POLITECNICO DI TORINO

CNR-Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni (CNR-IEIIT)

ENGINEERING MATHEMATICS COURSE

Enhancing Urban Mobility Infrastructure through Traffic Flow Modeling and Sensor-based Monitoring

MASTER THESIS

Relator: Fabrizio Dabbene

Correlator: Chiara Ravazzi

Student: Ingegneri Diego

ACCADEMIC YEAR 2022/2023

Contents

Chapter 1

Introduction

1.1 Background and Context

The traffic flow management problem represents one of the main challenges for the administration of large cities. Indeed, as urban areas become larger and more populated, and roads experience heavy traffic, congestion occurs more frequently. This has an impact on the livability of cities as well as on management costs. Technology has been relied upon to overcome these problems in recent years. In particular, "Intelligent Transportation Systems" (ITS) refer to the use of advanced technologies and applications to offer innovative solutions to the problem of traffic congestion, playing the role of traffic control by extracting data for transport planning and intervention.
Another key aspect for data mining is "Traffic State Estimation" (TSE). TSE refers to the use of partially observed and noisy traffic data to infer the values of traffic variables, such as flow, density, velocity, travel time, and other types of data. The TSE model has three principal components:

- A traffic flow model.
- The use of input data and partial data.
- The estimation approach.

These three components will be fundamental to model the main factors that we will evaluate in the context of urban traffic.

1.2 Scope and contribution

Aim of this thesis is to optimize traffic monitoring through the use of sensors. In particular, we optimise sensor placement by leveraging road layout to identify key sensor positions. This optimization aims to minimize the number of sensors required while ensuring complete observability of the traffic system. Another key task is to find a solution even in the case of partial data (i.e. with a limited number of sensors). Our contribution was to analyze the optimization methods and estimation approaches in the literature. Finally, apply them to case studies and real-life scenarios, demonstrating their correctness and efficiency through simulation and data validation.

1.3 Motivation

The primary cause of these issues is the overloading of road networks, resulting in urban congestion. These challenges manifest in the costs of managing urban areas, with damages from the excessive load on the urban road system being both perceptible and easily quantifiable. In particular, several impacts on individuals and the environment are readily observable:

- Traffic Jams and Stress:
  - Traffic jams can lead to increased stress and tension for both drivers and passengers. Long queues, slowdowns, and blocking situations can generate aggressive and dangerous behavior, escalating the risk of road accidents and fostering an overall atmosphere of dissatisfaction and frustration.
- Time Implications:
  - Italians dedicate an average of 10 hours and 40 minutes each to travel from Monday to Friday [15], translating to just over 2 hours a day on average. This considerable time investment diminishes productivity and efficiency, negatively impacting the overall quality of time, which, in turn, reflects on the productivity of a city.
- Economic and Lifestyle Effects:
  - Traffic and transportation inefficiencies significantly impact commercial activities and community quality of life. Challenges in accessing commercial areas, high transport costs, and time losses due to traffic jams can diminish the attractiveness of urban areas and shopping centers, adversely affecting the local economy and people's quality of life.

- Environmental Consequences:
  - Urban traffic generates a substantial environmental impact, contributing to greenhouse gas emissions and air pollution. Increased fuel consumption leads to elevated emissions of CO2 and other pollutants like nitrogen oxide and fine particles. These factors can exacerbate health problems, such as asthma and cardiovascular diseases, particularly affecting vulnerable groups such as children and the elderly.

Among the measures aimed at reducing traffic,at reducing emission levels and at encouraging the use of public transport, we can mention: car sharing and car pooling, taxi buses, limited traffic zones, network of runways cycle paths, creation of trolleybuses and tramways, underground metros, trains and surface "light" metros; the so-called Intelligent transportation systems (ITS).
Thanks to these systems, which connect digital technologies with vehicles, road infrastructures and more, it is possible to monitor, organize and manage public and private transport in an integrated way.

Traffic monitoring and management systems have become increasingly important in recent years to improve the quality of urban mobility. Thanks to real-time traffic monitoring, it is possible to control the flows of vehicles and pedestrians in the key points of the city, streamlining traffic and improving road safety and the efficiency of transport (public and private) and better monitoring of infrastructure.
Sensors are the ideal solution for traffic management activities, in fact each sensor, based on its characteristics, is able to detect information such as: vehicle speed, flow and road density. Depending on the monitoring objective, the sensors can focus on multiple lanes at once, single lanes, sections of the road or entire intersections. Since the problem of traffic monitoring is very recent, standard methods have not yet been developed. In particular, there is still a lot of confusion about which are the best sensors to use for data collection and which is the best localization. The latter represents the main problem of these technologies. For this reason, sensor localization will be the main problem we will address in this thesis as it represents a fundamental component in the development of intelligent urban mobility since it represents a subproblem of the problem known as network sensor location problem (NSLP).

1.4 State of the art

1.4.1 Network model

In the field of traffic dynamics modeling, three main types of models are commonly employed: microscopic, macroscopic, and mesoscopic models.

Microscopic models

Microscopic models delve into the intricate details of traffic behavior by meticulously describing the trajectory of each vehicle. These models capture the subtle interactions between vehicles, taking into account factors such as acceleration, deceleration and lane changes. In essence, microscopic models provide a detailed perspective, which offer insights into the specific movements and behaviors of vehicles within the traffic system. Each trajectory is described by means of an Ordinary Differential Equation (ODE).

Macroscopic models

On the other side of the spectrum, macroscopic models take a more holistic approach, depicting traffic as a continuous entity, like a fluid. Comparing the principles of fluid dynamics, macroscopic models provides a generalized representation of traffic flow. Instead of focusing on the details of individual vehicle movements, these models emphasize broad features such as traffic density and flow rates. In this way, they provide a comprehensive overview that is particularly useful for analyzing large-scale traffic patterns and making predictions about overall system behavior. The basis for any macroscopic traffic model is the conservation law of vehicles, which is expressed by means of a Partial Differential Equation (PDE).

Mesoscopic models

The mesoscopic model is a crossover between the first two models. The contrast between microscopic and macroscopic models shows the different ways in which traffic dynamics can be conceptualized and studied. While microscopic models excel at capturing the nuances of individual vehicle movements, macroscopic models offer a higher-level perspective, facilitating a more comprehensive understanding of the collective behavior of traffic as a continuous flow. This dual approach allows researchers and practitioners to navigate the complexities of traffic systems, from detailed interactions at the micro level to broader trends and patterns at the macro level.

1.4.2 Fundamental diagram

The development of the traffic models began with Greenshields et al.[29], which manually collected the data from a stretch of road for a period of time. By counting the vehicles and knowing the distance traveled and the road section he was able to obtain speed, flow and density. Plotting this data into a speed-density diagram, observed an inverse correlation between these variables proposing a linear function of velocity in terms of density, also known as Greenshield's Fundamental Diagram (FD).
The Fundamental Diagram is a pivotal concept in the realm of traffic flow theory. It delineates the observed connection between traffic variables under static traffic circumstances, often termed as steady or balanced traffic.
The typical FD theory assumes that the traffic state, with the idea by Lighthill and Whitham [30] and Paul I. Richards [31] they proposed the Lighthill-Whitham-Richards (LWR) model where in stationary traffic, both flow and velocity could be written as a function of density.

Therefore we can rewrite the conservation law as an equation of only the density.

1.4.3 Single road model

The traffic state of the network refers to the values of the density, flow and velocity which are defined for each road.
Since we are in the case when the entire network is in steady-state, so each road is characterized by a unique vehicular flow, that is, the outgoing vehicular flow is equal to the incoming vehicular flow .

1.4.4 Intersection model

Urban traffic networks consist of a collection of roads which are connected by intersections. In addition to this, network models require to specify how flow behaves at each intersection.
Therefore every vehicle will follow single road model and each route corresponds to an intersection model where the conservation law imposes constraints on the distribution of flows incoming and outgoing roads at each intersection.

1.5 Data collection

Data types are grouped into two categories according to the measurement methodology:

- stationary;
- mobile.

In our study we have been focus only on the types of stationary sensors since in our model, for the sake of simplicity, we have assumed steady state.

The traffic state of the network refers to the values of the density, flow and velocity which are defined for each road.
Variables that fixed sensors can collect include:

- Number of vehicles per unit time.
- Vehicle speed.
- Vehicle length or classification.

This category includes technologies such as: inductive loops, magneto dynamic sensors, infrared, passive acoustic sensors, ultrasonic sensors, radio sensors, triboeletric sensor, pneumatic tubes, radar [10]. (These sensors will be explored in depth in chapter 3).
For our study we have choosen to use sensors that only deal with the counting of vehicles passing by knowing the time frame in which data is collected. In this way we are able to calculate flows by dividing the number of vehicles counted by the time interval in which the count was made.
According to our model, two data sources are available:

- Flow sensors in single roads.
- Turning ratio sensors in intersections.

These types of sensors according to the model will be stationary and they can only collect data close to where they are installed. Through the data collected by the installed sensors we are able to collect information to calculate the flows in correspondence with the designated arches.
A Turning Ratio (TR) $r_{i,j}$ is the fraction of the flow in road i, that turns to road j at an intersection.

In the case of partial data we can calculate flows by applying the law of conservation of flow where in the case of the intersection model we also introduce the turning ratio $r_{i,j}$ into the equation.

1.6 Estimation approaches

Using the Graph representing the interested network we used optimization algorithms to find the optimal set of sensors to install for traffic monitoring in order to have full observability of the system using as few resources as possible.

This is known as the Network Sensor Location Problem (NSLP).
There are various types of approaches to address this problem:

- **Algebraic approaches**:
  - Hu et al.(2009) [12] put forward an assumption-free network sensor location problem and directly tackled it using the path-link incidence matrix.

  - ManWo Ng (2012) [14] redefined the problem using a link-node incidence matrix.

Unfortunately, these algebraic approaches are inefficient because these algebraic approaches did not fully leverage the topological tree-shaped attributes of the problem's solution.

For this reason in our study we will use the graphical approach which we will mention below and which we will delve into in chapters 5 and 6.

- **Graphic Approaches**:
  - Sheng-xue He (2013) [13] who provided an efficient algorithm based on the topology of the network and graph theory.

This approach offers the advantage of intuitively and visually discovering and proving properties related to the solution of NSLP. It provides a clear visualization that makes it easier to understand the solution. Moreover, it grants traffic management agencies the flexibility to select links for sensor installation according to practical needs.

Network sensor location problem

Since we said that we will use 2 sensor technologies, we respectively define $\mathcal{R}$ as the set of Turning Ratio sensors and $\mathcal{S}$ as the set of flow sensors. Our problem can be translated into the following optimization problem where $\mathcal{R}^*$ and $\mathcal{S}^*$ are respectively the optimal set of turning ratio sensors and flow sensors.

$$\begin{aligned} \mathcal{R}^*, \mathcal{S}^* = \underset{\mathcal{R},\mathcal{S}}{\operatorname{argmin}} \; & |\mathcal{R}| + |\mathcal{S}| \\ \textit{subject to} \quad & |\mathcal{R}| = n_{\mathcal{R}}, \\ & |\mathcal{S}| = n_{\mathcal{S}} \end{aligned} \tag{1.1}$$

where $n_{\mathcal{R}}$ and $n_{\mathcal{S}}$ represent the minimum number of sensors to be installed to have full observability.
To solve this optimization problem we chose to implement algorithms NSLP-based algorithms.

Hypothesis of the problem

We have chosen to break down problem (1.1) into 2 subproblems based on the following hypotheses:

1. There is a minimum number of sensors to be installed to have full observability linked to the network topologies.

2. we have a limited number of turning ratio sensors (i.e. we have a partial data collection).

Therefore the initial optimization problem is solved in 2 steps:

- **STEP 1**: (find the optimal set $\mathcal{R}^*$).
  Since we said that the TR sensors measure the turning ratio flow, we will look for the nodes that have the highest number of outgoing arcs in such a way as to minimize the number of flow sensors to be installed in the arcs later. Therefore we can see that the resolution of the problem is not only based on the minimum number of sensors to be installed, but also on the optimal position.

  This problem will be implemented and solved by algorithm 6.3 that we will introduce in chapter 6.

- **STEP 2**: (based on the partial data of $\mathcal{R}^*$ find the optimal set $\mathcal{S}^*$).
  In this optimization problem we calculate the spanning tree of the graph where there will be no need to install sensors. For the properties of the spanning tree, it is possible to derive the flow knowing only the incoming flow. After finding the optimal set $\mathcal{R}^*$ in step 1 and spanning tree, the remaining set on which to install the sensors will be the optimal set. $\mathcal{S}^*$.

  This problem will be implemented and solved by algorithm 6.4 that we will introduce in chapter 6.

1.7 Results and contributions

We exploited the properties of the algorithms based on spanning tree solution to uniquely obtain the flows of the system where sensors have not been installed. The algorithms studied and implemented ad hoc have been fundamental. We implemented the algorithms by taking cues from algorithms in the literature for studying graphs such as: Depth-first search (DFS) [34] is an algorithm for traversing or searching tree or graph data structures, the algorithms of Kruskal and Prim for finding the shortest path [32], Tarjan's algorithm to find the strongly connected components of a graph [33]. Based on these algorithms, we developed a method to save the coordinates of real-network in the graph. This made it possible to perform the calculations while maintaining the configuration of the initial map. Through this method we were able to show the graph within the chosen map.
The applications of these approaches are promising, as they enable the derivation of flows in areas without installed sensors. We achieved this by exploiting the constraints of the problem through the flow conservation law. However, significant limitations persist, primarily related to costs and implementation. The number of sensors required for traffic monitoring is intricately linked to the

topology of cities, introducing various physical and mathematical constraints. Despite notable performance improvements, sensor deployment and their optimal locations remain computationally and economically significant, albeit more feasible challenges. Data collection poses another fundamental limitation due to the availability of a limited number of sensors. Our subsequent chapters delve into the issue of partial data, where we discuss the number and location of sensors as crucial constraints. The results obtained are promising, demonstrating the ability to deduce flows through simulations even in areas without installed sensors, with a margin of error less than 10%. This was achieved using networks based on real maps with a limited number of turning ratio sensors (i.e., partial data). An interesting tradeoff emerged between flow sensors and turning ratio sensors, indicating that installing turning ratio sensors instead of flow sensors can reduce the total number of sensors needed for full observability
In particular, this study will make the following contributions:

- The use of graphical approach not only highlights the spanning tree feature of the subset of non-equipped links, simplifying the solution to NSLP, but also streamlines the process of link flow inference. Additionally, this approach offers the advantage of making the discovery and related proofs of solution properties to NSLP more intuitive and visual. Through this graphical approach, we can readily offer traffic management agencies flexibility in selecting links for sensor deployment in practical scenarios.

- Recognizing that traffic networks may already have sensors installed on certain links, we demonstrated how to address two problems from this new graphical perspective. One problem involves identifying potential links from which flows can be inferred based on sensor measurements. The other entails identifying the smallest subset of links that, when added to the subset of already equipped links, achieves full observability. Our research has shown that these two problems are interconnected, with the solution to the first providing a foundation for the second.

- We applied a graphical algorithm for an intriguing problem: determining the optimal locations for a given number of sensors in order to infer link flows on as many links as possible. To address this, we initially construct a spanning tree. Subsequently, by gradually adding links to the tree and striving to minimize the influence of emerging circuits on link flow inference, we arrive at the final subset of non-equipped links.

- In order to address the NSLP, we exploited the concept of a weighted graph and used existing algorithms to construct a minimum spanning tree. These tasks have made the solution to NSLP with prior requirements a routine and applicable practice, aligning with our approach of addressing problems from a graphical perspective.

1.8 Overview

In chapter 2 we give an introduction on urban mobility and its problems facing with the key points to solve it together with the role that the PNRR, which could have by impacting cities and their infrastructures. In chapter 3 we will give a description of the main sensors for traffic monitoring, listing the pros and cons and describing the role they have on urban mobility by exploiting their technology. In chapter 4 we will introduce the main concepts of traffic flow modelling by distinguishing their classification: microscopic, mesoscopic and macroscopic. We will introduce the law of conservation of flow and fundamental diagram (Greenshield' speed-density fundamental diagram) for all classification of traffic flow modelling. We will explain concepts such as concept as congestion, jam-density and capacity. Finally we will delve into sensor data collection and types of estimation approaches. In chapter 5 we will formulate mathematically all these concepts by introducing theorems, lemmas and propositions on the topology of the network so as to be able to develop the correct algorithms from an empirical point of view. Fundamental diagrams will be able to classify which data we have to use as input and what are the estimation approaches to collect them. Finally we will identify the best approach by implementing an algorithm and apply it to a case study. In chapter 6 we will further develop the approaches introduced previously both from a mathematical and algorithmic point of view, even in the case of partial data or with a limited number of sensors and using two different sensor technologies. After modelling the problem we will build and implement our algorithm in the case of partial data and test its correctness using a small graph. In chapter 7, with the help of commercial software we will carry out some simulations on real cases using the approaches previously tested with simulated data and we will compare them with theoretical cases showing their discrepancy. Finally, in Chapter 8 we will provide the conclusions of our study and where it can lead continuing this path.

Chapter 2

Urban Mobility and the Role of PNRR

In this chapter we will explore the challenges that urban mobility poses, we will analyze the infrastructure, the costs, the feasibility of the projects and the benefits that can be obtained from them. Based on this information we will see the role of the PNRR on urban mobility.

Essentially, urban mobility pertains to the movement of people and goods within a city. This term is particularly associated with large cities, as the concept, though seemingly straightforward, becomes intricate when considering factors such as infrastructure, transportation, and politics in megalopolises. Managing urban mobility poses a significant challenge for urban areas. Decision-makers and urban planners, in addition to financial limitations, grapple with multiple often conflicting demands. These include maintaining a high quality of life while fostering a conducive environment for economic activities, limiting traffic in sensitive zones without hindering essential movement of people and goods.
Urban mobility encounters numerous hurdles, with traffic congestion being one of the most formidable. Extensive economic studies underscore the societal cost of congestion, and smoother traffic flow in urban settings correlates with higher economic growth potential. Research indicates that optimized traffic flow can enhance worker productivity by up to 30 % [2] in heavily congested regions. In the European Union, road congestion frequently plagues urban areas and suburbs, impacting a large portion of EU citizens. However, historical trends demonstrate that expanding road capacity in urban regions results in increased traffic and subsequently worsened congestion. Consequently, alternative solutions must be explored.
Addressing a myriad of challenges is imperative, and the Recovery and Resilience Plan (PNRR) is expected to play a pivotal role in resolving or even enhancing the situation.

2.1 Urban Planning and Infrastructure Challenges

The relentless expansion of urban populations places ever-increasing pressure on urban mobility.[1] Presently, 56% of the global population resides in cities, and projections indicate that by 2050, this figure will soar to nearly 70%. Even cities not anticipating population growth are contending with surging transportation demands that strain urban infrastructure and space. An estimated 41 megacities are anticipated by 2030, with a projected 67.17 trillion passenger-kilometres in mass transit by 2050 [2].
The primary challenges confronting these cities are as follows:

- Traffic Congestion: Commencing with an examination of efficiency, the time commuters lose due to traffic congestion negatively impacts labor productivity. INRIX's 2019 Global Traffic Scorecard revealed that in the United States, drivers spent an average of 99 hours stuck in congestion, amounting to nearly USD 88 billion in lost productivity.[2]

- Public Health and Safety: Roughly 1.3 million individuals succumb to road accidents annually, with around 50 million sustaining injuries. More than half of these casualties involve pedestrians, cyclists, motorcyclists, and their passengers. Furthermore, traffic congestion leads to elevated levels of particulate matter in vehicle exhaust, a known health hazard. The World Health Organization states that these particles can deeply infiltrate lung passages and enter the bloodstream, causing cardiovascular, cerebrovascular, and respiratory issues. [4]

- Environmental Impact: Approximately 22% of global carbon dioxide (CO2) emissions are attributed to the transportation sector, and this percentage is predicted to increase with the continued rise in population [5]. In India, for instance, without substantial investments in public transportation and intelligent road technologies, CO2 emissions are projected to surge from 70 megatons in 2015 to 540 megatons by 2050.[6]

The themes of sustainable urban development have long played a pivotal role in shaping contemporary city projects and the management of urban transformations. These themes stem from heightened environmental concerns and the realization of the imperative for enhanced efficiency in utilizing available energy resources. This evolution is mirrored by the ongoing process of urban expansion and population growth, which is further exacerbated by ongoing climate change, biodiversity loss, and the depletion of natural resources. Consequently, the pursuit of sustainable development has become a paramount concern for an increasing number of European municipalities (Iveroth, Johansson, Brandt, 2013). In response to these pressing issues, urban sustainability and the concept of sustainable development have taken on central roles in crafting urban transformation strategies, shifting the paradigm towards the intricate notion of a "Sustainable ecosystem" (Newman & Jennings, 2008). Numerous European cities have embarked on a developmental trajectory marked by targeted

strategies and measures aimed at enhancing the comfort and liveability of urban environments, all while minimizing environmental degradation and adapting to the repercussions of climate change (Iveroth et al., 2013). The Leipzig Charter for Sustainable Cities, established in 2007, also marked the commencement of this fresh era focused on quality and urban sustainability (Cecchini, 2010). It underlined the imperative of amalgamating environmental, economic, and societal dimensions of urban quality, particularly emphasizing the role of public spaces. Achieving these goals necessitated the adoption of novel administrative efficiencies and a robust public mandate for transformative initiatives. In recent times, numerous European cities have confronted the task of formulating targeted strategies across various scales of intervention. These range from individual building scales (passive structures, bioclimatic design, optimal building orientation, energy generation through solar and photovoltaic systems) to neighborhood scales (both public and private green spaces, soil permeabilization, rainwater harvesting), and extending further to the overarching urban scale.
At this broader urban level, new possibilities emerge, encompassing strategies directed at fostering urban development centered around planning hubs. An overview of the challenges posed by urban planning and infrastructure can be synthesized in the subsequent table.

Urban Planning
· Light rail/tramway/bus · Public transport on water (hydrofoil/Ferry) · Carpooling (biogas/electric cars) and charging stations · Protected pedestrian paths, cycle paths and bike sharing · Integration between public/private transport modes · Integrated infrastructural arteries of mobility · Bridges/overpasses with green infrastructure · Establishment of the congestion charge · Limitation of parking spaces for private vehicles · Establishment of large no-traffic zones

2.2 The Actions of PNRR (Next Generation EU's Recovery and Resilience Facility)

The National Recovery and Resilience Plan (PNRR) is an integral part of the Next Generation EU program (NGEU), a comprehensive 750 billion euro package devised by the European Union in response to the pandemic crisis. This package consists of approximately half grants and includes the Recovery and Resilience Facility (RRF), with a six-year duration spanning from 2021 to 2026.

The RRF encompasses a total allocation of 672.5 billion euros (312.5 billion in grants and the remaining 360 billion in low-interest loans).[7]:

1. "Digitization, Innovation, Competitiveness, Culture": This mission earmarks over 49 billion euros (40.3 billion from the RRF and 8.7 billion from the Supplementary Fund). Its objectives involve propelling the nation's digital transformation, fostering innovation within the production sector, and directing investments towards two pivotal sectors for Italy—tourism and culture.

2. "Green Revolution and Ecological Transition": With a total of 68.6 billion euros (59.5 billion from the RRF and 9.1 billion from the Fund), this mission aims to enhance the sustainability and resilience of the economic system while ensuring inclusivity and fairness.

3. "Infrastructure for Sustainable Mobility": This mission allocates 31.5 billion euros (25.4 billion from the RRF and 6.1 billion from the Fund). Its primary objective centers on the development of modern, sustainable transportation infrastructure accessible across the nation.

4. "Education and Research": With a total of 31.9 billion euros (30.9 billion from the RRF and 1 billion from the Fund), this mission aims to fortify the education system, nurture digital and technical-scientific competencies, and advance research and technological transfer.

5. "Inclusion and Cohesion": This mission entails an allocation of 22.6 billion euros (19.8 billion from the RRF and 2.8 billion from the Fund). Its purpose is to facilitate labor market participation, including training, bolster active employment policies, and promote social inclusion

6. "Health": This mission designates 18.5 billion euros (15.6 billion from the RRF and 2.9 billion from the Fund) to reinforce prevention and health services in various regions, modernize and digitize healthcare systems, and ensure equitable access to care.

Of particular note is Mission 3, which deals with infrastructures for sustainable mobility. This mission is set to have 32 billion euros at its disposal to pursue the following objectives:

- Promoting rail freight and passenger transport and implementing the European Rail Transport Management System (ERTMS).

- Implement an advanced digital monitoring system.

- Environmental sustainability and energy efficiency of ports.

- Land and sea accessibility.

- Digitization of logistics systems.

For Mission 3, "Infrastructures for Sustainable Mobility," the allocation is 32 billion euros, distributed as follows:

- Railway works for mobility and fast connections in the country (26.7 billion euros).
- Safety and digital monitoring of roads, viaducts and bridges (1.6 billion euros).
- Ports of Italy integrated project (3.3 billion euros).
- Digitization of airports and logistics systems (0.4 billion euro).

Urban areas hold a primary position in the PNRR's priorities:

- Local public transport will be revitalized and reinforced, including the replacement of bus fleets and metropolitan trains.
- Sustainable mass rapid transport lines like tramways, trolleybuses, and rapid transit buses will be developed.
- Public mobility interventions will be aligned with strategic planning for urban area sustainability.
- Substantial technological investments are earmarked for major urban centers.
- The cities will see the funding of innovative programs for the quality of living (PINQUA) which will be judged worthy at the end of the evaluation of the proposals presented.

The PNRR's strategy revolves around the sustainability and resilience of network infrastructures and systems:

- Innovative passenger transport traction methods, including hydrogen experimentation for trains and the replacement of diesel vehicles with alternative traction buses, primarily electric ones.
- Development of tourist cycle paths within the National Network, traversing multiple regions, to bolster sustainable tourism and urban cycle path development.
- Rolling stock renewal through the acquisition of new trains.
- Implementation of a national cold ironing plan to electrify docks, as well as the Green Ports plan to enhance energy efficiency and waste management in ports.
- Energy efficiency of public residential buildings in the "Safe, green and social" program.

- Infrastructural digitization, including technological monitoring of road bridges and viaducts within the national network, digitization of airspace control systems by ENAV, sustainable logistics via dialogue platforms and artificial intelligence, the Brenner Digital Green Corridor project for CO2 emission reduction through hydrogen production from renewable energies, and the ERTMS system for railway traffic management, control, and onboard signaling.

2.3 Importance of Infrastructure for Mobility in Urban Areas

Enhancing urban mobility necessitates the development of robust infrastructure. The Commission, through the urban mobility package, aimed to bolster European cities in addressing their mobility challenges. This entailed a significant shift in approaching urban mobility to ensure the more sustainable growth of urban areas within the EU. Member States were encouraged to take coordinated and impactful actions in response.

Central to this package was the introduction of the Urban Sustainable Mobility Plan (PUMS), a strategic blueprint that leveraged existing planning tools. It diligently integrated principles of collaboration, public engagement, and assessment to cater to current and future mobility requirements, thereby enhancing the quality of life both within cities and their surroundings.

Policies and measures outlined in a PUMS encompassed the entirety of transport modes within urban areas – be it public or private, involving passengers or goods, motorized or non-motorized, and addressing circulation and parking issues. Moreover, the package advocated for more intelligent and regulated access systems in urban regions, along with the coordinated implementation of Intelligent Transport Systems (ITS) and improved road safety. By November 2019, an evaluation was underway to ascertain the package's effectiveness and its alignment with intended objectives. [8]

The first stride in infrastructure development involved crafting a set of standardized indicators for urban mobility. Pending a favorable impact assessment, this would be followed by legislative proposals urging Member States to systematically gather and present urban mobility data. This data would cover the adoption of Sustainable Urban Mobility Plans (SUMPs) in all core network and comprehensive TEN-T network urban nodes, as well as their surrounding areas. With this data in hand, regular reports would be compiled to track the progress made by Member States and urban nodes in their pursuit of more sustainable urban mobility.

Once again, the Urban Sustainable Mobility Plan (PUMS) came to the fore-

front. It constituted a strategic framework building upon existing planning tools and was meticulously guided by principles of integration, public engagement, and assessment. This plan effectively addressed the current and future mobility needs of people and goods, with the overarching goal of enhancing the quality of life within cities and their peripheries.
In a report issued by the European Commission [9], several crucial aspects are highlighted to underscore the significance of infrastructure.

The following key points are emphasized:

- The Commission's proposal to bolster assistance in the context of TEN-T urban nodes is commended. This includes nodes and connections situated in regions at the periphery and outermost areas of the EU. These nodes are vital for resolving issues related to missing links and inadequate connections, which remain substantial challenges. Notably, there is an emphasis on the necessity of enhancing seamless connectivity across diverse regions, spanning rural, mountainous, peri-urban, and urban zones. This requires a coherent framework of interoperable infrastructure, encompassing sustainable transportation modes such as railways, inland waterways, and intermodal hubs. The report underlines the significance of fostering connectivity, particularly through high-quality collective transport like public transport, active mobility options, and individual means of transportation.

