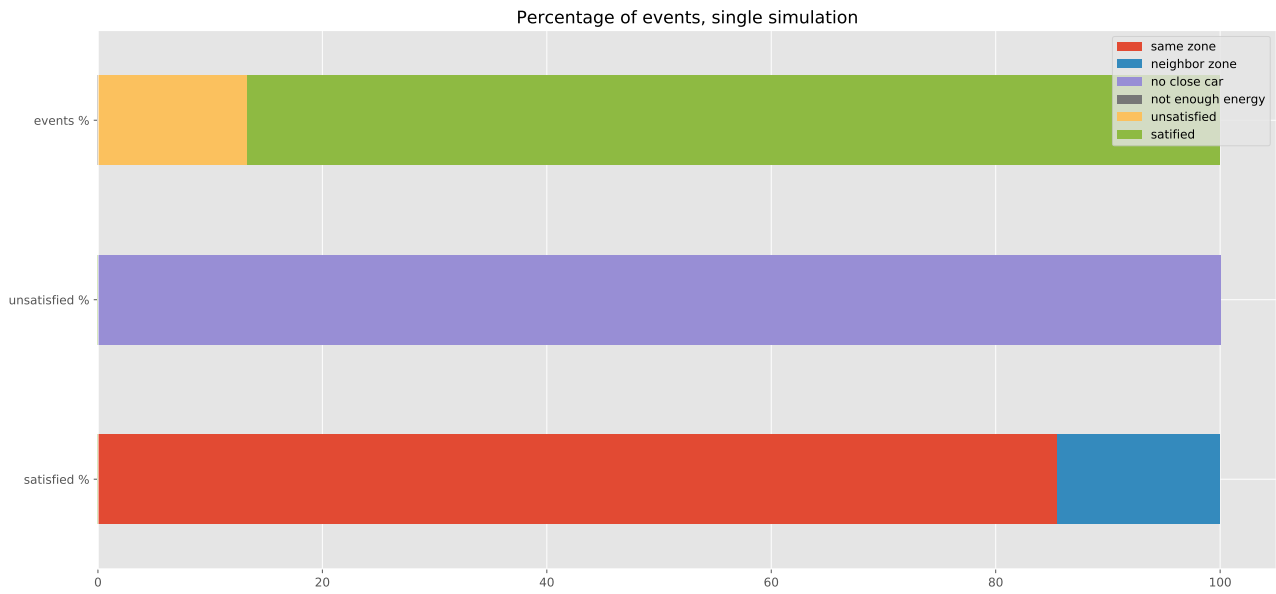


FIGURE 4.5: Events types percentages, original trace

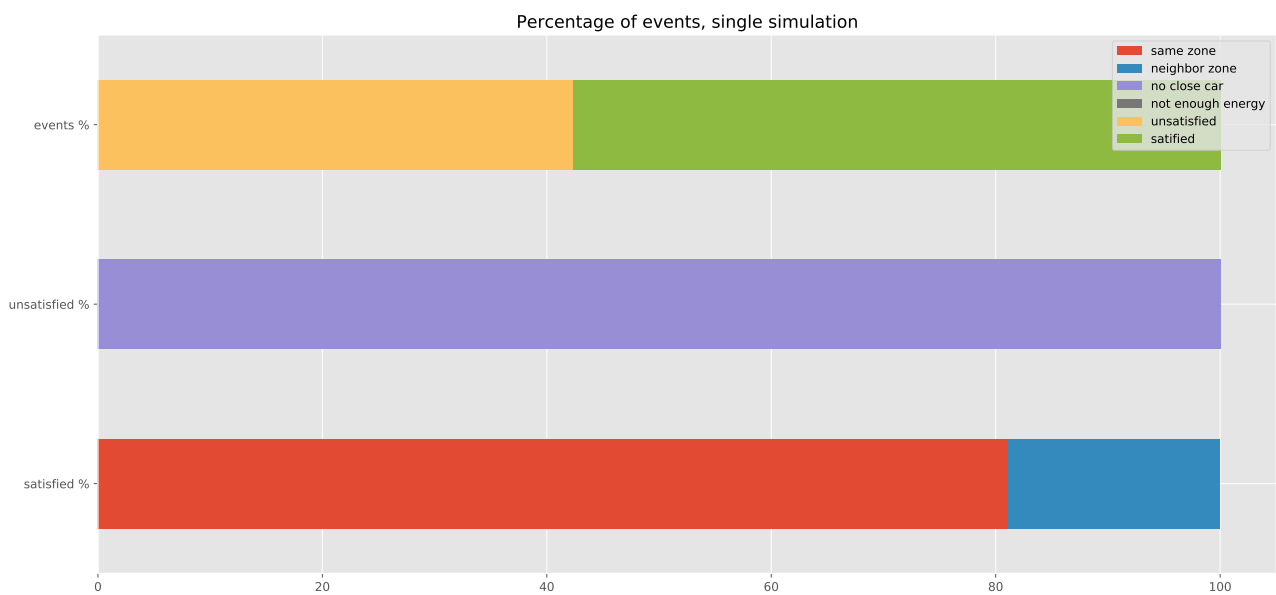


FIGURE 4.6: Events types percentages, simulated trace

The percentage of unsatisfied booking request using the input trace is between 15% and 20%, while in the case of simulated trace is above 40%, meaning that the demand model successfully uncorrelates cars and user requests locations.

Chapter 5

First case study: city of Turin

In this section, we will present simulation results for the city of Turin. First, we present a validation of the proposed mobility demand model. Then, we will present a simulation campaign for three different scenarios with multiple or specific configurations. The analysed scenarios are:

- Fully centralised charging infrastructure, without user contribution
- Fully decentralised charging infrastructure, with or without user contribution
- Hybrid charging infrastructure, with or without user contribution

The system parameters under study are:

- Number of cars in the fleet
- Charging capacity
- Upper charging threshold (β): a charging process is terminated when car's SOC reaches the value of β .
- Users' willingness to contribute (only with user contribution): probability that a user will contribute to charging processes when requested by system operator.

The quality of a configurations is measured using the following metrics:

- percentage of unsatisfied requests: gives an indication of the quality of the service in terms of cars availability for users' requests. An unsatisfied request is defined as a user request for which the user did not find an available car close to the origin of the request, or did not find a car with enough energy to perform the desired trip.
- Charging relocation cost: gives an indication of the cost of charging process in terms of time needed to drive cars to charge. When a car needs to be charged, the system has to physically move it to the closest charging point. Charging relocation cost measures the time needed to perform this move.

5.1 Single charging hub scenario

5.1.1 Parameters analysis

In this subsection, we will analyse results coming from a set of simulations for the city of Turin in which we imagine that the charging process is performed through a single, large charging hub. The location of the hub is set through the "num_parkings" strategy [36], namely the hub is located in the zone where the largest number of cars is usually parked. The parameters grid for this set of simulations is the following:

- **requests_rate_factor**: 1
- **n_cars_factor**: from 0.5 to 1.5, spaced by 0.2
- **time_estimation**: True
- **queuing**: True
- **alpha**: 25
- **beta**: from 60 to 100, spaced by 10
- **hub**: True
- **hub_zone_policy**: "num_parkings"
- **n_poles_n_cars_factor**: from 0.05 to 0.2, spaced by 0.005
- **relocation**: False
- **distributed_cps**: False

In figure 5.1 is shown the percentage of unsatisfied requests as a function of the number of cars per charging poles. The total number of cars is fixed to the 90% of the original fleet size and the different curves represent different values of β . In the left region of the chart, we can notice that β has little to no impact, probably due to the insufficient charging capacity of the system. The curves present a knee around the value of 15 cars per charging point ($n_poles_n_cars_factor = 0.07$) and stabilises progressively. We can observe that when there is enough charging capacity a lower value of β corresponds to a lower percentage of unsatisfied requests. This can be intuitively explained by the fact that cars stay in charge for less time on a average, which means they will also stay more time available in the city.

Figure 5.2 shows the relocation cost of charging operations in terms of hours needed for moving cars to charge. As before, the number of cars is fixed to the 90% of the original fleet size and the purpose is to study the impact of β . From the chart, it is clear that lower values of β correspond to higher relocation cost, because cars have to be charged more often. Furthermore, the relationship looks pretty linear: doubling the average charging time (i.e. doubling β) also the relocation cost doubles.

Let's now analyse the impact of fleet size. Figure 5.3 shows the percentage of unsatisfied requests as a function of the number of poles per car, with β set to 60 and different curves for different fleet size. As expectable, increasing the number of cars has a strong positive impact on unsatisfied demand. An interesting effect that this chart shows is also that increasing the number of cars the number of poles per car needed to reach the best operational region decreases. This means that, once that β and fleet size have been set, an increase in the absolute charging capacity is

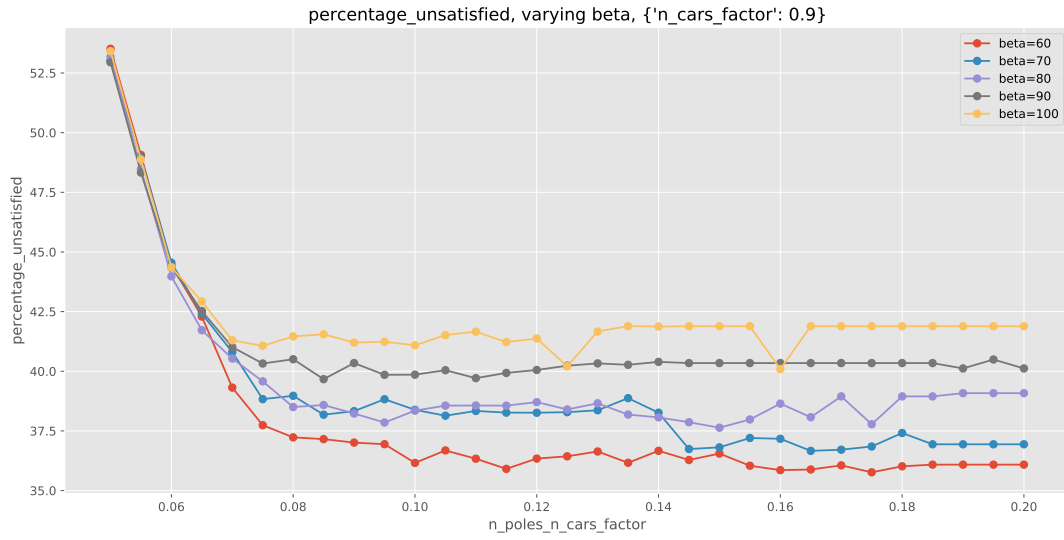


FIGURE 5.1: Percentage of unsatisfied events, n_cars_factor fixed, β variable

Reducing the value of β has the effect of making more cars available on average, therefore reducing the percentage of unsatisfied demand.

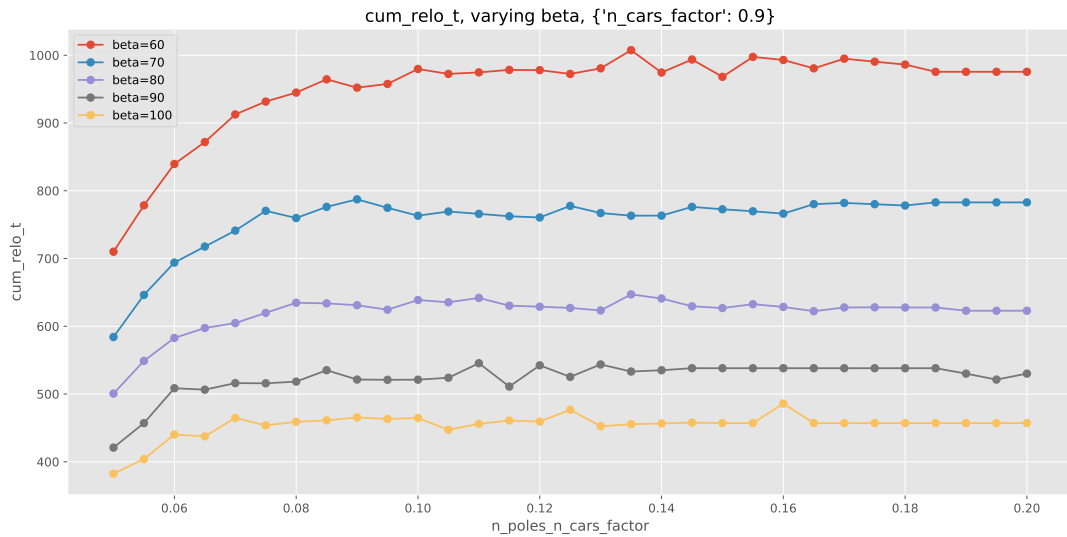


FIGURE 5.2: Relocation cost, n_cars_factor fixed, β variable

Reducing the value of upper charging threshold β has the effect of bringing cars to charge more often, therefore increasing charging relocation cost.

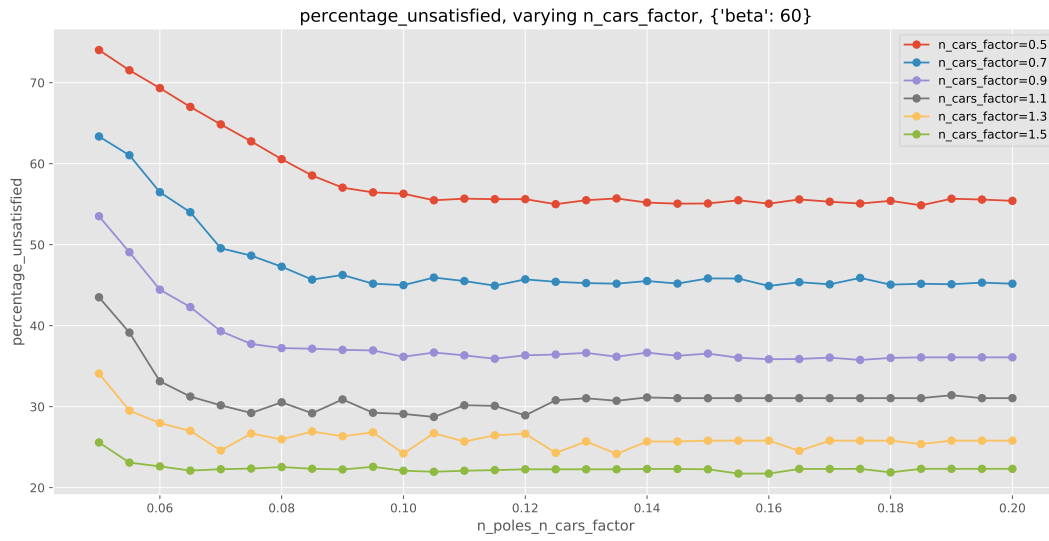


FIGURE 5.3: Percentage of unsatisfied events, β fixed, n_cars_factor variable

Increasing fleet size lets users find a car more likely, reducing the percentage of unsatisfied requests.

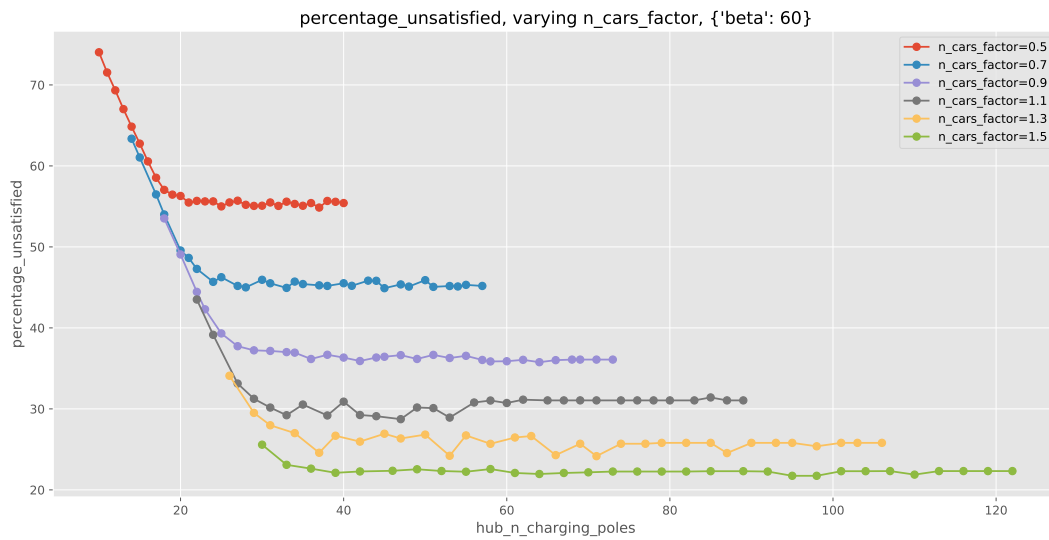


FIGURE 5.4: Percentage of unsatisfied events, β fixed, n_cars_factor variable, absolute number of charging poles

The absolute number of charging poles needed for running the system at its best increases along with fleet size.

not beneficial for the system. This effect is better visible by plotting the same chart against the absolute number of charging points, as in figure 5.4.

Figure 5.5 shows the relocation cost by plotting different curves for different fleet size. Increasing the number of cars increases also the relocation cost, likely because users will be able to perform more trips. Despite this, the effect of fleet size on the relocation cost seems to have less impact than the choice of β : doubling the number of cars, in fact, provokes an increase in the relocation cost which is less than the double.

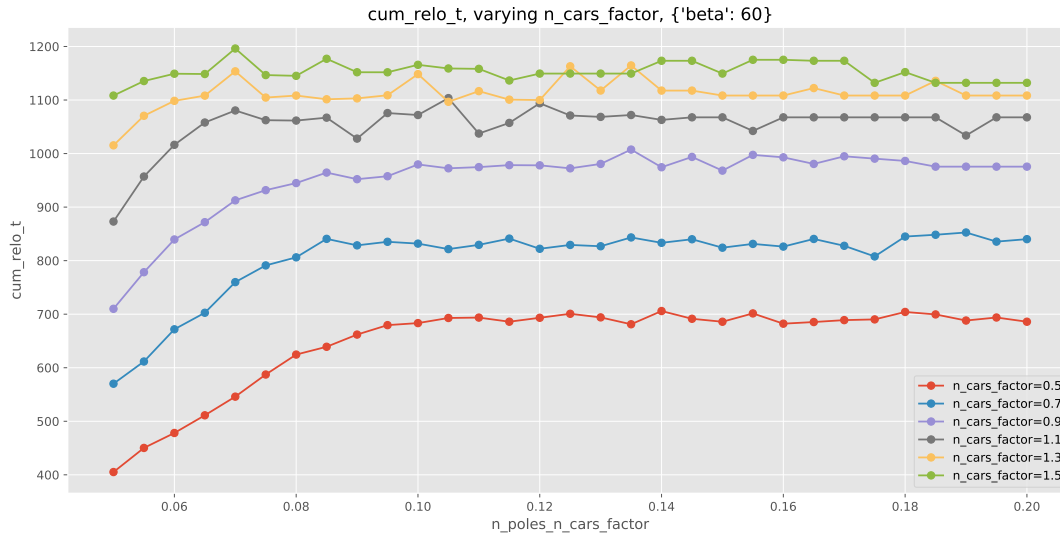


FIGURE 5.5: Relocation cost, β fixed, $n_{\text{cars_factor}}$ variable

The relocation cost of bringing cars to charge grows along with fleet size, but tends to grow less sharply when the fleet is bigger.

We can therefore deduce that, in this scenario, a positive correlation between the quality of service and the charging relocation cost exists: the best service from the users' point of view (i.e. minimum percentage of unsatisfied requests) is reached when the costs for the system are the highest, and viceversa.

The best configuration in terms of satisfied demand is composed by the following parameters:

- $\beta = 60$
- $n_{\text{cars_factor}} = 1.5$

Figure 5.6 shows the percentage of satisfied and unsatisfied demand for this optimal configuration as a function of $n_{\text{cars_n_poles_factor}}$. We can observe that satisfied demand reaches almost 80% of requests, with about 60% being "comfortable" trips (the car is found in the same zone where the request was made) and 20% "less comfortable" trips (the car is found within a neighbor zone). We can also notice that the most of unsatisfied demand, globally reaching slightly more than 20%, is given by users not finding a close car, while cases for which users found car without enough energy happen more rarely.

The best configuration in terms of relocation cost is composed by the following parameters:

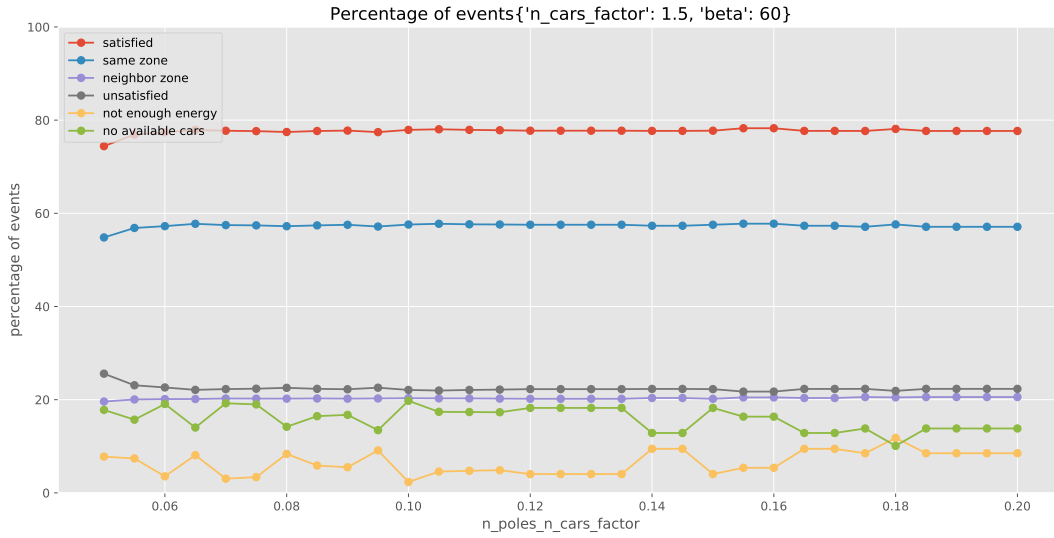


FIGURE 5.6: Events profile, minimum unsatisfied demand configuration

Once a good configuration of fleet size and charging threshold has been found, excess charging capacity has little impact on system performance.

- $\beta = 100$
- $n_cars_factor = 0.5$

Figure 5.7 shows the percentage of satisfied and unsatisfied demand for the minimum cost configuration as a function of $n_cars_n_poles_factor$. Satisfied demand is in this case only constituted by the 40% of requests, with about 30% being "comfortable" trips and 10% "less comfortable" trips. Once again, the most of unsatisfied demand is given by users not finding a close car, while cases for which users found car without enough energy happen in less than 10% of requests.