- The report welcomes the formulation of Sustainable Urban Mobility Plans (SUMPs) that interconnect urban nodes with developmental initiatives. These plans have the potential to include measures that integrate sustainable transportation modes in innovative ways. It is suggested that the EuroVelo network should be effectively integrated into the TEN-T network to maximize synergies between the two networks. Furthermore, during the construction or enhancement of TEN-T infrastructure within urban nodes, due consideration should be given to accommodating bicycle usage.

- Emphasis is placed on ensuring intelligent and sustainable connections between urban nodes, high-speed main lines, stations, and paths for high-speed trains within urban environments. The overarching goal is to maintain continuous and uninterrupted connectivity along the primary network lines. Where feasible, integrated mobility solutions are to be preserved in metropolitan areas. Moreover, easy and intelligent connectivity with urban centers, as well as urban and peri-urban areas, is a priority.

- The Commission is urged to expand the existing roster of urban nodes to encompass other cities located along TEN-T corridors. These additions should align with agreements reached with Member States. Additionally, further support should be provided to facilitate the creation and enhancement of Sustainable Urban Mobility Plans (SUMPs) in these new urban nodes. The report suggests that local authorities could play a more integral role in governing and planning TEN-T network corridors. They

could also contribute to defining pertinent criteria for evaluating SUMPs through collaborative mechanisms.

Chapter 3

Sensor Technologies for Traffic Monitoring

In this chapter we will introduce the main types of sensors for traffic monitoring. We will focus on traffic monitoring through the installation of fixed sensors and we will show the various methods of data collection and how other variables such as speed, density, flow, occupancy rate, etc. can be obtained from the counting. Finally we will show the advantages and disadvantages of the properties of each sensor and show how by combining different types of sensors you can get a complete picture of traffic monitoring.

An accurate understanding of the current state of network traffic is a fundamental requirement for intelligent transport systems. These systems are designed to be adaptable, capable of adjusting their operational characteristics in response to external conditions, all in pursuit of specific objectives.

One pivotal aspect of these systems involves recognizing the passage of vehicles. This recognition enables the estimation of travel times across the network by tracking these identified vehicles. This mechanism is employed by certain public transport companies to provide information at stops. On the road, companies like "Highway companies" employ it by tracing Telepass users through buoys placed at various points along motorway sections. Similarly, in Great Britain, TrafficMaster uses automatic license plate reading systems to identify and track users. Notably, this identification system has more recently found application in the railway sector through the Eurobalise system. This integration augments the existing automatic locking system that has been in use for decades.

3.1 Introduction to Traffic Monitoring Sensors

The environmental and street monitoring system serves a dual purpose. On one hand, it identifies external conditions critical for safety reasons, such as detecting the presence of ice, snow, water on the road, as well as fog or smoke that impairs visibility. On the other hand, it assesses the impact of the transportation system on the surrounding environment. This includes measuring pollutant concentrations and noise emissions. This data is crucial for implementing appropriate control measures aimed at mitigating these impacts.
This data acquisition system encompasses various components. First, there's the traffic monitoring system, which captures descriptive variables related to vehicle or user flow within the system. Then there's the environmental and street monitoring systems that monitor external conditions and their effects on the environment. Lastly, the system involves a localization component that tracks the positions of individual vehicles, people, or entire fleets.
The incorporation of automated counting devices within designated sections serves to broaden the system's objectives. Beyond merely identifying congestion points, it also strives to reduce congestion. This is achieved by utilizing monitoring data to provide timely information and implement regulatory measures in real-time. The advantages of traffic monitoring extend beyond congestion management. It can address various other needs, including understanding and minimizing other externalities associated with traffic, particularly in terms of safety, air quality, and noise pollution.

3.2 Types of Sensors for Traffic Data Collection

Back in 2002, the Ministry of Infrastructure and Transport introduced the "Guidelines for the Design of Traffic Monitoring Systems." These guidelines offer a framework for determining the appropriate size of the system, selecting the suitable technologies, and outlining the execution methods for monitoring. It's important to note that, beyond real-time monitoring, periodic assessments can also be conducted on specific aspects of the network. These assessments might involve mobile counting devices or manual counts.
In essence, an automatic monitoring system is comprised of the following components:

- a sensor, a device made of a material sensitive to a quantity descriptive of the phenomenon, emits a signal in correspondence with a variation of the value of the greatness;
- a detector (or a measuring device), a device, usually electronic, capable to codify the signal produced by the sensor;
- a transmitter, system of transmission of the information encoded by the sensor to the detector and from this to the processor;
- a processor, which supplies data processed from detector data.

Automatic monitoring techniques are based on the detection of one or more of the perturbations induced by the vehicular passage on the surrounding environment:

- Weight (WIM sensors, pneumatic tubes; triboelectric sensors).
- Emission of mechanical waves (passive acoustic sensors, ultrasonic sensors).
- Reflection of visible radiation (video sensors).
- Reflection of invisible radiations (infrared, microwave, radio sensors).
- Electromagnetic induction (inductive loops, magnetodynamic sensors).

To detect these perturbations, sensors have been made that use the most several physical phenomena, including:

- Elastic deformation (pneumatic tubes and load cell sensors).
- Electrification by friction (triboelectric cables).
- Piezoelectric properties of crystals (piezoelectric sensors).
- Geomagnetism (magnetodynamic sensors).
- Electrical capacitance (capacitive sensors).
- Doppler effect (radar).
- Electromagnetic radiation at different frequencies.

Weight-sensitive sensors are designed to detect the passage of individual axles, measuring their weight based on the specific physical principles they rely on. However, they don't directly identify the entire vehicle's passage. Certain devices, like loops that operate on magnetic induction, can discern when a vehicle enters a specific detection area, but they don't provide speed information. In contrast, Doppler radar devices utilize a vehicle's speed to detect its passage. This enables them to not only count vehicles but also measure their speed. Some sensors are designed to perform dual measurements across two consecutive sections of a street, allowing them to determine vehicular speed. Furthermore, these sensors can be categorized based on their installation methods. Fixed sensors fall into two main types: intrusive and non-intrusive sensors. Intrusive sensors are embedded within the road pavement and are essentially immovable once installed. On the other hand, non-intrusive sensors are positioned on supports at the side or above the road.

Moreover, sensor technologies vary in how they gather data. Some sensors take isolated measurements of specific aspects, while others capture a sequence of measurements along an axis or within an area. This sequential approach provides more comprehensive insights, allowing for a finer characterization of

passing vehicles by factors such as length or type. For example, magnetodynamic sensors and video systems can provide detailed information about the type of vehicles passing by, while video tracking systems offer data on vehicle trajectories..

3.2.1 Traffic monitoring with fixed sensor networks

- Pneumatic tubes represent one of the earliest and simplest methods for counting traffic. These tubes consist of rubber tubes affixed to the road, placed transversely in relation to the vehicle's direction of travel. When a vehicle's wheel crosses over the tube, it compresses the rubber, generating a pressure wave within. This wave triggers a switch, often in the form of a membrane, located at one end of the tube. The switch, part of an electric circuit powered by a battery, closes each time it's activated. This signal is then captured by a counter, which registers each pair of pulses as a single vehicle unit.

- Triboelectric cables operate on the principle of electrification through friction, characteristic of certain dielectric materials. The sensor itself comprises a coaxial cable featuring a central conductor (steel wire) surrounded by a dielectric material and enclosed by a ring of twisted steel wires. When a vehicle's wheels pass over the cable, the friction between the steel wires and the dielectric material generates an electric charge accumulation. This charge variance is detected, indicating the passage of the vehicle's axle.

- Piezoelectric sensors utilize the piezoelectric properties of quartz crystals. When subjected to dynamic stress, these crystals become electrified. Uneven distributions of electricity form on the crystal's surface, leading to a potential difference between its stressed faces. This potential difference, proportional to the stress intensity and direction, translates to an electric signal indicating stress variation. Piezoelectric sensors are especially useful for measuring weight as they're sensitive to dynamic changes in stress. Class I piezoelectric sensors provide weight measurements, while class II types only detect vehicle axle passage.

- Inductive loops remain a prominent traffic monitoring system, not only in Italy but globally. These loops consist of electrical conductors wound into the pavement, powered by alternating current generators and connected to a counting device. Electromagnetic induction serves as the operating principle. Passing vehicles induce eddy currents in the loop's conductive material, creating a magnetic field opposing the coil's magnetic field. This phenomenon of mutual induction forms a transformer-like interaction between the coil and the vehicle, aiding detection.

- Infrared sensors function by utilizing an optical system equipped with an infrared-sensitive material positioned on the focal plane. Within these

devices, an infrared-sensitive element serves as the sensor, converting reflected or emitted energy into electric signals.

  - Passive infrared sensors are comprised of a receiving device designed to detect the energy emitted or reflected by infrared radiations. These radiations emanate from the road pavement or the surfaces of vehicles passing through their monitored area. These sensors are primarily used for identifying vehicle passages. If observing multiple areas, they can also measure speed and length. For measuring traffic flow rates, these sensors are typically mounted above each lane's centerline, often on elevated structures overlooking the road.
    As a vehicle passes over the sensor, it detects changes in radiant energy within the infrared range. This energy discrepancy is distinct from what's emitted by the road surface when there's no vehicle present. The detection of this energy shift signals the passage of the vehicle. The energy difference captured by the sensor is directly linked to the vehicle's absolute temperature and the emissivity of its metal surface. Emissivity, in this context, refers to the ratio between the actual energy emitted by a material and the energy emitted by an ideal source at the same temperature.
  - Active type infrared laser sensor: In contrast to passive infrared devices, active type sensors offer more than just detecting vehicle presence or passage. They also directly measure speeds and classify them. This sensor, similar to the passive type, consists of a source and a receiver for infrared rays, typically two laser diodes with a wavelength of 0.9 micrometers. These diodes emit pulses slightly separated in time, creating a time interval between the vehicle passing through the first and second beams of infrared rays. This interval enables the calculation of the vehicle's speed. Additionally, by analyzing the reflections from the vehicle, these sensors generate two-dimensional or even three-dimensional images, allowing for vehicle classification based on their shape. Transceivers within these sensors can also transmit information to users by encoding and modulating infrared rays with relevant data.

- Magnetic sensors are devices that detect the presence of metal objects by sensing changes in the Earth's magnetic field caused by these objects. In road applications, there are mainly three types used:

  - Biaxial Saturable Core Magnetometers: These are stationary detectors that measure changes in both vertical and horizontal components of the Earth's magnetic field caused by passing vehicles. They consist of two sets of windings: one arranged vertically to detect changes in the vertical component and another aligned horizontally with the vehicular flow to sense vehicle presence. Anomalies in the Earth's magnetic field lead to voltage changes in the circuit. When this change surpasses a set threshold, it indicates a vehicle's passage. Similar

data as from inductive loops are provided, often used for temporary monitoring or locations where loop installation is challenging.

  - Induction Magnetometers: These devices involve a coil sensitive to shifts in the lines of force in the Earth's magnetic field. Passing vehicles with iron elements cause variations in the magnetic field's intensity, producing a voltage change. This allows the detection of vehicle passages. These sensors provide measures of volume, lane occupancy, and speed. They usually require a minimum vehicle speed and aren't primarily used for detecting vehicle presence.
  - Magneto-Dynamic Sensors or VMI (Vehicle Magnetic Imaging) Sensors: These represent a modern intrusive sensor type capable of detecting various traffic data. They appear as small rectangular plates with a microprocessor powered by rechargeable batteries. VMI sensors can detect vehicle passages, speed, length, occupancy time, and time spacing. For longer-term detections, the sensor is installed within a vertical crack in the road pavement. For shorter surveys, it can be placed on the road surface inside a metal housing. These sensors analyze variations in the Earth's magnetic field caused by metal components in passing vehicles. Changes in the magnetic field induce electrical signals in the Giant Magneto Ratio (GMR) circuits inside the sensor. These signals, proportional to the magnetic mass of the vehicle, are processed by the microprocessor and stored in its memory. The recorded data can be transferred to a computer, where analysis software organizes and provides traffic information like vehicle counts, speed measurements, occupancy times, vehicle lengths, and time spacings.

- Microwave radars, also known as Radio Detection and Ranging (RADAR) systems, utilize energy with wavelengths between 1 and 30 cm, which corresponds to frequencies ranging from 1 to 30 GHz (commonly around 10.5, 24.0, and 34.0 GHz in traffic monitoring). These radars transmit energy beams, and when a vehicle passes through the beam, part of the energy is reflected back to the radar's antenna, enabling vehicle detection. There are two main types of microwave sensors:

  - Microwave Radar with Doppler Effect Detection: This type of radar, also referred to as continuous wave radar, employs a directional antenna emitting focused electromagnetic waves at a constant frequency (around 10 GHz). The radar's antenna, which also receives signals, is typically installed outside the road area on a pole, existing portal, or overpass. The sensor relies on the Doppler-Fizeau effect, which modifies the frequency of electromagnetic waves due to relative motion between source and receiver. It not only counts passing vehicles but also directly measures their speed. The speed accuracy is approximately within 2 km/h up to 100 km/h, with a 10 km/h tolerance. Notably, the frequency difference between emitted and

reflected waves increases with higher emission frequencies, making high-frequency radars more sensitive to vehicle motion. However, Doppler microwave radars cannot detect stationary vehicles or those moving very slowly (around 3 km/h or less).

- Microwave Radar with Presence Detection: The second type is the true-presence microwave radar, also known as Frequency Modulated Continuous Wave radar. Unlike Doppler radar, this radar emits a continuously modulated frequency, resulting in a changing frequency over time. The radar detects vehicle presence based on the frequency difference between emission and reception times. It divides its observation field into modules defined by its elliptical footprint on the road, allowing for signal analysis. When a vehicle transitions from one module to the next, the time taken helps calculate its speed. Besides counting vehicles and measuring their speed, this sensor can detect stationary vehicles. When connected to a remote control unit, it enables real-time reporting of accidents.

- Sonic and ultrasonic sensors: In the field of traffic monitoring, two types of acoustic sensors are currently in use:

  - Ultrasonic Sensors: are highly utilized for acoustic detection. They can assess various traffic parameters similar to what magnetic loops can evaluate: vehicular flow rate, occupancy rate, and transit speed. These sensors are compact devices comprising a sound wave generator and receiver, operating within a frequency range of 25 kHz to 60 kHz. Unlike other sensors, ultrasonic sensors don't require modifications to the road surface. They are typically installed perpendicular to a portal, on an overpass above the road, or horizontally at the roadside. These sensors rely on the principle of sound wave reflection. The time it takes for a sound wave to travel from the source, bounce off a reflecting surface (positioned orthogonal to the wave's propagation), and return to the source is directly related to the distance between the source and the reflecting surface. Utilizing this phenomenon, the sensor gauges its distance from the reflective surface (which can be the road or the top of a vehicle), enabling it to detect passing vehicles.

  - Passive Acoustic Sensors: capture vehicle passage, presence, and speed. These non-intrusive detectors recognize vehicles based on the noise they generate due to mechanical components and tire-road interaction. Typically, an array of acoustic sensors is positioned at regular intervals alongside roads and connected to a signal processing system. When a vehicle traverses the detection zone, an increase in sound energy is recorded, indicating the presence of a vehicle. This signal persists as long as the vehicle is within the surveyed area. As the vehicle leaves the zone, the sound energy diminishes below the detection threshold, leading to the signal disappearing.

3.2.2 Vehicle counting and traffic measurements

Traffic monitoring devices perform vehicle counting measurements passing through a section or present in a survey area. From these measurements, it is possible to make estimates of vehicular outflow variables and consequently apply traffic analysis or regulation models.
A vehicle counting device, positioned at a road section, count the number n of transited vehicles, similarly to what a single observer would do at the roadside, and thus provides a measure of the flow φ over a period of time T:

$$\varphi = \frac{n}{T} \tag{3.1}$$

Conducting two surveys in consecutive sections using two sensors connected to a single computing device equipped with a stopwatch enables the measurement of speed, akin to the process of two observers stationed at the ends of a trunk of length L with chronometers.
The time interval between the passage of the generic vehicle i in the first and in the second section gives its travel time:

$$t_i = t_{2,i} - t_{1,i}$$

The total time spent by vehicles in the detection area (i.e. in the trunk of length L) is the sum of the individual journey times of the vehicles entering and exiting the time T:

$$\sum_{i=1}^{n} t_i = \sum_{i=1}^{n} t_{2,i} - t_{1,i} = \sum_{i=1}^{n} t_{2,i} - \sum_{i=1}^{n} t_{1,i}$$

The connection between the duration vehicles spend and the observation time T is termed the occupancy rate, a key parameter offered by non-point detectors:

$$\tau = \frac{1}{T} \sum_{i=1}^{n} t_i$$

The total time spent by vehicles in the detection area makes it possible to determine the average spatial speed of the detected vehicles:

$$v_m = \frac{L}{\frac{1}{n}\sum_{i=1}^{n} t_i} = \frac{n}{\sum_{i=1}^{n} \frac{t_i}{L}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{\nu_i}}$$

Observation 3.1. The meaning of measuring the average spatial velocity stems from the equation of state employed in a flowing traffic scenario. In this equation, the non-uniform vehicular speed is computed as a spatial average rather than a temporal one.

By using this equation and assuming minimal speed fluctuations over time, it's possible to derive an approximate estimation of vehicular density. This is achieved by dividing the average occupancy rate by the length of the monitored area:

$$\rho = \frac{\varphi}{v_m} = \frac{n}{T} = \frac{\sum_{i=1}^{n} \frac{t_i}{L}}{n} = \left(\frac{1}{T} \sum_{i=1}^{n} t_i \right) \frac{1}{L} = \frac{\tau}{L}$$

Observation 3.2. The direct relationship between occupancy and density enables the formulation of algorithms to estimate network conditions (like the California and Payne automatic crash recognition algorithms) and traffic control strategies (such as the Alinea access control technique) in terms of occupancy rather than density.

3.3 Advantages and Limitations of Different Sensor Types

Each traffic sensor has its pros and cons based on the technology it uses. For example: Pneumatic tube-based counting devices offer advantages in terms of cost-effectiveness and ease of installation and removal. They have been extensively used in the past, and to some extent, even today, for periodic monitoring due to their few days of autonomy. However, they come with significant drawbacks including high counting inaccuracy, especially for high traffic flows where errors can surpass 20%, and the inability to capture data from multi-axle vehicles. There is also a risk of mechanical pipe damage, particularly from heavy vehicles, posing a danger of the pipe detaching violently from the road surface. In contrast, triboelectric sensors function similarly to pneumatic tubes but are more durable, less visible, and can be permanently installed in the pavement, although they are slightly more costly. Piezoelectric plate sensors, like quartz sensors, maintain their properties over time and are not affected by temperature changes. They share similar accuracy and cost characteristics as load cells.
Magnetodynamic sensors offer easy installation and removal due to their small size, long counting periods due to battery operation (around 90 days), counting precision, and the ability to capture data from vehicles of varying speeds.
Two types of microwave radar sensors have the advantage of weather insensitivity, avoiding performance issues during bad weather or fog. However, they

tend to be pricier than traditional road surface detectors, although their lower maintenance costs could make them more cost-effective in the long run.
Let's summarize all the pros and cons of the devices listed earlier:

1. **Weight-based Sensors** (WIM sensors, pneumatic tubes):
   - **Pros**:
     - Accurate and reliable measurements of vehicle weight and pressure.
     - Suitable for vehicle counting and classification.
   - **Cons**:
     - Require physical infrastructure integrated into the road surface.
     - Vulnerable to physical and environmental damage.
     - Installation and maintenance can be costly.
2. **Emission of Mechanical Waves** (passive acoustic sensors, ultrasonic sensors):
   - **Pros**:
     - Non-invasive detection of vehicle presence.
     - Useful for real-time traffic monitoring.
   - **Cons**:
     - Accuracy may be affected by ambient noise.
     - May require calibration and regular maintenance.
3. **Reflection of Visible Radiations** (video sensors):
   - **Pros**:
     - Provide detailed visual data on vehicles.
     - Enable traffic monitoring and driver behavior analysis.
   - **Cons**:
     - Effectiveness may be reduced in low-light or adverse weather conditions.
     - Image processing requires computational resources.
4. **Reflection of Invisible Radiations** (infrared sensors, microwave sensors, radio sensors):
   - **Pros**:
     - Perform well in various weather and lighting conditions.
     - Offer remote detection and continuous monitoring.
   - **Cons**:

  - May require precise calibration.
  - Some sensors can be influenced by moving objects nearby.

5. **Electromagnetic Induction** (inductive loops, magnetodynamic sensors):
   - **Pros**:
     - High precision in vehicle detection.
     - Suitable for traffic signal control and vehicle counting.
   - **Cons**:
     - Require in-ground installation.
     - May be sensitive to magnetic interference.
6. **Elastic Deformation** (pneumatic tubes, load cell sensors):
   - **Pros**:
     - Reliable detection of vehicle weight and presence.
     - Often used for vehicle classification based on weight.
   - **Cons**:
     - Installation can be invasive and require frequent maintenance.
7. **Triboelectric Charging** (triboelectric cables):
   - **Pros**:
     - Simple and cost-effective method for vehicle detection.
     - Can be used to detect vehicle presence or passage.
   - **Cons**:
     - Sensitivity and accuracy may vary based on environmental conditions.
8. **Piezoelectric Properties of Crystals** (piezoelectric sensors):
   - **Pros**:
     - High sensitivity and rapid response to mechanical loads.
     - Used for vibration monitoring and light traffic.
   - **Cons**:
     - Limited in detecting heavier vehicles or slow movements.
9. **Geomagnetism** (magnetodynamic sensors):
   - **Pros**:
     - Sensitive to changes in the magnetic field caused by vehicles.
     - Used for vehicle counting and classification.
   - **Cons**:

  - May require calibration and initial configuration.

10. **Electrical Capacitance** (capacitive sensors):
    - **Pros**:
      - Detect vehicle presence without physical contact.
      - Used to detect vehicles in specific areas.
    - **Cons**:
      - Sensitivity may be influenced by environmental conditions.
11. **Doppler Effect** (radar):
    - **Pros**:
      - Detect vehicle movement and speed.
      - Suitable for traffic control and speed monitoring.
    - **Cons**:
      - Higher electrical power requirements.
      - Possible interference from surrounding objects.

3.4 Integration and Data Fusion of Sensor Data

Sensor data fusion occurs as a follow-up step after sensor integration. In this process, data from different sensor sources are combined to uncover more detailed, precise, and valuable insights than what each source could provide on its own. Fusion can happen at different levels, ranging from basic fusion that combines raw signals to more advanced fusion that extracts intricate information or behavioral patterns.
The benefits of sensor data integration and fusion include:

- Enhanced Accuracy: Combining data from various sources leads to a more accurate and dependable measurement of the phenomenon being studied.
- Diminished Uncertainty: Employing multiple data sources helps reduce uncertainties and ambiguities that individual datasets might carry.
- Augmented Robustness: In cases where one sensor malfunctions or provides inaccurate data, other sensors can step in to offer valid information, thus enhancing the overall system's resilience.
- Elevated Comprehension: Integrating and fusing data allows for a deeper and more thorough comprehension of an event or situation, thereby facilitating better decision-making.
- Identification of Concealed Patterns: Data fusion has the potential to uncover patterns or correlations that might remain hidden when analyzing individual data sources.

Nevertheless, it's worth acknowledging that the integration and fusion of sensor data come with their own set of challenges. These challenges encompass aligning data, handling uncertainties, and dealing with the computational intricacies of processing fused information. To overcome these hurdles, experts rely on techniques from signal processing, data fusion algorithms, machine learning approaches, and advanced mathematical models.
Multi-technology systems blend different sensing technologies to leverage the strengths of each while compensating for their weaknesses.
A prime example of an integrated sensor system is the Siemens ASIM TT266 triple technology, which consists of:

- A Doppler radar for detecting speeds exceeding 12 km/h.
- Two passive infrared sensors for detecting vehicle presence and speeds under 12 km/h.
- An ultrasonic sensor for measuring vehicle height and classification.

This setup allows for more accurate measurements of traffic flow and enhances the ability to classify vehicles correctly. To set up the system, a central unit needs to be suspended above the road, with a dedicated sensor for each lane. These sensors can be mounted on a portal structure or a walkway. Additionally, a technical room is necessary to accommodate the equipment and ensure a power supply is available.
The parameters that can be monitored include:

- Counting of vehicles in transit.
- Vehicle speed and length.
- Time spacing.
- Presence of stationary vehicles.
- Classification of vehicles (5 classes).

Chapter 4

State of the Art in Traffic Flow Modeling

In this chapter we will apply the state of the art to traffic flow modeling. We will give a graphical representation of the model by introducing intersection and route parameters that can model traffic patterns in a coherent manner. We will classify and compare different traffic flow models: macroscopic, microscopic and mesoscopic. We will show what the input data of our model are and through the use of the stationary sensors introduced in the previous chapter, what the different estimation approaches are which highlight a common problem to be addressed: the NSLP (Network sensor location problem). We will show that there are 2 main approaches, the node-based approach and the graph approach. We will analyze and compare the two approaches focusing on the graphical approach that most efficiently addresses the NSLP problem.

4.1 Traffic flow Modeling Overview

Urban traffic networks consist of a collection of roads which are connected by intersections, as shown in Fig. 4.1 take from [14]. Each road section follows described above.

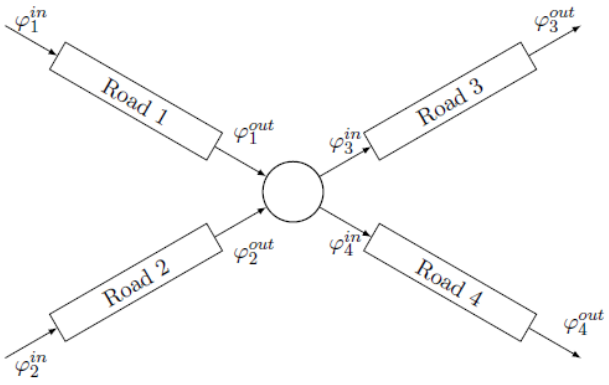

Figure 4.1: Flow intersection model

We define the following notation:

- $\varphi_i^{in} = \varphi_i(0,t)$ the inflow of the i-th road.
- $\varphi_i^{out} = \varphi_i(l_i,t)$ the outflow of the i-th road.

The primary element within junction models revolves around vehicle routing, entailing the task of efficiently distributing incoming traffic flows at intersections to their corresponding outbound roads based on user preferences.

An extensively employed technique, such as the Turning Ratio (TR) method, addresses this challenge through intersection parameters called turning ratios (denoted as $r_{i,j}$).
These ratios signify the proportion of traffic from road i, that is φ_i^{out}, that transitions onto road j at the intersection.
A parallel solution to this issue involves Traffic Assignment (TA), utilizing Origin-Destination (OD) matrices. These matrices, coupled with a specialized cost function, facilitate the computation of optimal routes. Modern strategies, including Dynamic Traffic Assignment (DTA), ascertain optimal routes in real-time by considering the current network state.
The second facet of junction models pertains to calculating road inflows and outflows, contingent on the densities of each road. One of the frequently adopted models is the flow-maximizing model, an extension of the CTM adapted for networks. This model assumes cooperative driving, aiming to maximize total intersection flow while abiding by structural constraints. Nevertheless, it tends to overestimate real-world flows, particularly failing to describe deceleration near intersections. Alternate models have been suggested, incorporating factors like traffic light cycles. Nonetheless, constructing realistic junction models remains an ongoing challenge in the realm of traffic network analysis.

Typically, the aforementioned models necessitate substantial computational resources.
As an alternative, region-based models divide the network into sections, treating them as cohesive units. Here, the state variables correspond to average density and flow for the entire region. The Macroscopic Fundamental Diagram (MFD) establishes a relationship between a region's average density and its outgoing flow. This concept has progressed significantly, accommodating heterogeneous traffic compositions like mixed car and bus traffic. The MFD framework has also been employed to address congestion propagation due to internal and external trips. A central issue with MFD involves instantaneous flow changes due to shifts in region density, potentially leading to vehicles with infinite speed. To mitigate this, MFD formulations accounting for delays have been proposed.

4.2 Traffic Flow Models Classification and Comparison

We start to model traffic dynamics. The principal three types of model are microscopic, macroscopic and mesoscopic where the latter refers to an intermediate model between the first two.

4.2.1 Microscopic models

A microscopic model concerns the description of the behavior of a vehicle, or rather of a pair of vehicles, in particular the trajectory of each individual vehicle. The model considers that each driver is influenced only by the vehicle in front in fact is known as car-following-models. Each trajectory is described by means of an Ordinary Differential Equation (ODE).
Let α the index of vehicle, we can write $s_\alpha(t)$ the position of vehicle at time t and we can describe trajectory as:

$$\begin{cases} \dot{s}_\alpha(t) = v_\alpha(t) \\ \dot{v}_\alpha(t) = a(s_\alpha - s_{\alpha-1}, v_\alpha, v_{\alpha-1}) \end{cases} \tag{4.1}$$

where $\alpha-1$ is the index of the vehicle in front, and a is the acceleration function. Microscopic model are not suitable to describe the general condition traffic or to serve for estimation or control techniques but they are suitable to describe the effect of single vehicle in the traffic flow therefor they are useful for providing realistic simulations that serve as data validation for other methods.