5.1.2 In-depth analysis

In this subsection, we will present the results of a single simulation run for the case of a fully centralised infrastructure in the city of Turin. The particular configuration chosen is the following:

- **requests_rate_factor:** 1
- **n_cars_factor:** 1
- **time_estimation:** True
- **queuing:** True
- **alpha:** 25
- **beta:** 80
- **hub:** True

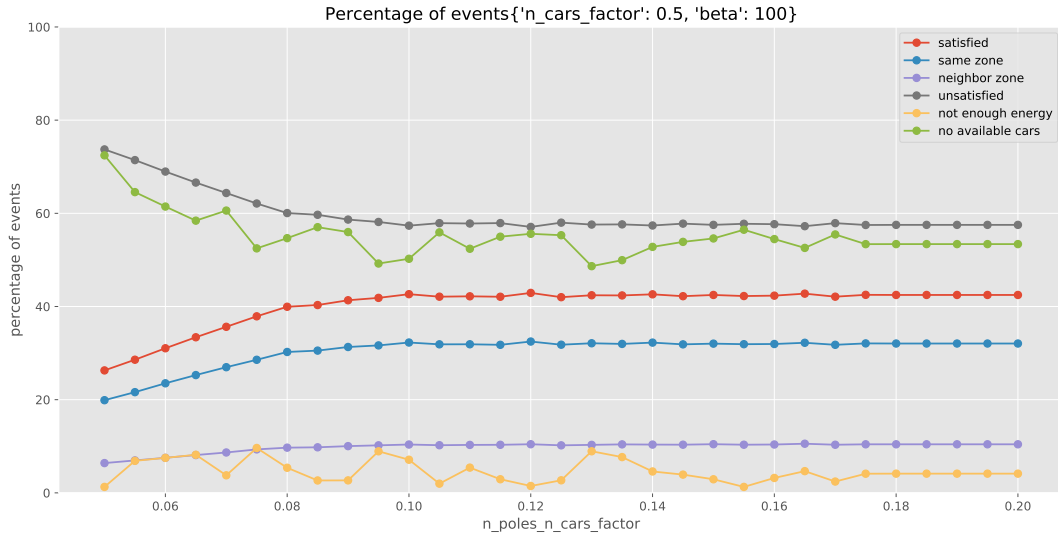


FIGURE 5.7: Events profile, minimum charging relocation cost configuration

Minimising charging relocation cost without taking into account other factors lead to a very inefficient system for users.

- **hub_zone_policy:** "num_parkings"
- **n_poles_n_cars_factor:** 0.1
- **relocation:** False
- **distributed_cps:** False

Figure 5.8 shows the location of the hub selected through the "num_parkings" strategy. We can consider this being a best case, ideal scenario, because in many practical situation will be impossible to build a big charging station in the very center of a city. However, eC2S allows to manually set the location of the hub in order to study other specific situations.

In figure 5.9 is represented an horizontal barplot with the percentages of successful and unsuccessful user requests. We can see that slightly more than 60% of user requests have been satisfied, while almost 40% corresponds to unsuccessful requests. Almost 70% of unsatisfied requests happened because no close car was found, while the remaining failed because close cars did not have enough energy to serve the request. We can also see that in almost 25% of successful requests, users had to walk to a neighbor zone in order to reach a car able to serve their trips.

In figure 5.10 is represented the hourly profile of charging events that the system has to manage. In particular, the chart refers to when a charging event starts. These events can be seen as notifications of the fact that a car reached a SOC under the level set by α . The peaks of charging requests happens at 5pm to 6pm, with the system having to manage an average of 8 charging events, with peaks of 14 and 16. During the morning and afternoon the average stays quite steady around 6 charging events. After 6pm, the number of charging events to manage start to decrease, and reaches a lower peak of 0 to 1 charging events during night at about 2am. We can see that

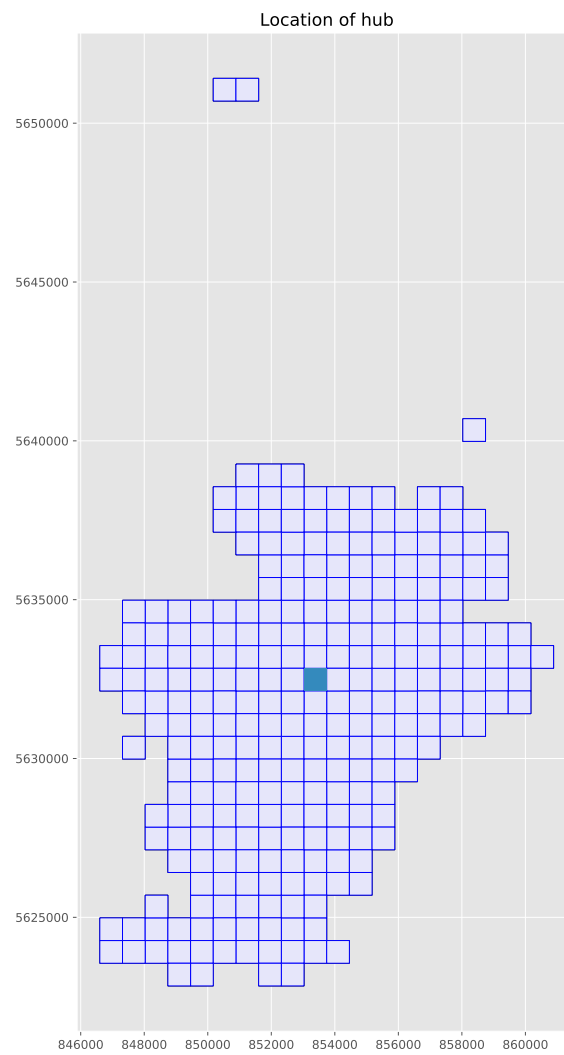


FIGURE 5.8: Hub location

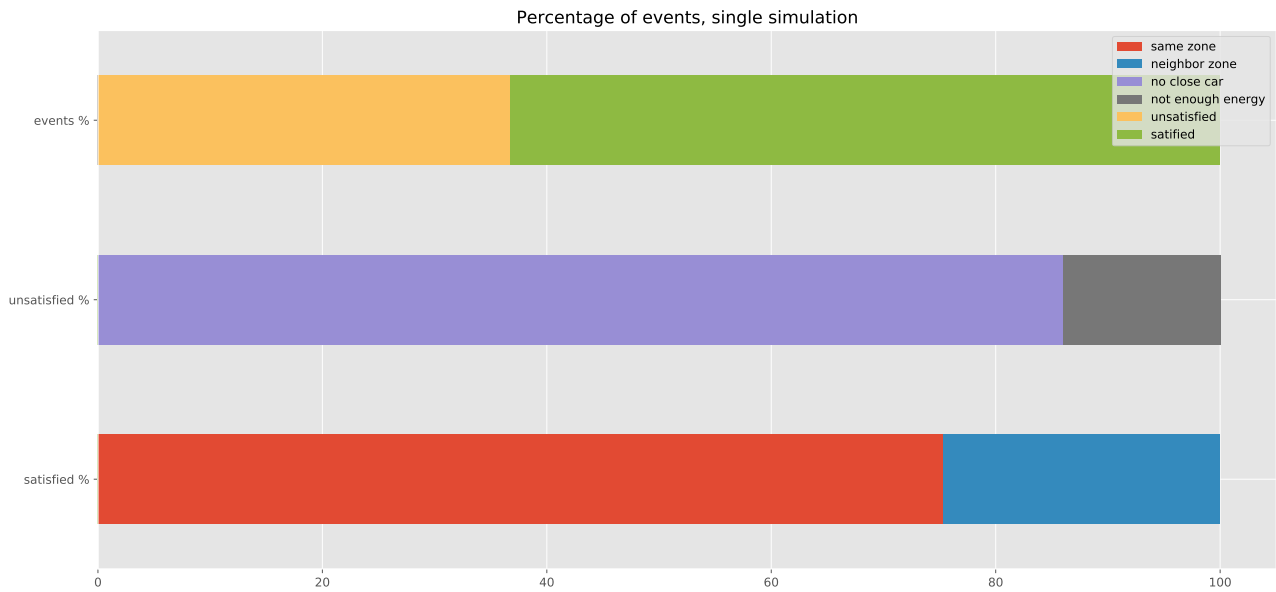


FIGURE 5.9: Events types percentages

it happens quite often that the number of charging events to manage in an hour is greater than 10 and, in a few cases, around 15 or more.

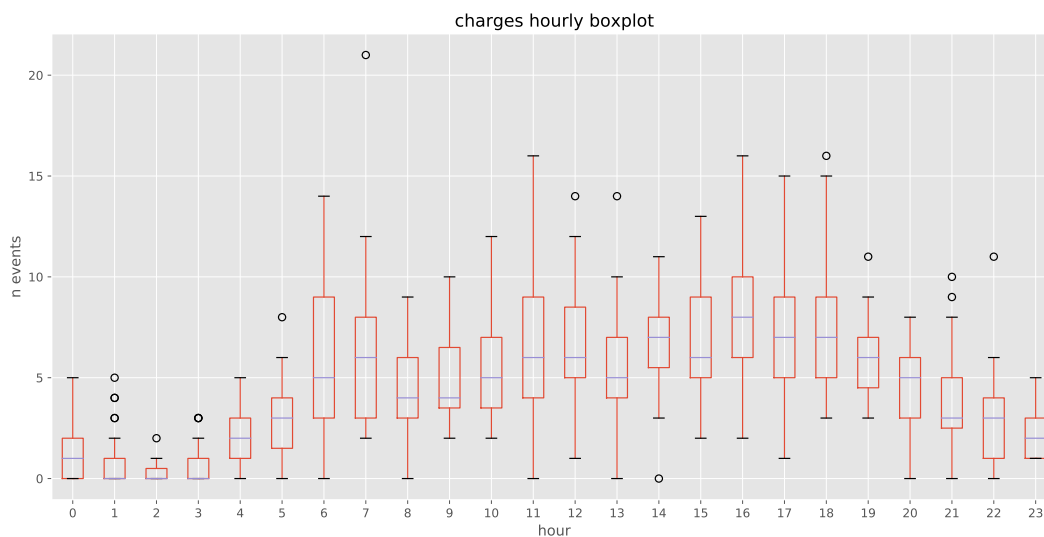


FIGURE 5.10: Hourly boxplot of number charging events managed by the system

In figure 5.11 is represented the number of cars charging at a given hour. We can observe that the average of each hour is below the maximum capacity of the system, but the upper quartiles are saturated almost all day long, exception made for the interval 1am-9am.

In figure 5.12 is represented the evolution in time of cars' status during simulation. We can notice that on a average there are about 350 cars constantly available,

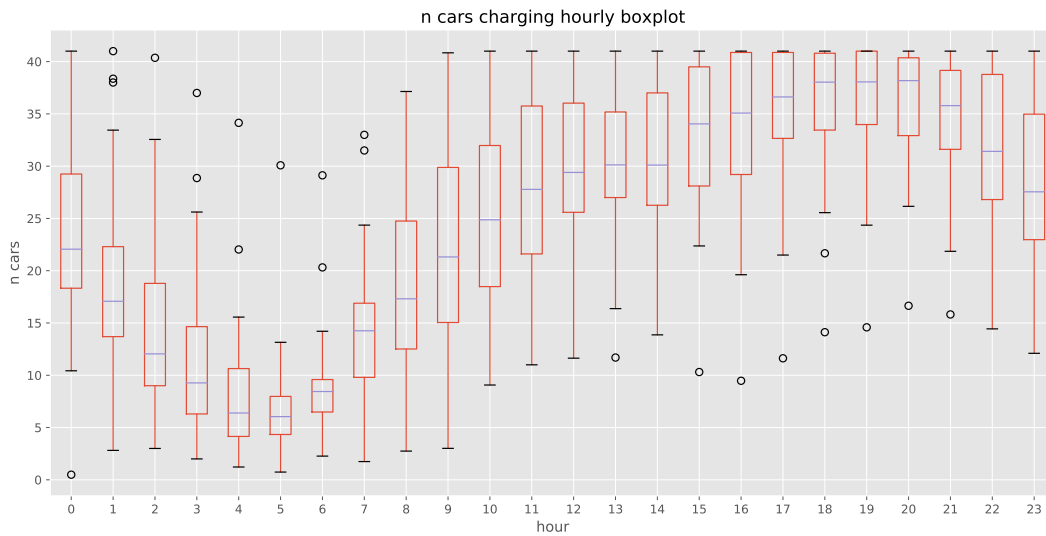


FIGURE 5.11: Hourly boxplot for number of cars charging

which can be a sign of inefficiency in fleet management. We can therefore think that it could be possible to maintain similar level of service even with a lower number of cars in the fleet. We can also notice the the system accumulates progressively "dead" cars, namely cars which are too far from system's charging infrastructure to be re-located. It would be probably necessary to implement "rescue" strategies for those cars.

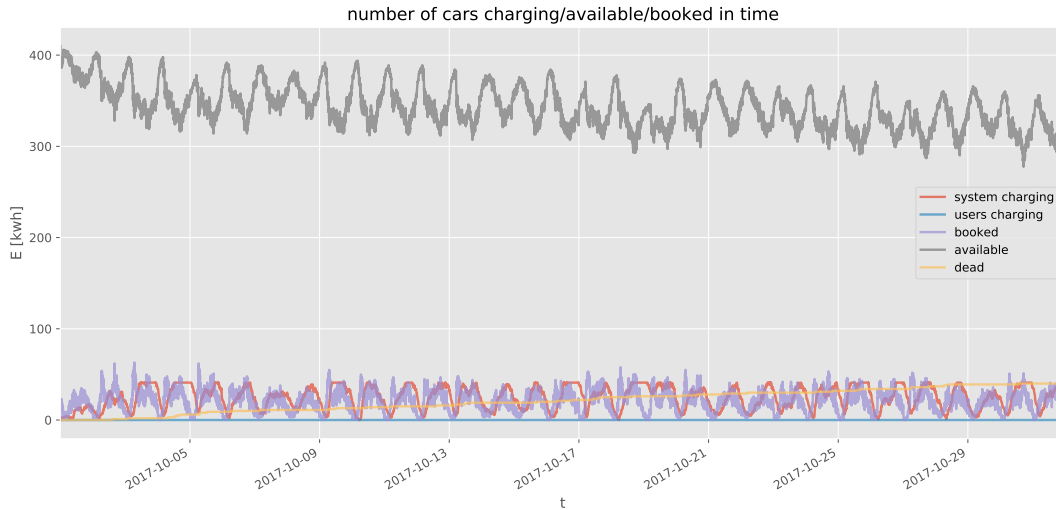


FIGURE 5.12: Fleet status in time

Figure 5.13 shows the charging relocation cost in terms of time needed for moving the cars to charging stations, as we are not considering post-charging relocation. We can see that the average between 6am and 6pm is around one hour, namely in

each hour the system need about one hour driving to bring cars to charge. It is also visible, however, that in many cases the upper quartiles are between 1 hour and 2 hours, and that there are also fewer cases in which the relocation cost has been more than two hours. In figure 5.14 and 5.15 are shown respectively the overall time and space distributions of charging requests. We can notice that charging relocation activity is more concentrated during daily hours and in the center. However, the airport of Caselle is the zone with the biggest number of charging requests (about 3% of the total), due to its geographical position.



FIGURE 5.13: Hourly boxplot for relocation cost expressed in hours

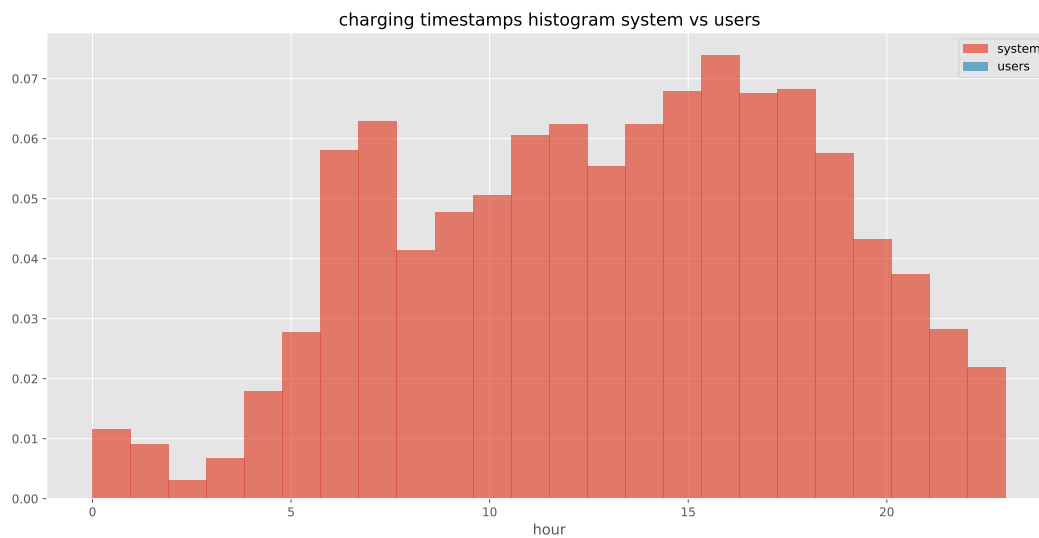


FIGURE 5.14: Histogram of charging events start timestamps

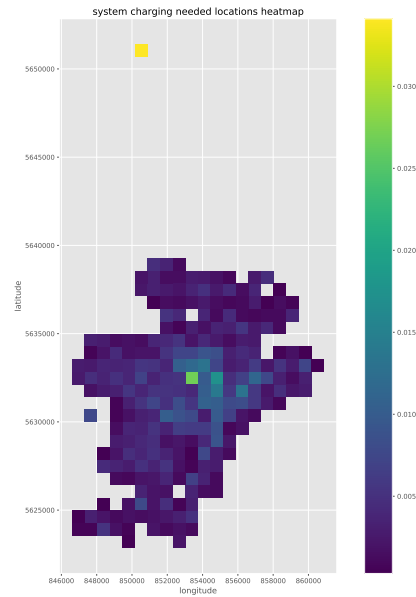


FIGURE 5.15: Clorophlet map of charging requests locations

Figure 5.16 shows the absolute hourly distribution for unsatisfied requests. We can see that the peak of unsatisfied requests is reached between 3pm and 5pm, when the average number of unsatisfied requests is about 70, with peaks around or more than 80. In relative terms, as it shown in figure 5.17, this corresponds to about 25% of the whole unsatisfied demand. Figure 5.18 shows instead the spatial distribution of unsatisfied demand, which is visibly concentrated in the center of the city.

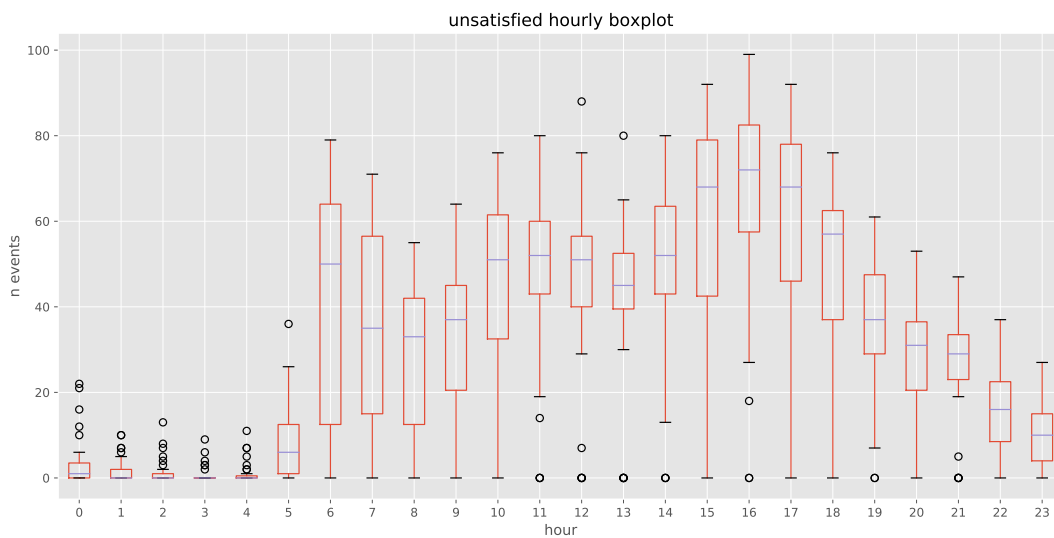


FIGURE 5.16: Hourly boxplot for unsatisfied requests

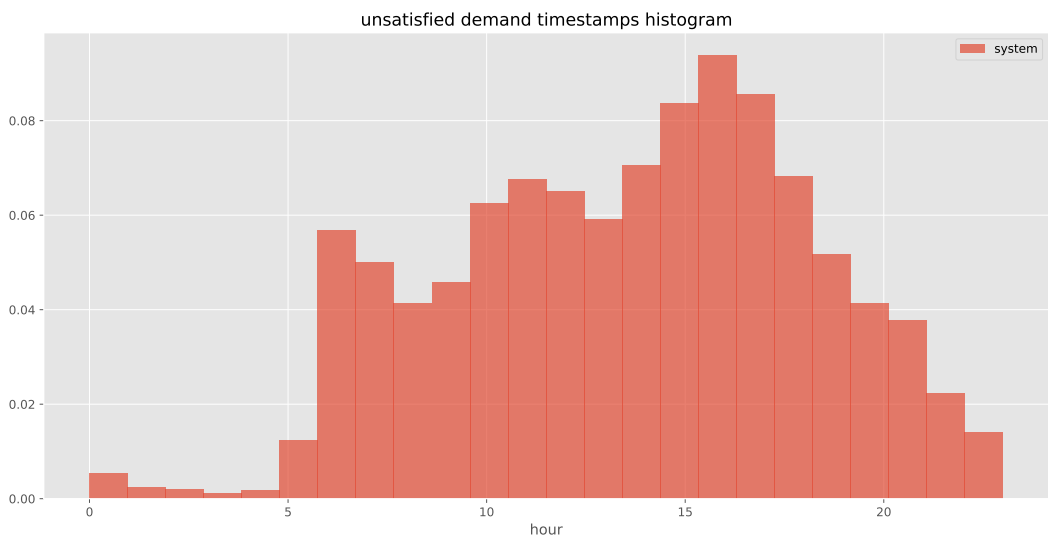


FIGURE 5.17: Histogram of unsatisfied requests timestamps

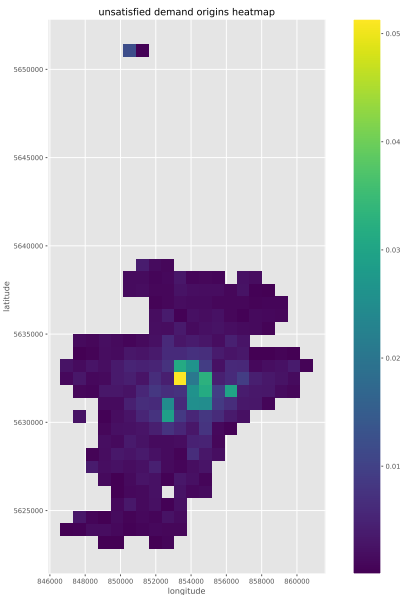


FIGURE 5.18: Clorophlet map of unsatisfied requests locations

5.2 Distributed infrastructure scenario

5.2.1 Parameters analysis

In this subsection, we will analyse results coming from a set of simulations for the city of Turin in the case that the charging process is performed through a distributed charging infrastructure. Charging point placement is decided through the "num_parkings" strategy [36], namely there will be more charging capacity in the zones where an higher number of cars is usually parked. The general assumption for charging relocation in this case is that the system brings a car to charge in the closest charging spot, and the car enters a queue if the charging spot is full. The parameters grid for this set of simulations is the following:

- **requests_rate_factor**: 1
- **n_cars_factor**: from 0.5 to 1.5, spaced by 0.2
- **time_estimation**: True
- **queuing**: True
- **alpha**: 25
- **beta**: from 60 to 100, spaced by 10
- **hub**: False
- **n_poles_n_cars_factor**: from 0.05 to 0.2, spaced by 0.005
- **relocation**: False
- **distributed_cps**: True
- **cps_placement_policy**: "num_parkings"
- **willingness**: from 0 to 0.99, spaced by 0.33

First, we want to investigate how a distributed infrastructure might impact system operations in comparison with a centralised hub. In order to do this, let's first plot event percentages profile with the best parameters of the centralised case, namely the minimum unsatisfied demand configuration, in figure 5.19, and the minimum relocation cost configuration, in figure 5.20. Surprisingly, performances in terms of satisfied demand are consistently worse (respectively 30% and 20%) than in the centralised case with the same parameters. This effect may be due, for example, to an imbalance in queues at charging stations, but needs further investigation to be properly characterised.