4.2.2 Macroscopic models

A macroscopic model, on the other hand, is suitable to describe the general condition traffic because it concerns the schematization of a traffic flow, like a fluid dynamics. In fact the principal variables are density, flow and velocity.

we consider $\rho(x,t)$ the traffic density at time t around a point x with unit of measure veh/km, $\varphi(x,t)$ the traffic flow at time t around x with unit of measure veh/h and the velocity $v(x,t)$ at the time t around x whit unit of measure km/h. The macroscopic traffic model is described by the conservation law of vehicles that is a partial differential equation (PDE):

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}\varphi(x,t) = 0 \tag{4.2}$$

Moreover the three variable mentioned above are related by hydrodynamic relation:

$$\varphi(x,t) = v(x,t) \cdot \rho(x,t) \tag{4.3}$$

From this relation an inverse correlation was observed between these variables and it can be obtained linear function between the flow and the velocity in terms of density.
This linear function is also known as the Greenshield's Fundamental Diagram (FD) (see Fig. 4.2).

With this idea will be proposed later the Lighthill-Whitham-Richards (LWR) model Showing the following relationships:

$$\varphi(x,t) = \Phi(\rho(x,t)) \quad , \quad v(x,t) = \Psi(\rho(x,t)) \tag{4.4}$$

which assumes that the velocity and flow are a function of the sun traffic density.

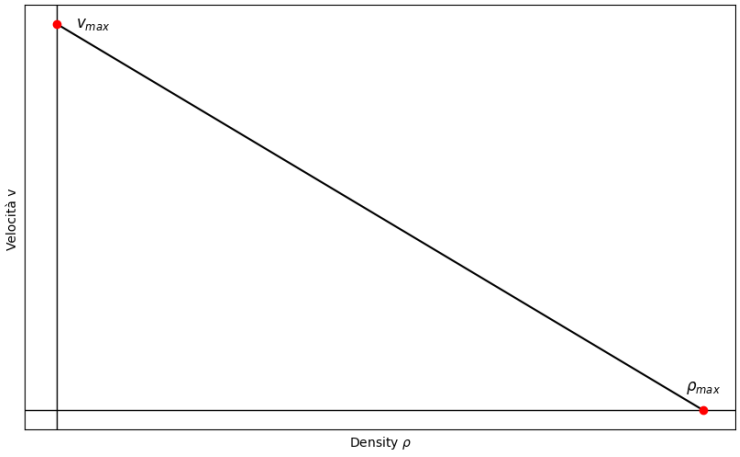

Figure 4.2: Greenshield's speed-density fundamental diagram

.

Where $\Phi(\rho(x,t))$ and $\Psi(\rho(x,t))$ are known respectively as the flow-density and speed-density FD.

Therefore we can rewrite the conservation law as an equation of only the density:

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}\Phi(\rho(x,t)) = 0 \tag{4.5}$$

From the figure 4.2 we can obtain several proprieties:

- $\Phi(0) = 0$.
- Let ρ^{max} the maximum density value, than $\Phi(\rho^{max}) = 0$. ρ^{max} is known as the $Jam\ density$.
- Φ is *convex* or if we are close jam density only unimodal.
- $v^{max} = \Phi(0)$ is the $free-flowspeed$ and it's a finite positive number that is the maximum vehicle speed.
- $w = \Phi(\rho^{max})$ is the *congested* wave speed and it's a finite negative number.
- φ^{max} is kwnow as the *capacity* and FD has a unique maximum value.

The LWR is the most commonly used model for traffic research applications and his discretization version is known as the Cell Transmission Model (CTM). In this model, a road is divided into an arbitrary number of cells of equal length. Therefore it is possible to obtain the density of the i-th cell from the one after as hypothesized for the traffic models. We can write as follows:

$$\rho_i[k+1] = \rho_i[k] + \frac{\Delta t}{l_i}(\varphi_{i-1}[k] - \varphi_i[k]) \tag{4.6}$$

where $\rho_i[k]$ is the density of cell i at time step k, $\varphi_i[k]$ is the flow between cells i and $i+1$, l_i is the cell lenght, and Δt is the discretization time. Let D_i the maximum flow that exit from cell ii an S_i the maximum flow that can enter in the cell. We can write as follows:

$$D_i = min(v^{max}\rho, \varphi^{max}) \tag{4.7}$$

$$S_i = min(w(\rho^{max} - \rho), \varphi^{max}) \tag{4.8}$$

While the inter-cell flow as:

$$\varphi_i[k] = min(D_i[k], S_{i+1}[k]) \tag{4.9}$$

This values are obtained by using a particolar FD with triangular chaper, ann CTM calculates D_i and S_i.
Initial-order models like the Lighthill-Whitham-Richards (LWR) model and the Cell Transmission Model (CTM) have limitations in accurately depicting certain real-world traffic behaviors, such as stop-and-go waves and capacity drops.

These models fall short because they immediately adjust vehicle speeds in response to density changes, following the Fundamental Diagram (FD).

An approach to address these limitations involves second-order models. These models introduce an additional equation alongside the conservation law to represent gradual changes in vehicle velocity. One prominent example of a second-order model is the Aw-Rascle-Zhang (ARZ) model:

$$\begin{cases} \frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(v\rho) = 0 \\ \\ \frac{\partial}{\partial t}(v + p(\rho)) + v\frac{\partial}{\partial x}(v + p(\rho)) = 0 \end{cases} \tag{4.10}$$

Where $p(\rho)$ is an increasing of function of ρ and is called the pressure.

4.2.3 Mesoscopic model

The third category of frameworks is known as mesoscopic models, situated between the broader macroscopic models and the more detailed microscopic models that focus on individual elements. Within these moderately detailed models, vehicles are grouped into packs sharing similar attributes such as speeds and travel times. This grouping facilitates streamlined calculation processes.
For computing the flow velocity of a vehicle pack, mesoscopic models employ expressions derived from outflow characteristic curves. These curves depict the relationships among average travel speed, flow/capacity ratio, and vehicular density. Capacity models, vehicle queuing models, and velocity/density models are used in mesoscopic models to depict the progression of vehicle outflow over time. Hence, accurate calibration of sub-model parameters is crucial for generating simulations that closely resemble reality.
Models within this category simulate network performance at an aggregated level, utilizing collective variables like capacity, flow, and density. Traffic flows are represented discretely by tracing individual packet movements, characterized by departure times and specific paths, often concentrated around particular points. This assumption becomes more realistic as packet sizes decrease. Due to their attributes, mesoscopic models are applicable to diverse networks and can even simulate phenomena like tail formation and backward propagation with reasonable computation times.
The Continuous-Time Model (CTM) is also categorized within this family of models

4.3 Input Data for Traffic Flow Models

Traffic data can be collected using stationary sensors and Floating Car Data (FCD).

4.3.1 Stationary sensors

They are fixed at pre-established points such that they can collect the right kind of data. The data types are collected based on the sensor type for example: inductive-loop detectors, microwave and infrared radars, and traffic cameras. Usually the most collected data are:

- Number of vehicle per unit time.
- Vehicle speed.
- Vehicle length or classification.
- Occupancy (percentage of time that the sensor detects a vehicle).

Depending on the technologies employed, a significant portion of this data might remain inaccessible. For instance:

- Single induction loops can offer precise flow measurements but cannot directly gauge vehicle speeds.
- Radar sensors excel at measuring vehicle velocities with a commendable degree of accuracy, but they struggle to identify stationary vehicles and might skip counting when multiple vehicles share the same speed.
- Traffic cameras gather exceptionally detailed traffic information, but their usage involves high costs not only in terms of the cameras themselves but also the extensive image processing required. Moreover, these cameras are susceptible to environmental factors like rain or fog.

Many stationary sensors lack the capability to differentiate between individual vehicles, thereby making it impossible to construct an Origin-Destination (OD) matrix. In contrast, Automatic Vehicle-Identifier (AVI) sensors, such as license-plate readers, are adept at collecting this specific type of data. Additionally, alongside traditional sensors, the proliferation of Wi-Fi and Bluetooth (BT) enabled devices has introduced novel sources of traffic data.

4.4 Estimation Approaches for Traffic Flow Modeling

The installation and upkeep of stationary sensors in a large urban area come with substantial expenses. Efficiently determining the right quantity and strategic placement of these sensors is crucial for cost reduction while maintaining data accuracy. This challenge is recognized as the Network Sensor Location Problem (NSLP).

It's widely acknowledged that solving this type of problem falls into the category of NP-complete, which means finding optimal solutions can be highly

complex and time-consuming. Consequently, various heuristics have been introduced over the years to provide practical and effective approaches for addressing this challenge.

- One approach involves formulating the problem by considering all potential network paths, constructing a matrix of constraints to establish boundaries on the required sensor count for comprehensive flow estimation.
- Another technique centers on flow conservation equations at intersections instead of path flows. This method avoids path enumeration and calculates sensor requirements based on graph topology, nodes, and edges.
- An algorithm for flow sensor placement has been proposed, leveraging network topology and graph theory. However, due to its minimal network assumptions (primarily topology), it may entail a high sensor count for practical scenarios.
- More robust strategies for addressing this issue have been explored.
- Some studies have tackled the problem of measurement noise, suggesting a trade-off between sensor count and flow estimation resilience against noise.
- Initially, Traffic State Estimation (TSE) methods were devised for one-dimensional roads like highways and freeways.
- An alternate method utilized an Extended Kalman Filter (EKF) combined with the Cell Transmission Model (CTM) to estimate highway section densities, utilizing data from stationary sensors at specific sections. This involves linearizing the CTM around a current state and comparing measurements with linearized model predictions. Subsequently, GPS data was integrated to enhance highway state estimation through a Lagrangian model as input to the EKF. More recently, connected vehicle data was incorporated, further refining state estimation.
- Another prevalent TSE technique involves Luenberger-like observers, switching between free-flow and congested states, then correcting estimates based on measurements. Particle Filters (PF) have shown superior outcomes compared to Kalman Filter-based methods, especially when augmented with connected vehicle data. Although PF offers precise estimation, it comes with increased computational demands.
- Over the years, various methods like Unscented Kalman Filters (UKF) and Bayesian probabilistic model-based Expectation–Maximization Extended Kalman Filters (EM-EKF) have been used. However, urban networks have received less attention compared to highways due to the complexity of modeling traffic dynamics at intersections, managing entry and exit flows, and the limited sensor data availability.

4.5 Problematics and Contributions in Traffic Flow Modeling

4.5.1 Network sensor location problem

This intriguing issue of determining the optimal positions for deploying a reduced number of counting sensors to deduce flows on other connections and their variations is referred to by Hu et al. (2009) and Castillo et al. (2010, 2011) as Network Sensor Location (NSLP) matters. Within the literature, NSLP has predominantly been approached as a subset of broader problems, such as estimating the origin-destination (O-D) matrix. In contrast to treating NSLP as a distinct problem, a foundational linkage approach using the incidence matrix of connecting paths has been proposed. However, the necessity for path enumeration leads to only approximate conclusions.
To circumvent the cumbersome path enumeration, Man Wo Ng in Ng (2012) recently introduced a node-centric approach to NSLP employing the incidence matrix of nodes, significantly streamlining the computations. Nevertheless, even this algebraic approach, akin to the one presented by Hu et al. (2009), faces challenges when applied to real-world, large-scale networks. Although both Hu et al. (2009) and Ng (2012) recognized the potential of granting traffic management agencies flexibility in selecting connections for sensor deployment, they lack an effective methodology due to their failure to fully exploit topological features for identifying such subsets.
This limitation also hampers the efficient resolution of other sensor placement issues. The two models proposed by Hu et al. (2009) and Ng (2012) each possess their advantages and limitations. Recent studies, such as those by Gentili and Mirchandani (2012), conclude by outlining several promising directions for future research

4.5.2 A node based approach

The approach of Ng(2012) makes important contributions in the continuation of the research:

- The approach successfully reconfigures the issue of connection observability, initially introduced by Hu, Peeta, and Chu, utilizing the node-link incidence matrix. This transformation simplifies the laborious task of enumerating routes by instead enumerating nodes, all the while avoiding the introduction of new complexities.

- While Hu postulated the potential existence of an upper threshold for the minimum number of sensor-equipped connections required to ensure full observability across all connections, Ng also speculated that this upper limit is "dictated by the network's structure, regardless of the total count of links within the network." Through the utilization of the node-based approach, Ng substantiated this notion by deriving a clear expression for

this upper limit (thus confirming the first part of the conjecture) and further demonstrated that this upper limit is contingent upon the network's size, both in terms of nodes and links (thereby disproving the second part of the conjecture).

- Lastly, considering the possibility of pre-existing sensors on transport networks, Ng illustrated, via the node-based approach, the feasibility of:
  - Identifying an additional subset of links that should be equipped with sensors.
  - Inferring the specific link streams that can be deduced from these newly equipped sensors.

4.5.3 A graphical approach

A graphical approach to identify sensor locations for link flow inference has been treated by in [13] by Hu, Peeta and Chu (2013). They introduce a graphical strategy aimed at determining the most compact set of links within a traffic network where counting sensors should be installed. This installation facilitates the inference of flows across all other unmonitored links. This study highlights the inherent topological tree characteristics of solutions, enabling traffic management authorities to conveniently and flexibly select links for sensor deployment in practical scenarios. The approach addresses four key problems:

- The first two concerns address traffic networks that already feature sensors on specific links. They involve identifying the subset of links from which link flows can be extrapolated based on sensor measurements. Additionally, the task involves pinpointing the smallest subset of links necessitating counting sensor installation to accurately infer link flows on all unmonitored, non-equipped links.

- The third problem revolves around identifying optimal positions for a predetermined number of sensors. The aim is to deduce flows across the maximum number of links while incrementally expanding the scope of links included within circuits.

- The final challenge revolves around identifying the smallest set of links that should host sensors. This strategy seeks to simultaneously fulfill prior requirements while deducing flows across all remaining links. This is achieved through constructing a minimum spanning tree. These methodologies find applications in various long-term planning scenarios and link-centric applications within traffic networks.

This study addresses the challenge of Traffic State Estimation (TSE) within urban networks, employing diverse data sources such as stationary flow sensors.

- Initially, we tackle the issue of estimating flow and density for individual roads within an urban network while considering stable conditions.

Here, we introduce an approach to determine the optimal placement and minimum number of flow and Turning Ratio (TR) sensors. Notably, our method overcomes the complexities of the Network Sensor Location Problem (NSLP).

- The second aspect involves the dynamic evolution of density and flow on all roads within the urban network. Our primary innovation lies in presenting a data-driven TSE technique tailored for general urban networks, eliminating the need for a Fundamental Diagram (FD). This approach takes into account heterogeneous data originating from flow sensors and TR sensors.

Chapter 5

Density and Flow Estimation in Steady-State Conditions

In this chapter we will give mathematical formalization to our model by introducing theorems, lemmas and propositions by analyzing its main properties (the spanning tree property will be fundamental which will help us in the efficiency of the calculations in the previous chapters). We will address the problem of the complete observability of our model which, since we will use a graphical approach, will have limitations given by the topology of the network. These limitations translate into the constraint of installing a minimum number of sensors to have full observability that derive from the topology of the system, i.e. it will depend on the number of nodes and edges of our graph.
By exploiting the topological properties of the graph we will develop a technique for the full observability of our system using the minimum possible number of sensors exploiting the spanning tree and the flow conservation law. By combining these two techniques we can obtain the flow even in arcs where no sensors have been installed, thus limiting the number of sensors to be installed but always having full observability of the problem. Finally we will implement the algorithm and show that it works using a known case study.

5.1 Fundamentals of Density and Flow Estimation

We consider a directed graph where the node of the graph are partitioned in two disjoint sets $\mathcal{C}$ and $\mathcal{N}$. $\mathcal{C}$ corresponds to source and sink nodes of the network and $\mathcal{N} = \{1, 2, ..., n_N\}$ corresponds to intersections which are not able to generate or store vehicular flow. The edge $\mathcal{E} = \{1, 2, ..., n_\mathcal{E}\}$ corresponds to the set of roads of the network. Denote $\mathcal{I}(k)$ as the set of incoming edges to some node k

and $\mathcal{I}(k)$ as the set of outgoing edges from k. We define a partition of $\mathcal{E}$ in three sets: $\mathcal{E}_{in} = \bigcup_{k\in\mathcal{C}} \mathcal{O}(k)$ are the boundary incoming roads, $\mathcal{E}_{out} = \bigcup_{k\in\mathcal{C}} \mathcal{I}(k)$ are the boundary outgoing roads, and $\mathcal{E}_{net} = \mathcal{E} \setminus (\mathcal{E}_{in} \cup \mathcal{E}_{out})$ are the internal roads of the network. The following definitions are taken from [14]:

Definition 5.1.
A feasible traffic network is a directed graph $\{\mathcal{C} \cup \mathcal{N}\ ,\ \mathcal{E}\}$ such that the following conditions are met:

- Every edge is part of a path that starts with an edge from $\mathcal{E}_{in}$ and ends with an edge from $\mathcal{E}_{out}$.
- The graph contains no self loops.
- $\forall k \in \mathcal{N}$ we have $\mathcal{I}(k) \cap \mathcal{E}_{out} = \emptyset$ and $\mathcal{O}(k) \cap \mathcal{E}_{in} = \emptyset$.
- There is no production or storage of vehicles in the nodes in $\mathcal{N}$.

5.1.1 Single road model

The traffic state of the network refers to the values of the density, flow and velocity which are defined for each road. Denote by ρ_i the density of road i. From the conservation law, we have:

$$\frac{d}{dt}\rho_i(t) = \frac{1}{l_i}(\varphi^{in}(t) - \varphi^{out}(t)) \tag{5.1}$$

where φ^{in} is the incoming (or upstream) flow and φ^{out} is the outgoing (or downstream) flow.
This chapter is limited to the case when the entire network is in steady-state. This is formally true only when external boundary flows are constant and the network dynamics reach equilibrium, or it is approximately true when the external boundary flows slowly evolve in time.
This conditions establishes that:

$$\rho_i(t) = \rho_i,\ \varphi^{in}(t) = \varphi^{in},\ \varphi^{out}(t) = \varphi^{out} \qquad \forall i \in \mathcal{E}\ \forall t \tag{5.2}$$

This in turn implies,

$$0 = \varphi^{in} - \varphi^{out} \tag{5.3}$$

so each road $i \in \mathcal{E}$ is characterized by a unique vehicular flow:

$$\varphi_i = \varphi^{out} = \varphi^{in} \tag{5.4}$$

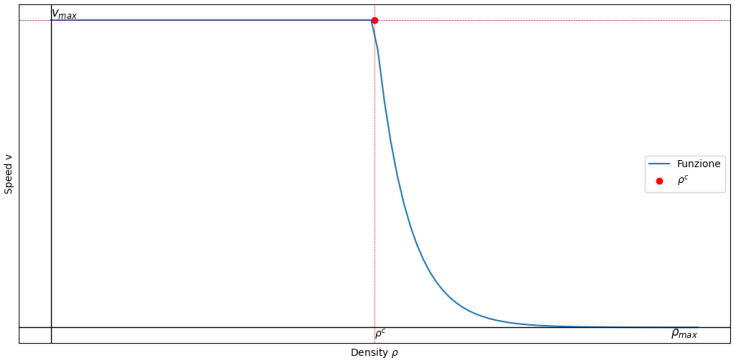

Figure 5.1: Speed-density relation triangular fundamental diagram

Furthermore, we adopt the triangular fundamental diagram to model the relationship between density and flow in equilibrium:

$$\varphi_i = \Phi(\rho_i) = \begin{cases} v_i^{max}\rho_i & if \quad 0 \leq \rho_i \leq \rho_i^c \\ w(\rho^{max} - \rho_i) & if \quad \rho_i^c \leq \rho_i \leq \rho_i^{max} \end{cases} \tag{5.5}$$

as shown in Fig. 5.1 this FD has as parameters the free-flow speed v_i^{max} , the congestion-wave speed w_i, the jam density ρ^{max}, the critical density ρ_i^c , and the maximum road capacity φ_i^{max} .

Note that these parameters are related to each other, and knowledge of just three is sufficient to calculate the rest.

Taking the relationship $\varphi_i = v_i\rho_i$ where vi is the space-mean speed of road i, the speed-density fundamental diagram is given by

$$v_i = \Psi(\rho_i) \begin{cases} v_i^{max} & if \quad 0 \leq \rho_i \leq \rho_i^c \\ w_i(\frac{\rho_i^{max}}{\rho_i} - 1) & if \quad \rho_i^c \leq \rho_i \leq \rho_i^{max} \end{cases} \tag{5.6}$$

5.1.2 Intersection Model

If a road has a density higher than its critical value, it is said to be in congested regime. Otherwise, the road is said to be in free-flow. At the network level, the

conservation law imposes constraints in the flow distribution of the incoming and outgoing roads in each intersection.

Consider an internal intersection $k \in \mathcal{N}$, such that it is neither a source or sink of traffic flow. Therefore, flow conservation requires that the total incoming flow must be equal to the total outgoing one,

$$\sum_{j \in \mathcal{O}(k)} \varphi_j - \sum_{i \in \mathcal{I}(k)} \varphi_i = 0 \tag{5.7}$$

5.2 Complete Observability Problem in Traffic Flow Estimation

We initiate the process of reconfiguring the network by introducing a solitary virtual centroid.
Within a transportation network, there exists at least one pivotal point from which traffic originates or towards which it gravitates. Designate the node where traffic movement originates as the "origin," and the node where flows are drawn as the "destination." In this network context, the origin and destination nodes function as centroids, while the remaining nodes are referred to as non-centroids or standard nodes. Adhering to the principles of flow conservation, it becomes evident that nodes other than centroids in the network cannot have a degree of one or zero. In this context, the degree of a node signifies the count of links interfacing with the said node.
To capitalize on the preservation of flow equilibrium at nodes and the uniformity of traffic requirements (inclusive of the total trip initiations and total trip attractions at centroids), we will revamp the traffic network by introducing a virtual centroid capable of supplanting numerous original centroids.
In a traffic network featuring just a single virtual centroid, for any given node, if we possess information about the flow across all links intersecting with that node except for one link, we can deduce the flow across the omitted link based on the principle of flow preservation within the respective node. Should this sequential process persist across nodes, we can anticipate ultimately deriving the flow of links across all connections. In actuality, there is a particular structural arrangement within the network that facilitates the realization of the aforementioned notion. This configuration is commonly known as a spanning tree. Pertinent concepts are expounded upon as follows. The following definitions are taken from [12] and [13]

Definition 5.2.
In an undirected graph, a **circuit** is a closed walk starting and ending at the same node, during the walk there is no link that has been traversed more than

once. A graph is said to be circuit-free if, and only if, it has no non-trivial circuits which consists of a single node. A graph is called a **tree** if, and only if, it is **circuit-free** and connected.
A **spanning tree** for a graph G is a subgraph of G that contains every node of G and is a tree.

Proposition 5.3.
Any tree that has more than one node has at least one node of degree 1.

Proof.
we expose the various steps:
Step 1: Pick a node k of T and let e be a link incident on v.
Step 2: While $deg(k) > 1$, repeat steps 2a, 2b, and 2c ($deg(k)$ denotes the degree of k):
Step 2a: Choose e' to be a link incident on v such that $e' \neq e$.
Step 2b: Let k' be the node at the other end of e from k.
Step 2c: Let $e = e'$ and $k = k'$.

The algorithm terminate because the set of tree nodes is finite a and is circuit-free. When it terminates a node of degree 1 will have been found. □

Proposition 5.4.
After removing one node of degree 1 and the link incident on it in any tree that has more than one node, the remaining part of the tree still is a tree.

Proof.
Let T be a particular but arbitrarily chosen tree that has more than one node. We can see that removing one node of degree 1 and the link incident on it in T will not change the connectivity of remaining nodes and the remaining part of the tree still is circuit-free. So according to the definition of tree, the remaining part of the tree still is a tree. □

Lemma 5.5.
If G is any connected graph, $\mathcal{C}$ is any nontrivial circuit in G, and one of the links of $\mathcal{C}$ is removed from G, then the graph that remains is connected.(Epp, 2004).

Proposition 5.6.
Every connected graph has a spanning tree.

Proof.
Suppose G is a connected graph. If G is circuit-free, then G is its own spanning tree and we are done. If not, then G has at least one circuit $\mathcal{C}_1$. By Lemma 5.5, the subgraph of G obtained by removing a link from $\mathcal{C}_1$ is connected. If this subgraph is circuit-free, then it is a spanning tree and we are done. If not, then it has at least one circuit $\mathcal{C}_2$, and, as above, a link can be removed from $\mathcal{C}_2$ to obtain a connected subgraph. Continuing in this way, we can remove successive links from circuits, until eventually we obtain a connected, circuit-free subgraph

T of G. Also, T contains every node of G because no nodes of G were removed in constructing it. Thus T is a spanning tree for G. □

Proposition 5.7 (Existence and multiplicity of the smallest subset of installed links for full observability)**.**
There exists a smallest subset of links on which to install sensors for inferring link flows on all remaining links. But such a subset is not unique.

Proof.
Since the traffic network with only one centroid is connected, by Proposition 4 there is a spanning tree for this network. The links outside of the spanning tree form the smallest subset of links we need according to the spanning tree algorithm given above. Then the existence of result is proved. From the process of constructing the traffic network with only one centroid, we know that there exists at least one circuit composed of a path connecting an O–D pair (or part of the path losing its one or two endpoints and the incident links) and the new added links (or links with changed endpoint) that connect the virtual centroid with the path. In the proof of Proposition 4 presented above, one link in a circuit which will be chosen to remove can be any link of the circuit. So we can expect different spanning trees appearing with different links chosen in the circuit to remove. Then the multiplicity of result is proved. □

Proposition 5.8 (Minimum number of sensors for full observability)**.**
Let m and n denote the number of links and the number of nodes in the traffic network with only one centroid, respectively. The minimum number of links on which sensors need be installed to infer flows on all remaining links is $m-n+1$ and the maximum number of links which can be left without installed sensors is $n-1$.

Proof.
For any positive integer i, any tree with i nodes has $i-1$ links (Epp, 2004). Since the spanning tree covers all nodes of the traffic network with only one centroid, it has $n-1$ links. If we add any other link into the spanning tree, there will appear at least one circuit. If we increase or decrease the link flows on all links of the circuit by any same suitable amount, the flow conservation will still hold at all nodes. That will lead to the impossible of inferring flows on all links. So the maximum number of links which can be left without installed sensors must be$n-1$. Because there are m links in the traffic network with only one centroid, the minimum number of links on which sensors need be installed to infer flows on all remaining links is $m-n+1$. □

Proposition 5.9 (Integrality of inferred link flows)**.**
If the link flows on installed links are integral, the inferred link flows on the other links must be integral as well.

Proof.
According to the spanning tree algorithm presented above, we use flow conservation condition to calculate unobserved link flows one by one. In this way,

if the observed link flow is integral, the inferred link flows must be integral as well. □

5.3 Techniques for Solving the Complete Observability Problem

With the aforementioned propositions, we can introduce a spanning tree algorithm for identifying the minimal subset of links onto which sensors should be deployed to achieve complete observability.
This will allow the inference of link flows across all remaining connections.
The outlined steps of the **spanning tree algorithm** are as follows:

Algorithm 5.1.

- ***STEP 1****: Disregarding the specific direction of the connections, create a spanning tree for the traffic network, focusing on a single central point. Mark the links within this spanning tree as "uninstalled." Proceed to place sensors on the remaining links outside the spanning tree, marking them as "installed."*
  [Given that the traffic network centered around a single point is interconnected, in accordance with Proposition 4, a spanning tree exists for this network. Various established algorithms for generating spanning trees can be applied here. The installed links collectively form the most compact set of connections suitable for sensor installation to achieve complete observability.]
- ***STEP 2****: Scanning the spanning tree or the remaining part of the tree, find the set $\mathcal{S}$ of nodes of degree 1 and repeat following steps:*
  - ***Step 2a****: If set $\mathcal{S}$ is devoid of elements, the algorithm reaches its conclusion. Otherwise, for any node k from the set $\mathcal{S}$ employ the flow conservation equation to calculate the flow values on the links within the spanning tree or the unrevealed segment of the tree's links. [Drawing from the procedure of establishing the traffic network centered around a sole centroid, we ascertain the validity of the flow conservation condition across all nodes within the network].*
  - ***Step 2b****: Exclude node k from the set $\mathcal{S}$ as well as from the spanning tree (or the remaining tree segment) simultaneously. Eliminate the link that connects node k to the other section of the tree from the tree (or the remaining part of the tree). If set $\mathcal{S}$ is not now devoid of elements, return to Step 2a. Alternatively, scanning the tree (or the remaining tree part) to discover an other node $k \in \mathcal{S}$ with a degree of 1, and then proceed to Step 2a.*
    [By virtue of Proposition 2, it is evident that the remaining tree section remains a tree. Hence, the task of identifying nodes in $\mathcal{S}$ with

a degree of 1 can persist until flow values across all links have been deduced].

5.4 Case Studies and Results

We assume that the traffic networks we're studying are connected and don't have any loops, which are links with just one end. This is usually true in real-world scenarios.
In Figure 5.2, there's a diagram of a traffic network with 4 main points, 18 connections, and 10 spots. Points 1 and 2 are where the traffic starts, and points 9 and 10 are where it ends. Because there are only ways out from points 1 and 2 and only ways into points 9 and 10, we have to create a virtual point to replace the original starting and ending points. This means we have to move some connections around, such as links (1,3), (1,4), (2,4), (2,5), (8,9) and (8,10). The new version of the network, which now has only one main point labeled as 'Virtuale', has 18 connections and 7 spots. In this updated network, we can apply some rules called flow conservation conditions at all the spots.