Let's now analyse the effect on users' willingness to contribute on charging process. We assume to adopt the "free floating" strategy [36], in which a user plug a car at destination zone only if there is an available charging spot in the same zone. Furthermore, a car plugged by users remains in charge until it reaches a SOC equal to β . These assumptions can be changed in the simulator to see how different strategies may impact performance metrics.

The best configuration in terms of satisfied demand is composed by the following parameters:

- $\beta = 70$

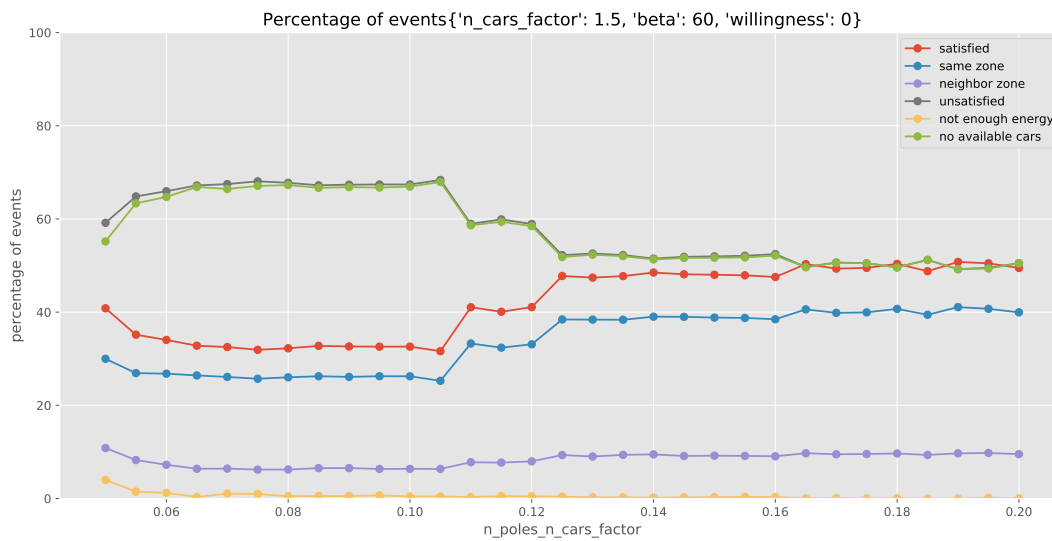


FIGURE 5.19: Events profile, no users, best centralised configuration

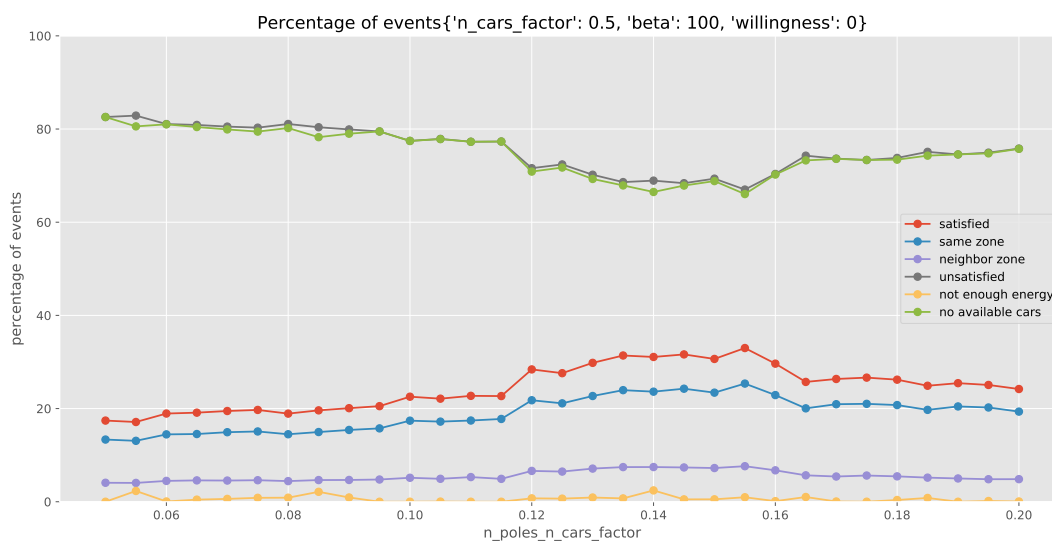


FIGURE 5.20: Events profile, no users, best centralised configuration

- $n_cars_factor = 1.5$

The best configuration in terms of relocation cost is composed by the following parameters:

- $\beta = 100$
- $n_cars_factor = 1.3$

Figure 5.21 shows the impact of users' willingness on the percentage on unsatisfied demand. Generally, as expectable, a greater willingness corresponds to a lower percentage of unsatisfied requests, but this difference decreases when the total charging capacity increases. This effect is stronger for higher values of β , as shown in figure 5.22, corresponding to the minimum relocation cost configuration.

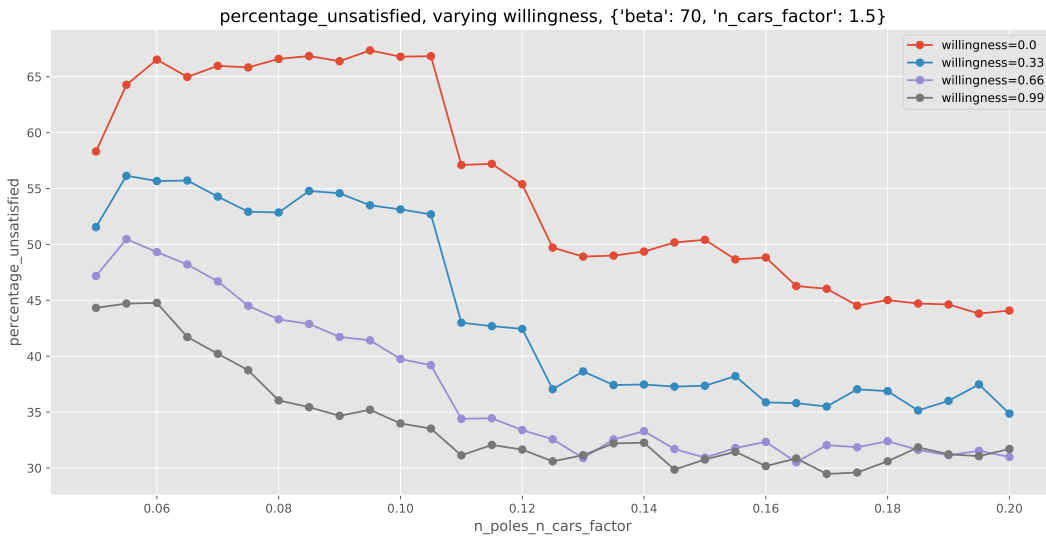


FIGURE 5.21: Percentage of unsatisfied events, best case configuration, varying willingness

Figures 5.23 and 5.24 show the impact of users' willingness on the total relocation cost, respectively for the minimum unsatisfied demand configuration and the minimum relocation cost configuration. From both figures it is clear that user contribution might consistently contribute to a reduction in overall relocation costs, with a global minimum of only 45 hours of relocation in a month, in case of full users contribution.

Despite the big improvements in terms of relocation costs, the percentage of satisfied requests does not reach the best levels obtained in the centralised scenario, as shown in figure 5.25. This result can however be biased due to the fact that the hub is placed in an optimal zone, namely where most of the trips' origins and destinations are. Results might be different with another hub placement policy.

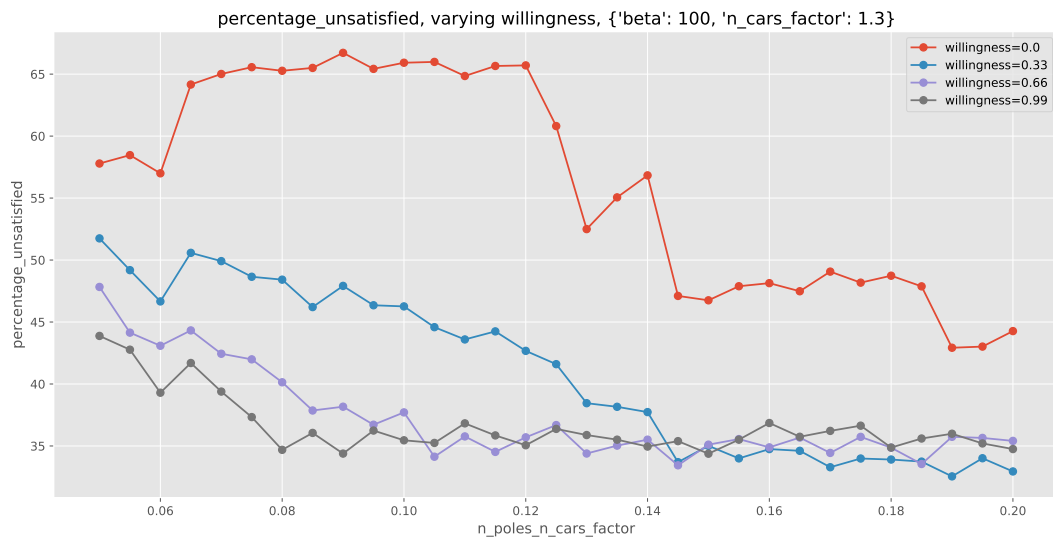


FIGURE 5.22: Percentage of unsatisfied events, best case configuration, varying willingness

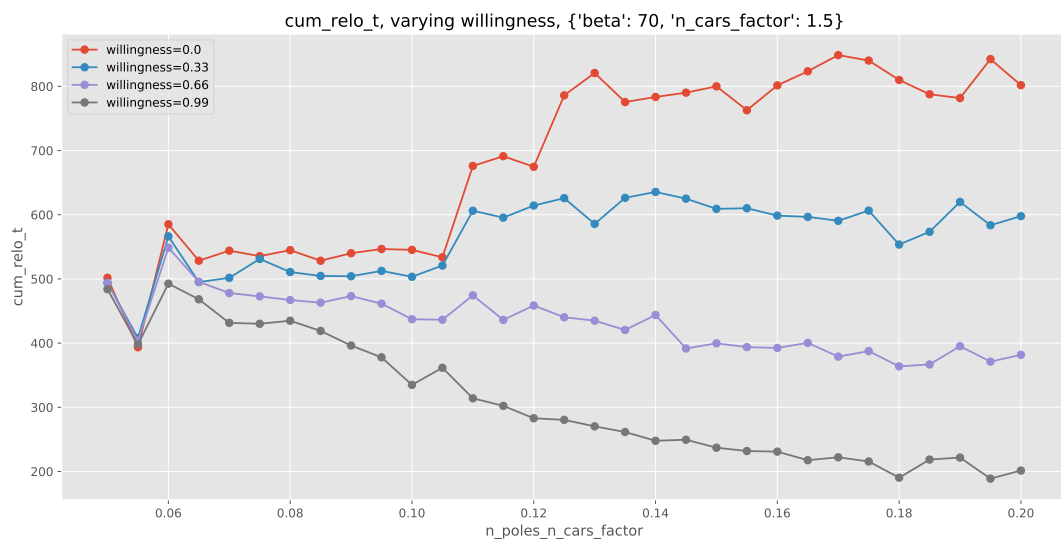


FIGURE 5.23: Percentage of unsatisfied events, best case configuration, varying willingness

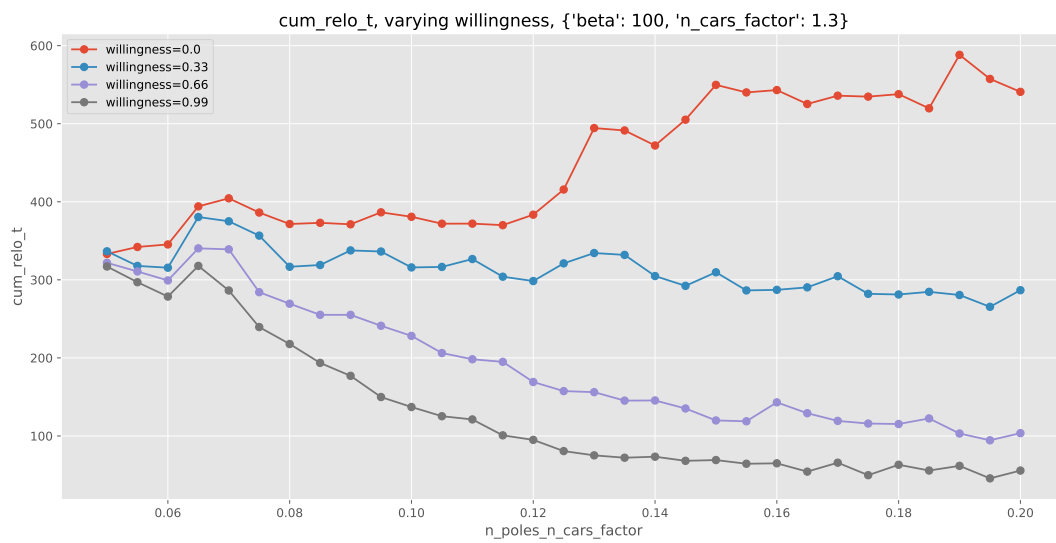


FIGURE 5.24: Percentage of unsatisfied events, best case configuration, varying willingness

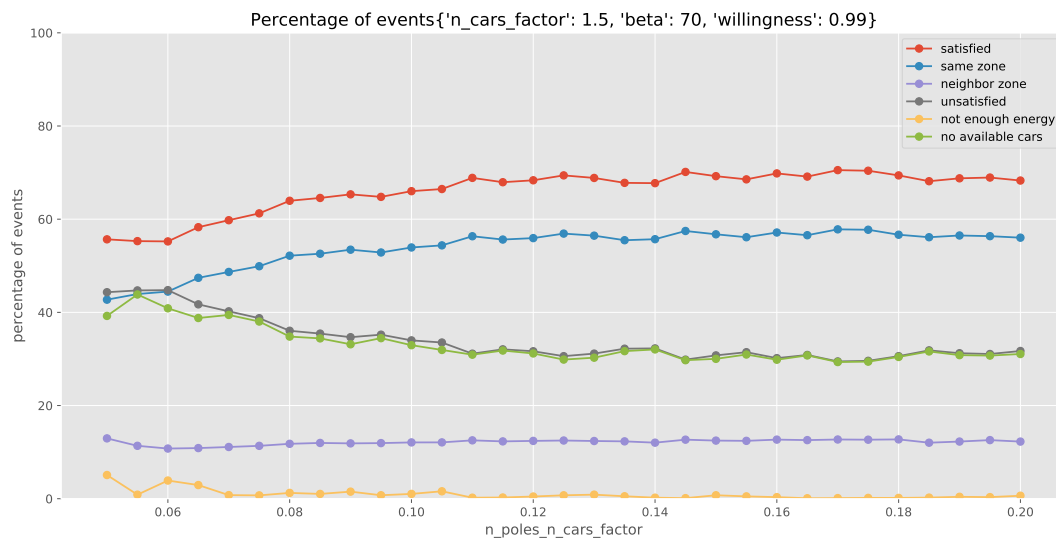


FIGURE 5.25: Events profile, best satisfied demand configuration

5.2.2 In-depth analysis

In this subsection, we will present the results of a single simulation run for the case of a decentralised charging infrastructure in the city of Turin. The particular configuration chosen is the following:

- **requests_rate_factor**: 1
- **n_cars_factor**: 1.3
- **time_estimation**: True
- **queuing**: True
- **alpha**: 25
- **beta**: 100
- **hub**: False
- **n_poles_n_cars_factor**: 0.1
- **distributed_cps**: True
- **user_contribution**: True
- **willingness**: 0.66

Figure 5.26 shows the locations of zones having at least one charging poles, selected through the "num_parkings" strategy.

In figure 5.27 is represented an horizontal barplot with the percentages of successful and unsuccessful user requests. We can see that the profile is quite similar to the one of the centralised case shown in figure 5.9, exception made for the percentage of unsatisfied requests caused by cars with no enough SOC. In the distributed case, the percentage of this events is around 2%, while in the centralised charging scenario was more than 15%.

In figure 5.28 is represented the hourly profile of charging event happening across the distributed charging infrastructure. Contrarily to the case of centralised charging shown in 5.10, where the peaks of charging requests happens at 5pm to 6pm, the highest number of charging events triggers can be observed around 6am. This is very likely due to the presence of user charging, as during morning users are more often moving to the center, where most of charging poles are, and therefore it is possible to apply the "free floating" charging strategy. It is also possible to observe a general increase in the number of charging events in comparison with the centralised case, where there are only charging events managed by the system.

In figure 5.29 is represented the number of cars charging at a given hour for charging events managed by the system. The profile is much smoother than the one of the centralised case shown in figure 5.11, and does not present evident peaks. This effect can be probably more advantageous for logistic operations. In figure 5.30 is shown the chart for user-managed charging events, and it confirms the intuition made for figure 5.28, namely that the peak in morning charging events is mostly due to users mobility patterns and the "free floating" charging strategy.

In figure 5.31 is represented the evolution in time of cars' status during simulation. The profile has two main differences with respect to the centralised charging case. First, the greater number of user-managed charging events than system-managed ones, already observed in figures 5.29 and 5.30. Second, the weaker trend of "dead" cars accumulation, and therefore lower "rescue" costs.

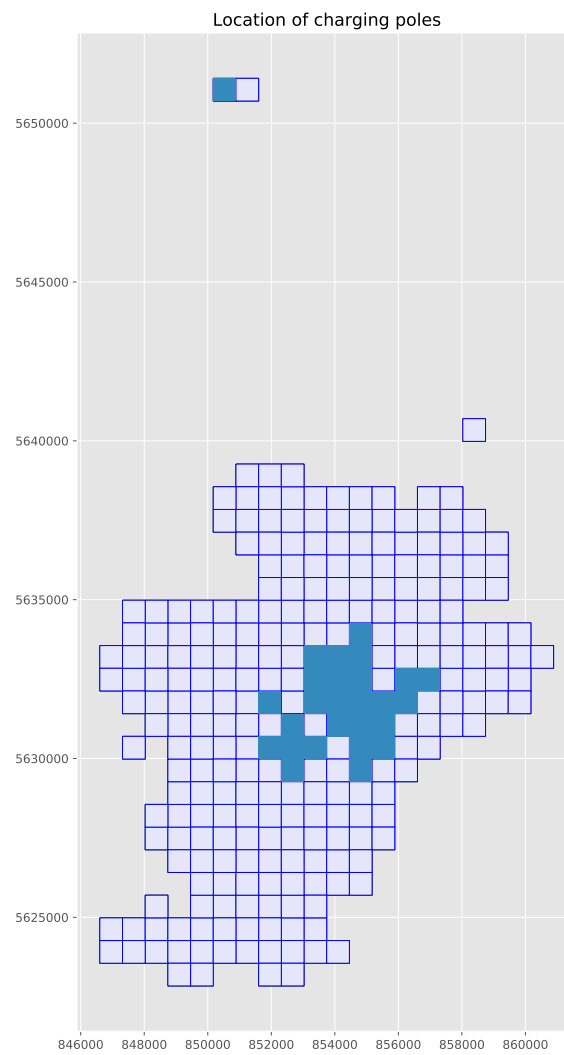


FIGURE 5.26: Charging zones location

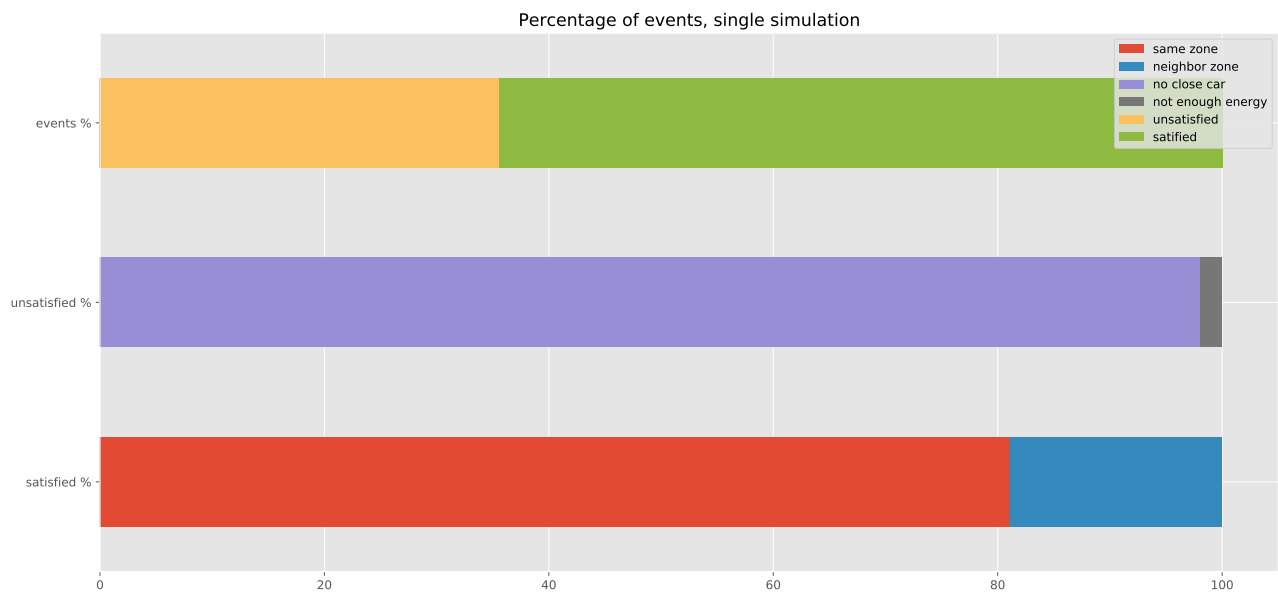


FIGURE 5.27: Events types percentages

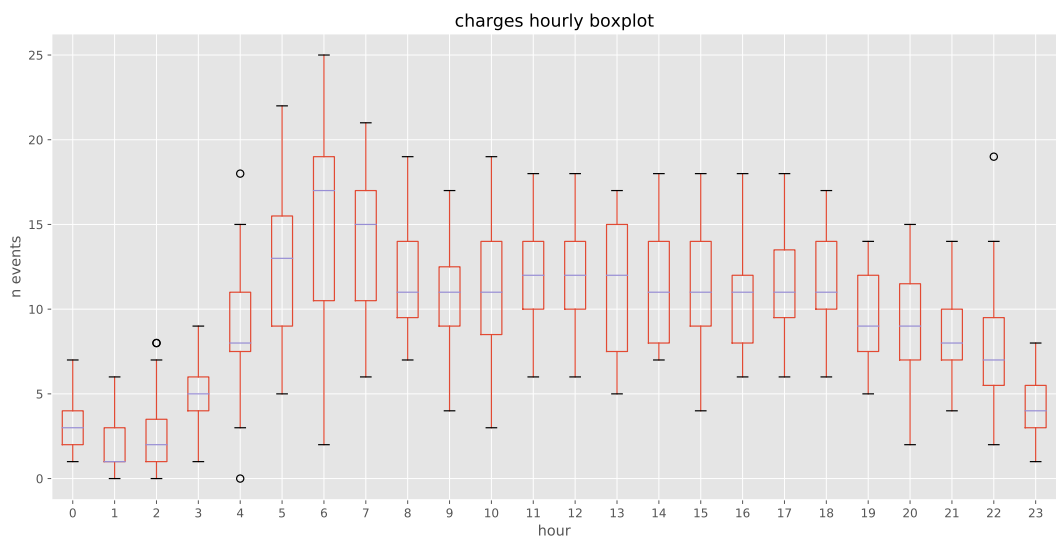


FIGURE 5.28: Hourly boxplot of number charging events

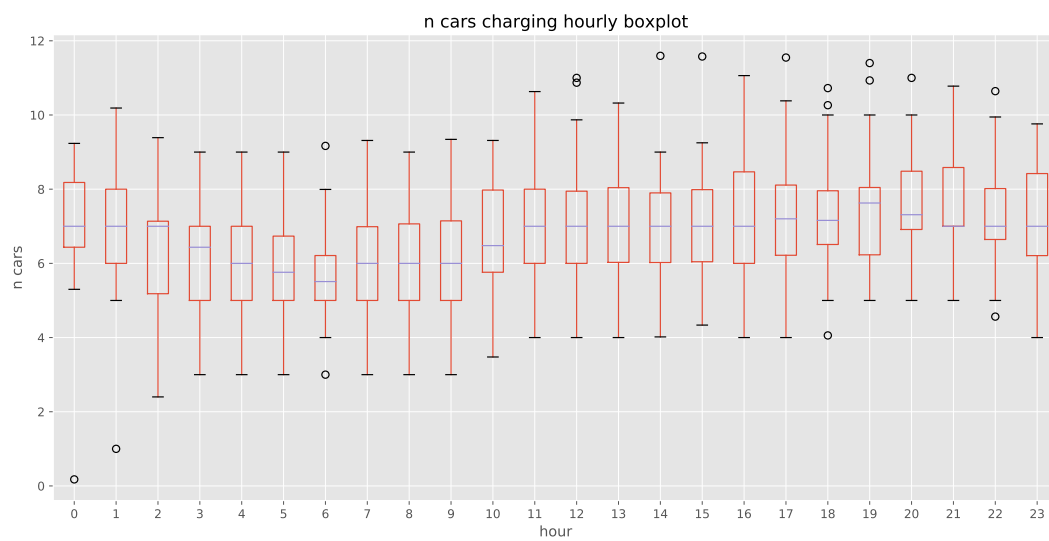


FIGURE 5.29: Hourly boxplot for number of cars charging in events managed by the system