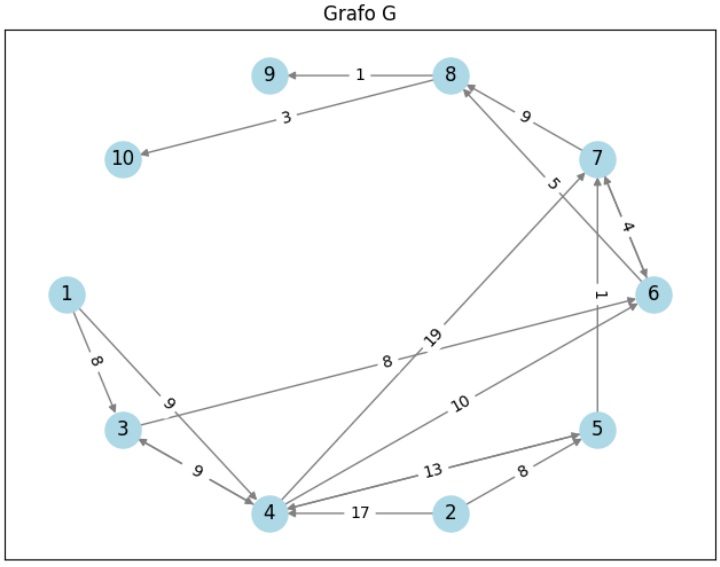

Figure 5.2: Original Graph

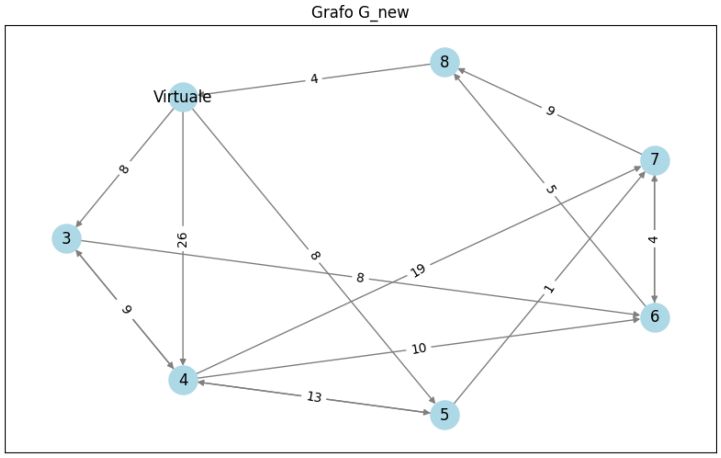

Figure 5.3: Centroid Graph

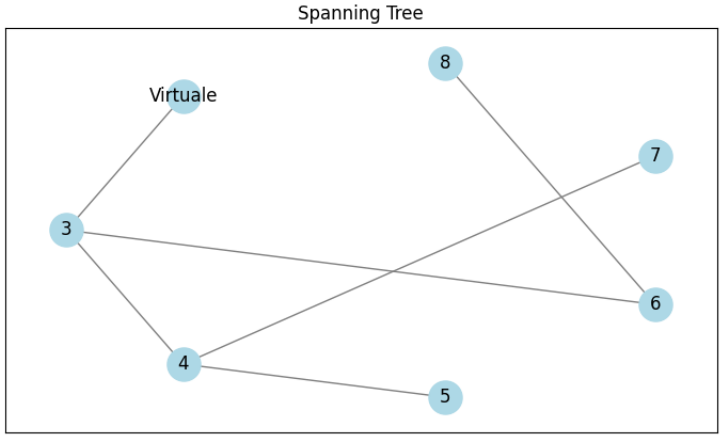

Figure 5.4: Graph with no installation fo sensors

After obtaining the spanning tree for the modified fishbone network, we can employ the same method to approximate the traffic flow across all connections.

As an example, when using the spanning tree composed of links ('Virtuale',3), (3,4), (3,6), (4,5), (4,7) and (6,8). We pinpoint nodes 'Virtuale', 5, 7 and 8 with a degree of 1 within the spanning tree. By applying principles of flow conservation at these nodes, we can infer the traffic volumes on links ('Virtuale',3), (4,5), (4,7) and (6,8). The remaining segment of the original spanning tree encompasses links (3,4), (3,6). Within this section, we encounter nodes 3 and 4, each with a degree of 1, enabling us to deduce the traffic flows on links (3,4) and (3,6) through the application of flow conservation principles.

The spanning tree method fully leverages the network's topological attributes when addressing the fundamental NSLP (Network Sensor Location Problem). This confers certain advantages that are absent in algebraic approaches.

- Firstly, the spanning tree technique brings forth the advantage of seamless applicability to real-world, extensive networks through the utilization of established algorithms for erecting spanning trees. This obviates the intricacies entailed in matrix computations.
- Secondly, this innovative approach bestows upon traffic management agencies a remarkable degree of flexibility in sensor placement. Through the introduction of fresh links into the spanning tree to forge circuits, operators possess the latitude to excise distinct links within these circuits, yielding a novel resolution that mirrors their inclinations. To illustrate, should the management agency resolve to deploy sensors on links (3,6), (4,6), (4,7), and (5,7), and given a spanning tree comprised of links (4,3), (5,4), (3,6), (4,7), (7,8), and (8,'Virtuale'), the removal of links (3,6) and (4,7) from the spanning tree, along with the addition of new links in the residual section, will yield a reconfigured spanning tree devoid of links (3,6), (4,6), (4,7), and (5,7). This process may be accomplished in two stages: initially, by introducing link (2,5) to formulate a circuit encompassing links (2,5), (5,4), (4,7), (7,8), and (8,'Vituale'), followed by the elimination of link 11 to derive a fresh spanning tree constituted by links (2,5), (4,3), (5,4), (3,6), (7,8), and (8,'Vituale'). In the subsequent phase, the incorporation of link (6,8) into the revised spanning tree, culminating in the creation of a circuit, precedes the removal of link (3,6). Ultimately, we attain a spanning tree comprising links (2,5), (4,3), (5,4), (5,7) (6,8), (7,8), and (8,'Virtuale'), aligning with the requisite sensor deployment on links (3,6), (4,6), (4,7), and (5,7).
- Thirdly, this pioneering method lends itself to the critical evaluation of a solution's rationality and the potential for adjustments by discerning whether the unobserved links have the capacity to coalesce into a spanning tree.

In summation, the spanning tree methodology introduces a novel vantage point in tackling an array of network sensor location predicaments.

Chapter 6

Density and Flow Estimation from Partial Data

In this chapter we will develop the approach techniques introduced in the previous chapter in the case of partial data. We will analyze the graphical approach using 2 sensor technologies: turning ratio sensors installed at the intersections and counting sensors installed at the arcs. The case of partial data results in having a limited number of sensors. Always giving mathematical formalization we will introduce theorems, lemmas and propositions to find in such a way as to have the properties of our model clear. We will see that our problem will become an optimization problem, where having a limited number of turning ratios, in addition to the constraint of the minimum number of sensors to be installed as we saw in the previous chapter, the position of the sensors will also be fundamental, which depends on the topology of the system.
We will show that the optimization problem can be solved, even with partial data, by running two algorithms in 2 steps. The first step tries to find the optimal solution for the set $\mathcal{R}^*$ of turning ratio sensors which have a limited number and with the information obtained from the output of the first algorithm in the second step we will find the optimal set for the set $\mathcal{S}^*$ of the counting sensors so as to be able to achieve complete observability of the problem. We will show how to obtain the flows where no sensors have been installed using the information given by our optimization problem which will give us the optimal location of both the turning ratio sensors and the flow sensors and exploiting the spanning tree of the system together with the conservation equation of the flow, we will be able to obtain the missing flows.
Finally we will apply these methods to a graph, verifying their correctness.

6.0.1 Intersection model

Let's consider an intersection labeled as k, belonging to the set $\mathcal{N}$. In this scenario, we possess information regarding the proportions of traffic making turns at this intersection. These proportions are defined as Turning Ratios (TR) denoted by $r_{i,j}$. Here, $r_{i,j}$ signifies the fraction of traffic that moves from road i (part of the set $\mathcal{E}$) to road j (also part of the set $\mathcal{E}$), specifically at the intersection k found in the set $\mathcal{N}$, under the condition that i is among the roads entering the intersection ($\mathcal{I}(k)$) and j is among the roads exiting the intersection ($\mathcal{O}(k)$).
With access to this information, we're enabled to express the traffic flow on each outgoing road from the intersection as a mathematical combination that is linear, utilizing the traffic flows on the incoming roads.

$$\varphi_j - \sum_{i \in \mathcal{I}(k)} r_{i,j}\varphi_j = 0 \quad , \quad \forall j \in \mathcal{O}(k) \tag{6.1}$$

Furthermore we have the condition $\sum_j r_{i,j} = 1$

6.0.2 Measurements and linear constraints

Now we consider that two partial data sources are available:

- Flow sensors in single roads
- TR sensors in intersections

The presence of these sensors in particular locations provides different levels of information, which are represented as constraints in the flow distribution. Let $\mathcal{R} \subseteq \mathcal{N}$ be the set of intersections equipped with TR sensors. For each of the outgoing edges of the intersections in $\mathcal{R}$, an equation following (6.1) can be written.
Define $\mathcal{A}(R)$ as the matrix that collects the resulting equations for all intersections in $\mathcal{R}$:

$$\mathcal{A}(\mathcal{R})_{i,j} = \begin{cases} 1 & if \quad e_i = j \\ -r_{j,e_i} & else \end{cases} \tag{6.2}$$

Let's work with the given information.

Suppose we have an ordered set of elements denoted as e_i, where each e_i corresponds to the i-th element in the collective set formed by taking the unions of $\mathcal{O}(k)$ for all k in the set $\mathcal{R}$.
Since an edge can be attributed to only one source node, it becomes evident that the sets $\mathcal{O}(k)$ for every node k in the set $\mathcal{N}$ do not intersect with one another. Consequently, the count of rows in the matrix $\mathcal{A}(R)$ equals $|\bigcup_{k \in \mathcal{R}} \mathcal{O}(k)|$, which in turn is the sum of the out-degrees (the number of outward edges) for each node k in the set $\mathcal{R}$. The out-degree of node k is represented as $deg^{out}(k)$.

Now, let's consider a set $\mathcal{U}$, which encompasses intersections not covered by measurements. Parallel to the earlier discussion involving set $\mathcal{R}$, each intersection in set $\mathcal{U}$ is associated with an equation similar to the one mentioned as (6.1). To organize these equations, we define $\mathcal{B}(U)$ as the matrix that aggregates these equations, where each equation corresponds to a node k in set $\mathcal{U}$.

$$\mathcal{B}(\mathcal{U})_{i,j} = \begin{cases} 1 & if \quad j \in \mathcal{O}(k_i) \\ -1 & if \quad j \in \mathcal{I}(k_i) \\ 0 & else \end{cases} \tag{6.3}$$

Consider k_i the i-th element in the ordered set $\mathcal{U}$. Verify that the dimensions of $\mathcal{B}(\mathcal{U})$ are $(n_{\mathcal{N}} - n_{\mathcal{R}})$ by $n_{\mathcal{E}}$, where $n_{\mathcal{R}} = |\mathcal{R}|$.
For the representation of flow sensors, let's denote $\mathcal{S}$ as a subset of $\mathcal{E}$, comprising roads equipped with flow sensors. The size of this set is denoted as $|\mathcal{S}| = n_s$. We define a matrix $C(\mathcal{S}) \in 0, 1^{n_{\mathcal{S}} \times n_{\mathcal{E}}}$, where $C(\mathcal{S})i, j = 1$ if the i-th sensor is positioned on the j-th road. It's important to note that we can express $C(\mathcal{S})$ as $[ue1, ..., u_{e_{n_{\mathcal{S}}}}]^T$, where e_i stands for the i-th element in the set $\mathcal{S}$, and u_i represents the standard basis vector corresponding to the i-th coordinate. We assume that the elements within $\mathcal{S}$ are distinct, thus leading to $rank\ C(\mathcal{S}) = n_{\mathcal{S}}$. With the help of these matrices, we can formulate the linear constraints as follows:

$$\begin{bmatrix} L(\mathcal{R}) \\ C(\mathcal{S}) \end{bmatrix} \varphi = \begin{bmatrix} 0 \\ \varphi_m \end{bmatrix} \tag{6.4}$$

6.0.3 Flow reconstruction

To ascertain the flow values across all roads in the network, we employ the inversion of the system described in Equation 6.4. Previous studies have demonstrated that the outcomes produced by Algorithms 6.3 and 6.4 yield in a square matrix $\begin{bmatrix} L(\mathcal{R}^*) \\ C(\mathcal{S}^*) \end{bmatrix}$ with complete rank. Without loss of generality, assume an indexing of roads such that $C(\mathcal{S}^*) = [0, \mathbb{I}]$. The flow vector φ can be split into two components, namely the measured flow φ_m and unmeasured flows φ_u, such that $\varphi = \begin{bmatrix} \varphi_u \\ \varphi_m \end{bmatrix}$. Additionally, consider a partition $L(\mathcal{R}^*) = [L_u, L_m]$, where L_u is a square matrix. Because the matrix $\begin{bmatrix} L_u & L_m \\ 0 & \mathbb{I} \end{bmatrix}$ is full rank, it easy to show that L_u is invertible. After strainghtforward calculations, it can be seen that

$$\varphi_u = -L_u^{-1} L_m \varphi_m \tag{6.5}$$

Given the known values of φ_m, the complete flow vector can be reconstructed.

These findings align with similar studies in the literature, which also entail matrix inversion. It's worth noting, however, that He's (2013) devised algorithm demonstrated the capability to reconstruct road flows with quasi-linear complexity, albeit restricted to the case where $\mathcal{R} = \emptyset$. Whether this approach can be expanded to the broader scenario of $\mathcal{R} \subseteq \mathcal{N}$ for complexity reduction remains uncertain. These potential extensions will be subject to scrutiny in forthcoming research endeavors.

6.1 Challenges in Estimating Traffic Density and Flow from Partial Data

In real-world scenarios, we often encounter traffic networks with certain links already equipped with sensors. This situation gives rise to two distinct challenges:

- The first challenge involves determining the most compact set of links suitable for sensor deployment. This set needs to be combined with the links where sensors were previously installed, enabling the inference of flow information across the remaining links.
- The second challenge is to identify a subset of links for which flow information can be inferred using measurements from sensors installed on a portion of the links.

Assuming that we have already solved the second challenge, resolving the first challenge becomes relatively straightforward through the employment of the spanning tree approach detailed in Section 5. Initially, we eliminate all links that are equipped with sensors and those for which flow inference is possible (matching the solution from the second challenge). Additionally, we remove nodes where the flow of incident links can be either measured by sensors or inferred from measured link flows. Within the remaining network, we apply the spanning tree algorithm outlined in Section 5. The minimal set of links identified by the spanning tree algorithm, designed for sensor installation, represents the smallest link subset needed to address the first challenge. The logical progression parallels the reasoning presented in Section 5.
For the resolution of the second challenge, it becomes necessary to establish the validity of the subsequent proposition. The following propositions are take from [13]

Proposition 6.1 (Ambiguity of link flows in a circuit))**.**
In a graph, considering any connected subgraph composed of nodes whose degrees are all equal to or greater than 2, the link flows on the links of the connected subgraph cannot be determined unambiguously.

Proof.
We can partition the connected subgraphs into two types.
Type 1 denotes the subgraphs composed of nodes whose degrees are all equal

to 2.
Type 2 denotes the subgraphs which are composed of nodes whose degrees are all equal to or greater than 2 and need satisfy the condition that at least one node degree is greater than 2. Suppose $\mathcal{C}$ is a particular but arbitrarily chosen subgraph of type 1 with at least 2 nodes, and consider the following algorithm:

- Step 1: Pick a link e of $\mathcal{C}$ and let k and k'' be two endpoints of e.

- Step 2: While v is not identical with k'' , repeat steps 2a, 2b, and 2c:
  - Step 2a: Choose e' to be a link incident on k such that $e' \neq e$.
  - Step 2b: Let k' be the node at the other end of e' from k.
  - Step 2c: Let $e = e'$ and $k = k'$.

The algorithm as explained earlier is certain to eventually halt because the collection of nodes forming the subgraph $\mathcal{C}$ is both finite and interconnected. Upon termination, a circuit encompassing all the nodes within $\mathcal{C}$ will have been discovered. Remarkably, this circuit coincides with $\mathcal{C}$ itself. If the flow values on all links within the $\mathcal{C}$ circuit are simultaneously increased or decreased by a proportional and appropriate amount, the principle of flow conservation will continue to be upheld across all nodes. Consequently, the flow values of the links within any subgraph of type 1 cannot be definitively ascertained.
Regarding a subgraph of type 2, it becomes evident that the quantity of links surpasses the number of nodes. When no other preliminary assumptions or knowledge apart from the flow conservation stipulations at all nodes are considered, the quantity of unknown variables becomes greater than the quantity of equations denoting flow conservation conditions. In this context, the unknown variables' count corresponds to the number of links within the subgraph of type 2, while the equation count corresponds to the number of nodes within the same subgraph of type 2.
Consequently, the flow values on the links present in any subgraph of type 2 cannot be conclusively determined either. □

Leveraging Proposition 6.1, we can employ the following procedure to pinpoint the set of links for which flow values can be deduced from sensors installed on a subset of links beforehand. Initially, we designate links already equipped with sensors as "observed," while the remaining links are labeled as "unobserved." Subsequently, we inspect the collection of nodes to determine whether any node exists for which only a single "unobserved" link is incident. By applying the flow conservation condition at the identified node, we deduce the flow value for the "unobserved" link and then relabel it as "observed." This process of scanning through the node set and inferring link flows is repeated until it becomes unfeasible to locate a node where only a single "unobserved" link is incident.

Theorem 6.2 (The unique topological feature)**.** *[13].*
If there is no extra information about actual used paths and flow productions and attractions at centroids in a traffic network, the unequipped links for full observability will form a spanning tree for the corresponding reformulated network where the flow conservation conditions hold at every node.

Proof.
As outlined in Proposition 5.8, in a redefined network containing n nodes, the count of links without equipment must amount to $n-1$. In cases where the collection of unequipped links fails to constitute a spanning tree within the reformulated network, it implies the presence of a circuit within the subgraph formed by these unequipped links and their associated endpoints. This outcome can be derived from the fact that any connected graph G with n nodes and $n-1$ links is inherently a tree, as demonstrated by Epp (2004). As per Proposition 6.1, the unambiguous determination of link flows within a circuit is unattainable. Consequently, even if we possess knowledge regarding link flows across all remaining links encompassing equipped and unequipped links situated outside the circuit, the exact flow values for links within the circuit remain indeterminable. Thus, the set of unequipped links necessary for complete observability must align with the link set of a spanning tree within the corresponding redefined network, where the constraints of flow conservation are upheld at every node. □

6.2 Partial Data Estimation Techniques and Algorithms

The subsequent notions are requisite for our algorithm to determine the most suitable positions for a specified count of sensors.

Definition 6.3. [13]

- A **weighted graph** is a graph for which each link has an associated real number **weight**. The sum of the weights of all the links is the **Total weight** of the graph.
- A **minimum spanning tree** for a weighted graph is a spanning tree that has the least possible total weight compared to all other spanning tree for the graph.

Our objective here is to tackle the challenge of pinpointing the optimal positions for a given quantity of sensors in order to deduce the maximum possible number of link flows across the remaining connections. Let m and n represent the counts of links and nodes, respectively, within the traffic network G, which revolves around a single central point. Our task involves installing t sensors strategically to enable the inference of link flows across as many of the remaining links in G as feasible.

We now understand that two primary factors play a pivotal role in determining the choice of sensor locations. Firstly, the quantity of sensors holds significance, and secondly, the topological characteristics of the chosen sensor deployment sites also come into play. Considering the interplay of these two factors, we present an algorithm devised to address this issue. The ensuing steps encapsulate the associated approach:

Algorithm 6.1.

- ***STEP 1****: Build a spanning tree T of G by using the spanning tree algorithm given in Section 2. Let H and W denote the set of links in T and the set of the remaining links outside of T, respectively. Let C denote the set of links being part of any circuit of subgraph T. If* $t \geq m - n + 1$*, go to Step 2; or else, go to Step 3.*
- ***STEP 2****: Choose any* $t + n - m - 1$ *links in T and then remove them from T. Go to Step 5.*
- ***STEP 3****: Let* T_{emp} *be a subgraph of G identical with T. Let* H_{temp} *and* V_{emp} *denote the set of links of Temp and the set of nodes of Temp, respectively. U is an empty link set. Assign 0 to the weights of all links of G. Let* $M_{emp} = 2|H_{emp}|$ *and S: = U (where we have used* $|S|$ *to denote the cardinality of the set S).*

  While *(*$H_{emp} \neq \emptyset$*).*

  - ***Step 3a****: Only considering the links in Temp to calculate the degrees of nodes, add nodes of Temp whose degrees are equal to 1 into S.*
  - ***Step 3b****: For every node v e S, there is a link e connecting v to the other part of* T_{emp} *and set the weight of e to* M_{emp}*.*
  - ***Step 3c****: Let* M_{emp}*: =* $M_{emp} - 2|S|$*. Delete all of nodes of S and the links connecting nodes of S to the other part of Temp from* H_{emp} *and* V_{emp}*, respectively. Renew graph Temp and let* $S := \emptyset$

  and ***While***

- ***STEP 4****: for* $i := 1$ *to* $m - n - t + 1$

  - ***Step 4a****: Choose a link e from W such that comparing with choosing another link from W, having e to add into T will make the total weight of renewed C the minimum among all the choices.*
  - ***Step 4b****: Remove e from W and add e into T. Then renew C.*

  next i

- ***STEP 5*** *:Except the links in the new subgraph T, the other links make up the subset of links that determines the optimal locations for the given t sensors. Now we are done.*

It's important to highlight that the solution derived from the aforementioned algorithm can yield a local optimum linked to the initial construction of the spanning tree. Since there are multiple ways to generate distinct spanning trees for a connected graph, different starting points could lead to varied outcomes. These outcomes signify optimal sensor installation locations relative to the employed spanning trees. However, due to the substantial number of potential spanning trees, efficiently listing all of them remains an unfeasible task. For example, a complete graph having l nodes could generate as many as l^{l-2} distinct spanning trees. A practical alternative involves running the algorithm several times, each with a different initial spanning tree, and then comparing the results to determine the most favorable choice for the final sensor installation locations.

In line with Proposition 5.8, if t is equal to or greater than $m - n + 1$, it becomes possible to infer link flows across all connections. However, if this is not the case, there will exist links for which flow inference is restricted due to the limited number of sensors. Derived from the proof of Proposition 6.1, we understand that links forming part of circuits cannot have their flows inferred. While utilizing the spanning tree algorithm detailed in Section 2, we establish that there exists a sequence of links for calculating flow values across the uninstalled links. Should a link, requiring its flow calculation prior to others, be implicated in a circuit, it obstructs the subsequent flow calculations for the succeeding links. This is the underlying reason for our endeavors to minimize the weight of the final set $\mathcal{C}$ and to conscientiously shape circuits – unless it's entirely unavoidable – among uninstalled links, where the flow calculations will take place further down the sequence.

In the execution of Step 4a, a meticulous approach involves testing every link within W and subsequently comparing the resulting weight of $\mathcal{C}$ following different modifications. Post removal of a link e from T but outside of $\mathcal{C}$, an assessment is carried out to verify if the graph composed of links within T (but external to $\mathcal{C}$) and nodes connected to links that aren't all part of $\mathcal{C}$ remains interconnected. This helps in determining whether the excluded link e has become integrated into a circuit. In practical scenarios, given that most roads in a traffic network support two-way traffic, a more pragmatic approach can be adopted to establish circuits, ultimately simplifying the operations involved in Step 4a.

Certain prerequisites often necessitate fulfillment when aiming to deploy sensors on the smallest subset of links to deduce flow information across all remaining links. Two of these prerequisites involve minimizing the associated cost of sensor installation and promptly equipping sensors on designated links. We can address both these prerequisites uniformly by employing a weighted graph.

The algorithm presented below aims to resolve our challenge of generating spanning trees that adhere to these prior requirements. As previously noted, enumerating all possible spanning trees of a graph, and subsequently comparing them to extract all minimum spanning trees, poses a substantial challenge. Nevertheless, our objective is to identify at least one minimum spanning tree, which fortunately proves sufficient.

Fortuitously, we have access to efficient algorithms for constructing minimum spanning trees. Among these algorithms, Kruskal's and Prim's methods can be harnessed to achieve worst-case complexities of $O(m \log_2 m)$ **and** $O(n^2)$, respectively, for graphs containing n nodes and m links. The Prim's algorithm for constructing minimum spanning trees (Epp, 2004) is integrated into our algorithm, as detailed in the subsequent steps:

Algorithm 6.2 (DFS)**.**

- ***STEP 1****: Create the corresponding weighted graph representing the prior requirements. If the primary requirement is to minimize installation costs, we can assign the negative of the installation-related cost as the weight of the respective link. If the primary requirement is to equip sensors on specific links as early as possible, we can assign a large real number as the weight for these specified links and a small real number for the other links. This yields a weighted graph G with n nodes.*
- ***STEP 2****: Build a minimum spanning tree using Prim's Algorithm. Select a node v from G and let T be the graph with only one node, v, and no links. Let V be the set of all nodes in G except v. for i from 1 to n - 1 (where n is the number of nodes in G)*
  - ***Step 2a****: Find a link e of G such that (1) e connects T to one of the nodes in V, and (2) e has the least weight of all links connecting T to a node in V. Let w be the endpoint of e that is in V.*
  - ***Step 2b****: Add e and w to the sets of links and nodes in T, and remove w from V, end for i, The resulting graph T is a minimum spanning tree for G.*
- ***STEP 3****: Apart from the links in the new subgraph T, the remaining links constitute the subset of links on which to install sensors for full observability. Infer link flows on the links in T using step 2 of the spanning tree algorithm outlined in algorithm (6.1).*

6.2.1 Number and location of flow and TR sensors

Our goal is to determine the smallest possible quantity of flow and Turning Ratio (TR) sensors, along with their specific placements. Initially, we'll examine a scenario where the count of TR sensors, denoted as $|\mathcal{R}| = n_{\mathcal{R}}$, is fixed, while our focus is on minimizing the count of flow sensors $|\mathcal{S}|$.

The core challenge can be defined as identifying the most optimal sets, denoted as $\mathcal{R}^*$ and $\mathcal{S}^*$, which effectively address the given problem.

$$\begin{aligned} \mathcal{R}^*,\ \mathcal{S}^* \quad &= \quad \underset{\mathcal{R},\mathcal{S}}{\operatorname{argmin}} \quad |\mathcal{S}| \\ \textit{subject to} \quad & \quad rank\left(\begin{bmatrix} L(\mathcal{R}) \\ C(\mathcal{S}) \end{bmatrix}\right) = n_{\mathcal{E}} \\ & \quad |\mathcal{R}| = n_{\mathcal{R}} \end{aligned} \tag{6.6}$$

The following definitions are taken from [14]

Proposition 6.4.
For any $n_{\mathcal{R}} \in \{0, 1, ..., n_{\mathcal{N}}\}$, problem (6.5) has at least one feasible solution.

Proof.
Consider $\mathcal{S} = \mathcal{E}$ for which $C(\mathcal{E}) = \mathbb{I}_{n_{\mathcal{E}}}$. It is stringhtforward that $\begin{bmatrix} L(\mathcal{R}) \\ C(\mathcal{S}) \end{bmatrix} = n_{\mathcal{E}}$ for matrix $L(\mathcal{R})$. This selection of sets $\mathcal{R}$, $\mathcal{S}$ satisfies the constraints, hence the feasible set is non empty. □

This issue exhibits a combinatorial nature. Employing a straightforward approach, where the cost is computed for every feasible combination and the best one is selected, would demand a significant number of operations, specifically around $O(n_{\mathcal{N}}/n_{\mathcal{R}}2^{n_{\mathcal{E}}})$, making it a computationally expensive solution method. Consequently, we examine the possibility of breaking down this problem into two separate problems, each with a reduced level of intricacy.
The initial problem entails determining the optimal collection of TR sensors independently. This endeavor involves finding the best set of sensors that minimizes certain criteria as

$$\begin{aligned} \mathcal{R}^* \quad &= \quad \underset{\mathcal{R}}{\operatorname{argmax}} \quad L(\mathcal{R}) \\ \textit{subject to} \quad & \quad |\mathcal{R}| = n_{\mathcal{R}} \end{aligned} \tag{6.7}$$

after we found $\mathcal{R}^*$, the second problem consists in finding any set $\mathcal{S}^*$ that have to satisfy the following conditions:

$$\begin{gathered} \text{find any} \quad \mathcal{S}^* \subseteq \mathcal{E} \\ \text{such that } rankC(\mathcal{S}^*) = n_{\mathcal{E}} - rankL(\mathcal{R}^*) \end{gathered}$$

$$rank\left(\begin{bmatrix} L(\mathcal{R}) \\ C(\mathcal{S}) \end{bmatrix}\right) = rankC(\mathcal{S}^*) + rankL(\mathcal{R}^*) \tag{6.8}$$

Lemma 6.5.
For any given $\mathcal{R}^* \subseteq \mathcal{N}$, problem(6.7) has at last one solution.