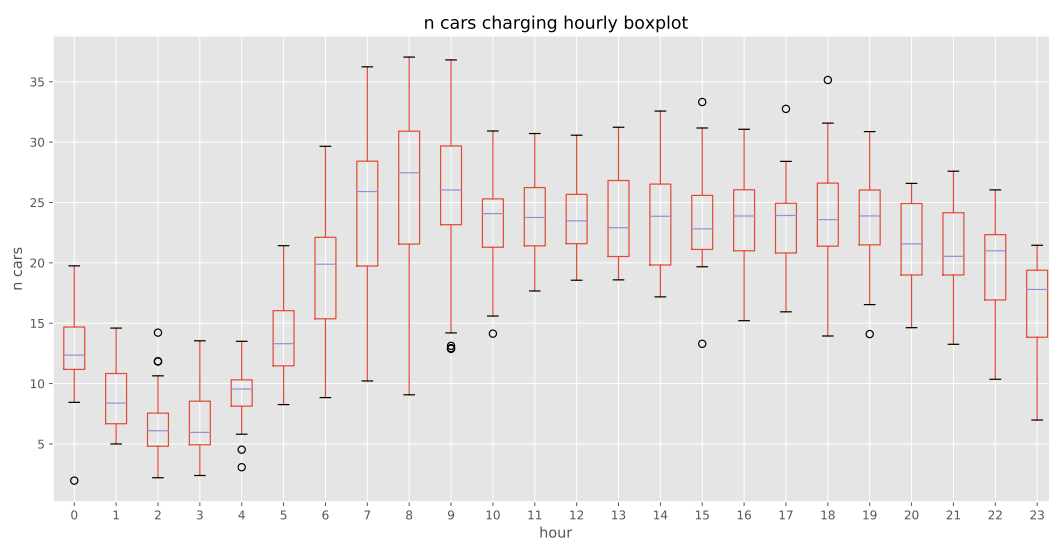


FIGURE 5.30: Hourly boxplot for number of cars charging in events managed by users

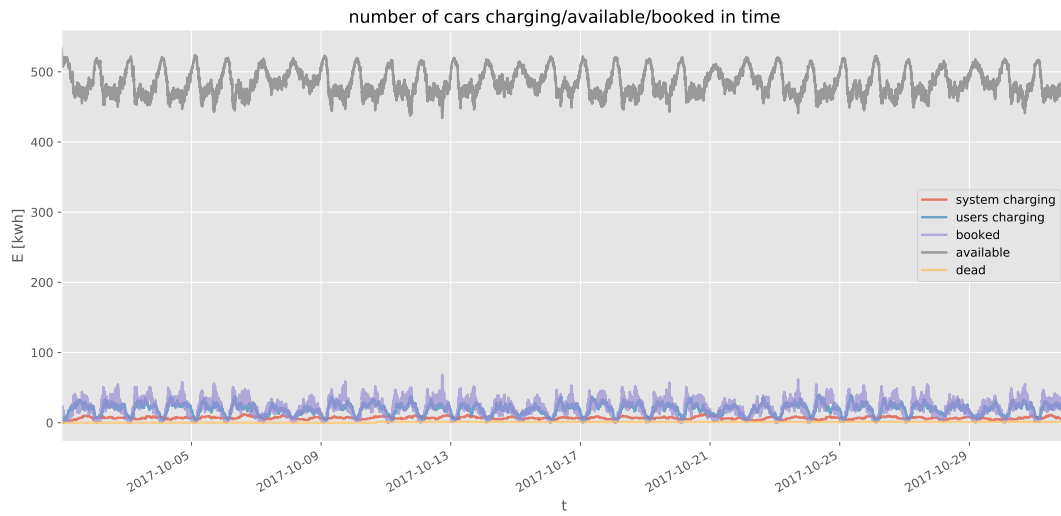


FIGURE 5.31: Fleet status in time

Figure 5.32 shows the charging relocation cost in terms of time needed for moving the cars to charging stations. We can observe a dramatic reduction of relocation time in comparison with the centralised case shown in figure 5.13, both in terms of average and peaks. It is also notable that, during the timeframe going from 1am to 4am, the time needed for relocation is very close to zero, exception made for some outliers.

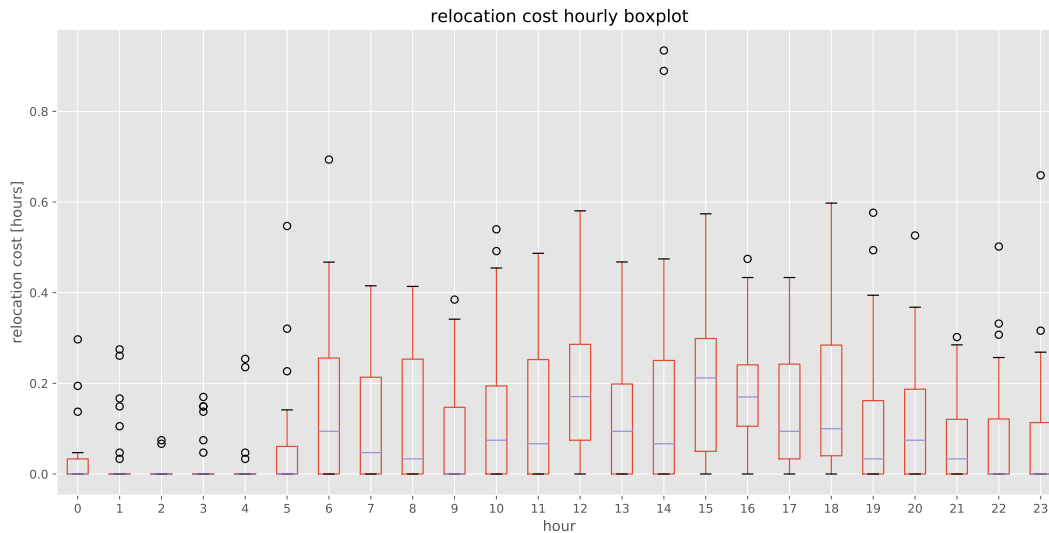


FIGURE 5.32: Hourly boxplot for relocation cost expressed in hours

5.3 Charging hub and distributed infrastructure scenario

5.3.1 Parameters analysis

In this subsection, we will analyse results coming from a set of simulations for the city of Turin in which the charging infrastructure is composed by a centralised hub, managed by a system operator, and a distributed infrastructure where only users can plug cars. The parameters grid chosen for this set of simulations is the following:

- **requests_rate_factor**: 1
- **n_cars_factor**: from 0.5 to 1.5, spaced by 0.2
- **time_estimation**: True
- **queuing**: True
- **alpha**: 25
- **beta**: from 60 to 100, spaced by 10
- **hub**: True
- **hub_zone_policy**: "num_parkings"
- **n_poles_n_cars_factor**: from 0.05 to 0.2, spaced by 0.005
- **relocation**: False
- **distributed_cps**: True
- **cps_placement_policy**: "num_parkings"
- **willingness**: from 0.33 to 0.99, spaced by 0.33

In order to analyse the quality of this scenario, we will once again consider the best configuration in terms of satisfied demand and the best configuration in terms of relocation cost. The best configuration in terms of satisfied demand is composed by the following parameters:

- $\beta = 60$
- **n_cars_factor** = 1.5
- **willingness** = 0.33

The best configuration in terms of relocation cost is composed by the following parameters:

- $\beta = 100$
- **n_cars_factor** = 1.3
- **willingness** = 0.99

Figure 5.33 shows the impact of users' willingness on the percentage on unsatisfied demand. For lower charging capacities, a greater willingness corresponds to a lower percentage of unsatisfied requests, but interestingly this effect is reversed for higher charging capacities: a greater user contribution corresponds to a lower percentage of satisfied demand. This is probably due to the fact that cars plugged by users remain unavailable until they reach the threshold β , and therefore when charging capacity increases more cars stay unavailable. Also, this trend tends to strengthen with higher values of charging capacity and also with higher values of β , as observable in figure 5.34, corresponding to the minimum relocation cost configuration.

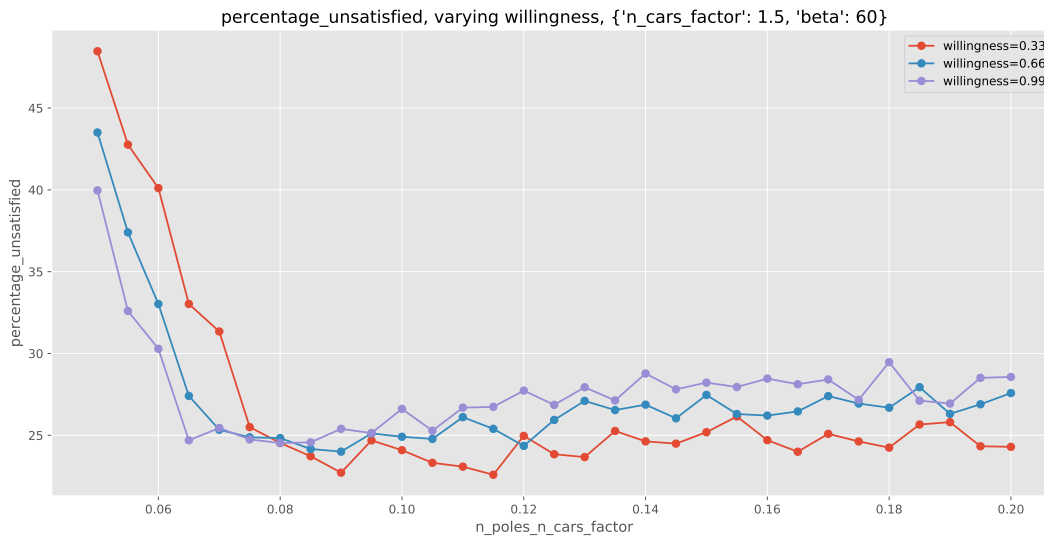


FIGURE 5.33: Percentage of unsatisfied events, best case, varying willingness

Figures 5.35 and 5.36 show the impact of users' willingness on the total relocation cost, respectively for the minimum unsatisfied demand configuration and the minimum relocation cost configuration. As for the case of fully decentralised infrastructure, it is clear that users contribution might play a decisive role in relocation cost reduction. Figure 5.37 shows instead the events percentage profile in the best satisfied demand case, which is very similar to the fully centralised case, a part from a more evident transient in case of low charging capacity.

For this scenario, we do not show the in-depth analysis as results are very similar to what observed previously.

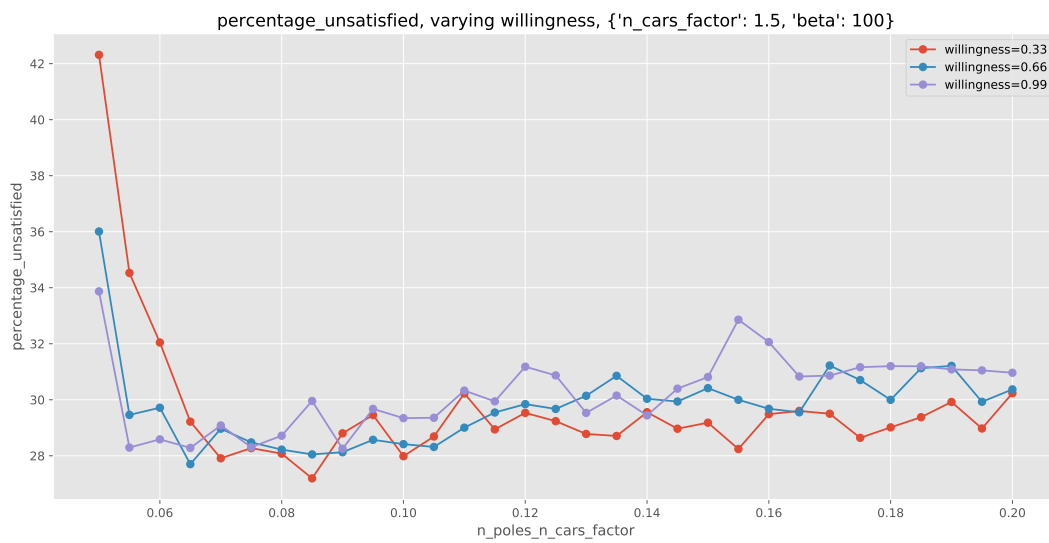


FIGURE 5.34: Percentage of unsatisfied events, min cost, varying willingness

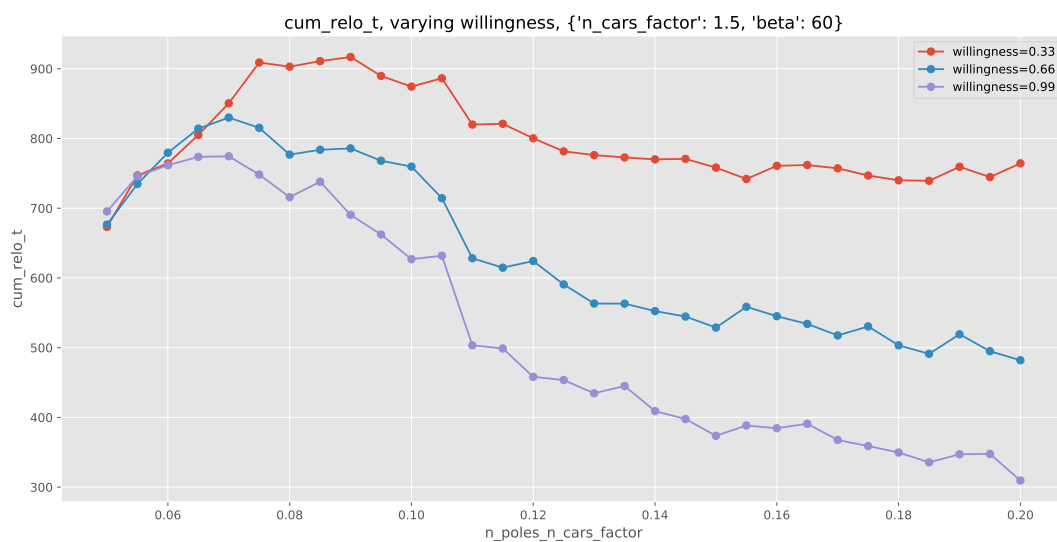


FIGURE 5.35: Relocation cost, min unsatisfied, varying willingness

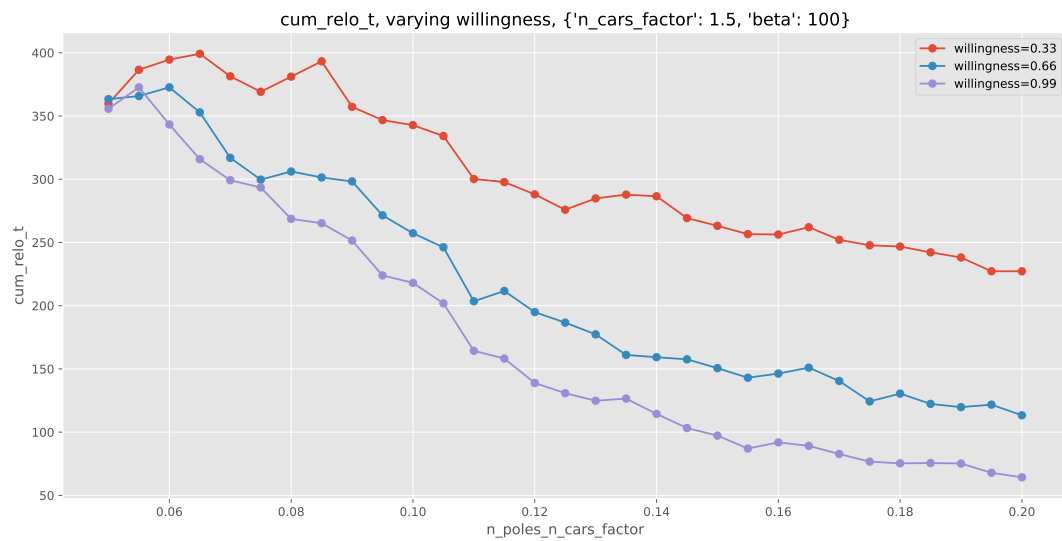


FIGURE 5.36: Relocation cost, best case, varying willingness

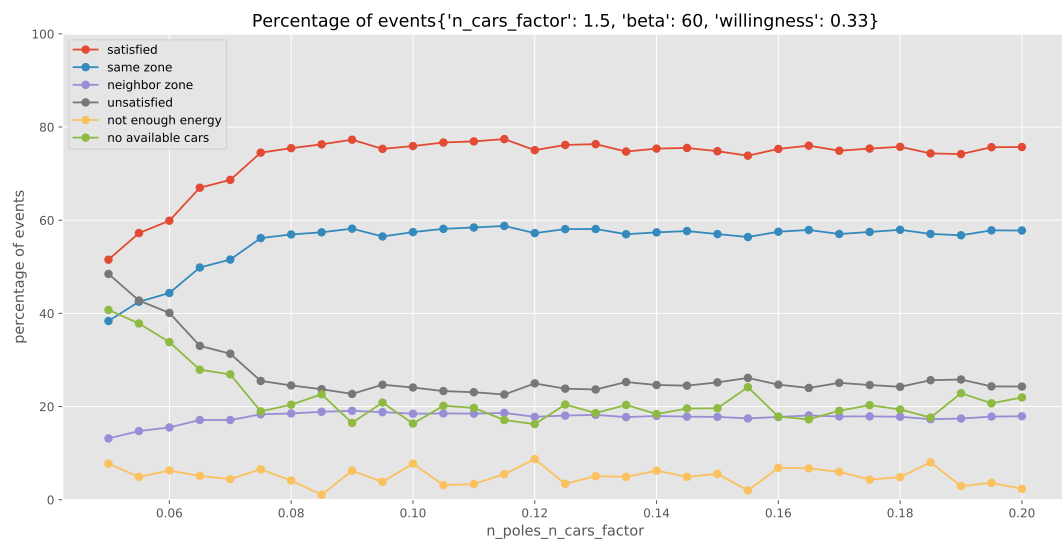


FIGURE 5.37: Events profile, min unsatisfied

Chapter 6

Second case study: Cities comparison

The goal of this chapter is to put in evidence analogies and differences of EFFCS systems deployment in different cities, considering the same charging scenarios presented in the previous chapter, namely:

- Fully centralised charging infrastructure, without user contribution
- Fully decentralised charging infrastructure, with or without user contribution
- Hybrid charging infrastructure, with user contribution

In particular, we will present simulation results for the following cities:

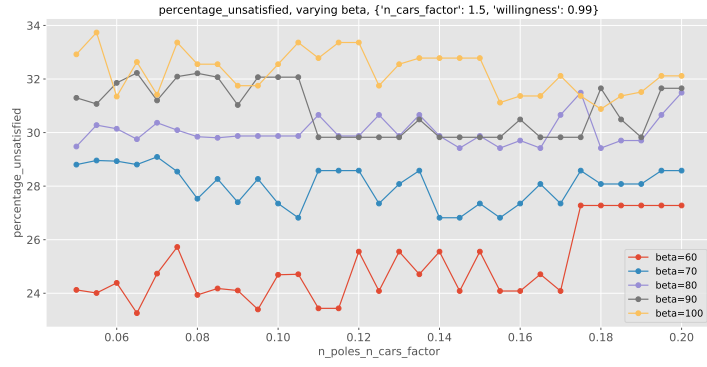
- Brooklyn, NYC, USA
- Milano, Italy
- Berlin, Germany
- Vancouver, Canada

6.1 Centralised infrastructure scenario

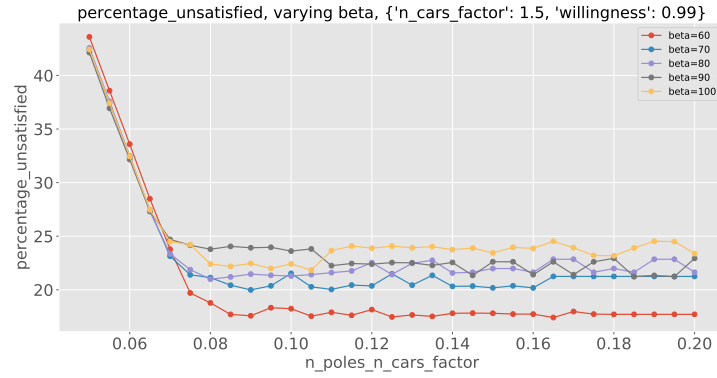
In this section, we will analyse results coming from a set of simulations per each city, in which we imagine that the charging process is performed through a single charging hub. Simulation hypotheses and parameters configuration are the same presented for the city of Turin.

Figure 6.1 shows the effect of β on the percentage of unsatisfied demand in the best case for each city, namely using the configuration which lead to the maximum satisfied demand. In the case of Brooklyn 6.1a, Milano 6.1b and Vancouver 6.1c, higher values of β lead to higher percentage of unsatisfied demand, accordingly to what happened to Turin. In the particular case of Brooklyn, there is no initial transient involving charging capacity, meaning that 20 cars per charging pole is already a good ratio for the size of the hub. Instead, Berlin presents an inversion of the trend for β : lower values correspond to a lower satisfied demand. This effect might be related with the size of the operational area for Berlin, which is the biggest among cities under study.