Proof. from definition of $L(\mathcal{R}^*)$, it can be seen that

$$0 < rankL(\mathcal{R}) \geq \sum_{k \in \mathcal{N}} deg^{out}(k) < n_\epsilon \tag{6.9}$$

Consider $\mathcal{S} = \mathcal{E}$, that $C(\mathcal{E}) = \mathbb{I}$, and $rankC(\mathcal{E}) = n_\mathcal{E}$.
It follows that $rankC(\mathcal{E}) + rankL(\mathcal{R}^*) > n_\mathcal{E}$.
However, $rank\left(\begin{bmatrix} L(\mathcal{R}) \\ C(\mathcal{S}) \end{bmatrix}\right) = n_\mathcal{E}$, and some rows of $rankC(\mathcal{E})$ are linear combinations of the other rows.

We can find a solution with the following iterative process:

1. Initialize $\mathcal{S} = \mathcal{E}$.
2. While $rankL(\mathcal{R}^*) + rankC(\mathcal{E}) > n_\mathcal{E}$.
   (a) Find $s \in \mathcal{S}$ such that corresponding row of $C(\mathcal{S})$ is a linear combination of the other rows of the matrix.
   (b) Assign $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s\}$.
3. Define $\mathcal{S}^* \leftarrow \mathcal{S}$.

As the process only removes redundant rows, the rank of augmented matrix is still equal to $n_\mathcal{E}$. Additionally, the algorithm only ends when $rankL(\mathcal{R}^*) + rankC(\mathcal{E}) = n_\mathcal{E}$. Therefore, for given $\mathcal{R}^*$, it is always possible to find$\mathcal{S}^*$ with satisfies the constraints. □

Lemma 6.6.
Let $\mathcal{R}^*$, $\mathcal{S}^*$ be an optimal solution to problem (6.6). Then, the rows of matrices $L(\mathcal{R}^*)$ and $C(\mathcal{S}^*)$ are independent.

Proof.
We proceeded by contradiction. Assume that the rows of $C(\mathcal{S}^*)$ are not linearly independent to the rows of $L(\mathcal{R}^*)$. There exists at least one row of $C(\mathcal{S}^*)$, $\mathbf{u}_s^t$ such that $s \in \mathcal{S}^*$, which is a linear combination of the rows of $\begin{bmatrix} L(\mathcal{R}^*) \\ C(\mathcal{S}*) \end{bmatrix}$.
Define $\mathcal{S}' = \mathcal{S}^* \setminus \{s\}$. Matrix $C(\mathcal{S}')$ is the same as $C(\mathcal{S}^*)$ but with row $\mathbf{u}_s^t$ removed. It is evident that $\left(\begin{bmatrix} L(\mathcal{R}^*) \\ C(\mathcal{S}') \end{bmatrix}\right) = n_\mathcal{E}$ and the constraints are still satisfied.
Additionally, $|\mathcal{S}'| = n_\mathcal{S}^* < |\mathcal{S}^*|$, making $\mathcal{S}'$ non optimal, which is a contradiction. □

Theorem 6.7.
Let $\mathcal{R}^*$,$\mathcal{S}^*$ is a solution of problem (6.5) if and only if $\mathcal{R}^*$ is a solution to problem (6.6) and $\mathcal{S}^*$ is a solution to problem (6.7).

Proof.
Recall,by construction, $rankC(\mathcal{S}) = |\mathcal{S}|$ for any $\mathcal{S} \subseteq \mathcal{C}$.
First, we prove implication. Assume that $\mathcal{R}^*$, $\mathcal{S}^*$ is a solution to problem 6.5. From lemma 6.6 and the constraints of problem 6.5, it is evident that $\mathcal{S}^*$ is a solution to problem 6.7. By way of contradiction, assume there exists $\mathcal{R}'$, $|\mathcal{R}'| = n_{\mathcal{R}}$ such that $rankL(\mathcal{R}') > rankL(\mathcal{R}^*)$. Because Lemma 6.5, we can find $\mathcal{S}'$ satisfying problem 6.7. It easy to check that the pair $\mathcal{R}'$, $\mathcal{S}'$ lie in the feasible region of problem 6.5, and that $|\mathcal{S}'| < |\mathcal{S}^*|$. This implies that $\mathcal{S}^*$ is not an optimal solution. This is a contradiction, then this $\mathcal{R}'$ cannot exist and therefore, $rankL(\mathcal{R}^*) \geq rankL(\mathcal{R})$ for any $\mathcal{R}$, $|\mathcal{R}| = n_{\mathcal{R}}$, therefore $\mathcal{R}^*$ is a solution to problem (6.6).
Now we proceeded to necessity. Assume that $\mathcal{R}^*$ is a solution to problem (6.6) and $\mathcal{S}^*$ is a solution of problem (6.7), then $|\mathcal{S}^*| = n_{\mathcal{E}} - rankL(\mathcal{R}^*)$. Consider that another pair $\mathcal{R}'$, $\mathcal{S}'$ is an optimal solution to problm (6.5), therefore $|\mathcal{S}'| \geq |\mathcal{S}^*|$. By construction, it must be that $rankL(\mathcal{R}^*) > rankL(\mathcal{R}')$, and using Lemma (6.6) $rankL(\mathcal{R}^+*) \geq n_{\mathcal{E}} - |\mathcal{S}'|$. This implies that$|\mathcal{S}^*| \leq |\mathcal{S}'|$, and thus, $|\mathcal{S}'| = |\mathcal{S}^*|$ and pair $\mathcal{R}^*$, $\mathcal{S}^*$, is also an optimal solution to problem (6.5) □

6.2.2 Optimal location of turning ratio sensors

The previous propositions allows to write the rank of $L(\mathcal{R})$ for any $\mathcal{R}$ as a function of the number of intersections in the network and their out-degrees, i.e.

$$rankL(\mathcal{R}) = n_{\mathcal{N}} - n_{\mathcal{R}} + \sum_{k \in \mathcal{R}} deg^{out}(k) \tag{6.10}$$

Using (6.10), problem (6.6) can be rewritten as

$$\begin{aligned} \mathcal{R}^* = \underset{\mathcal{R}}{\text{argmax}} \ & \sum_{k \in \mathcal{R}} deg^{out}(k) \\ subject \ \ to \quad & |\mathcal{R}| = n_{\mathcal{R}} \end{aligned} \tag{6.11}$$

To solve this problem, the following algorithm is proposed:

Algorithm 6.3 (Location of TR sensors)**.**

Inputs: Directed graph $\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\}$ and number of TR sensors $n_{\mathcal{R}}$.

Output: Set of intersection $\mathcal{R}^*$.

1. *Calculate the vector of out-degrees: $\boldsymbol{d}(k) \leftarrow deg^{out}(k), \ \forall k \in \mathcal{N}$.*

2. *Sort $\boldsymbol{d}$ from highest to lowest and return the sorting vector λ, i.e. $\lambda(1)$ is the index of the highest element of $\boldsymbol{d}$, $\lambda(2)$ is the index of the second highest, and so on.*

3. *Construct $\mathcal{R}^* \leftarrow \{\lambda(1), \lambda(2), ..., \lambda(n_{\mathcal{R}})\}$.*

Remark: Several solutions are possible depending on the multiplicity of the out-degrees of the nodes of the network, however, all these solutions are optimal. The following propositions show that problem (6.7) can be converted into problem (6.10).

Lemma 6.8.
For any feasible traffic network $\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\}$ *and any set of intersections* $\mathcal{R} \subseteq \mathcal{N}$, $A(\mathcal{R})$ *is full row rank.*

Proof.
Without loss of generality assume that the ordering of the elements of $\mathcal{E}$ is such that the smaller indexes correspond to $\mathcal{E}_{in}$, followed by $\mathcal{E}_{net}$ and ending with $\mathcal{E}_{out}$. Denote $|\mathcal{E}_{in}| = n_{in}$ and $|\mathcal{E}_{out}| = n_{out}$.
For now, let $\mathcal{R} = \mathcal{N}$. Using the proposed indexing, we can write:

$$\mathcal{A}(\mathcal{N})_{i,j} = \begin{cases} 1 & if \quad i = j + n_{in} \\ -r_{i=j+n_{in}} & else \end{cases} \tag{6.12}$$

for $i = 1, 2, ..., n_{\mathcal{E}} - n_{in}$ and $j = 1, 2, ..., n_{\mathcal{E}}$. We can split $\mathcal{A}(\mathcal{N}) = [X\ Y]$ where Y is a square matrix of size $n_{\mathcal{E}} - n_{in}$, and can be written as $Y = -\mathbb{I} - R^T$ with $R_{i,j} = r_{i+n_{in},j+n_{in}}$ for $i, j \in \{1, 2, ..., n_{\mathcal{E}} - n_{in}\}$.

Matrix R has the following properties:

1. The diagonal entries of R are zero as $r_{i,j} = 0$.
2. All roads in $\mathcal{E}_{out}$ have no downstream neighbor, then $r_{i,j} = 0\ \forall i \in \mathcal{E}_{out}$. Thus, the last n_{out} rows of R are zero. Nevertheless by flow conservation, all the other rows of R sum to 1.
3. For every row i of R there is a sequence of nonzero elements of R of the form $R_{i,i_1}, R_{i_1,j_2}, ...R_{iq,j}$ such that $j \in \mathcal{E}_{out}$.

Properties 1 and 3 imply that Y^T is weakly diagonally dominant matrix i.e. $Y_{i,i} \geq |\sum_{j \neq i} Y_{i,j}|$ where the strict inequality is true only for the last n_{out} rows. Nevertheless, property 3 states that for every row i, there is sequence of nonzero elements that connect row i to one of the last n_{out} rows. Because of this, Y^T is a special type of matrix called "weakly chained diagonally dominant", which is known to be non-singular [28]. Thus all rows of $\mathcal{A}(\mathcal{N})$ form linearly independent set.
For any arbitrary $\mathcal{R} \subseteq \mathcal{N}$, $\mathcal{A}(\mathcal{R})$ is just a reduced version of $\mathcal{A}(\mathcal{N})$ with some of its rows removed. As the rows of $\mathcal{A}(\mathcal{N})$ are linearly independent, it is straightforward that $\mathcal{A}(\mathcal{R})$ is full row rank as well. □

Lemma 6.9.
For any feasible set traffic network $\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\}$ *and any set of intersections* $\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\} \subseteq \mathcal{N}, B(\mathcal{U})$ *is full row rank.*

Proof.
Initially assume that $\mathcal{U} = \mathcal{N}$. By construction, each column of $B(\mathcal{N}) \in \{-1, 0, 1\}^{n_\mathcal{N} \times n_\mathcal{E}}$ has only 2 non-zero entries which sum to 0. The only exceptions are columns corresponding to indexes $\mathcal{E}_{in}$ which have only one non-zero element equal to -1, and columns corresponding to indexes $\mathcal{E}_{out}$ with one non-zero entry equal to 1. Note that $B(\mathcal{N})$ is the incidence matrix of $\{\mathcal{N}, \mathcal{E}\}$.We proceed by contradiction. Assume that $B(\mathcal{N})$ is not full rank. This implies that there exist two sets $\mathcal{U}_1 \subset \mathcal{N}$ and $\mathcal{U}_2 \subset \mathcal{N}$ such that $\sum_{k\in\mathcal{U}_1} B(\mathcal{N})_{k,j} = -\sum_{k\in\mathcal{U}_2} B(\mathcal{N})_{k,j}$, $\forall j$, which is equivalent to $\bigcup_{k\in\mathcal{U}_1} \mathcal{O}(k) = \bigcup_{k\in\mathcal{U}_2} \mathcal{I}(k)$ and $\bigcup_{k\in\mathcal{U}_1} \mathcal{I}(k) = \bigcup_{k\in\mathcal{U}_2} \mathcal{O}(k)$.

This means that for every node in $\mathcal{U}_1$, any downstream neighbor must be a member of $\mathcal{U}_1$ or $\mathcal{U}_2$. Subsequently, any downstream chain of nodes must be contained in one of these two sets. As every edge is part of path ending in an element of $\mathcal{E}_{out}$, the previous statements imply that there exists a node $k \in \mathcal{N}$ such that $\mathcal{I}(k) \cap \mathcal{E}_{out} = \emptyset$, which is a contradiction. Hence, the rows of $B(\mathcal{N})$ form a linearly independent set.
Following a similar discussion as at the end of the proof of Lemma 6.8, we note that for $\mathcal{U} \subseteq \mathcal{N}$,the corresponding $B(\mathcal{U})$ implies only the removal of rows from $B(\mathcal{N})$, so the remaining rows are still linearly independent, and thus $B(\mathcal{U})$ is full row rank.

□

Theorem 6.10.
For any feasible network $\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\}$, and sets $\mathcal{R} \subseteq \mathcal{N}$ and $\mathcal{U} = \mathcal{N} \setminus \mathcal{R}$, it holds that $rank L(\mathcal{R}) = rank A(\mathcal{R}) + B(\mathcal{U})$.

Proof.
For any arbitrary $k \in \mathcal{N}$, we can check that $\mathbb{I}^T A(\{k\}) = B(\{k\})$. This is due to the fact that $\sum_i r_{i,j} = 1\ \forall j$ and that $r_{i,j} = 0$ if j is not downstream neighbor of i. Conversely, $\mathbb{I}^t A(\{k\})$ and $B(\{p\})$ share no non-zero entries for $k \neq p$, because $\mathcal{O}(k) \cap \mathcal{O}(p) = \emptyset$ and $\mathcal{I}(k) \cap \mathcal{I}(p) = \emptyset$ if $k \neq p$. Thus, for arbitrary set $\mathcal{K} \subseteq \mathcal{N}$, each row of $B(\mathcal{K})$ can be obtained by combination of unique rows of $\mathcal{A}(\mathcal{K})$. As $\mathcal{R} \cap \mathcal{U} = \emptyset$, no row of $B(\mathcal{U})$ can be written as a combination of rows of $\mathcal{A}(\mathcal{R})$, so both matrices have linearly independent rows. □

6.2.3 Optimal location of flow sensors

In this section we discuss an efficient solution to problem (2.15). The following Corollary provides a way to calculate the optimal number of flow sensors n_s^*.

Corollary 6.11.
For any feasible traffic network$\{\mathcal{C} \cup \mathcal{N}, \mathcal{E}\}$ and any set of intersection $\mathcal{R}^* \subseteq \mathcal{N}$ with cardinality $n_\mathcal{R}$, the minimum number of flow sensors required to infer all flows in the network is

$$n_s^* = n_\mathcal{E} - n_\mathcal{N} + n_\mathcal{R} - \sum_{k\in\mathcal{R}} deg^{out}(k). \tag{6.13}$$

Proof.
Let $\mathcal{S}^*$ be optimal solution to 6.7, then $n_{\mathcal{S}^*} = |\mathcal{S}^*| = rankC(\mathcal{S}^*) = n_{\mathcal{E}} - rankL(\mathcal{L}(\mathcal{R}^*))$. Thus, the corollary follows directly from Theorem 6.7. □

To solve the sensor location problem we require a method that locates the number of sensors indicated in Corollary 6.11 in such a way that the rows of $\mathcal{C}(\mathcal{S}^*)$ and $\mathcal{L}(\mathcal{R}^*)$ are linearly independent. Because of the efficiency and simplicity of graph-based approaches such as the one of [11], we decided to expand these techniques to include partial information of TRs. We propose Algorithm 6.3 which makes use of the topological structure of the traffic network and the information given by the set $\mathcal{L}(\mathcal{R}^*)$. The algorithm creates spanning trees of the input graph by using the well known Depth First Search (DFS), which is a well known graph traversal algorithm.

Algorithm 6.4.

Inputs: Directed graph $\{\mathcal{C} \bigcup \mathcal{N}, \mathcal{E}\}$ *and set* $\mathcal{R}^* \subseteq \mathcal{N}$.

Outputs: Set of measured edges $\mathcal{S}^*$.

1. *Replace all nodes in* $\mathcal{C}$ *with a single node* v_0 *such that* $\mathcal{E}_{in} = \mathcal{O}(v_0)$ *and* $\mathcal{E}_{out} = \mathcal{I}(v_0)$.
2. *Initialize* $\mathcal{E}_r \leftarrow \emptyset$ *and* $\mathcal{E}' \leftarrow \mathcal{E}$. *For each* $k \in \mathcal{R}^*$:
   (a) *Find* $\{e_1, e_2, ..., e_q\} = \mathcal{O}(k)$.
   (b) *Assign* $\mathcal{E}_R \leftarrow \{e_2, ..., e_q\}$.
   (c) *Assign* $\mathcal{E}' \leftarrow \mathcal{E}' \setminus \mathcal{E}_R$
3. *Ignoring edge direction, perform a DFS over* $\{\mathcal{N} \bigcup \{v_0\}, \mathcal{E}'\}$ *starting at* v_0. *Denote* $\mathcal{N}_T$ *and* $\mathcal{E}_T$ *as the vsited nodes and edges, respectively.*
4. *While* $\mathcal{R}^* \setminus \mathcal{N}_T \neq \emptyset$:
   (a) *For each* $k \in \mathcal{R}^* \setminus \mathcal{N}_T$:
      i. *Construct* $\mathcal{M}_k = \{m \in \mathcal{N} | \exists j \in \mathcal{O}(k) \bigcap \mathcal{I}(m)) \bigcap \mathcal{E}_R\}$.
      ii. *if* $\exists m \in \mathcal{M}_k \bigcap \mathcal{N}_T$:
         A. *Find* $j \in \mathcal{O}(k) \bigcap \mathcal{I}(m)$.
         B. *Return* k *and* j. *EXIT LOOP.*
   (b) *Find* $e_1 \in \mathcal{O}(k) \bigcap \mathcal{E}'$
   (c) *Assign* $\mathcal{E}' \leftarrow \mathcal{E}' \setminus \{e_1\}$
   (d) *Assign* $\mathcal{E}_R \leftarrow (\mathcal{E}_R \setminus \{j\} \bigcup \{e_1\})$
   (e) *Ignoring edge direction, perform a DFS over* $\{\mathcal{N} \bigcup \{v_0\}, \mathcal{E}'\}$ *starting at* k. *Denote* $\mathcal{N}_k$ *and* $\mathcal{E}_k$ *the visited nodes and edges, respectively.*
   (f) *Assign* $\mathcal{N}_T \leftarrow \mathcal{N}_T \bigcup \mathcal{N}_k$.

(g) Assign $\mathcal{E}_T \leftarrow \mathcal{E}_T \bigcup \mathcal{E}_k \bigcup \{j\}$.

5. *Assign* $\mathcal{S}^* = \mathcal{E}' \setminus \bigcup \mathcal{E}_T$.

Remark: For the case $\mathcal{R}^* = \emptyset$, , only steps 1, 3 and 5 are performed, and our algorithm becomes the same as the one presented in [11].

The first step of the algorithm aggregates the sources and sinks into a single node. The resulting node will satisfy the flow conservation equations and will make the graph strongly connected. Step 2 removes all but one of the outgoing edges for each of the nodes in $\mathcal{R}^*$. Then, step 3 constructs a tree from the remaining edges. However, because of the removal of edges in step 2, the resulting graph may become disconnected, then the DFS algorithm may not reach all of the nodes in the graph. Step 4 redoes the removal of outgoing edges from $\mathcal{R}^*$ such that the resulting graph is connected, and finishes the construction of a spanning tree of the original graph. Finally, flow sensors are located in edges which are not included in the spanning tree or the removed edges.

Remark: The removal of links in step 2 and the construction of spanning trees using the DFS algorithm are not unique, and alternative node and edge indexing might yield different sensor configuration. Nevertheless, all multiple solutions provide the same number of sensors, and are thus equally optimal.

6.3 Comparative Analysis of Partial Data Estimation Approaches

We now demonstrate the approaches introduced above using a follow graph. we can see that it has a source denoted with '$v+$' and a sink denoted with '$v-$'. We can denoted the set of sinks/source $\mathcal{C} = \{'v+','v-'\}$ and set of intermediate node $\mathcal{N} = \{1, 2, 3, 4, 5, 6\}$.

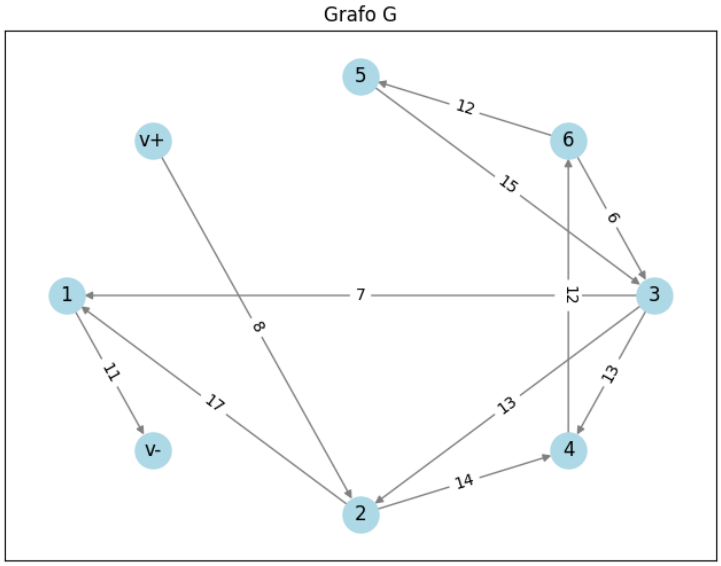

Figure 6.1: Original graph

Now using algorithm 5.1 we can build the graph with a single centroid and we sum the flow oh its edge.

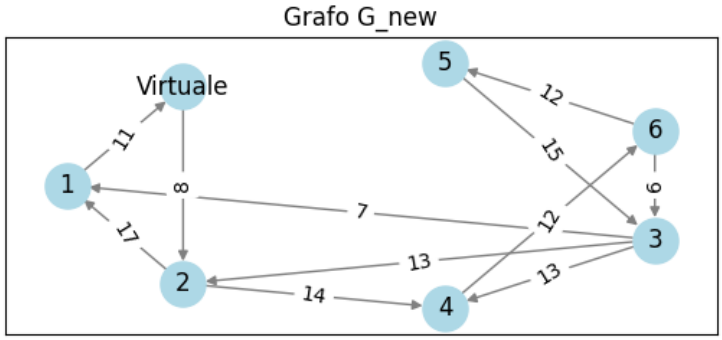

Figure 6.2: Centroid graph

Now let's assume we have partial data, in particular we have information from turning ratio sensors on only two nodes. Using algorithm 6.3 we can obtain the optimal position of turning ratio sensors that is the set fo nodes $\mathcal{R}^* = \{2, 3\}$. now having partial information on the turning ratios, using algorithm 6.4 we can still obtain the optimal set of flow sensors and their positions thanks to the spanning tree technology. We show an intermediate step of the algorithm. Step 2 removes all outgoing edges except one for each of the nodes $\mathcal{R}^*$ that is edges (3,1), (3,2) and (2,4) while step 3 builds a tree from the remaining edges.(Due to step 2 resulting graph may be disconnected, hence the DFS algorithm may not reach all nodes in the graph).

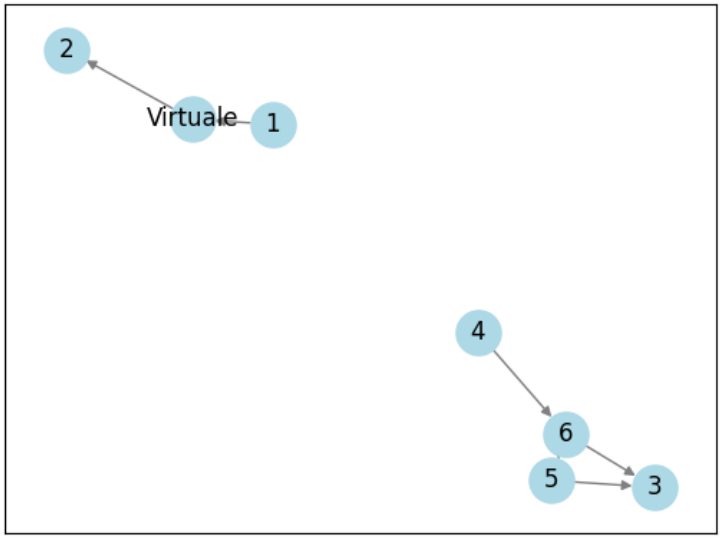

Figure 6.3: Intermediate graph

Step 4 repeats the removal of the outgoing edges from $\mathcal{R}^*$ such that the resulting graph is connected and finishes the construction of a spanning tree of the original graph. In this case we connected edge (4,2) and we go to step 2.

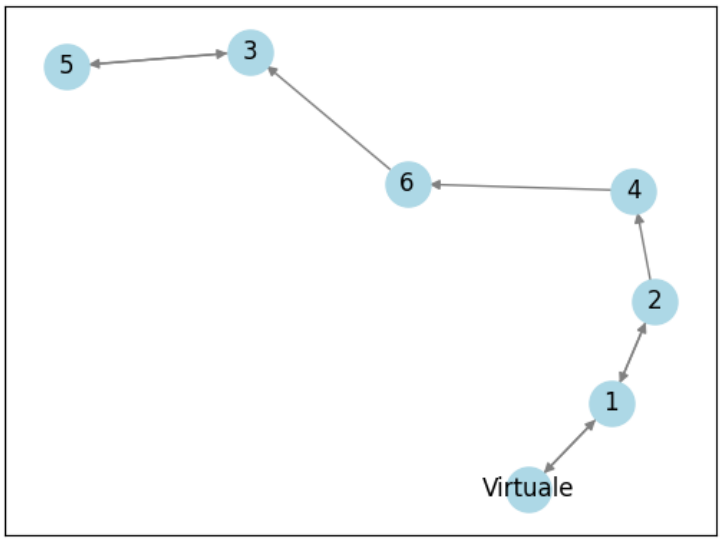

Figure 6.4: Spanning tree with $\mathcal{R}^*$

With step 5 the flow sensors are located in the arches which are not included in the spanning tree or removed edges. Then we can find the set of optimal flow sensor $\mathcal{S}^* = \{(\text{"}Virtuale\text{"}, 2), (6, 5)\}$.

Chapter 7

Simulation and Validation

In this chapter we will apply the algorithms introduced in the previous chapter to real cases, in particular using graphs built on networks taken from real maps and on the same network we will carry out simulations with the aid of commercial software. We will start by applying our approaches in the case of a map by evaluating the number and type of sensors to install. With the data collected we will carry out simulations to validate our model. Finally, we will carry out analyzes comparing the results of the theoretical model with the results of the simulation model, highlighting the discrepancies, trends and error percentages.

7.1 Real-world Application and Validation

In this section, we will apply the algorithms set out previously in real-world case by taking a map of the city of Catania. In particular we will take a portion of the city that includes in the east entrance area of port. We chose the portion in the east entrance area of Catania because vehicle slowdowns were recorded due to the increase in port traffic [18].

We will use two sensor technologies:

- counting sensors: types of sensors that are placed in correspondence with the arches and simply count the passage;
- turning ratio sensors: of the types of sensors that are placed at intersections and in addition to counting the number of vehicles, they also calculate the turning ratio from which the incoming and outgoing flow can be obtained.

Based on previous algorithms, we will use the Python programming language using the Networkx package to model the Graph. "NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks" [16]. we will obtain the spanning tree of the turning ratio and the counting sensors to have full observability and knowing

only the number of turning ratio sensors to install we will obtain the number of counting sensors to install, in addition to the number we will also obtain the optimal position. We will get all the information regarding the structure of our network under consideration from the opensource platform Openstreetmap.

7.1.1 The OpenStreetMap data structure

OpenStreetMap (OSM) is the largest free and editable geographic database in the world, built by volunteer work and released under a free license. It is a gigantic collaborative project, with millions of registered users around the world, whose aim is to create and provide free geographic data to anyone who wants to use it. Thanks to this platform it is possible to select the latitude and longitude coordinates so that the map frames the interested area and export the file. In particular, as we can see in the following figure, we must select:

- max latitude= max_{lat}.
- min latitude= min_{lat}.
- max longitude= max_{lon}.
- min longitude= min_{lon}.

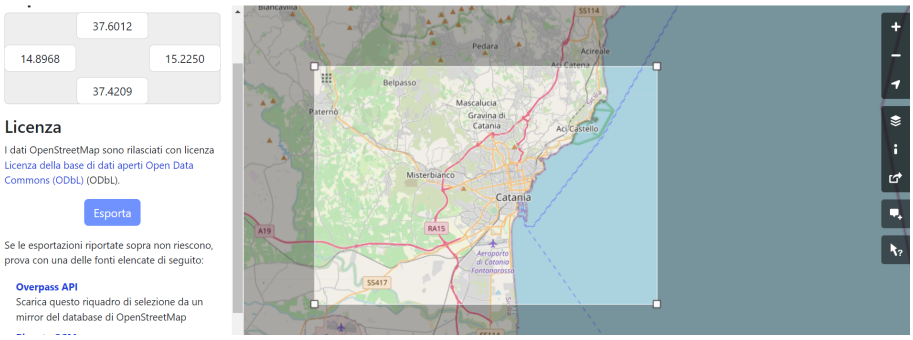

Figure 7.1: Catania

To select our portion of the Catania section which also included the port we entered the following values as parameters:

- $min_{lat} = 37.5114$.
- $min_{lon} = 15.0877$.
- $max_{lat} = 37.5206$.