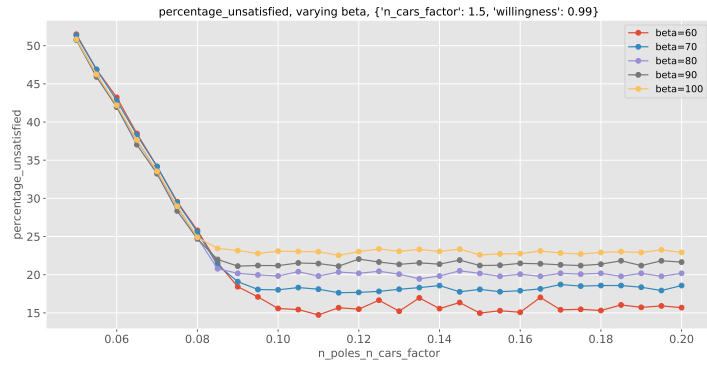
Figure 6.2 shows the effect of β on the cost of charging relocation. In this case, results are compliant with the one of Turin, with each city presenting a higher relocation cost for lower values of β . We can also notice that the cost tends to stabilise



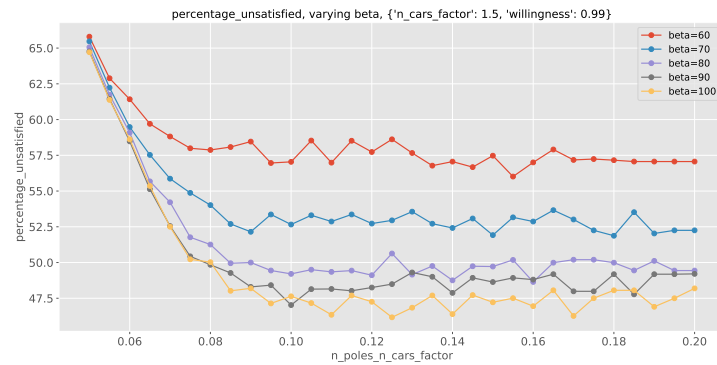
(A) Brooklyn



(B) Milano



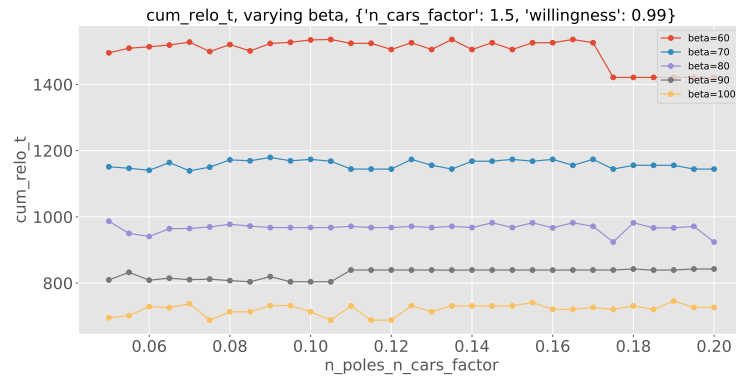
(C) Vancouver



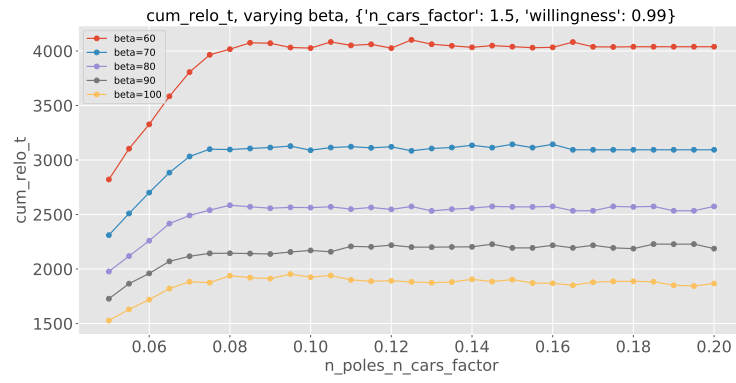
(D) Berlin

FIGURE 6.1: Percentage of unsatisfied demand, varying β , cities comparison

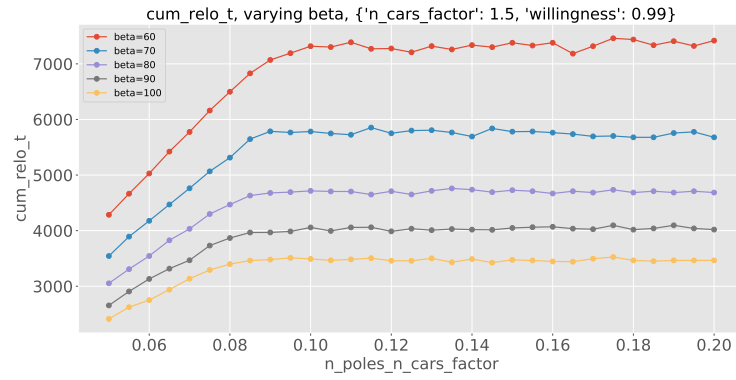
In the case of Brooklyn, Milano and Vancouver, higher values of β lead to higher percentage of unsatisfied demand, accordingly to what happened to Turin. In the particular case of Brooklyn, there is no initial transient involving charging capacity, meaning that 20 cars per charging pole is already a good ratio for the size of the hub. Instead, Berlin presents an inversion of the trend for β : lower values correspond to a lower satisfied demand.



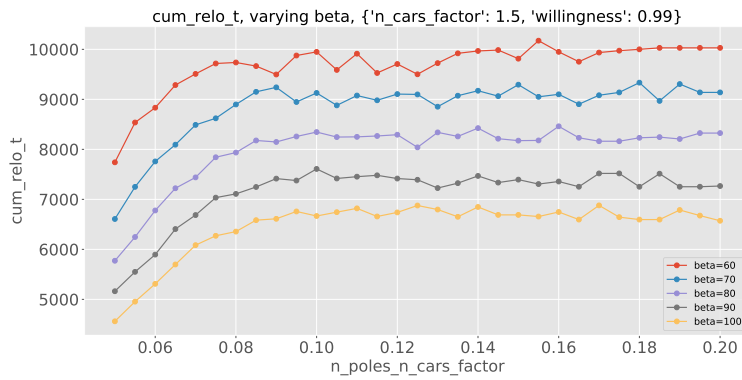
(A) Brooklyn



(B) Milano



(C) Vancouver



(D) Berlin

FIGURE 6.2: Charging relocation cost, varying β , cities comparison

Effect of β on the cost of charging relocation. Lower values of β correspond to higher relocation cost. Different cities have different trends about the initial transient and cost variation with β .

from a certain value of charging capacity on, with different trends depending on the specific city.

Figure 6.3 and figure 6.4 show the effect of fleet size on, respectively, the percentage of unsatisfied requests and charging relocation cost. Results are intuitive and analogous to the ones obtained for Turin, a part from intrinsic differences due to city-specific features.

6.2 Distributed infrastructure scenario

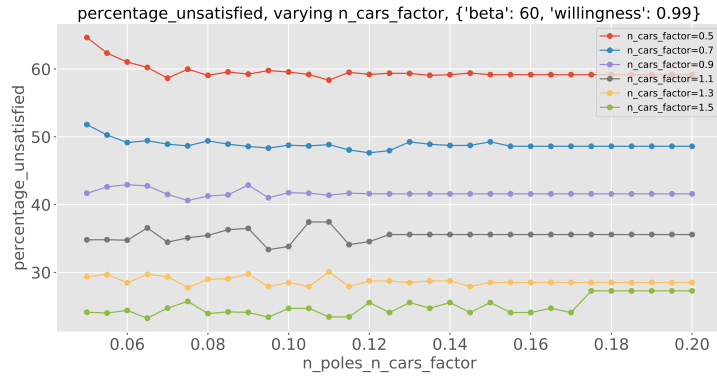
In this section, we will analyse results of a fully distributed charging scenario, with the same hypotheses and parameters configuration set for the city of Turin.

First, let's explore the effect of users' willingness to cooperate with charging operations. Figure 6.5 shows the percentage of unsatisfied requests with different values of willingness in the four cities under study. The positive influence of users contribution is evident in every city, and it increases when the global charging capacity is higher. We can also notice that there is a small difference between the performance obtained with willingness equal to 0.66 and 0.99, meaning that it is not strictly necessary that all users contribute in order to achieve improvements in the quality of the service. Berlin is the only city for which the difference between unsatisfied demand in case of willingness equal to 0.66 or to 0.99 remains visible also for high values of charging capacity.

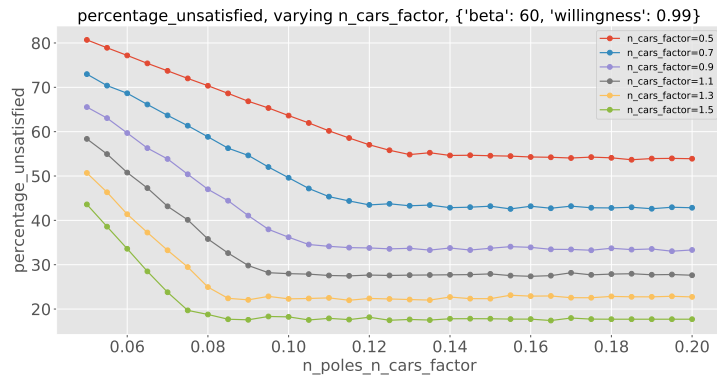
Figure 6.6 shows charging relocation cost in the four cities under study for different values of willingness. Besides lowering charging relocation cost in general, higher values of willingness produce an inverse relationship between relocation cost and charging capacity: in fact, when willingness is sufficient, higher values of charging capacity are associated with lower values of relocation cost. In all the four cities, a willingness of 0.99 guarantees a descending trend of relocation cost with charging capacity, while in the case of willingness equal to 0.66 only Berlin constitutes an exception to this behavior.

Let's now investigate the influence of the system parameter β . Figure 6.7 shows levels of unsatisfied demand with different values of β for the cities under study. Despite higher values of β , contrarily to the centralised charging scenario, corresponds in general to lower unsatisfied demand, this difference tends to mitigate while increasing charging capacity. At the same time, charging relocation cost is in generally heavily affected by β , as higher values of the parameter cause appreciably lower values of charging relocation cost, as shown in figure 6.8.

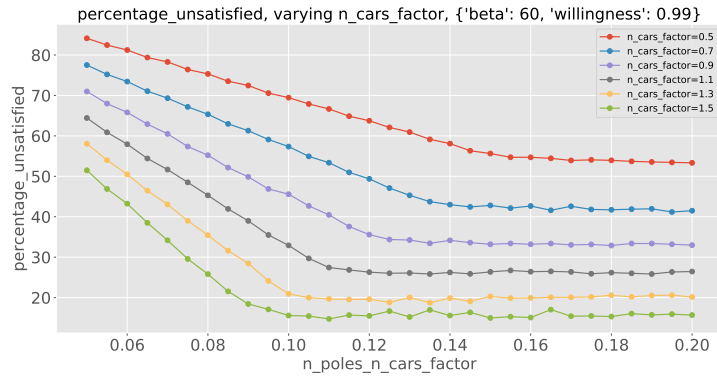
Finally, let's analyse the impact of fleet size. Figure 6.9 shows the variation of unsatisfied demand as a function of charging capacity and fleet size. Similarly to what observed for Turin, and to what intuition might suggest, fleet size has a very strong repercussion on cars availability, and therefore on satisfied mobility demand. In some cases, namely Milano and Vancouver, when the fleet is big enough, increasing charging capacity over a certain threshold is not beneficial for satisfied demand. For which concerns the charging relocation cost, shown in figure 6.10, charging capacity has instead a considerable significance: a higher number of charging stations makes charging relocation cost for different fleet size converge to similar values. In case of a distributed infrastructure, it might be therefore possible to reduce variable costs due to relocation at the price of increasing fixed costs associated with vehicles and charging infrastructure, at the benefit of the overall quality of service.



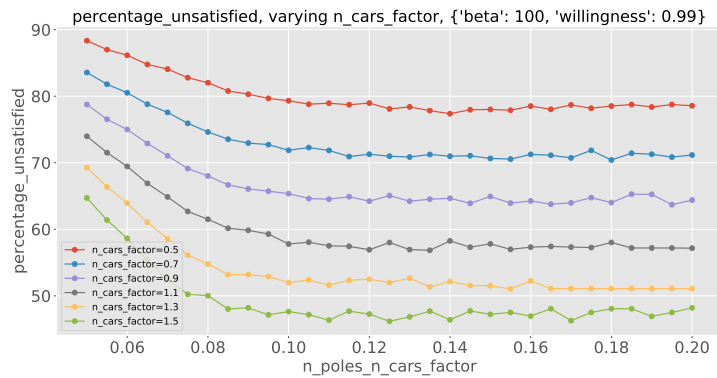
(A) Brooklyn



(B) Milano



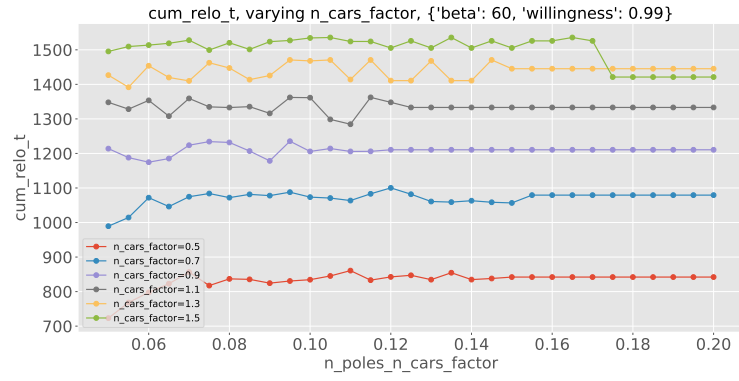
(C) Vancouver



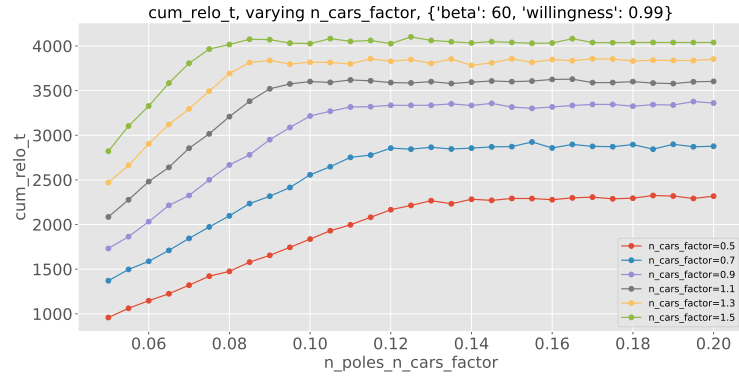
(D) Berlin

FIGURE 6.3: Percentage of unsatisfied demand, varying n_cars_factor , cities comparison

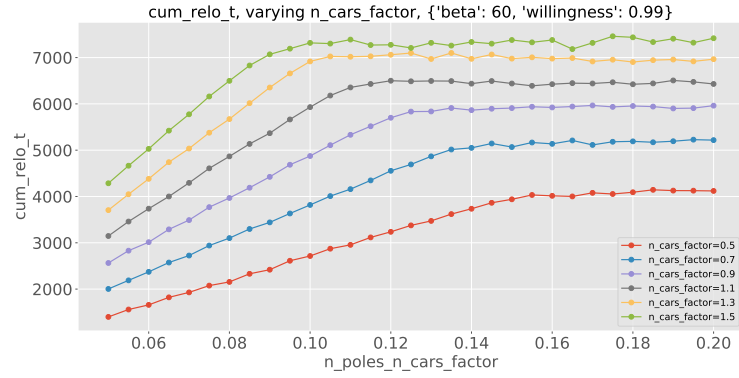
Effect of β on the cost of charging relocation cost. Lower values of β correspond to higher relocation cost. Different cities have different trends about the initial transient and cost variation with β .



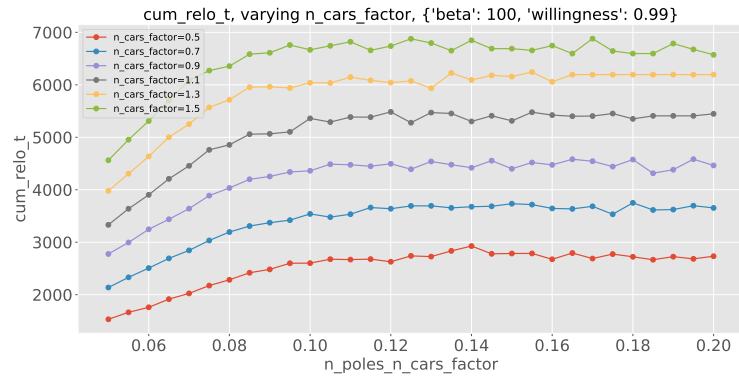
(A) Brooklyn



(B) Milano



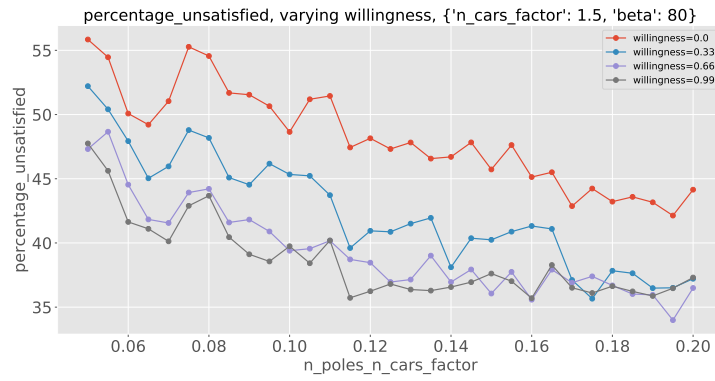
(C) Vancouver



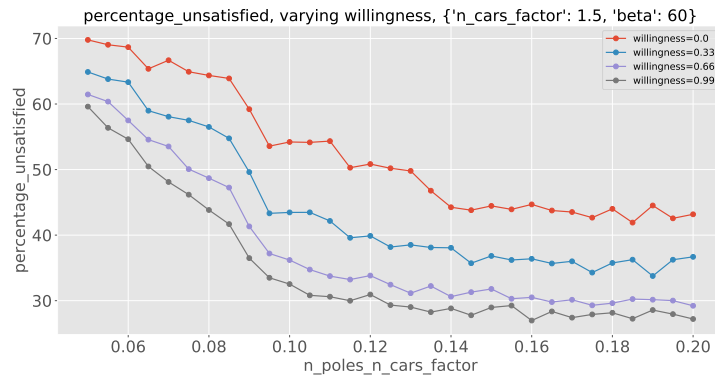
(D) Berlin

FIGURE 6.4: Charging relocation cost, varying n_cars_factor, cities comparison

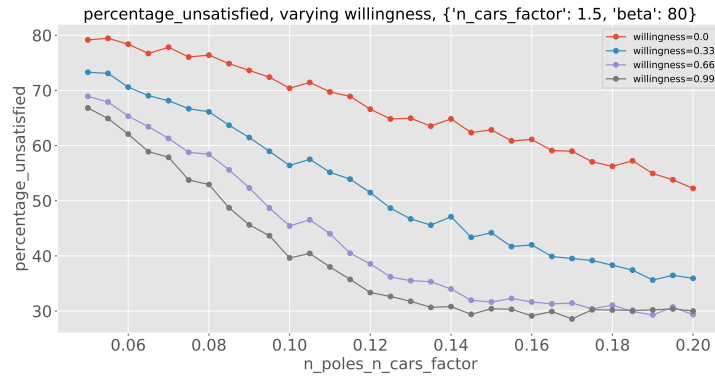
Effect of β on the cost of charging relocation cost. Lower values of β correspond to higher relocation cost. Different cities have different trends about the initial transient and cost variation with β .



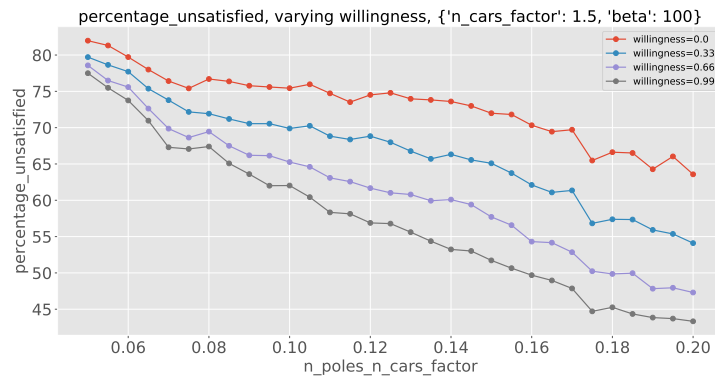
(A) Brooklyn



(B) Milano



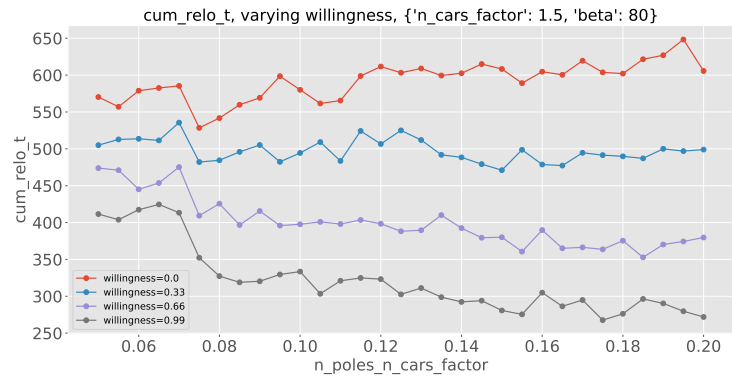
(C) Vancouver



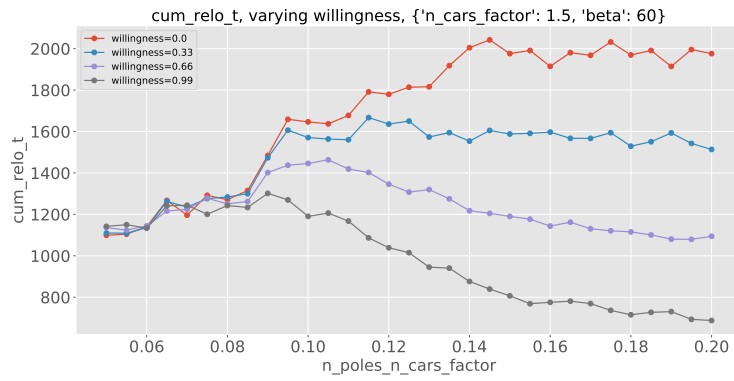
(D) Berlin

FIGURE 6.5: Percentage of unsatisfied demand, varying willingness, cities comparison

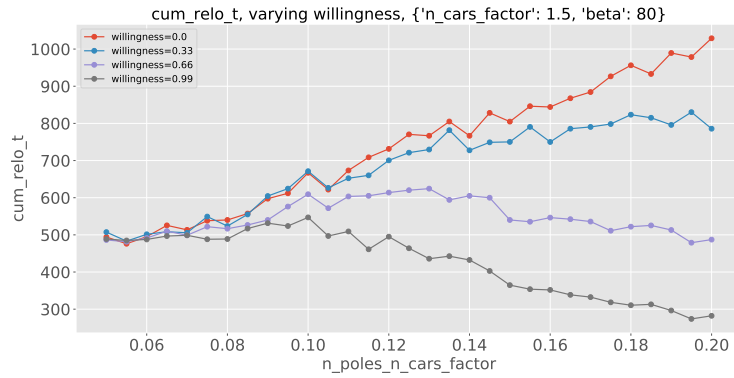
Effect of users willingness on unsatisfied demand. Higher values of willingness correspond in general to lower values of unsatisfied demand. However, with enough charging capacity, with willingness of 0.66 the service achieves almost the same performance as with a willingness of 0.99. In Berlin, instead, the difference between those values of willingness is still quite visible.