- $max_{lon} = 15.1077$.

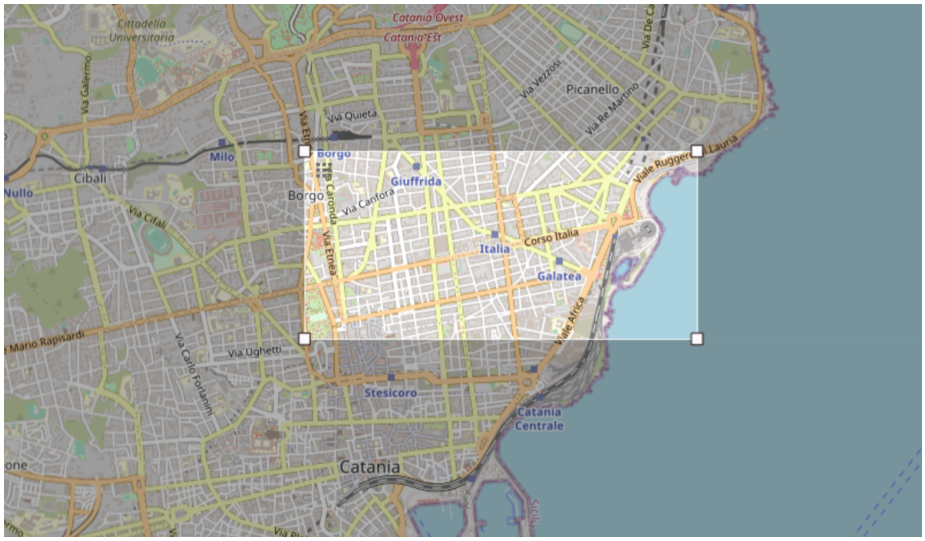

Figure 7.2: Section of Catania's porto

OpenStreetMap is a database, all the elements that can be inserted are of four types:

- Point (node): single point. Used to indicate objects "punctual" such as trees, traffic lights, mountain peaks or even to indicate a commercial activity in a simplified way, a restaurant, etc... .
- Line (way): a non-closed set of points. the path (way) is a segment between points that can describe a street, a river, the route of a ferry, a wall, etc.. .
- Area (polygon) : a closed line, usually tagged area=yes . In OSM there is no "primitive" polygon: if we want to represent a building, a lake or something that has an area we need to draw a path that closes on itself.
- Relation (relation): a set of the previous elements, for example a bus line that is made up of multiple roads and its stops. The relationship is not a real object but a container used to define complex objects. Who is just starting out with OSM it won't create or modify relationships, but at the moment it's right at least know that they exist.

To describe the objects mentioned above, the use of tags that follow an international standard is essential. Every object must therefore add at least

the main label that identifies it. They are always composed of a pair of elements. The first is said key (key), the second value (value). Usually the key describes a family of characteristics, while the value goes more specifically. In OpenStreetMap the communication routes are classified as follows:

- Roads
  - Main roads.
  - Urban roads .
  - Roads and streets for non-motorized vehicles.
  - Agricultural and forestry roads.
  - Various roads.
- Railways .
- Airways.
- Waterways.
- Water traffic.
- Docks.
- Transport of energy and goods.
- Barriers.
- Nature.

For our research object we will focus only on roads, in particular on the main streets of a road network which can be ordered according to a scale of importance: motorway,trunk, primary, secondary, tertiary and quaternary. Scale of importance *highway*:

- $highway = motorway$: The most important arteries in a road network. Equivalent to freeway (UK, USA), Autobahn (Germany), etc... .
- $highway = trunk$: Extra-urban road; The most important roads in a road network, other than motorways.
- $highway = primary$: Primary road.
- $highway = secondary$: Secondary road.
- $highway = tertiary$: "Tertiary" road. Road of local importance.
- $highway = unclassified$:"Quaternary" road. Base grid road.

While we will leave aside other types of communication routes such as:

- $highway = residential$:Residential street.

- *highway* = *pedestrian*: Pedestrian street.
- *highway* = *cycleway*: Bicycle lane.
- *railway* = *rail*: Railway for passenger trains.

The map includes a lot of information, only the coordinates of intersections and streets have been used for this work. With the aid of python we transformed the map file into a graph maintaining the initial configurations of the longitude and latitude coordinates. This is possible through the use of python's OSMnx library. OSMnx is a Python package to download, model, analyses, and visualize street networks and other geospatial features from OpenStreetMap. You can download and model walking, driving, or biking networks with a single line of code then easily analyze and visualize them. You can just as easily work with urban amenities/points of interest, building footprints, transit stops, elevation data, street orientations, speed/travel time, and routing. [17] and through this library we have to model the graph in such a way as to be able to represent it and superimpose it on the map with only the information of the arcs and intersections.

We can see in the following figure our graph superimposed on the map where the coordinate information has been kept. We note that the "blue" lines represent the arcs and the "red" points the intersections. The graph represented contains 1121 nodes and 2669 edges.

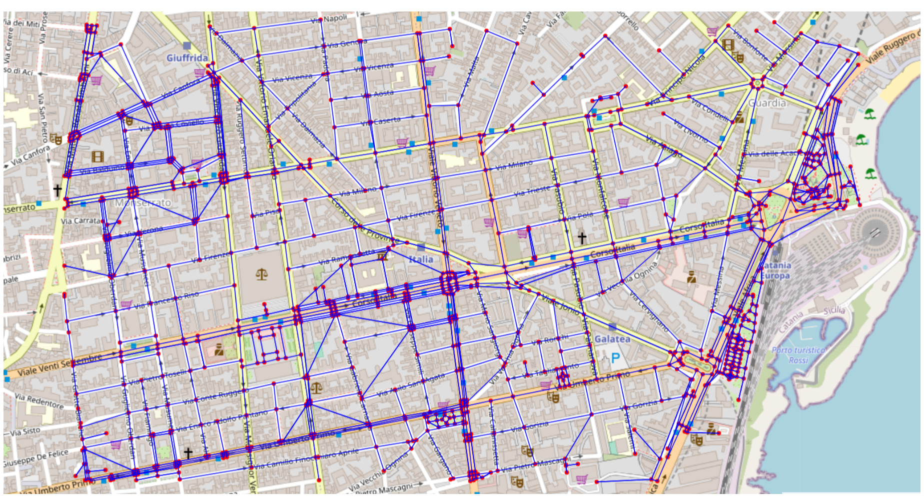

Figure 7.3: Reconstruction of Graph

Knowing only the number of turning ratios that means to have only the information from N=40 flows in correspondence of intersection with the algorithm

6.3 we can find the optimal position of TR sensors where the installation of the turning ratios is represented by the "blue" dots on the map.

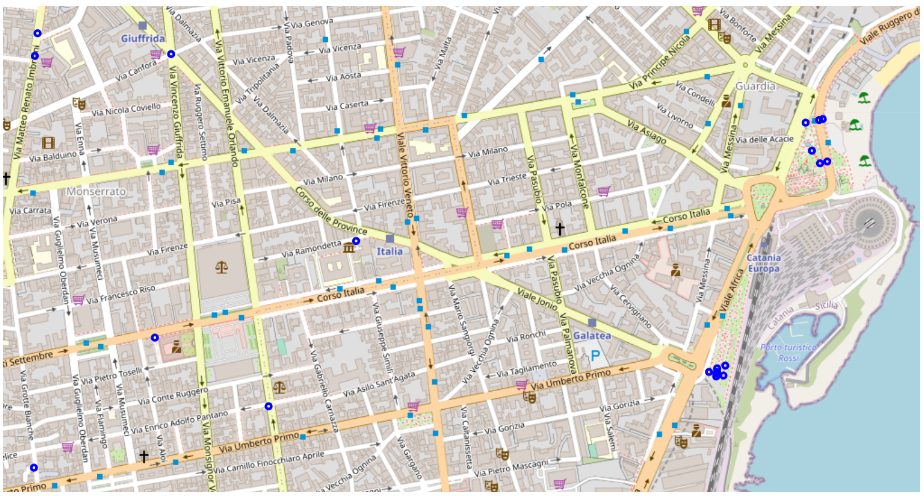

Figure 7.4: Optimal position of TR sensors

After having found the set of optimal positions of the turning ratios, we are ready to apply Algorithm 6.4.Through this algorithm, having only the information on the optimal position of the turning ratio sensors, we can obtain the spanning tree where there will be no need to install the counting sensors, and where, after installing these sensors in the remaining tree we could obtain the flow using the law of conservation of flow. In the following figure we can see that the intersections are represented by the "red" points while the arcs by the "blue" lines. We can immediately notice, observing the density of the blue lines, that the number of arcs represented from which we can obtain the flow without the installation is notable.

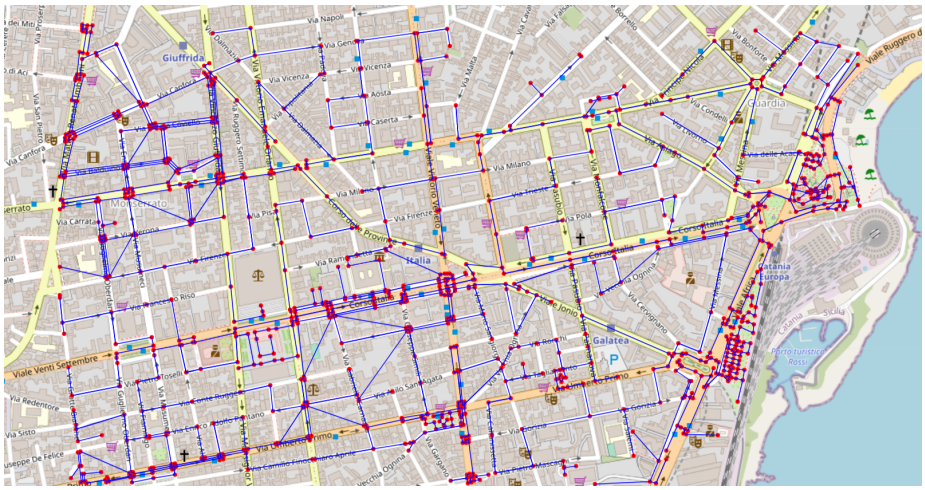

Figure 7.5: Spanning tree of Graph with $\mathcal{R}^*$

The number of edges representing the spanning tree together with the turning ratio information is 1371 which is exactly the number of counting sensors that we will not have to install despite having the flow information. By excluding the spanning tree from the initial graph we will find the arcs on which we will have to install the flow sensors. We can see in the following figure that the "green" lines superimposed on the map represent the counting sensors to be installed on the arches.

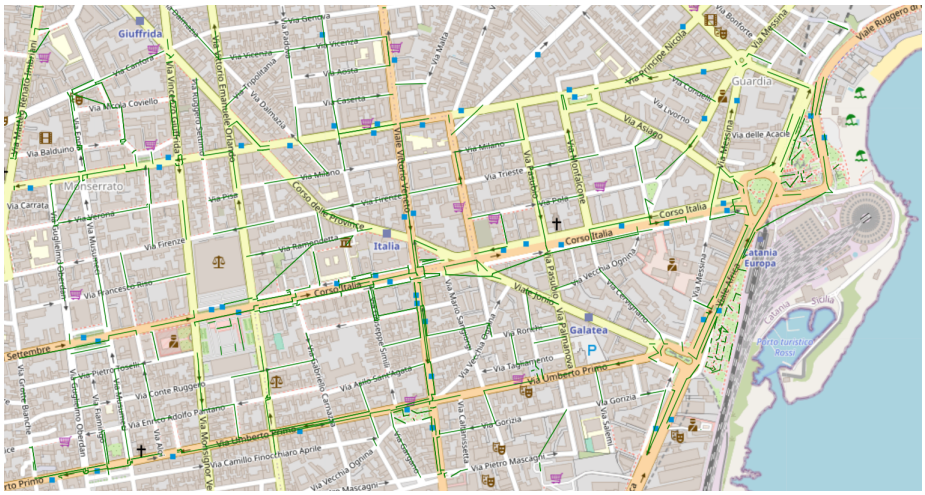

Figure 7.6: Flow sensor to install

In particular,we have to install about 1050 flow sensor These numbers respect the values of equation 6.13 from corollary 6.11. We can already see that using this algorithm halves the number of sensors to be installed to obtain the flow. If we compare with the initial Figure 6.6 of the graph we can see graphically in the following Figure 6.9 the much lower density of the arcs in green color.

We can see the Trade off between turning ratio and flow sensors for the simulated network. The x-axis represents the number of turning ratio for monitored intersections and the y-axis is the number flow sensors for for complete observability.

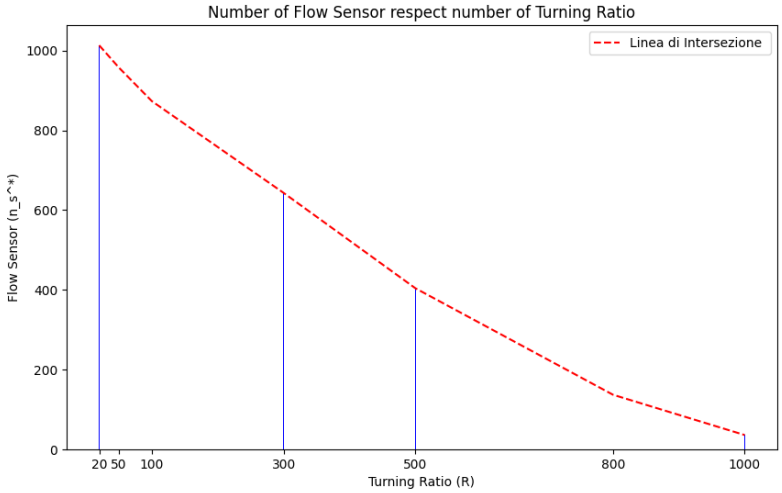

Figure 7.7: Trade off between TR and flow sensors

we can see that the number of counting sensors to be installed is inversely proportional to the number of turning ratio sensors. In particular, it is advisable to install more turning ratio sensors as they require a lower number in proportion to the flow sensors, even if they are more expensive, so we try to find a middle ground. We can show this trade of also using the sensors of map. We can see the difference with $R = 20$ and $R = 500$ the number of flow sensor used for monitoring traffic.

We can see how by using 480 more turning ratio sensors, we can save 621 counting sensors. In fact, if we look at the maps we can notice that as the

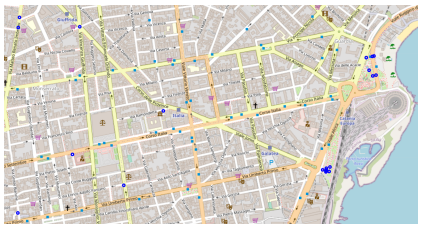

Figure 7.8: R=20

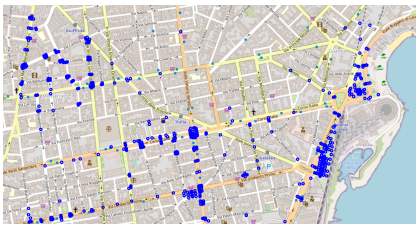

Figure 7.9: R=500

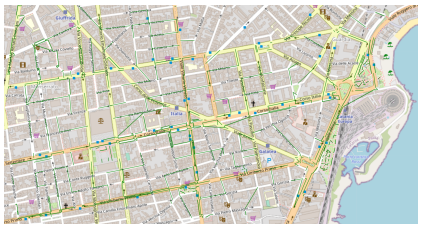

Figure 7.10: S=1037

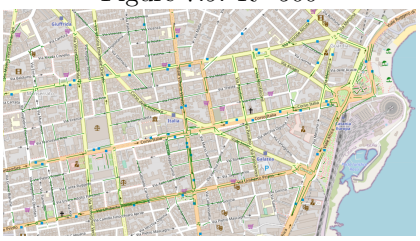

Figure 7.11: S=416

density of turning ratio sensors increases at the nodes, the density of flow sensors at the arcs decreases.

7.2 Preparation for the simulation of the traffic scenario

In this section we will show the preliminary steps to carry out our simulation. To prepare our network we will use the PTV Visum software which will standardize the network and then carry out the Simulation with the help of PTV Vissim. The two software are part of the PTV Group package, a leading company in the transport planning sector. Let's start by giving a brief description of the two software we are going to use:

- PTV Visum stands as a paramount transportation planning software, setting the standard for simulations and macroscopic modeling of transport networks. It encompasses transport demand analysis, public transport planning, and the formulation of transport strategies and solutions. With PTV Visum, one crafts transportation models that serve as the bedrock for both far-reaching strategic blueprints and nimble short-term operational decisions.

- PTV Vissim is recognized as the leading multimodal traffic simulation software globally. It accurately replicates the movement of all road users in a digital environment. Carefully evaluate and improve the effectiveness of your traffic systems. The results provide a solid basis for traffic planning

decisions, effectively addressing challenges such as traffic congestion and emissions.

In this work the use of PTV Visum is exclusively for the design of our network in such a way as to correctly import our map onto PTV Vissim for our simulation. We will therefore only use and limit ourselves to showing the main functions and tools we use to standardize our network:

- Input Data:
  - Offer model (Network graph with element attributes):
    - Arc length: real distance between two nodes.
    - Lanes and mileage: number of lanes present for each direction of travel.
    - Speed: Design speed of the road.
    - Capacity: Maximum volume that can be supported by the bow.
    - types of vehicles: the type of vehicle that can cross that road (e.g. car, bicycle, pedestrian, bus).
- Output Data:
  - Standardized network in ANM format.

Now let's take a smaller portion of the port to carry out our simulations, based on official documents from the municipality of Catania [21] on the hourly density of vehicles passing through the port area that connect with the city, we have identified a portion that includes a section of "Corso Italia" where the traffic density is quite high as we will see in the following figure.

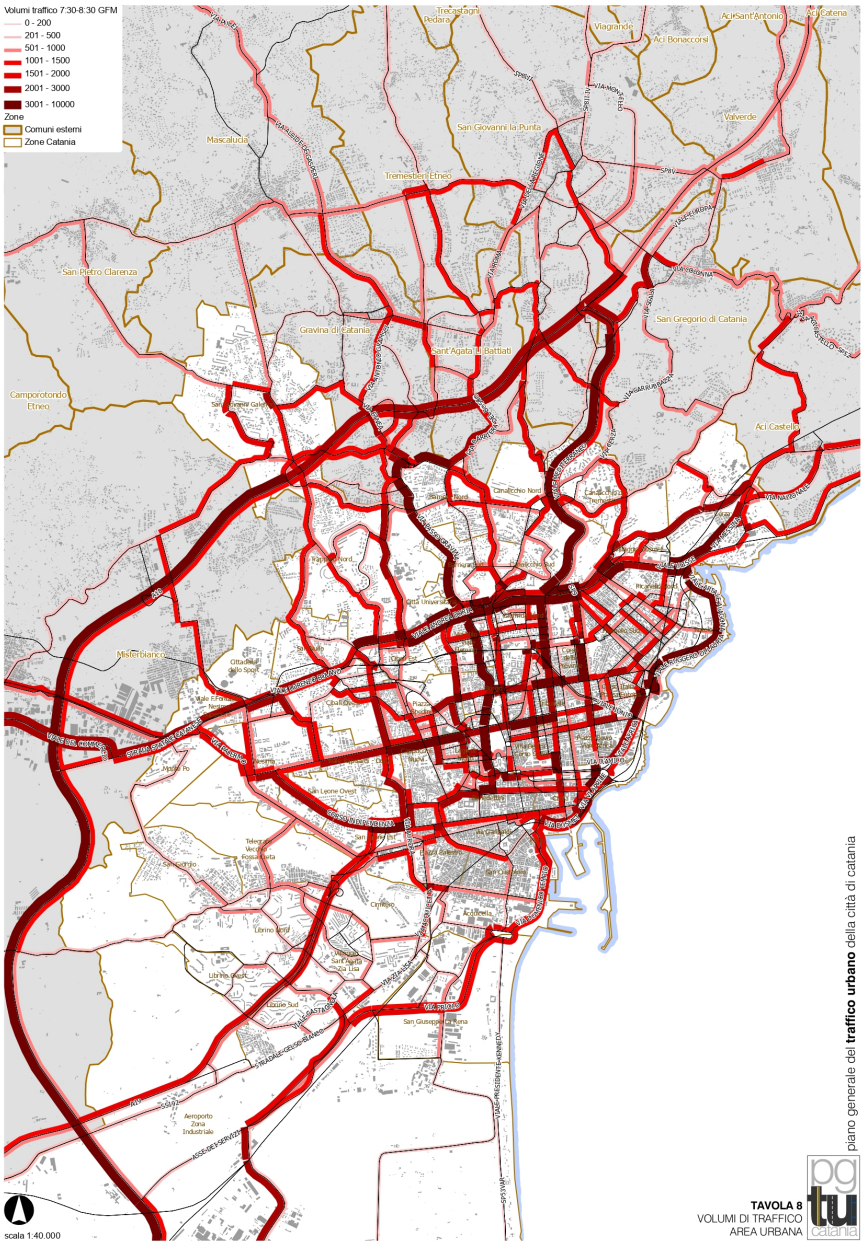

Figure 7.12: Traffic volume

With the help of Openstreetmap we are going to select a smaller portion of the

previous map so that the graphic representation is clearer and does not overload our simulation too much as we are interested in the validation of our simulations and not in a real representation of the phenomenon even if we will strive through research and in-depth analysis to be as close as possible to a real application.

We enter the following parameters to export the file in osm format:

- $min_{lat} = 37.5141600$.
- $min_{lon} = 15.0943800$.
- $max_{lat} = 37.5153200$.
- $max_{lon} = 15.0968800$.

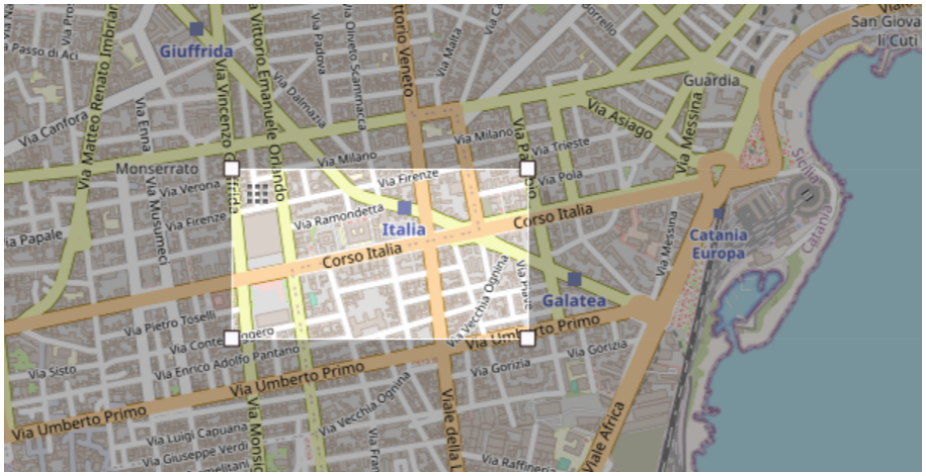

Figure 7.13: Section of Corso Italia

After downloading the osm file, we open the PTV Visum software and import the file. Using the interface of our software we will check that all the information in the osm file has been imported correctly.

Observation 7.1. It is important to note that importing the file can lead to some reading problems therefore we will provide the following PTV Visum functions to modify any errors in such a way as to represent the map with all the initial real values.

Based on what was said in the previous section, since we are only interested in urban roads that do not include pedestrian paths, cycle paths, railway lines and residential streets, we are going to select "Urban road network" in the

import configuration. We also choose to include all the information from the OSM file (for example primary, secondary road attributes, etc.).

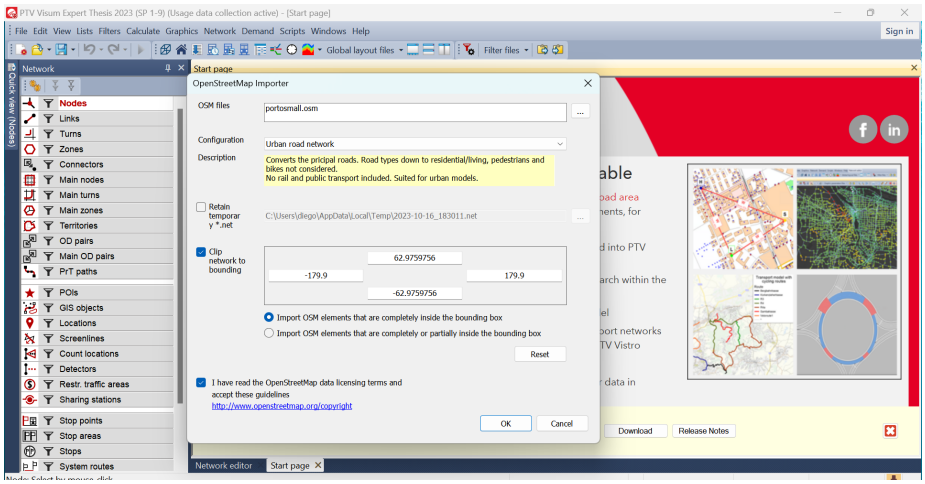

Figure 7.14: Import parameters

As soon as the file is imported, the map will appear in the "Network editor". The following figure shows the map imported with the standard graphics of our program. We can see that it distinguishes between a thicker primary road in "yellow" and a thinner secondary road in "blue". You can see how the intersections that correspond to the nodes of the graph have also been numbered. We can also see the presence of two traffic lights at the beginning of the streets at the top right of the network which we will then eliminate.

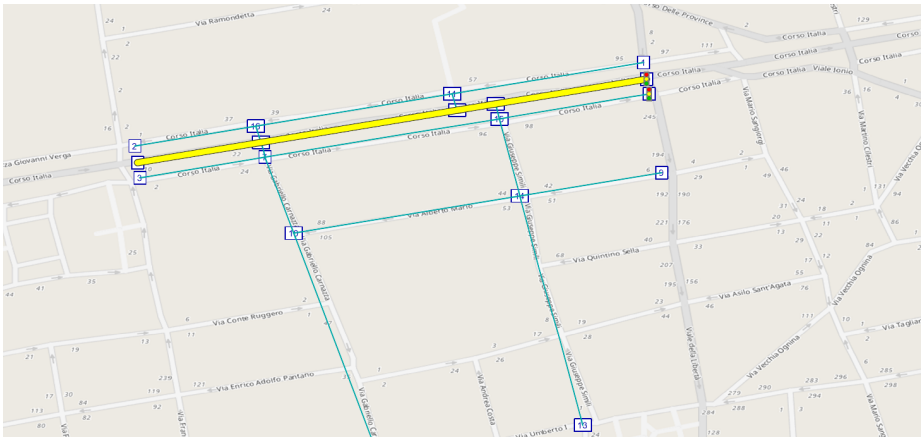

Figure 7.15: Map view

Now we can define the set of vehicles. in this case we define the set of vehicles that cross the links formed by the following vehicles:

- Car.
- Bus.
- HGV (Large goods vehicle).

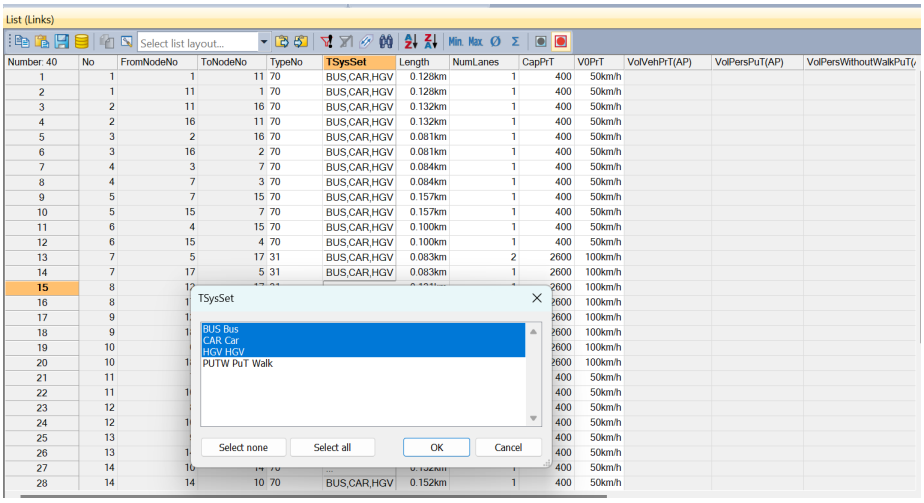

Figure 7.16: Types of vehicles

After defining the set of vehicles, we deal with the topology of the network, in particular we will check that there are no isolated points, that all the links are connected and that the direction of the links is consistent with the topological structure.

Furthermore there are other specific checks to do such as:

- capacity of links;
- zones and connectors;
- consistency of network;
- validity of matrix values;
- turn that make sense;
- other variable of interest.

After verifying the previous properties, we can do the network check to see if there are still errors.

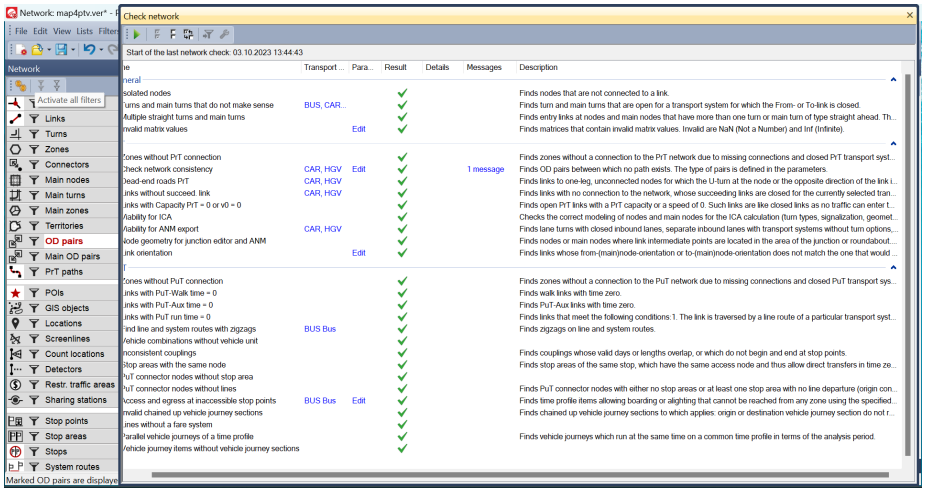

Figure 7.17: Network check

Now our network is ready to carry out the simulations, so we export the file in ANM format and open it on PVT Vissim.

7.3 Validation of Traffic Flow Models and Estimation Methods

In this section we will carry out a simulation on the portion of the map that we have previously processed using the PTV Visum software. Based on the information we have, we will apply the algorithms of section 6 and with the help of the PTV Vissim software we will simulate the traffic within the previously chosen map by collecting information on the flows in the intersections and in the arcs indicated by the outputs of the algorithms. Finally we will validate the results by calculating the missing flows and comparing them with those of the simulation.
As we said in the previous section PTV Vissim is a multimodal traffic simulation software.
Now let's try to give an overview of this program and understand how it works:

- **Input Data**:
  - Infrastructure:
    - Geometric dimensions (e.g. length and width of the lanes).
    - Areas of conflict/intersections (rules of precedence, traffic light controls, etc).
    - Related parameters/signage (limit speed, traffic light cycles, precedence and stop, pedestrian crossings, lane separation, maneuverability signs, etc).
  - Users:
    - Type and number of vehicles (standard, heavy vehicles, public transport, bicycles, pedestrians, etc...).
    - Routes, times and vehicle flows.
    - Specific areas and car parks (dimensions, characteristics, etc).
  - Map display and scale.
  - Simulation parameters.
- **Output Data**:
  - Traffic data: number of vehicles, speed, traffic density, distance between vehicles, queues, travel times, congestion etc... .
  - Agent data (vehicles and pedestrians): behavior of vehicles and pedestrians (paths, speed, accelerations, decelerations, overtaking decisions, etc...).
  - Analysis reports: simulation results, graphs, tables and statistics.
  - Environmental data: Information on emissions, fuel consumption and other related environmental metrics.

We use the previously processed file and export it to Python through the OSMnx library which allows us to maintain the coordinates of the original file so that we can visualize our graph in the map. We can see that they have a slightly different configuration, but this is dictated by the fact that they are 2 different programs, from a resolution and calculation point of view they arrive at the same result. The graph examined is made up of 52 nodes and 135 edges. Where we can see the lines in "blue" representing the arcs and the points in "red" representing the intersections.

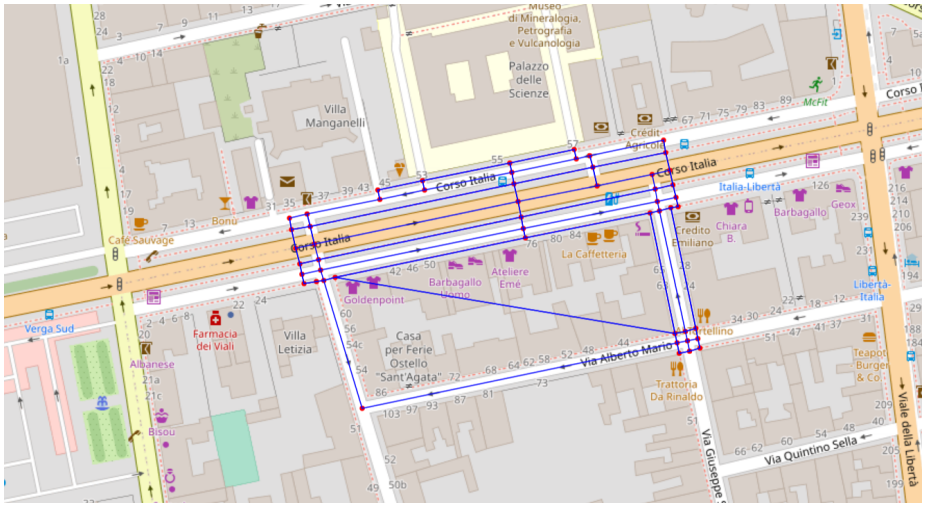

Figure 7.18: Original graph

Having only the information on the Turning ratio sensors with R=20, we can use algorithm 6.3 for the optimal position of the TR's sensors. We can see that the intersections where the turning ratio sensors will be installed are marked on the maps with "blue" points.

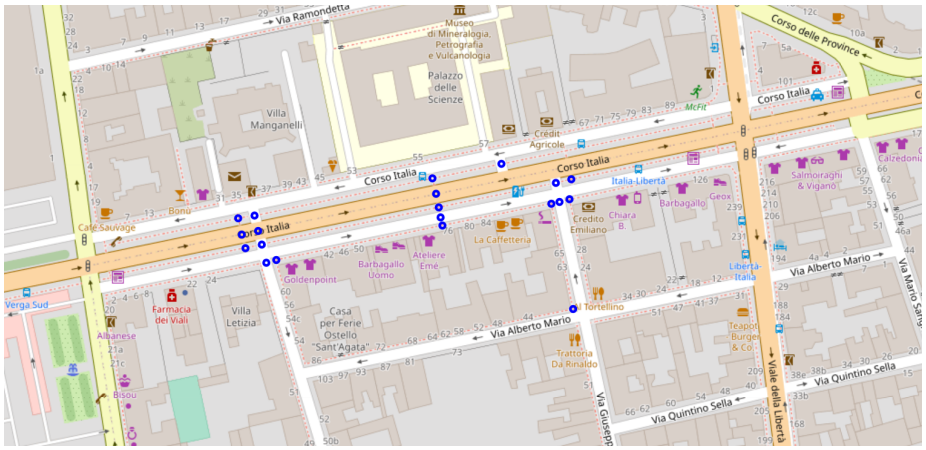

Figure 7.19: Optimal locaiton TR sensors

We save the results in the follow table. The first column denoted with "Number" simply counts the number of sensors, while the second column denoted with "Position Nodes" represents the coordinates of the intersection where the installed sensor is located.

Observation 7.2. The number indicating the intersection "Position Nodes" is nothing more than a unique identification code of the node of the original graph exported from the Openstreetmap file. We have chosen to keep this code so that it can then be compared with the Openstreetmap maps as it represents a unique code of the position of the graph node relative to the chosen map.

Number	Position Nodes
1	272872448
2	1859036801
3	2283181314
4	1859036803
5	1859036804
6	945509385
7	2283181335
8	2283181337
9	2283181341
10	2283181342
11	2283181345
12	945509410
13	2283181348
14	945509417
15	2283181353
16	945509419
17	2283181357
18	2283181367
19	2283181382
20	1859036772

Table 7.1: Optimal position of TR sensors.

Using the information on the number and optimal position of the set of turning ratio sensors by applying algorithm 6.4 we can find our spanning tree where we could obtain the flows without installing sensors. We can see that the "blue" lines represent arcs, while the "red" points represent intersections.

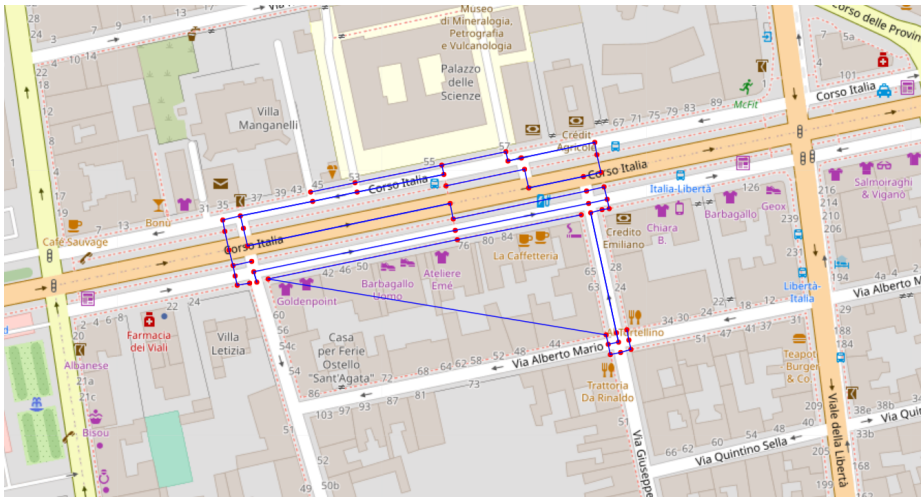

Figure 7.20: Spanning tree with $\mathcal{R}^*$

Together with the spanning tree, Algorithm 6.4 also provides us with the optimal set of number and position of counting sensors to install to have full observability of the flow of our network. We can see that the "red" lines represent the arcs where the flow sensors will be installed.

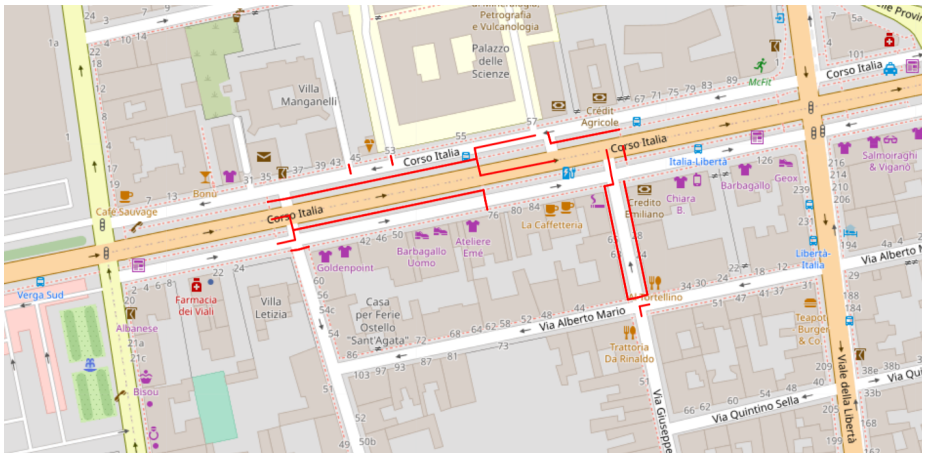

Figure 7.21: Optimal locaiton Flow sensors

Figure 7.8 represents the set of counting sensors to be installed to detect the traffic flow. we can see that the set of counting sensors to be installed is S=51.

We save the results in the follow table. The first column denoted with "Number" simply counts the number of sensors, while the second column denoted with "Position Edges" represents the initial and final node. Therefore it uniquely represents the arch on which the counting sensors will be installed to obtain information relating to vehicle flow.

Number	Position Edges	Number	Position Edges
1	2283181342 → 2283181345	26	2283181376 → 2283181357
2	1859036806 → 2283181382	27	1859036800 → 945509410
3	2283181367 → 2283181353	28	945509417 → 274252586
4	945509419 → 2283181348	29	8306875174 → 8306875172
5	2283181341 → 2283181337	30	2283181335 → 272872448
6	1859036803 → 1859036772	31	2283181337 → 1859036803
7	2283181382 → 2283181414	32	2283181367 → 2283181345
8	8306875181 → 8306875178	33	274252586 → 945509419
9	2283181367 → 2283181382	34	945509385 → 2283181348
10	1859036804 → 1859036776	35	1859036776 → 1859036804
11	2283181367 → 2283181374	36	2283181353 → 2283181367
12	1859036803 → 945509417	37	9455094117 → 1859036804
13	1859036774 → 1859036772	38	283181310 → 2728723306
14	2283181353 → 2283181367	39	1859036765 → 310880162
15	2283181348 → 945509385	40	945509385 → 272872330
16	1859036801 → 945509410	41	1859036774 → 1859036778
17	2283181345 → 2283181367	42	1859036772 → 1859036803
18	1859036774 → 1859036776	43	2283181357 → 2283181376
19	1345621267 → 2283181374	44	8306875178 → 8306875181
20	1859036759 → 310880162	45	1859036763 → 'Virtuale'
21	945509410 → 1859036801	46	1945509419 → 282757151
22	2283181348 → 2283181341	47	8306875172 → 8306875174
23	1859036776 → 1859036774	48	1859036759 → 1859036761
24	272872448 → 945509385	49	1859036776 → 1859036774
25	2283181345 → 2283181342	50	2283181354 → 2283181357

Table 7.2: Position of flow sensors

Table 7.3: Position of flow sensors

After extracting the information we need, that is:

- $\mathcal{R}^*$=Optimal set of turning ratio sensors
- $\mathcal{S}^*$= Optimal set of counting sensors to install

We will use the information from Tables 7.1 and 7.2 to carry out our simulation on PTV vissim. Let's start by importing the PTV Visum file previously processed into PTV Vissim. We see in the following image the figure of our file in standard format. We note that the "blue" lines represent the links, while the "purple" lines represent the lines of intersections which in this program are

recognized as "connectors" or trajectories from which it is possible to connect with other roads. We note that the representation is different from PTV Visum, so we will process all the information that will be needed for our simulation.

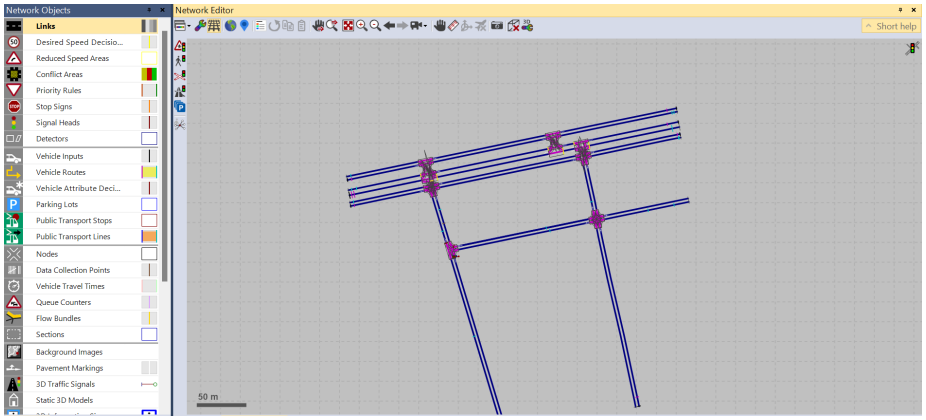

Figure 7.22: Standard view

There are different types of visualization of our graph. In this case we will use the editor "Maxar 2023, Microsoft corporaiton" where you can see a real reproduction of the city in the background.

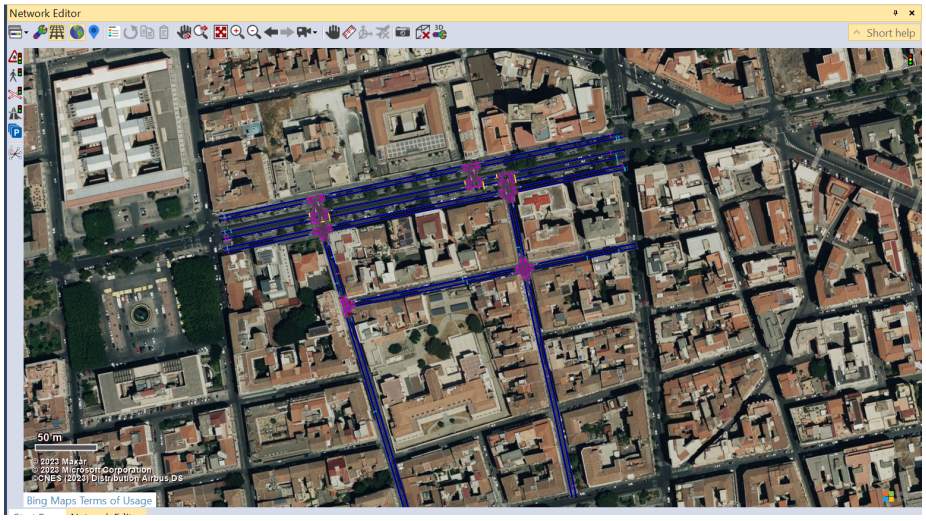

Figure 7.23: Custom view

Figure 7.10 represents the Openstreetmap file previously represented and implemented on ptv Visum. First we need to set an input for the number of vehicles,that is, how many cars,bus and HGV we expect to pass that road. we will do this for each source. The number entered in the "volume" column represents the number of vehicles input from that source in an hour, therefore the unit of measurement used is veh/h.
While anticipating that we are not interested in a real estimate of the flows at the level, but only in the method of estimation and calculation through the sensors, we will try to use data that is as truthful as possible, but we reiterate that we are only interested in the method from a qualitative and not quantitative. The source from which we obtained the information is the municipality of Catania through a public document of the general urban traffic plan [21]. As we can see in the previous map of Figure 7.12, in correspondence with the chosen portion, that of "Corso Italia", the map indicates a flow of 2000-3000 Veh/h, therefore we will insert a number of input vehicles that are in that interval, inserting the various inputs into every street that is connected to "Corso Italia" means that almost every vehicle will pass through there. Therefore we can add all the flows entered and choose the case in which we find ourselves in the highest traffic flow, i.e. 3000 Veh/h.

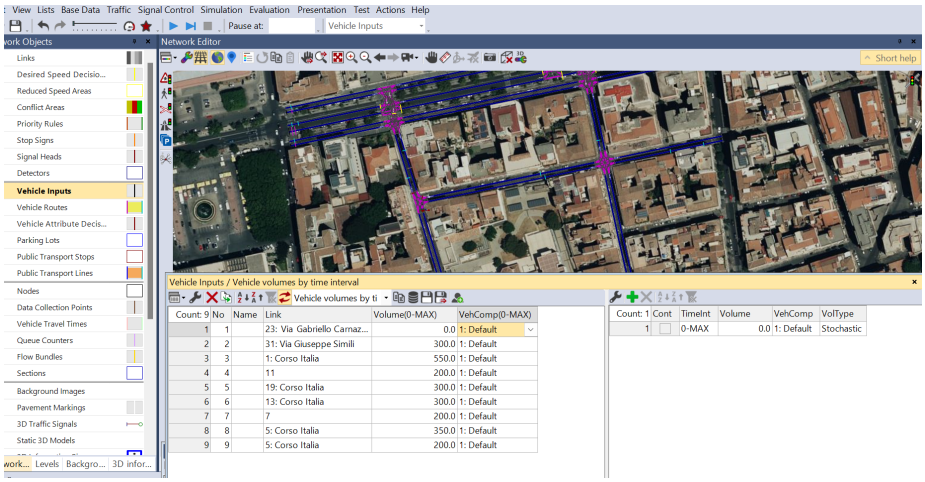

Figure 7.24: Vehicle inputs

After introducing the inputs we will also set the routes they will take. Each vehicle has a possible pre-set route, which must respect the shape of the road and traffic. In the following figure we can see in "yellow" the path that each vehicle will follow starting from the initial position denoted with a "purple" rectangle and arriving at the final position denoted with a "light blue" rectangle.

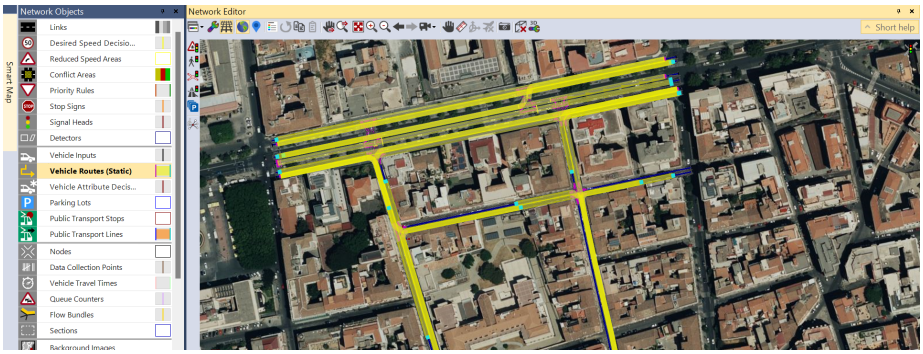

Figure 7.25: Vehicle routes

In addition to the routes that the vehicles must follow, there are also other parameters to be set that outline their behaviors. Such as the "Relative Flow rate" in PTV Vissim is a measurement that indicates the relationship between the flow of vehicles on a certain road or stretch of road and the theoretical maximum capacity of that road. Or the "REflow rate", i.e. cars remaining in the system before finishing their route. We can see some of the parameters set in the following figure.

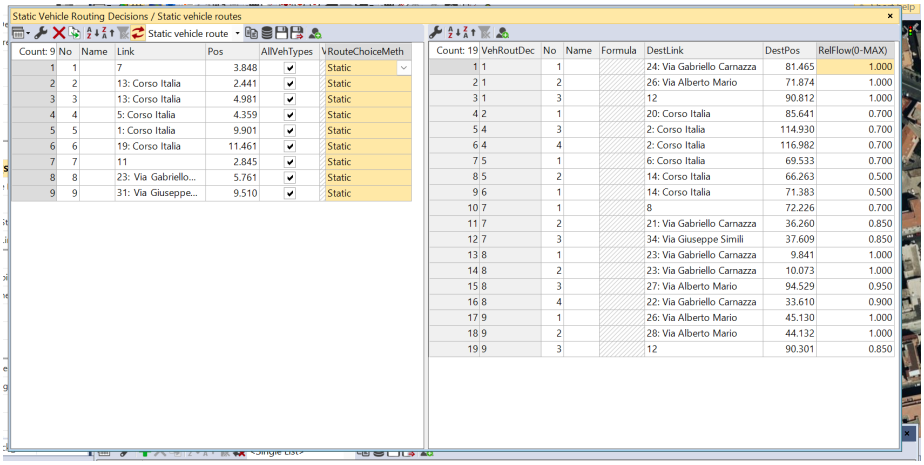

Figure 7.26: Vehicle behavior

Furthermore we will set the behavior as "static" which means that they will follow the same behavior in every area of the map.
After entering the previous parameters we can see the various areas of conflict.

Conflict areas are those areas where vehicles collide, so you need to set which vehicle has the right of way and which does not when their path intersects. In the following figure we can see how the conflict areas are of 3 types which manifest themselves based on color:

- Green= precedence.
- Red = give precedence.
- Yellow= no precedence.

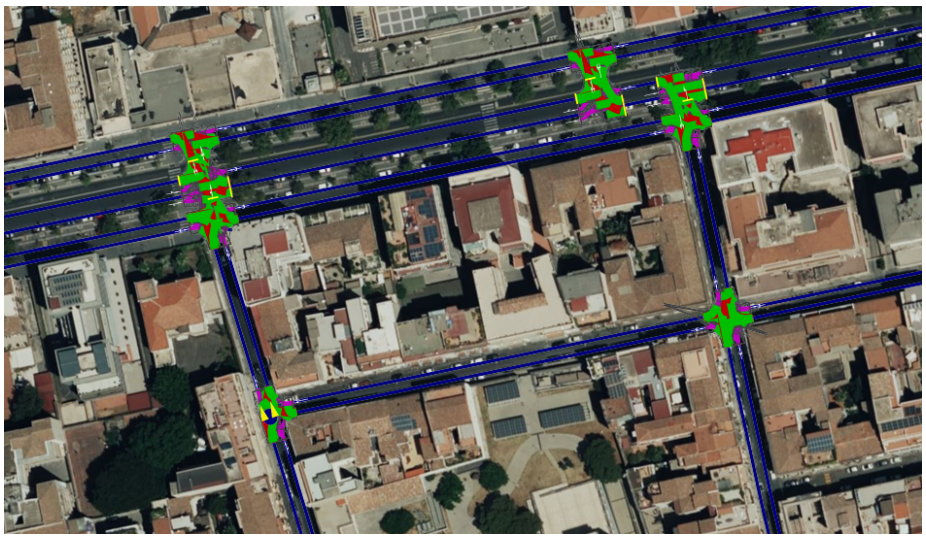

Figure 7.27: Conflict area

By carrying out a network check the areas of conflict that generate problems, for simplicity we set a traffic light control we can set all conflict areas to "passive", selecting "conflict area". In this way we will avoid configuration problems because as we have already reiterated we are only interested in the flow of vehicles and not in their behavior.

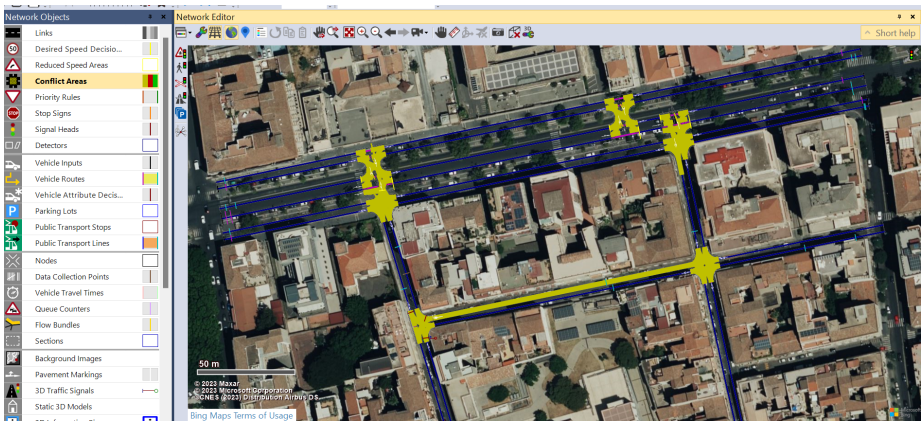

Figure 7.28: Passive conflict area

We can see that the behavior of vehicles can still be regulated by traffic light control as often occurs in real cases.
After setting all the conflict areas to "passive" we can see a 3D example of traffic light control. where at every turn a traffic light is inserted and a duration time is set. We can set follow parameters on a traffic light control of three traffic lights positioned at the intersection of three roads which has a cycle duration of 84 seconds:

- Traffic light 1: from 0 to 23 seconds green, 3 seconds yellow, 55 seconds red.

- Traffic light 2: from 26 to 53 seconds green, 3 seconds yellow, 55 seconds red.

- Traffic light 3: from 56 to 81 seconds green, 3 seconds yellow, 55 seconds red.

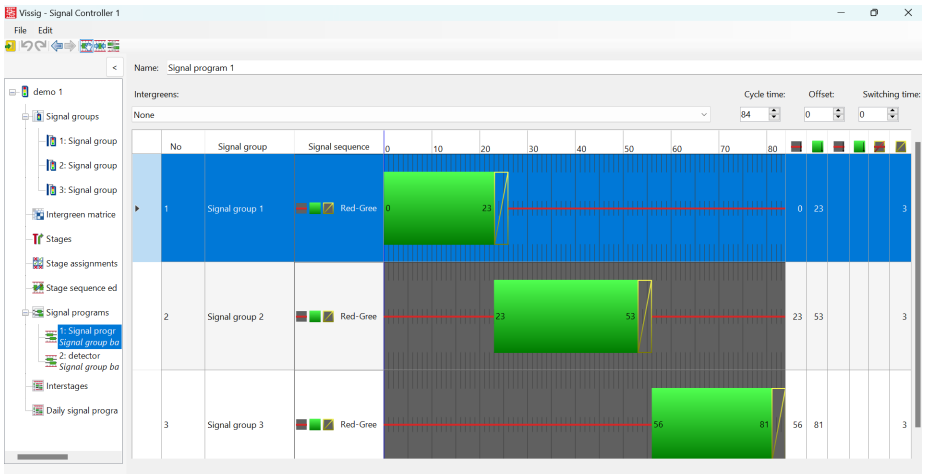

Figure 7.29: Traffic light parameters

In the following figure we can see a 3D representation of a traffic light control that includes an intersection between three roads.

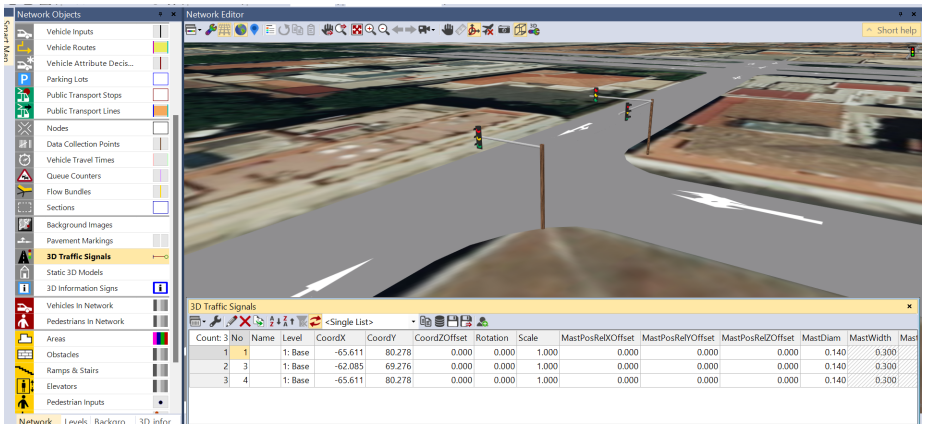

Figure 7.30: 3D view

Now using the data collected from Table 7.2 obtained from the previous algorithms, we will position the counting sensors in correspondence with the optimal positions. In PTV Vissim the sensors for data collection are called "Data collection points". These tools come on map routes and will contain any vehicles that pass by them. There are several functions that we can attribute to these tools

but we will limit ourselves to counting only the vehicles as we are interested in the flow.