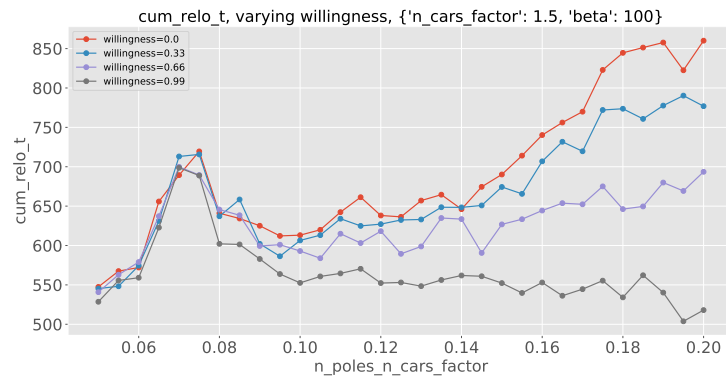
(A) Brooklyn



(B) Milano



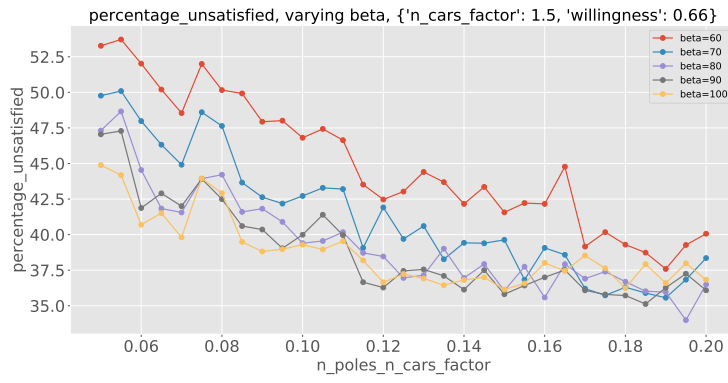
(C) Vancouver



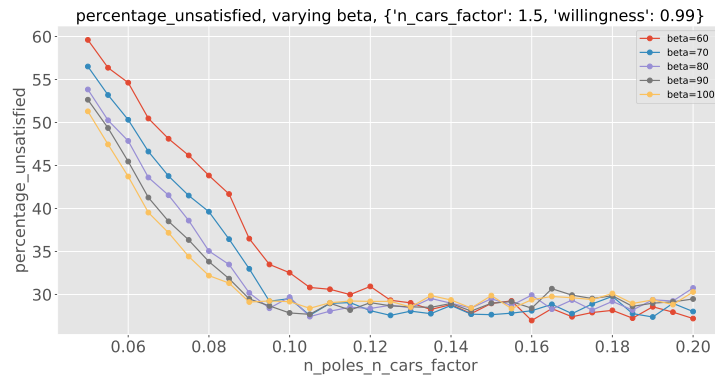
(D) Berlin

FIGURE 6.6: Charging relocation cost, varying n_{cars_factor} , cities comparison

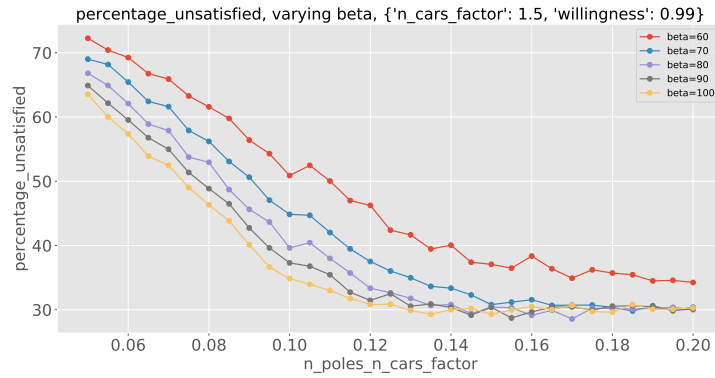
Effect of users willingness on the cost of charging relocation. Higher values of willingness correspond to lower relocation cost. Different cities have different trends about the initial transient, but in general high willingness causes charging relocation cost to decrease with charging capacity



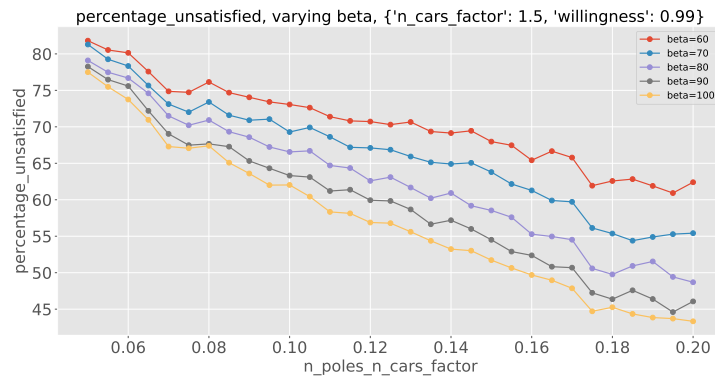
(A) Brooklyn



(B) Milano



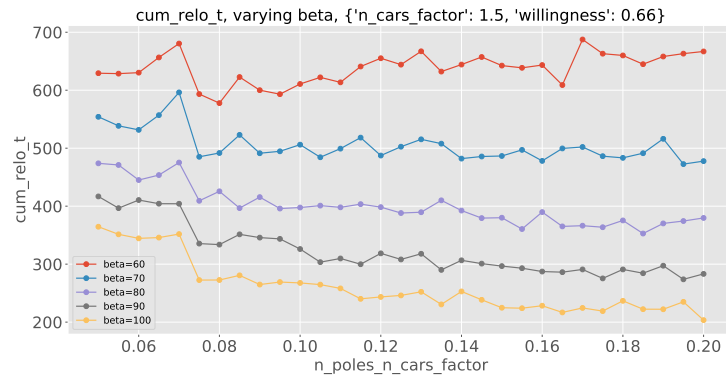
(C) Vancouver



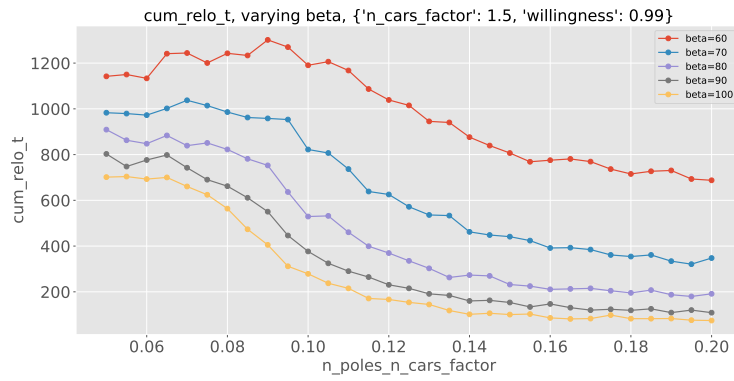
(D) Berlin

FIGURE 6.7: Percentage of unsatisfied demand, varying β , cities comparison

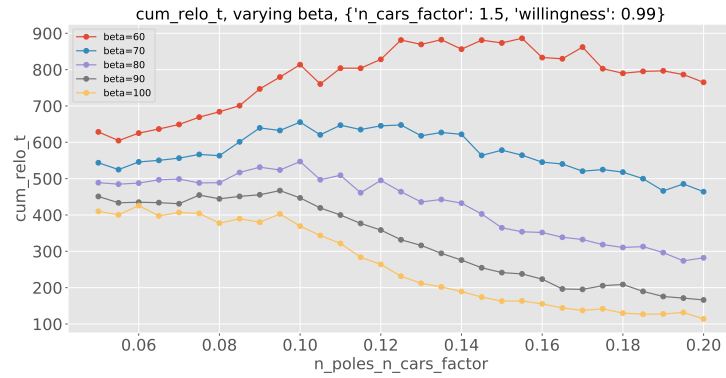
Effect of β on unsatisfied demand. Higher values of β correspond in general to lower values of unsatisfied demand, but the difference becomes more blurry while increasing charging capacity.



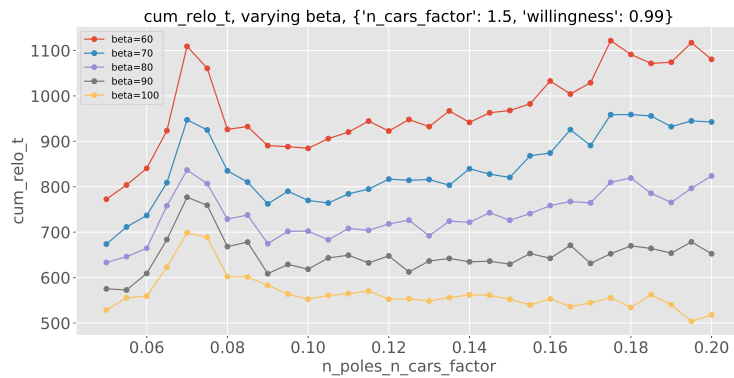
(A) Brooklyn



(B) Milano

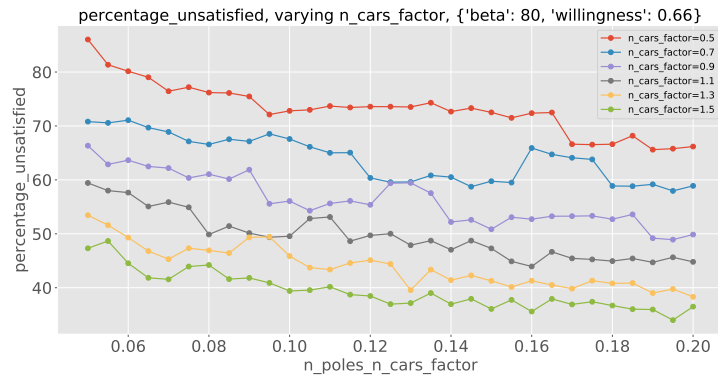


(C) Vancouver

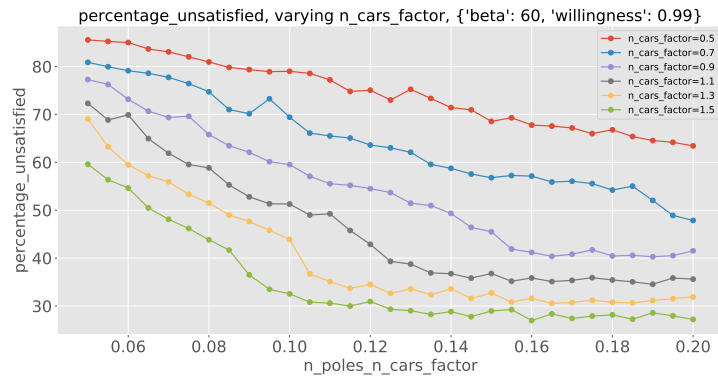


(D) Berlin

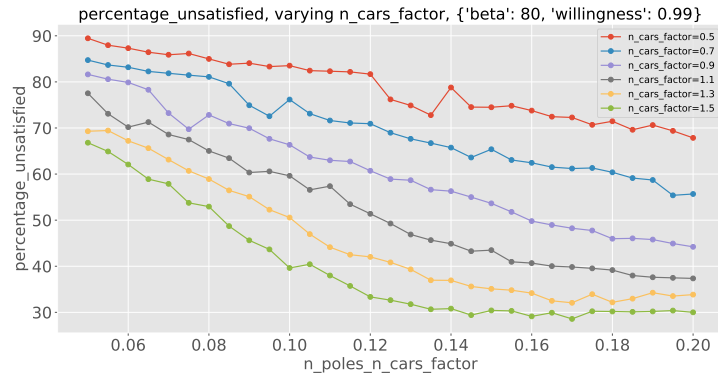
FIGURE 6.8: Charging relocation cost, varying β , cities comparisonEffect of β on the cost of charging relocation.



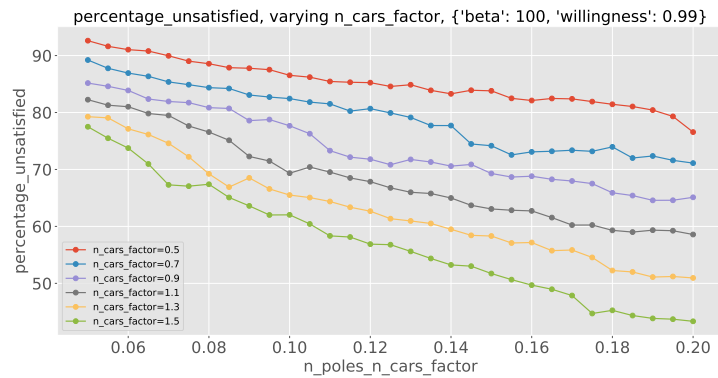
(A) Brooklyn



(B) Milano



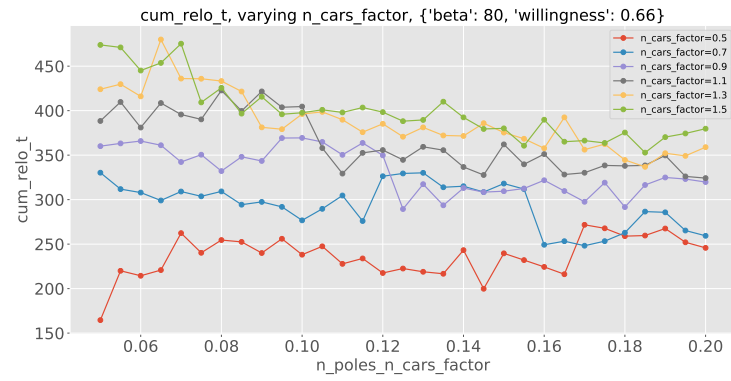
(C) Vancouver



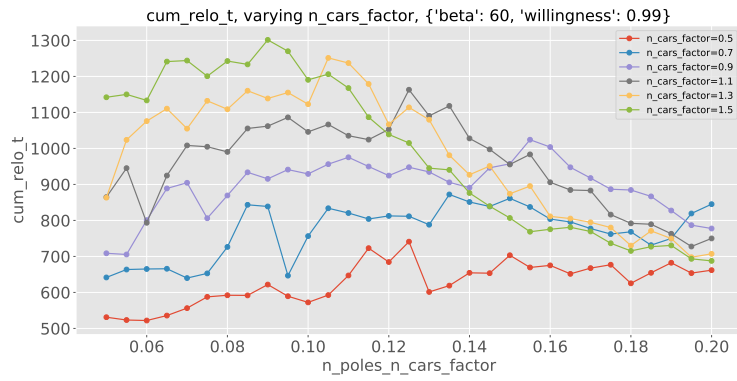
(D) Berlin

FIGURE 6.9: Percentage of unsatisfied demand, varying n_cars_factor, cities comparison

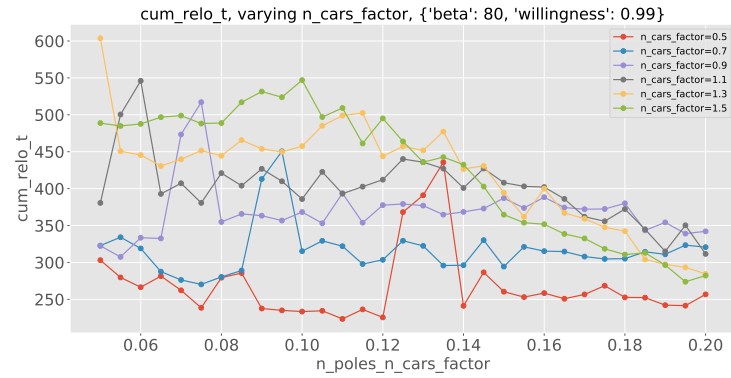
Effect of fleet size on unsatisfied demand.



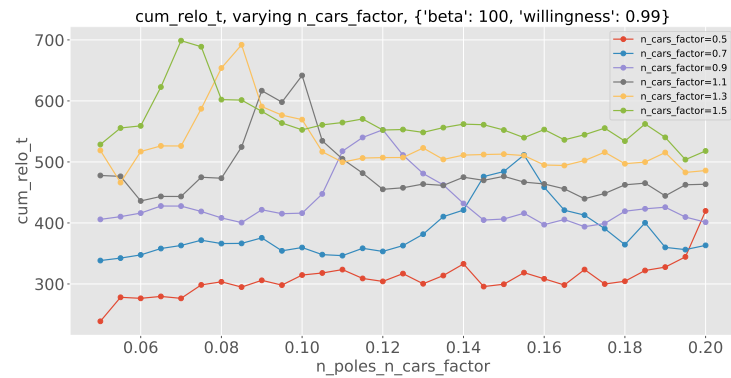
(A) Brooklyn



(B) Milano



(C) Vancouver



(D) Berlin

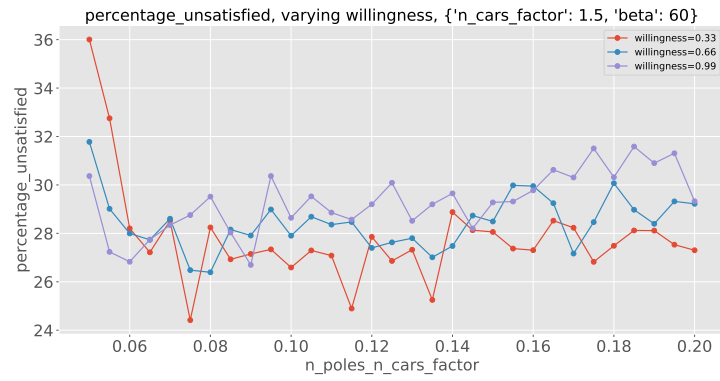
FIGURE 6.10: Charging relocation cost, varying n_cars_factor, cities comparison

Effect of fleet size on the cost of charging relocation cost.

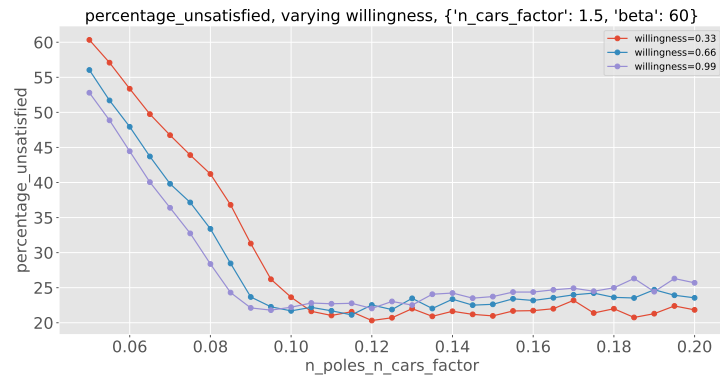
6.3 Charging hub and distributed infrastructure scenario

In this section, we present results for a scenario in which the charging process is performed through a hybrid infrastructure: the system operator manages recharges through a centralised hub, but users are allowed cooperate, given their willingness, using the available distributed infrastructure.

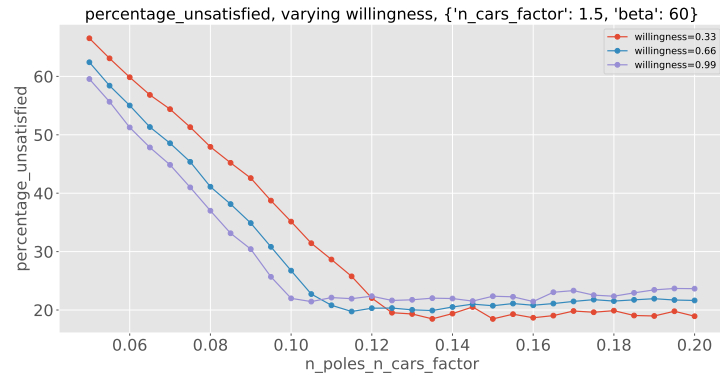
As in the previous section, let's first analyse the effect of users willingness for charging contribution. Figure 6.11 shows different curves of unsatisfied demand for different values of user willingness as a function of charging capacity. Interestingly, in the case of Brooklyn, Milano and Vancouver, users willingness has little to no impact on satisfied demand when charging capacity is sufficiently high, and higher values of willingness may even lead to a counterproductive effect, probably due to the higher number of cars charging, and therefore unavailable, at a given time. However, when total charging capacity is not very high, higher values of willingness correspond to a systematically higher level of satisfied demand. In Berlin, differently than the other cities, higher values of willingness appear being always beneficial for quality of service, and the trend does not revert while varying charging capacity. For which concerns charging relocation cost, shown in figure 6.12, higher values of willingness are associated in general with lower relocation costs, but with different trends depending on the city. In fact, for cities with bigger operational are, namely Milano, Vancouver and Berlin, users willingness does not impact relocation cost until a certain threshold of charging capacity is reached. In the case of Brooklyn, instead, higher willingness is always beneficial for satisfied mobility demand. The impact of system parameter β and fleet size, shown in figures 6.13, 6.14, 6.15 and 6.16, follows a trend which is very similar to the centralised case, exception made for a general reduction of the relocation cost due users contribution. However, such reduction is not comparable with the one obtained in the fully decentralised scenario, despite the global performance on satisfied demand are generally better.



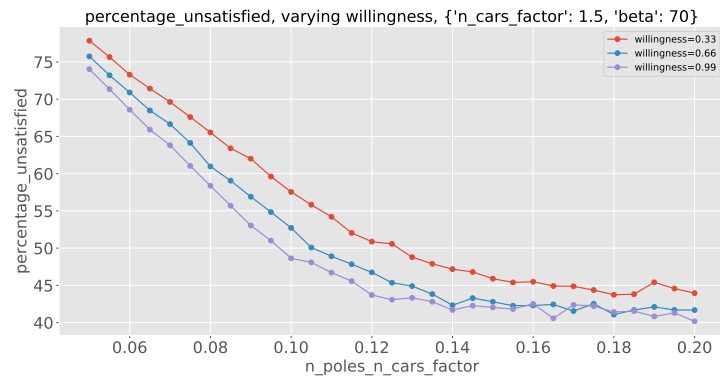
(A) Brooklyn



(B) Milano



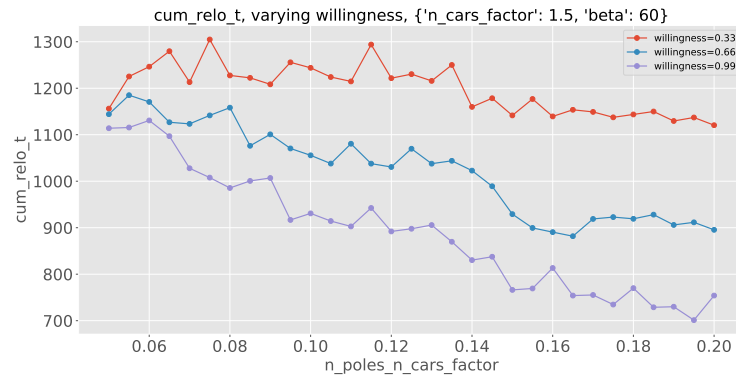
(C) Vancouver



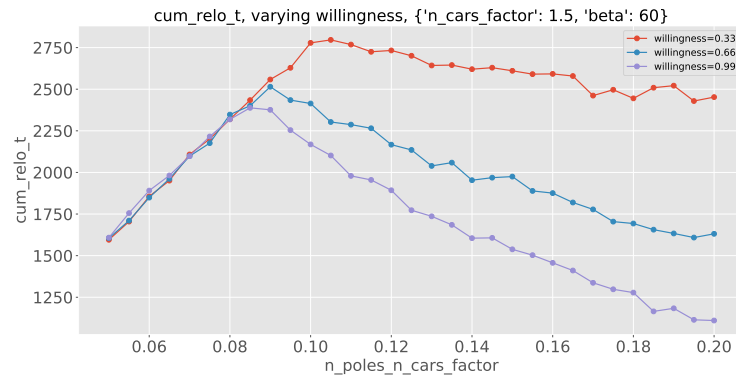
(D) Berlin

FIGURE 6.11: Percentage of unsatisfied demand, varying willingness, cities comparison

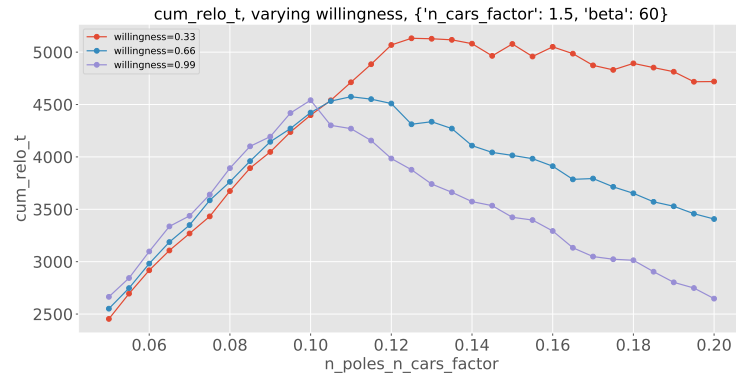
Effect of users willingness on unsatisfied demand.



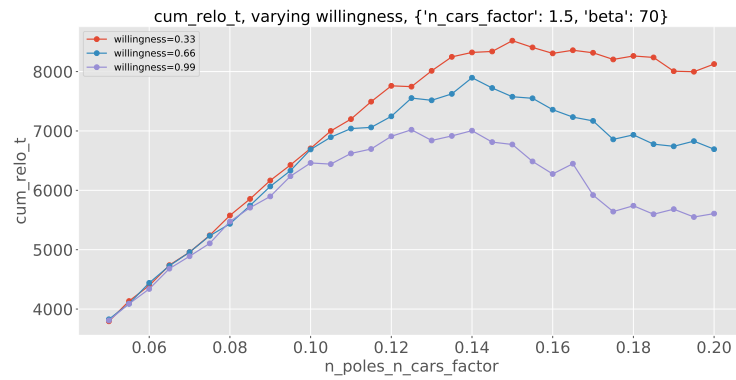
(A) Brooklyn



(B) Milano



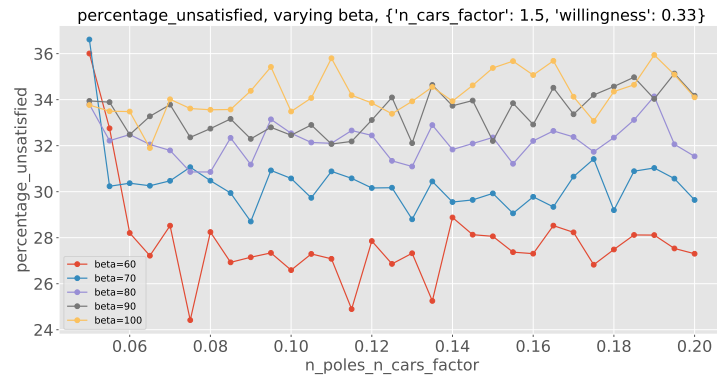
(C) Vancouver



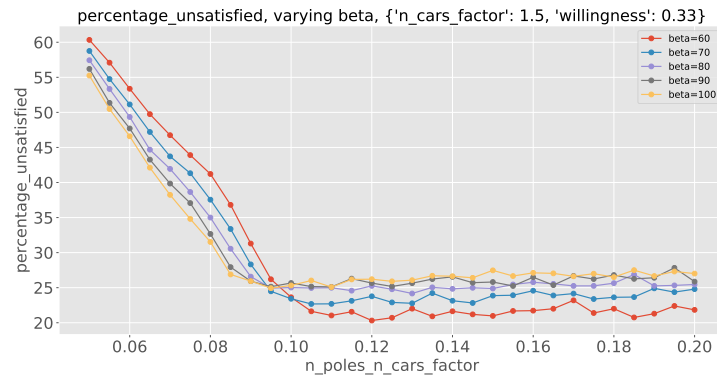
(D) Berlin

FIGURE 6.12: Charging relocation cost, varying willingness cities comparison

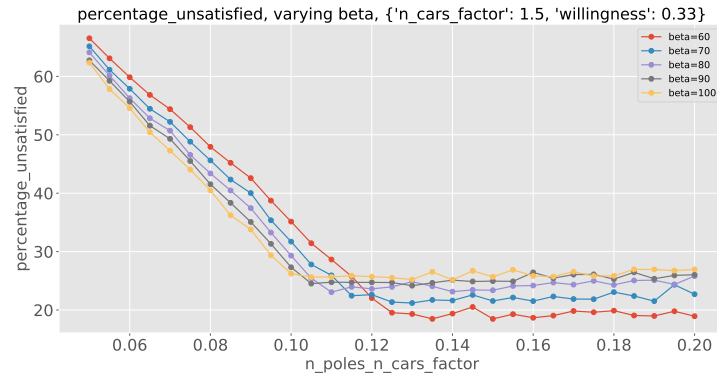
Effect of users willingness on the cost of charging relocation.



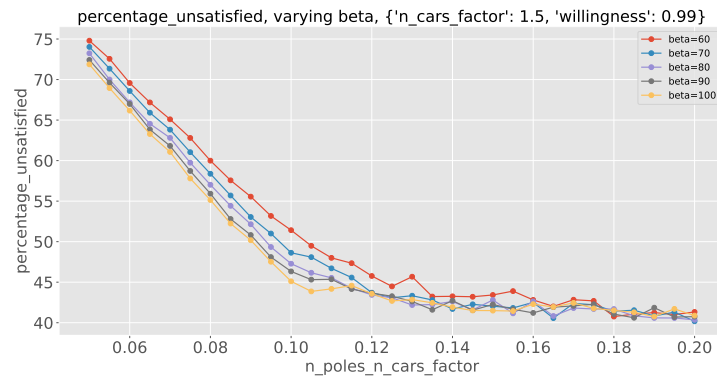
(A) Brooklyn



(B) Milano

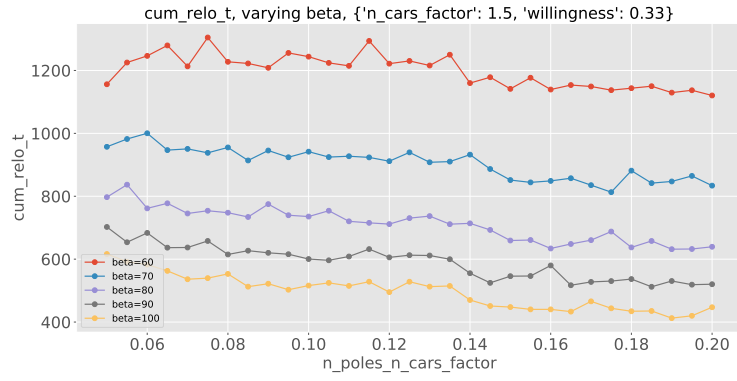


(C) Vancouver

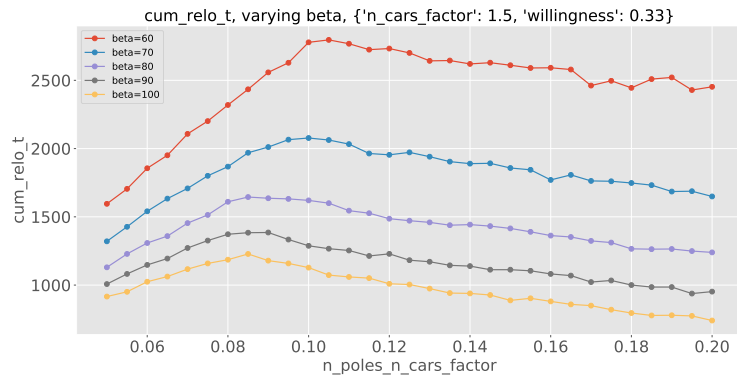


(D) Berlin

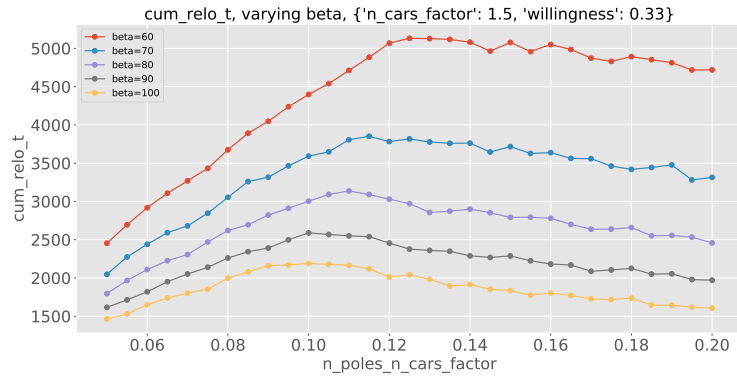
FIGURE 6.13: Percentage of unsatisfied demand, varying β , cities comparisonEffect of β on unsatisfied demand.



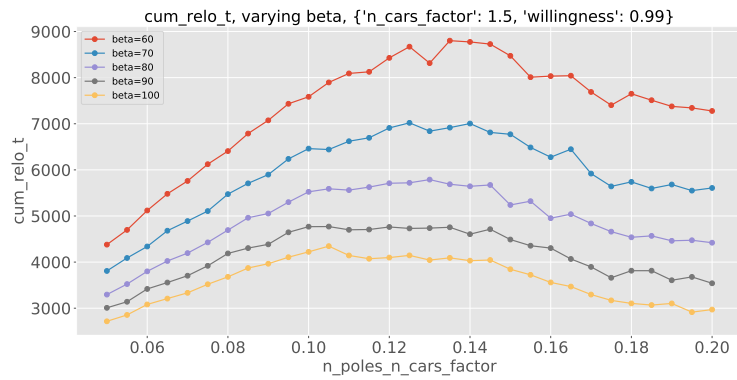
(A) Brooklyn



(B) Milano

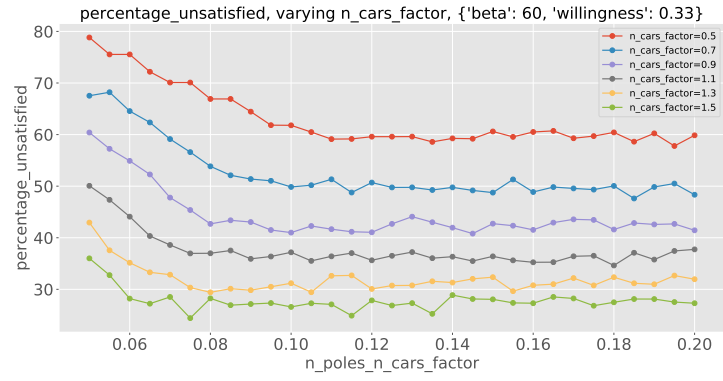


(C) Vancouver

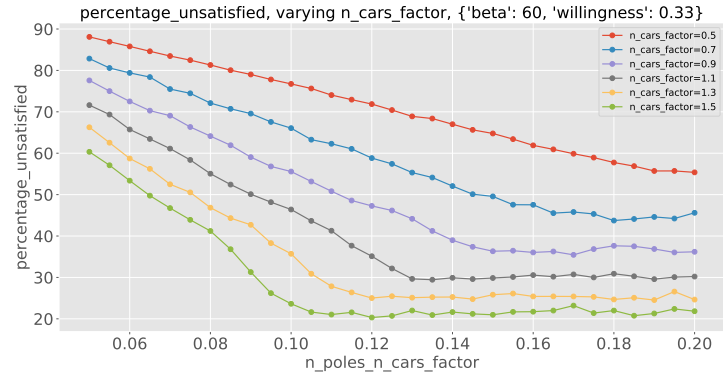


(D) Berlin

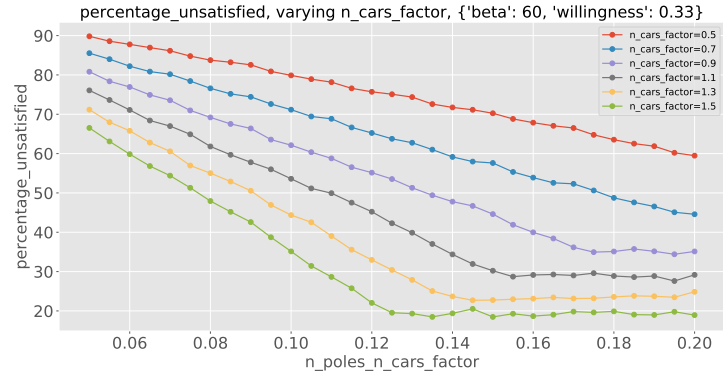
FIGURE 6.14: Charging relocation cost, varying β , cities comparisonEffect of β on the cost of charging relocation.



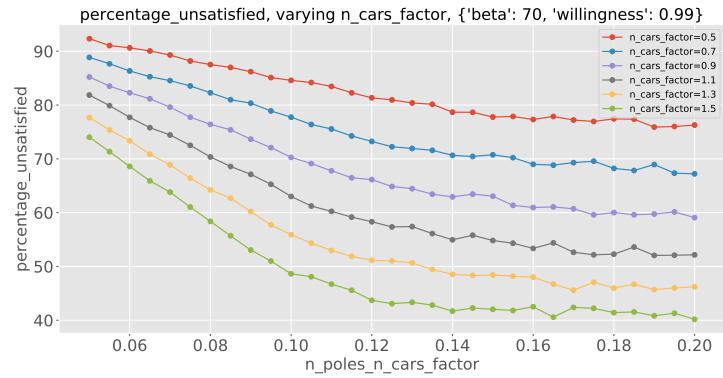
(A) Brooklyn



(B) Milano



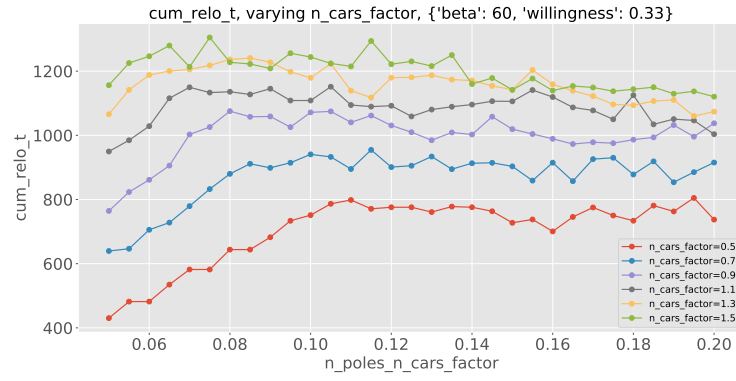
(C) Vancouver



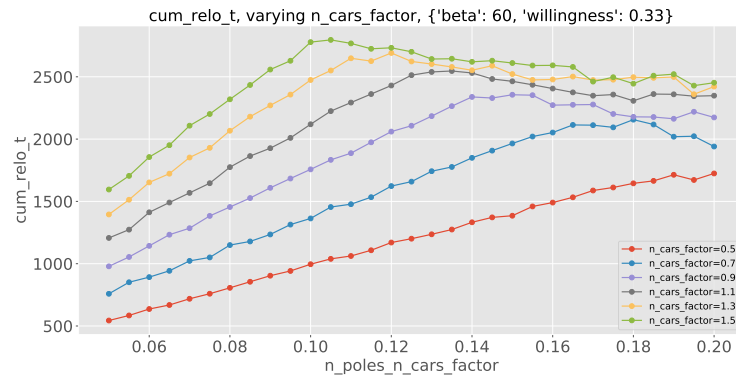
(D) Berlin

FIGURE 6.15: Percentage of unsatisfied demand, varying n_cars_factor, cities comparison

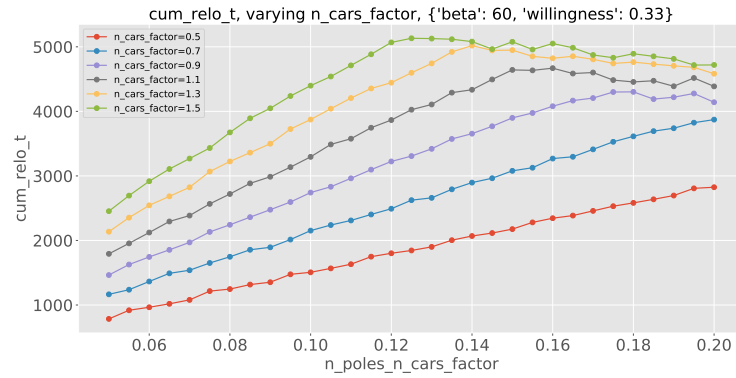
Effect of n_cars_factor on unsatisfied demand.



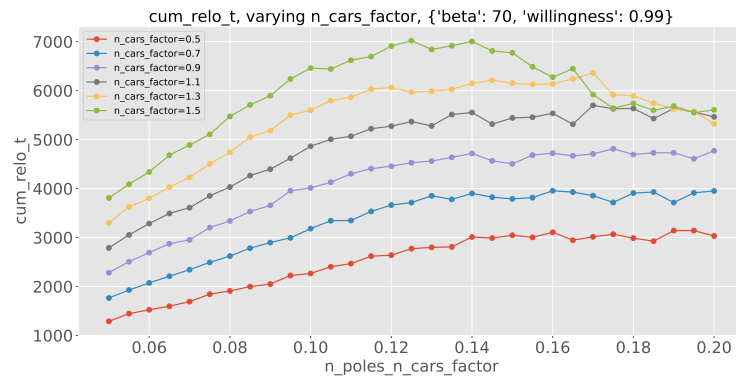
(A) Brooklyn



(B) Milano



(C) Vancouver



(D) Berlin

FIGURE 6.16: Charging relocation cost, varying n_cars_factor, cities comparison

Effect of n_cars_factor on the cost of charging relocation.

Chapter 7

Conclusions

The goal of this thesis is comparing different charging scenarios and fleet management strategies for EFFCS systems, with the aid of real world data coming from CS services and collected using exclusively ICT-based platforms. For this scope, we implemented eC2S : a software for modelling, simulating and designing EFFCS systems with a focus on charging infrastructure and management. The main contributions of this thesis can therefore be summarised as:

- Development of data-driven, city-specific mobility demand models for CS systems.
- Development of an event-based simulator for EFFCS systems.
- Analysis and visualisation of simulation output.

Our demand model is composed by a parametric estimation of inter-arrival times between mobility requests, based on Poisson processes, and a non-parametric estimation of OD matrices based on Kernel Density Estimate. For which concerns time estimation, we used a non-homogeneous Poisson process with a time-dependent arrival rate, in order to model the different profile of users requests over different hours of the day. Such model performs very good in terms of request profile estimation, with errors lower than 0.5% in each hourly slot for every city considered. The spatial estimation has peculiar characteristics in this work, because it has to preserve spatial patterns of different zones, while uncorrelating mobility demand from cars locations at a given time. This issue is triggered by the fact that input data describe only successful bookings, while we do not have any concrete information about user requests the system was not able to satisfy. Our model for OD matrices estimation, based on KDE, is able to estimate spatial patterns with an error lower than 3% for each single zone of cities under study, while uncorrelating users requests and cars locations through the usage of a Gaussian kernel.

The core simulation module, implemented in this thesis using the SimPy library, runs under the assumptions presented in chapter 3, but eC2S is designed to be flexible and modular, allowing to modify existing rules and to add easily other constraints.

We run a simulation campaign over 5 different cities in order to evaluate performance of three different charging infrastructure scenarios: fully centralised, fully distributed, hybrid. Results show that the presence of a centralised hub in a highly dynamic hotspot, namely where many trips start or end, is beneficial from the mobility viewpoint, as more user requests tend to be satisfied. On the other side, the management of charging processes is expensive because each car needs to be driven to the hub in order to be charged, no matter its position. With a decentralised infrastructure, the increase in management complexity is traded with a much lower

operational cost, as stations are widespread around the city. Furthermore, a distributed infrastructure allows for users contribution in charging processes, which can be primarily important for a further reduction of operational cost without hardly impacting users comfort.

Although these considerations apply in a fairly similar way across different cities, it is also possible to observe specific patterns and behaviors, probably due to structures and activities proper of each city.

Possible future works to extend the outcomes of this thesis include:

- Study the scalability of simulated systems with respect to the intensity of users requests.
- Implement demand-aware post-charging relocation strategies.
- Implement a rigorous cost analysis complete with energy market information.
- Consider renewable energy production close to charging stations.
- Consider different approaches for charging processes, such as battery swap.
- Consider mixed fleets with different type of vehicles, and mixed charging infrastructure with different types of poles.
- Improve and extend the comparison between cities.

Appendix A

Model validation for different cities

A.1 Milano

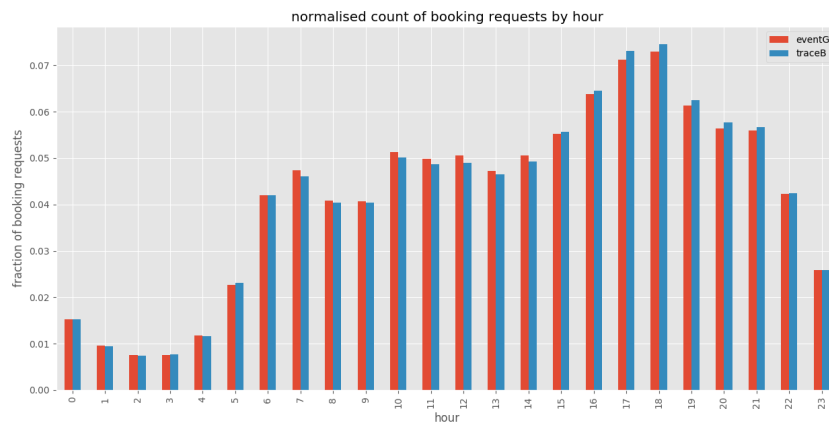


FIGURE A.1: Percentage of bookings by hour for the city of Milano

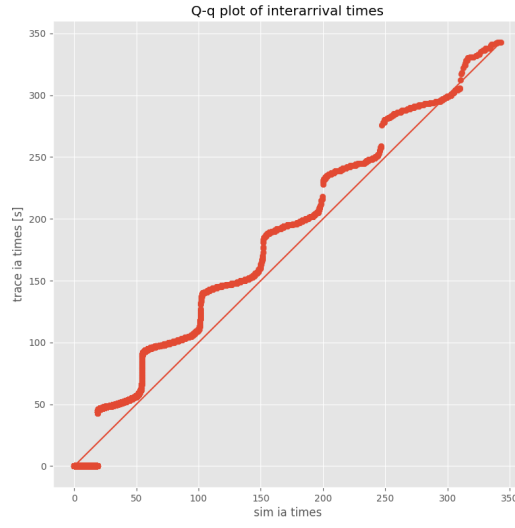
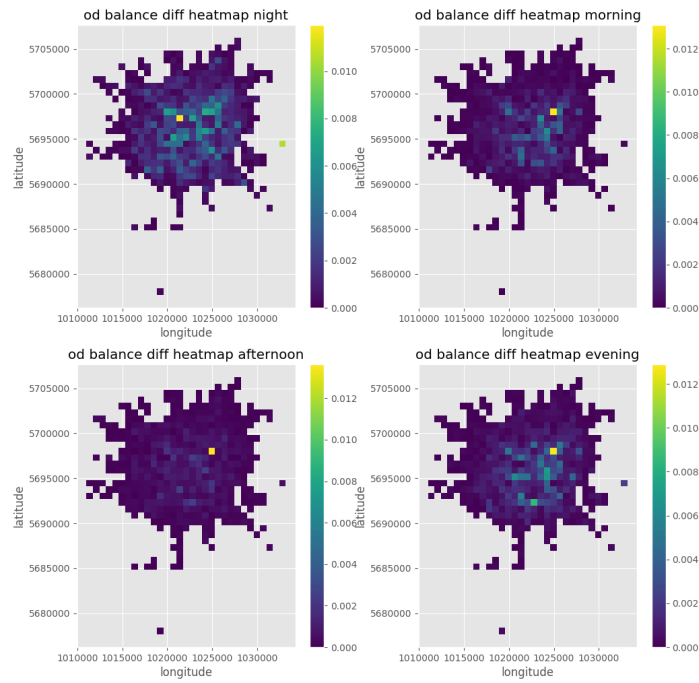


FIGURE A.2: Q-Q plot of inter-arrival times for the city of Milano

FIGURE A.3: $\epsilon_{i,t,\%}^{od}$ for the city of Milano during night, morning, afternoon and evening

A.2 Berlin

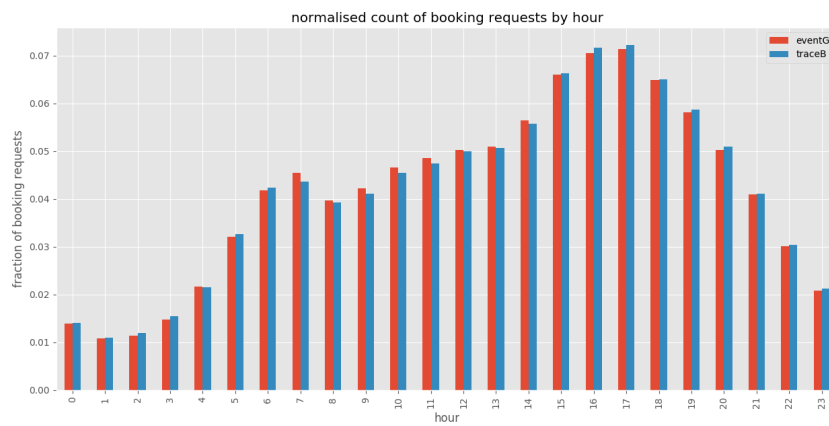


FIGURE A.4: Percentage of bookings by hour for the city of Berlin

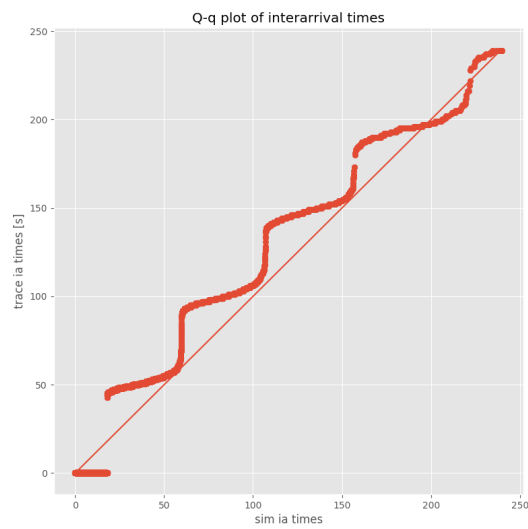


FIGURE A.5: Q-Q plot of inter-arrival times for the city of Berlin

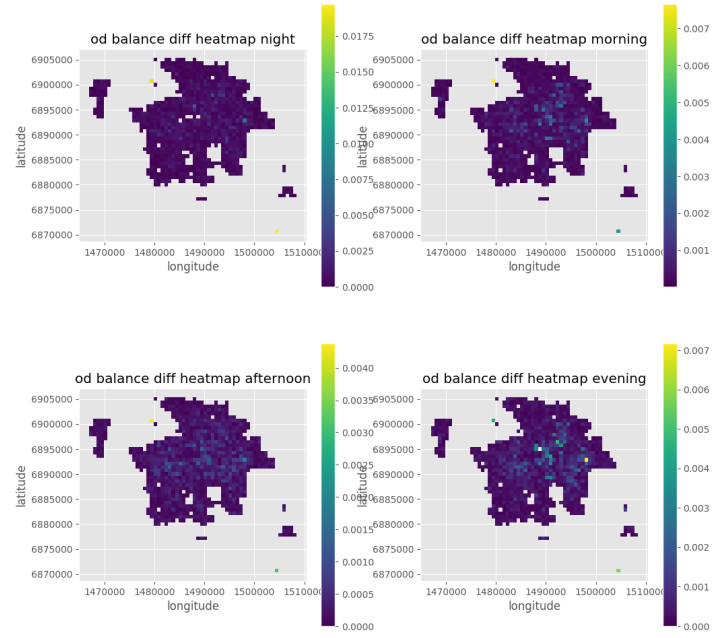


FIGURE A.6: $\epsilon_{i,t,\%}^{od}$ for the city of Berlin during night, morning, afternoon and evening

A.3 New York City (Brooklyn)

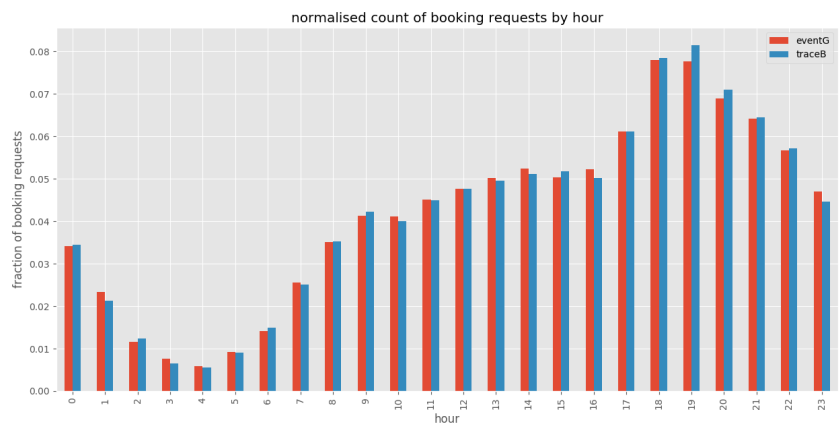


FIGURE A.7: Percentage of bookings by hour for NYC

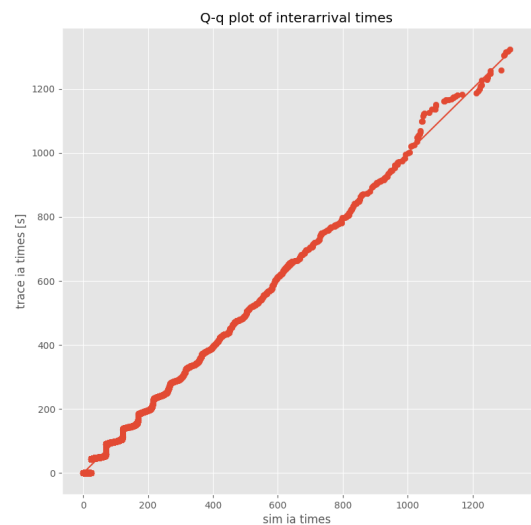


FIGURE A.8: Q-Q plot of inter-arrival times for NYC

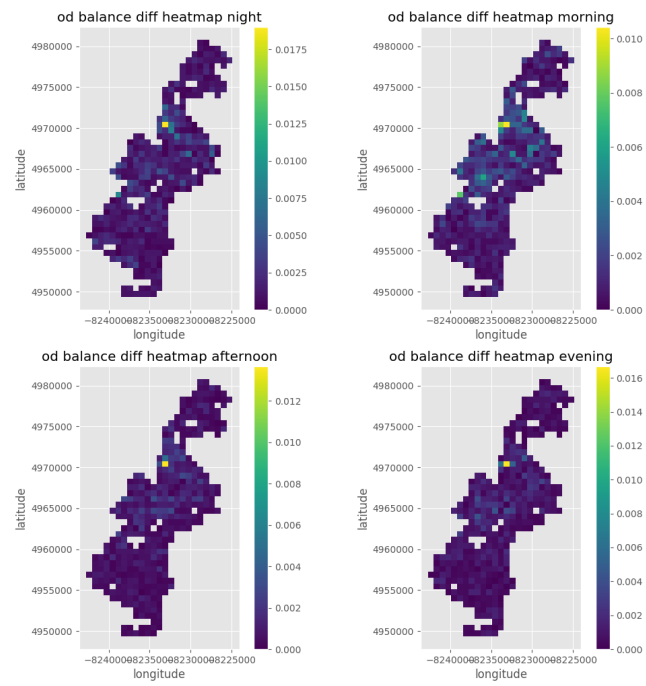


FIGURE A.9: $\epsilon_{i,t,\%}^{od}$ for NYC during night, morning, afternoon and evening

A.4 Vancouver

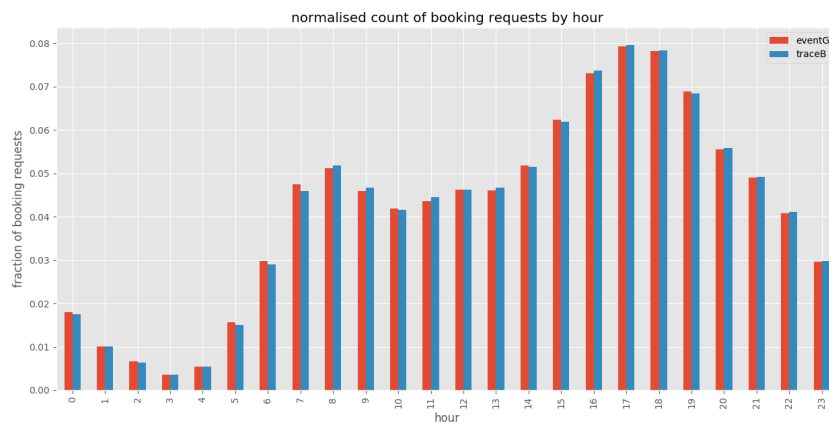


FIGURE A.10: Percentage of bookings by hour for the city of Vancouver



FIGURE A.11: Q-Q plot of inter-arrival times for the city of Vancouver

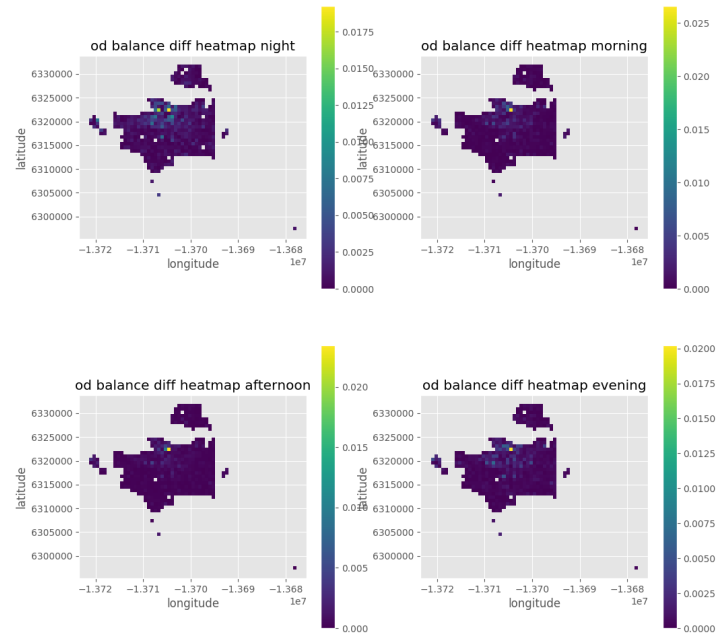


FIGURE A.12: $\epsilon_{i,t,\%}^{od}$ for the city of Vancouver during night, morning, afternoon and evening

Bibliography

- [1] URL: <https://github.com/AleCioc/DELCASIM>.
- [2] URL: https://en.wikipedia.org/wiki/Poisson_point_process#References.
- [3] URL: <https://scikit-learn.org/stable/modules/density.html#kernel-density-estimation>.
- [4] URL: https://en.wikipedia.org/wiki/Kernel_density_estimation#References.
- [5] URL: <https://towardsdatascience.com/simple-example-of-2d-density-plots-in-python-83b83b934f67>.
- [6] URL: <https://github.com/mongodb/>.
- [7] URL: <https://meanjs.org/>.
- [8] URL: <https://www.python.org/>.
- [9] URL: <https://www.datacamp.com/community/blog/python-scientific-computing-case/>.
- [10] URL: <https://www.stat.washington.edu/~hoytak/blog/whypython.html>.
- [11] URL: <https://2.python-requests.org/en/master/>.
- [12] URL: <https://api.mongodb.com/python/current/>.
- [13] URL: <https://pandas.pydata.org/>.
- [14] URL: <http://geopandas.org/>.
- [15] URL: <https://numpy.org/>.
- [16] URL: <https://matplotlib.org/>.
- [17] URL: <https://www.scipy.org/>.
- [18] URL: <https://simpy.readthedocs.io/en/latest/>.
- [19] URL: <https://www.spyder-ide.org/>.
- [20] URL: <https://www.jetbrains.com/pycharm/>.
- [21] URL: <https://docs.conda.io/en/latest/>.
- [22] International Energy Agency. 2018. URL: <https://www.iea.org/statistics/co2emissions/>.
- [23] International Energy Agency. *World Energy Outlook 2018 - Executive Summary*. 2018. URL: <https://webstore.iea.org/download/summary/190?fileName=English-WEO-2018-ES.pdf>.
- [24] International Renewable Energy Agency. "Climate Change and Renewable Energy: National policies and the role of communities". In: (2019).
- [25] Danilo Giordano Marco Mellia Andrea Morichetta Andrian Putina Flavia Salutari Alessandro Ciociola Michele Cocca. "UMAP: Urban Mobility Analysis Platform to Harvest Car Sharing Data". In: 37 (Aug. 2017), p. 8.

- [26] Maria Niestadt with Amalie Bjørnåvold. *Electric road vehicles in the European Union - Trends, impacts and policies*. Apr. 2019.
- [27] McKinsey Amsterdam Roundtables Foundation and Company. *Electric vehicles in Europe: gearing up for a new phase?* Apr. 2014.
- [28] Janssens D Wets G Arentze T Timmermans H. Bellemans T Kochan B. "Implementation Framework and Development Trajectory of FEATHERS Activity-Based Simulation Platform". In: *Transportation Research Record Journal of the Transportation Research Board* (2010).
- [29] Georg Brandstätter et al. "Overview of Optimization Problems in Electric Car-Sharing System Design and Management". In: vol. 22. Sept. 2016, pp. 441–471. ISBN: 978-3-319-39118-2. DOI: [10.1007/978-3-319-39120-5_24](https://doi.org/10.1007/978-3-319-39120-5_24).
- [30] Filipe Brandão, Ricardo Mendes Correia, and Alexandra Paio. "Measuring Urban Renewal: A Dual Kernel Density Estimation to Assess the Intensity of Building Renovation—Case Study in Lisbon". In: *Urban Science* 2 (Sept. 2018), p. 91. DOI: [10.3390/urbansci2030091](https://doi.org/10.3390/urbansci2030091).
- [31] Alfred Benedikt Brendel et al. "Improving electric vehicle utilization in car-sharing: A framework and simulation of an e-carsharing vehicle utilization management system". In: *Transportation Research Part D: Transport and Environment* 64 (2018). The contribution of electric vehicles to environmental challenges in transport. WCTRS conference in summer, pp. 230–245. ISSN: 1361-9209. DOI: <https://doi.org/10.1016/j.trd.2018.01.024>. URL: <http://www.sciencedirect.com/science/article/pii/S1361920916307556>.
- [32] Maurizio Bruglieri, Alberto Colorni, and Alessandro Lué. "The vehicle relocation problem for the one-way electric vehicle sharing". In: *ArXiv abs/1307.7195* (2013).
- [33] Xuejiao Cai, Zhifeng Wu, and Jiong Cheng. "Using kernel density estimation to assess the spatial pattern of road density and its impact on landscape fragmentation". In: *International Journal of Geographical Information Science* 27.2 (2013), pp. 222–230. DOI: [10.1080/13658816.2012.663918](https://doi.org/10.1080/13658816.2012.663918).
- [34] Car2Go. "Five reasons why carsharing plays a decisive role in the breakthrough of electric mobility". In: (Apr. 2018).
- [35] M Clemente et al. "Modeling and simulation of an electric car sharing system". In: *25th European Modeling and Simulation Symposium, EMSS 2013* (Jan. 2013), pp. 587–592.
- [36] Michele Cocca et al. "Free floating electric car sharing design: Data driven optimisation". In: *Pervasive and Mobile Computing* 55 (Feb. 2019). DOI: [10.1016/j.pmcj.2019.02.007](https://doi.org/10.1016/j.pmcj.2019.02.007).
- [37] Felix Creutzig et al. "Energy and environment. Transport: A roadblock to climate change mitigation?" In: *Science (New York, N.Y.)* 350.6263 (Nov. 2015), 911–912. ISSN: 0036-8075. DOI: [10.1126/science.aac8033](https://doi.org/10.1126/science.aac8033). URL: <https://doi.org/10.1126/science.aac8033>.
- [38] Luis G. Willumsen Juan de Dios Ortuzar. *Modelling transport*. John Wiley and Sons, 2011, p. 608.
- [39] Joni Downs. "Time-Geographic Density Estimation for Moving Point Objects". In: vol. 6292. Sept. 2010, pp. 16–26. DOI: [10.1007/978-3-642-15300-6_2](https://doi.org/10.1007/978-3-642-15300-6_2).

- [40] Sokona E. Farahani S. Kadner K.A. Adler I. Baum S. Brunner P. Eickemeier B. Kriemann J. Savolainen S. Schlömer C. von Stechow T. Zwickel Edenhofer O. R. Pichs-Madruga Y. and J.C. Minx (eds.) "Summary for Policymakers. In: Climate Change 2014: Mitigation of Climate Change". In: 2014.
- [41] EEA. 2016. URL: <https://www.eea.europa.eu/publications/climate-change-impacts-and-vulnerability-2016>.
- [42] Rosano Mariangela Ferrero Francesco Guido Perboli and Vesco Andrea. "Car-sharing services: An annotated review". In: *Sustainable Cities and Society* 37 (Oct. 2017). DOI: [10.1016/j.scs.2017.09.020](https://doi.org/10.1016/j.scs.2017.09.020).
- [43] Guy Fournier et al. "Car-Sharing With Electric Vehicles: A Contribution To Sustainable Mobility?" In: *Interdisciplinary Management Research* 11 (2015), pp. 955–975.
- [44] Raúl Tomás García et al. "The kernel density estimation for the visualization of spatial patterns in urban studies". In: June 2015. DOI: [10.5593/SGEM2015/B21/S8.111](https://doi.org/10.5593/SGEM2015/B21/S8.111).
- [45] Stefan Illgen and Michael Höck. "Electric vehicles in car sharing networks – Challenges and simulation model analysis". In: *Transportation Research Part D: Transport and Environment* 63 (2018), pp. 377–387. ISSN: 1361-9209. DOI: <https://doi.org/10.1016/j.trd.2018.06.011>. URL: <http://www.sciencedirect.com/science/article/pii/S1361920918300865>.
- [46] Margaret O'Mahony John Brady. "Modelling charging profiles of electric vehicles based on real-world electric vehicle charging data". In: *Sustainable Cities and Society* 26 (2016).
- [47] Peter T Doran William R L Anderegg Bart Verheggen Ed W Maibach Stuart Carlton Stephan Lewandowsky Andrew Skuce Sarah A Green Dana Nuccitelli Peter Jacobs Mark Richardson Barbel Winkle Rob Painting Ken Rice John Cook Naomi Oreskes. "Quantifying the consensus on anthropogenic global warming in the scientific literature". In: *Environmental Research Letters* (Apr. 2016).
- [48] Sarah A Green Mark Richardson Rob Painting Robert Way Peter Jacobs Andrew Skuce John Cook Dana Nuccitelli. "Quantifying the consensus on anthropogenic global warming in the scientific literature". In: *Environmental Research Letters* (May 2013).
- [49] Eric O'Shaughnessy Eric Wood Joyce McLaren John Miller and Evan Shapiro. "Emissions Associated with Electric Vehicle Charging: Impact of Electricity Generation Mix, Charging Infrastructure Availability, and Vehicle Type". In: *National Renewable Energy Laboratory report* (Apr. 2016). URL: www.nrel.gov/publications.
- [50] Benedikt Jäger, Michael Wittmann, and Markus Lienkamp. "Analyzing and Modeling a City's Spatiotemporal Taxi Supply and Demand: A Case Study for Munich". In: *Journal of Traffic and Logistics Engineering* (Jan. 2016). DOI: [10.18178/jtle.4.2.147-153](https://doi.org/10.18178/jtle.4.2.147-153).
- [51] Maja Kalinic and Jukka Krisp. "Kernel Density Estimation (KDE) vs. Hot-Spot Analysis - Detecting Criminal Hot Spots in the City of San Francisco". In: June 2018.
- [52] D. Kypriadis et al. "Minimum Walking Static Repositioning in Free-Floating Electric Car-Sharing Systems". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 1540–1545. DOI: [10.1109/ITSC.2018.8569912](https://doi.org/10.1109/ITSC.2018.8569912).

- [53] Junghoon Lee, Hye-Jin Kim, and Gyung-Leen Park. "Relocation Action Planning in Electric Vehicle Sharing Systems". In: *Multi-disciplinary Trends in Artificial Intelligence*. Ed. by Chattrakul Sombattheera et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 47–56. ISBN: 978-3-642-35455-7.
- [54] Aditya Krishna Menon and Young Lee. "Predicting Short-Term Public Transport Demand via Inhomogeneous Poisson Processes". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. New York, NY, USA: ACM, 2017, pp. 2207–2210. ISBN: 978-1-4503-4918-5. DOI: [10.1145/3132847.3133058](https://doi.org/10.1145/3132847.3133058). URL: <http://doi.acm.org/10.1145/3132847.3133058>.
- [55] Richard Mounce and John D. Nelson. "On the potential for one-way electric vehicle car-sharing in future mobility systems". In: *Transportation Research Part A: Policy and Practice* 120 (2019), pp. 17–30. ISSN: 0965-8564. DOI: <https://doi.org/10.1016/j.tra.2018.12.003>. URL: <http://www.sciencedirect.com/science/article/pii/S0965856417315331>.
- [56] Ansar-Ul-Haque Yasara Yves Vanrompay Tom Bellemansa Davy Janssens Geert Wetsa Muhammad Usmana Luk Knapena. "A coordinated framework for optimized charging of EV fleet in smart grid". In: (2016).
- [57] NASA. 2017. URL: <http://https://climate.nasa.gov/scientific-consensus/>.
- [58] British Petroleum. *BP Energy Outlook 2019*. Feb. 2019. URL: <https://www.bp.com/en/global/corporate/news-and-insights/press-releases/bp-energy-outlook-2019.html>.
- [59] Pedro J. Zufiria R. Alvaro-Hermana Jesús Fraile-Ardanuy and Luk Knapen Davy Janssens. "Algorithm development for night charging electric vehicles optimization in big data applications". In: (2016).
- [60] Pedro J. Zufiria R. Alvaro-Hermana Jesús Fraile-Ardanuy and Luk Knapen Davy Janssens. "Peer to peer energy trading with electric vehicles". In: *IEEE Intelligent transportation systems magazine* (2016).
- [61] Vikas C. Raykar, Ramani Duraiswami, and Linda H. Zhao. "Fast Computation of Kernel Estimators". In: *Journal of Computational and Graphical Statistics* 19.1 (2010), pp. 205–220. ISSN: 10618600. URL: <http://www.jstor.org/stable/25651308>.
- [62] Martin Repoux, Burak Boyacı, and Nikolas Geroliminis. "Simulation and optimization of one-way car-sharing systems with variant relocation policies". In: Jan. 2015.
- [63] Sheldon M. Ross. *Stochastic processes*. Wiley, 1996. URL: https://en.wikipedia.org/wiki/Poisson_point_process#References.
- [64] Ralph Sims et al. "Transport. In: Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change". In: Sept. 2014. DOI: [10.1017/CB09781107415416.005](https://doi.org/10.1017/CB09781107415416.005).
- [65] Qingguo Tang and Rohana J. Karunamuni. "Fast and accurate computation for kernel estimators". In: *Computational Statistics and Data Analysis* 94 (2016), pp. 49–62. ISSN: 0167-9473. DOI: <https://doi.org/10.1016/j.csda.2015.07.015>. URL: <http://www.sciencedirect.com/science/article/pii/S0167947315001723>.

- [66] Stephen H. Schneider William R. L. Anderegg James W. Prall. "Expert credibility in climate change". In: *Proceedings of the National Academy of Sciences of the United States of America* (Apr. 2010).
- [67] Wenhao Yu and Tinghua Ai. "The visualization and analysis of urban facility pois using network kernel density estimation constrained by multi-factors". en. In: 20 (Dec. 2014), pp. 902 –926. ISSN: 1982-2170. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1982-21702014000400902&nrm=iso.
- [68] Rabih Zakaria. "Optimization of the car relocation operations in one-way car-sharing systems". Theses. Université de Technologie de Belfort-Montbéliard, Dec. 2015. URL: <https://tel.archives-ouvertes.fr/tel-01619291>.