Observation 7.3. At the same time we will install the flow sensors in all the arches and then validate them with those calculated by theory, but we will only show those belonging to the table of the optimal set of flow sensors to be installed.

N=50 Data collection points will be placed as indicated in table 7.2 in correspondence with the interested arcs. In the following figure you can see a portion of the screen that will be shown. We have chosen to show both the total count and some intervals to monitor the system as it develops.

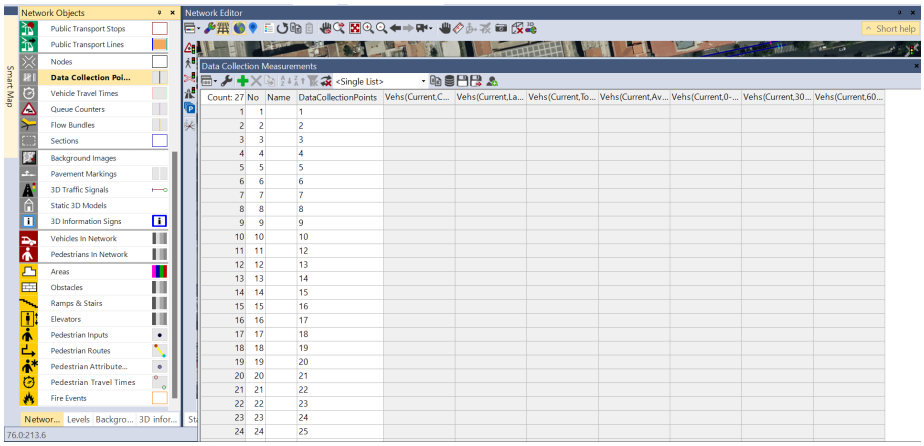

Figure 7.31: Data collection points

After placing the sensors in each arch we are ready to carry out our simulation. There are many parameters to set, we will show some of them below We set the simulation for a time n= 1080 seconds that is 18 min. This means that the number of vehicles to be contained is represented in a time span of 18 minutes.

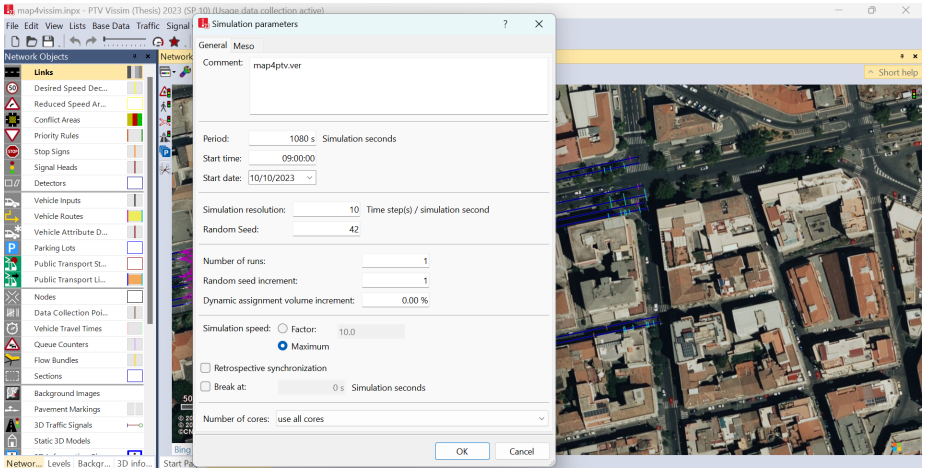

Figure 7.32: Simulation parameters

The data that each individual sensor provides us relates only to the number of sensors that are crossed, but since we are also aware of the time we can consider them as flows After having set all the parameters of the simulation we can click "run" and wait for the sensors to collect the information. Once the simulation is over these are the following results of the vehicle flows in 18 minutes. Figure 7.20 and 7.21 show the number of vehicles that crossed that road in 18 minutes corresponding to the arches where according to theory we must install the flow sensors.

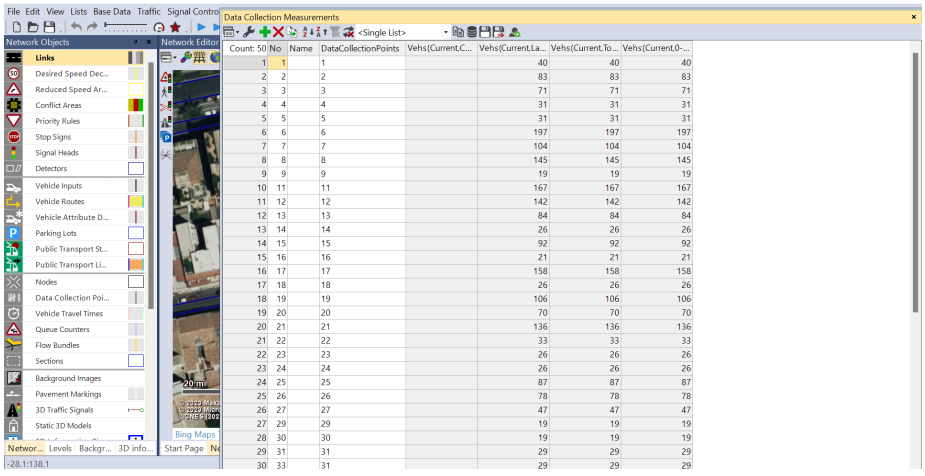

Figure 7.33: Data collection measurement (page 1)

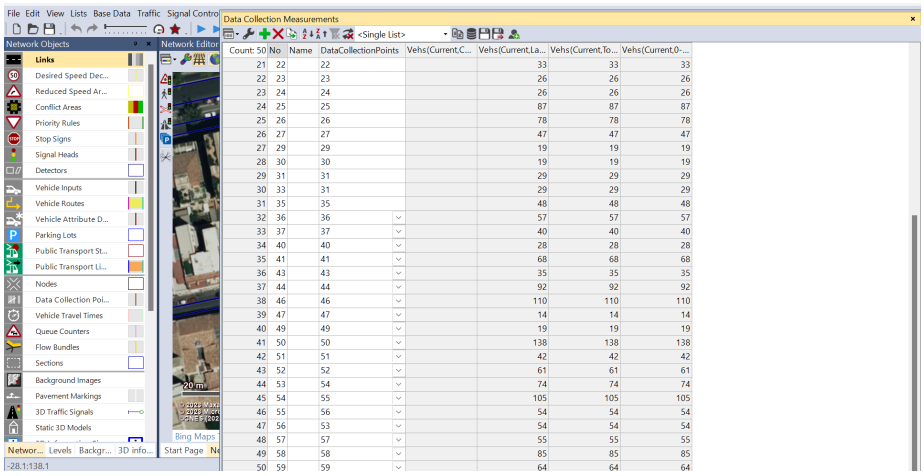

Figure 7.34: Data collection measurement (page 2)

Now that we have obtained the flows of our simulation, by applying algorithm 6.4 we can obtain the flows where the flow sensors have not been installed. We collect these values in tables where in the first column "Arcs" we find the arc and in the second column "Flow" the corresponding flow. In the flows column we have denoted in "red" the flows corresponding to the sensors installed in our simulation which are obviously equal to the theoretical flows, while in "black" the theoretical flows calculated through the algorithm.

Arcs	Flow
945509419 → 2283181348	31
1345621267 → 2283181374	27
1859036758 → 1859036759	26
1859036759 → 1859036758	169
1859036761 → 1859036765	112
1859036763 → 'Virtuale'	105
1859036765 → 1859036761	99
1859036772 → 1859036803	197
1859036772 → 1859036774	26
1859036772 → 1859036801	113
1859036774 → 1859036776	106
1859036774 → 1859036778	138
1859036776 → 1859036804	68
1859036776 → 1859036774	26
1859036801 → 1859036772	178
1859036803 → 1859036772	42
1859036803 → 2283181337	77
1859036804 → 1859036776	111
2283181306 → 2283181310	110
2283181310 → 2283181306	130
2283181310 → 2283181314	87
2283181314 → 2283181310	76
2283181314 → 2283181335	107
2283181335 → 2283181314	144
2283181337 → 1859036803	29
2283181337 → 2283181341	31

Table 7.4: Flow network (Part 1).

Arcs	Flow
2283181341 → 2283181348	105
2283181341 → 2283181337	170
2283181342 → 2283181349	164
2283181342 → 2283181345	40
2283181345 → 2283181342	78
2283181348 → 2283181353	26
2283181348 → 2283181341	26
2283181348 → 945509419	175
2283181349 → 2283181342	160
2283181352 → 2283181354	144
2283181353 → 1859036805	172
2283181353 → 2283181367	35
2283181353 → 2283181348	131
2283181354 → 2283181352	46
2283181354 → 2283181357	64
2283181357 → 2283181354	11
2283181357 → 2283181376	61
2283181367 → 2283181374	142
2283181367 → 2283181353	71
2283181367 → 2283181382	19
2283181374 → 2283181367	70
2283181376 → 2283181357	47
2283181376 → 2283181414	178
2283181382 → 2283181414	104
2283181382 → 2283181367	51
2283181414 → 2283181376	52

Table 7.5: Flow network (Part 2).

Arcs	Flow
2283181414 → 2283181382	56
8306875172 → 8306875181	183
8306875172 → 8306875174	54
8306875174 → 8306875172	148
8306875178 → 8306875181	74
8306875181 → 8306875178	145
272872330 → 2283181341	10
1859036801 → 945509410	158
272872330 → 945509385	19
272872330 → 2283181310	46
272872448 → 2283181353	103
272872448 → 2283181345	120
272872448 → 945509385	87
272872448 → 2283181335	29
274252586 → 2283181354	49
274252586 → 945509419	40
274252586 → 945509417	19
274252586 → 2283181341	144
282757151 → 2283181376	134
282757151 → 945509419	54
282757151 → 1859036805	197
282763248 → 1859036806	192
282763248 → 2283181414	136
282763499 → 2283181349	175
2283181367 → 2283181345	57

Table 7.6: Flow network (Part 3).

Arcs	Flow
282763499 → 8306875181	96
310880162 → 1859036763	130
310880162 → 1859036774	118
945509410 → 1859036801	33
945509410 → 1859036800	19
945509417 → 1859036804	92
945509417 → 1859036803	84
945509417 → 1859036774	117
945509419 → 2283181357	87
1345620364 → 1345621267	101
1345621267 → 1859036806	28
1859036758 → 1859036763	9
1859036759 → 1859036761	85
1859036763 → 1859036772	55
1859036765 → 1859036776	117
1859036800 → 2283181306	13
1859036801 → 2283181337	35
1859036804 → 2283181352	99
1859036805 → 2283181382	186
1859036806 → 2283181382	83
2283181335 → 2283181342	101
2283181345 → 2283181367	26
2283181374 → 8306875172	28
8306875174 → 8306875178	73

Table 7.7: Flow network (Part 4).

7.4 Insights and Findings from Validation Results

In this section we will validate our simulation results with methods for calculating flows. We will show the discrepancy between the flows calculated through the algorithms and simulations by showing some graphs and insights. In particular with the help of the python libraries we will show the goodness of our method through percentage of error between the theoretical calculation and the simulation. We will show both the punctual and system discrepancy and show the trend generated both from a theoretical and simulation point of view by comparing them in every aspect. Let's start by showing the graph of the two separate flows. On the left in "blue" the graph of the theoretical flows is shown, while on the right in "orange" the graph of the simulation flows. At first glance the graphs look similar. On the x-axis we have the number of arcs, while on the y-axis we have the flow, i.e. the number of vehicles passing through that arc in 18 minutes.

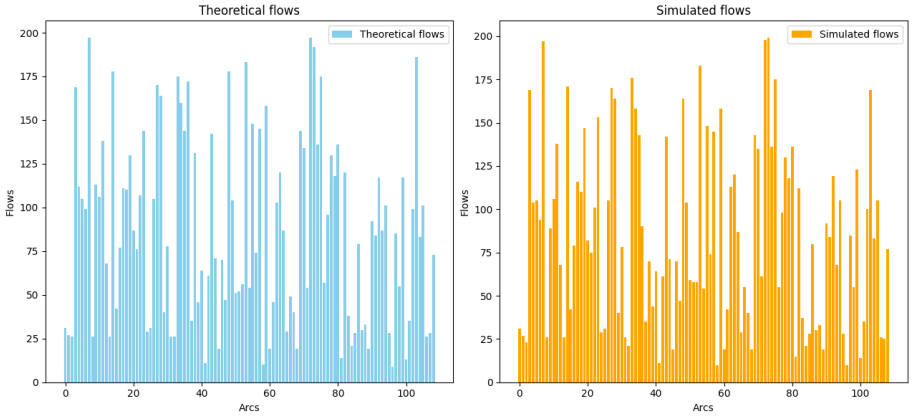

Figure 7.35: Flow graph

After generating the flow graphs of the two systems we can compare them.
In figure 7.36 we show the two graphs superimposed to better show the discrepancy. We note that in correspondence with the arches on which the sensors have been installed the flows are identical, while for some arches the values are not perfectly coincident, in particular already from the graph we notice two arches in particular where the discrepancy is significant, while others where it is less significant, but still relevant.

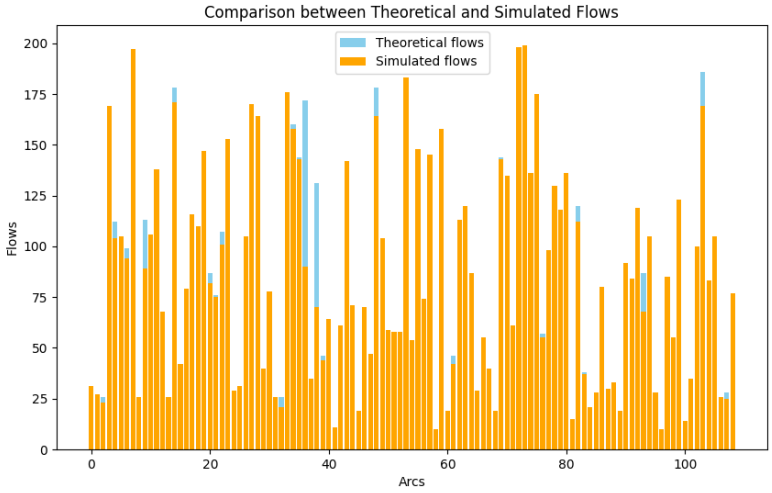

Figure 7.36: Graph of overlaid flows

To better show this discrepancy we will carry out a quantitative analysis by calculating the difference in flows in percentage arc by arc in such a way as to be able to better identify the margin of error. So we took the flow of each theoretical arc and compared it to the corresponding flow of the simulation arc. Let's define:

- φ_t = Theoretical flow.
- φ_s = Simulated flow.

We are going to calculate the percentage of the i-th edge denoted by p_i come:

$$p_i = \frac{\varphi_{t_i} - \varphi_{s_i}}{\varphi_{t_i}} \cdot 100 \tag{7.1}$$

Calculating the percentage of punctual error between the two flows we can see that most flows have an error rate ranging from 1% to 10%, while a smaller set of ten arcs has an error rate ranging between 10% and 20%. Finally only 2 arcs have an error percentage that is in a range between 40% and 50%. After calculating the percentage difference of the flows point by point we show graphically how it is arranged in the system.
In figure 7.37 we show the number of arcs on the x-axis while on the y-axis the percentage difference between the theoretical flow and the flow given by the simulation.

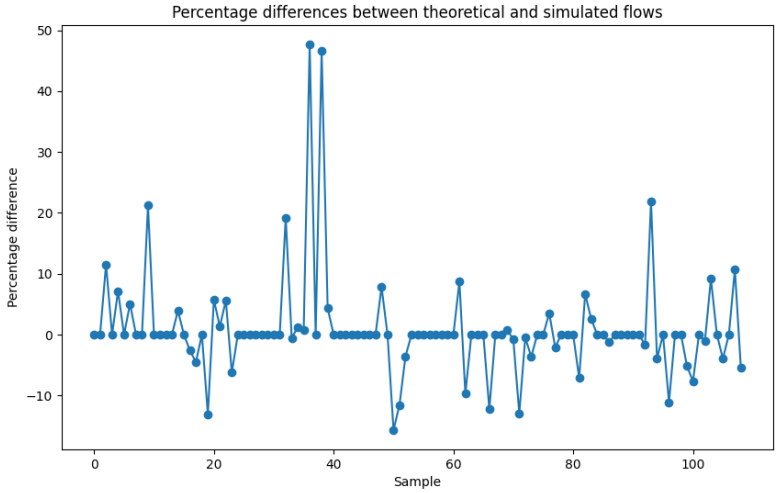

Figure 7.37: Precise percentage difference

Now let's see how much these discrepancies affect the percentage difference of the entire network by calculating the percentage difference of the flow of all the arcs.
Let's calculate the average of the percentage difference of the entire system. Let be the theoretical flow and the simulation flow respectively φ_t and φ_s, we denote the total mean difference of the system as:

$$Ad = \frac{1}{N} \sum_{i}^{n} |p_i| \tag{7.2}$$

Where N represents the total number of edges and the i-th percentage difference is taken in absolute value.
Considering all the calculations that were made on the basis of the flows recorded by the sensors, the discrepancy between the theoretical and simulation flows is only 3.68% which is a very promising number as we have a very low error rate considering that no sensors were installed in those arches. This means that the estimate can vary by approximately 37 machines per 1000 passes.
Figure 7.38 shows the trend of the percentage difference

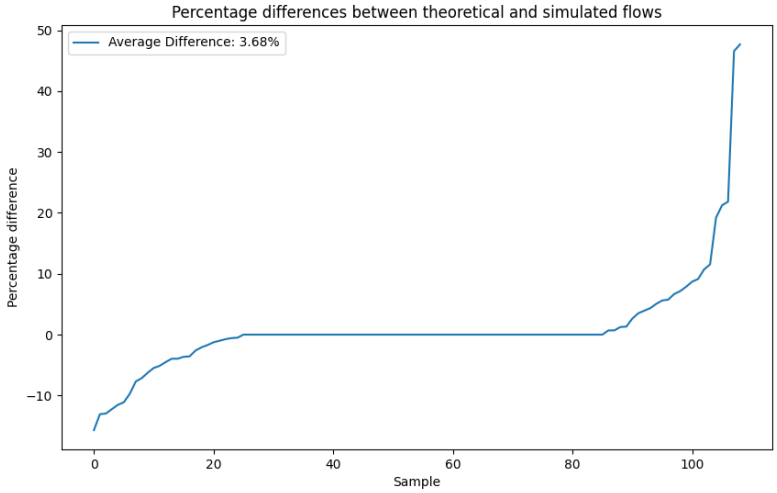

Figure 7.38: Average percentage difference

Although the previous percentage may seem to be true, however, we made the calculation also considering the flows of the arches on which the sensors were installed and this greatly affects the precision of the model as they do not vary from the theoretical flow to the simulation flow. Therefore, to have a more accurate estimate of the flow discrepancy due to our model calculations, we will eliminate the flows of the arcs where the flows where the sensors have been installed and we will recalculate the system average flow percentage difference. We can see in figure 7.39 that the average percentage difference rises dramatically to 8.36% which can still be considered a good error percentage considering that no sensor needs to be installed to estimate the flows in those arches.

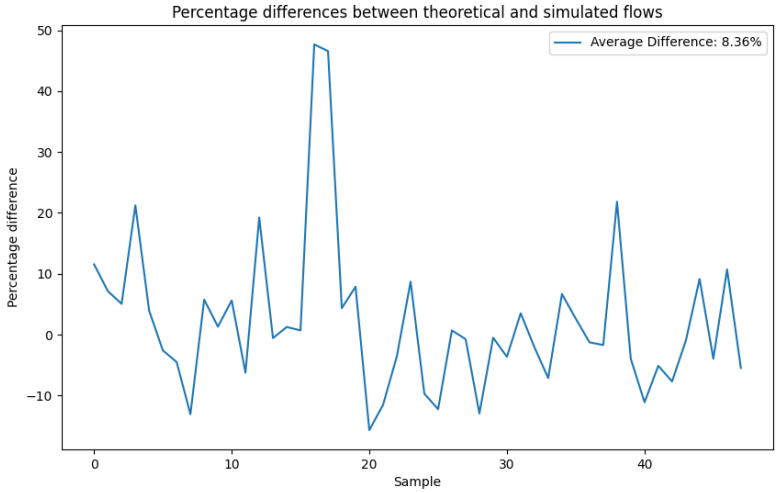

Figure 7.39: Average percentage difference filtrated

Now let's try to interpolate all the values of the theoretical flows and the simulation in such a way as to be able to show their trend.
We respectively define the average flow of the system as:

$$\varphi_m = \frac{1}{n} \sum_{i=1}^{N} \varphi_i \tag{7.3}$$

where N is the number of edges.

We denote respectively with φ_{m_t} the theoretical average flow and φ_{m_s} the average flow of the simulation.
In figure 7.40 you can see the trend of the theoretical flows on the left in "blue", while on the right in "red" the trend of the simulation flows.

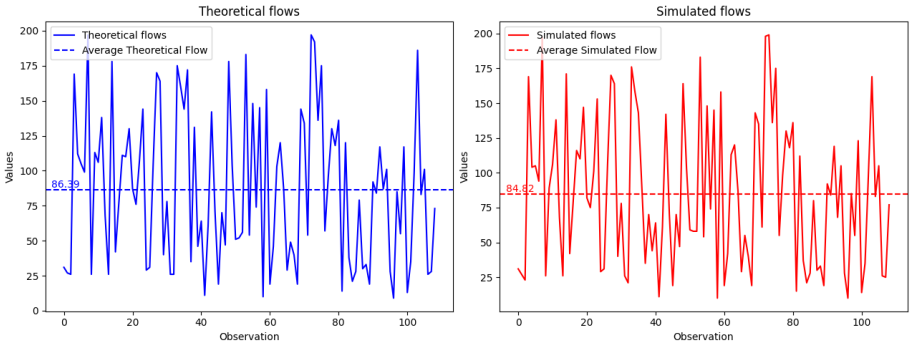

Figure 7.40: Flow trends

We can see how the theoretical approach overestimates the flows compared to the simulation with the theoretical average flow $\varphi_{m_t} = 86,39$ and the simulation average flow $\varphi_{m_s} = 84,82$ respectively.
Therefore, if we divide the theoretical average flow by the average flow of the simulation we obtain, we arrive at the conclusion that the theoretical approach overestimates the simulation approach by 1.85%.

$$\left(\frac{\varphi_{m_t}}{\varphi_{m_s}} - 1\right) \cdot 100 = \left(\frac{86,39}{84,82} - 1\right) \cdot 100 = 1.85\%$$

Finally, showing the overlapping trends in figure 7.41 we can see how they are very similar, confirming the fact that there are no major discrepancies.

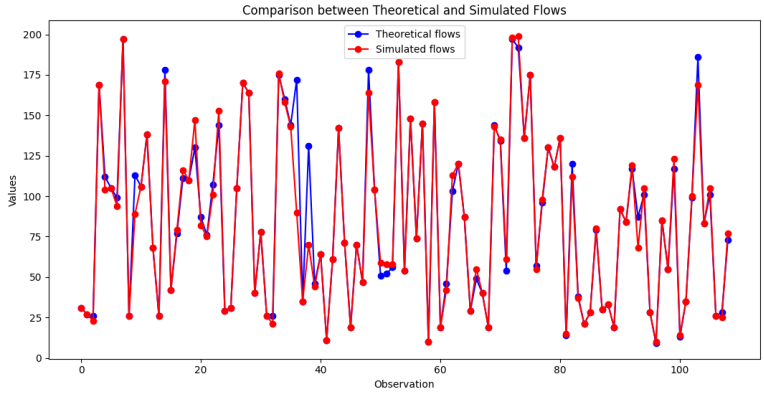

Figure 7.41: Overlapping flow trend

It can be seen how the theoretical approach outperforms the flow in some places. This certainly happens because in real cases there are certain factors that can influence the traffic trend which are not considered in the theoretical model. But surely this overestimation, being not very significant, can be ignore since the theoretical model overestimates the average flow by 1.85% this means that it errs by exceeding 2 vehicles per 100 which in conditions of traffic congestion are completely negligible. We can state that rather than an overestimation of the theoretical model, there is an underestimation of the simulation model as factors have been included which are not considered in the theoretical model, such as the REflow or the recycling rate of vehicle routes. Another factor, as we have already said before, can be congestion or slowdowns due to traffic light control. We can do this check by looking at the simulations at certain moments of time if these two factors occur. In figures 7.42 and 7.43, screenshots have been taken at certain moments of time at two intersections and we see that these two phenomena occur, respectively on the left a slowdown due to traffic light control and on the right a traffic congestion.

Figure 7.42: Light control

Figure 7.43: Congestion

Chapter 8

Conclusion

In this chapter we will give our conclusions on the study carried out, highlighting its strengths and limitations. Thanks to the information collected we will try to think about the implications it could have and future research to improve it even further.

8.1 Recapitulation of Research Objectives

We have seen how NSLP problems, despite being computationally very expensive, can be solved through ad hoc algorithms and heuristics without losing too much in efficiency costs, even for large-scale cities, but certainly for portions of cities. After evaluating different approach and resolution models, we have come to the conclusion that the graphical method is the most efficient at this moment which on the one hand offers easy interpretation while on the other it has constraints due to the topology of the Network itself. Using these methods we have seen how it was possible to save a large number of sensors to have full observation of the network, even if the number of sensors to be used remains significant, this method certainly represents a great improvement in the sector. We also showed how this method is very mutable, for example if we have a limited number of turning ratios with partial information from the system we can create a model by estimating the optimal set and the positions of these sensors and exploiting the partial information. find the optimal number of the position of the flow sensors to still obtain full observability of the system by exploiting the smallest number of sensors. All this with the elegance of graphic representation, being able to easily identify what interests us.
Thanks to the analyses carried out on our simulations, we have seen that it is possible to make a priori estimates with a very low margin of error and with a graphic representation of the phenomenon which allows us to make assessments on the structure of the problem and be able to change them during construction. Furthermore, a database should be set up that collects all the information monitored by the sensors and can exploit this information to build statistics, mobility

demand, traffic characteristics, etc... .

8.2 Summary of Findings and Contributions

In summary, we have seen that traffic monitoring is a solvable problem, even if it is very complex. The graphical methods we used are very easy to interpret for anyone even if they have a limiting factor from the network topology. Thanks to previous research on graphical representation, we have created and implemented algorithms that can be used without the aid of commercial software which exploit the graphical representation and give easy interpretation of the phenomenon.Furthermore we have given a measure in terms of error of the methods used compared to the "real cases", in our case the simulations. Showing that the methods used can also be used from a theoretical point of view using real data. We have given a theoretical method with an efficient graphical representation that can be applied to any network.Therefore, thanks to the use of these methods it is possible to carry out an a priori estimate and complete planning without having to physically intervene first on the network. Furthermore, the graphic representation and the tools used allow changes to be made without damaging the structure of the problem and having immediate feedback on any application within the network.

8.3 Implications for Urban Planning and Infrastructure Development

we have seen how with a sufficiently large number of sensors, taking advantage of the algorithms and heuristics that we have introduced, all the information for traffic monitoring can be obtained very easily and directly. Unfortunately, the minimum limit number for the installation of sensors, as we have seen in theory, is strictly linked to the topology of the network and only limitation of this problem is the number of sensors which leads to high implementation costs, but it is certainly possible to state that with traffic monitoring and road planning the costs due to congestion would be reduced as well as by increasing the quality of the life. Therefore we can say that using these technologies, even if expensive, would drastically improve the management costs of cities and the quality of life and the environment. We have seen how the use of just two sensor technologies such as Turning ratio sensors and counting sensors are sufficient to have full observability of the network and as seen in chapter 3 the installation of these sensors does not require exaggerated interventions for installation. An estimate of the initial purchase and installation costs and subsequent maintenance costs should certainly be made. We can certainly say that when there is a large-scale use of these types of technologies, the costs would be drastically reduced. Furthermore, as we saw in section 2, there are many funds that administrations that encourage the use of these technologies can draw on.

8.4 Future Research Directions

Future research could address the problem of the opportunity cost of being able to eliminate congestion in the city by exploiting these technologies and the high costs of sensors for traffic monitoring and urban planning. In particular, through a detailed research of the costs faced by a large city and its citizens due to traffic congestion, evaluate whether it is economically convenient to apply these methods despite being very expensive, as well as being able to find cheaper methods. Surely from an environmental point of view there would only be improvements and this would be the first incentive, but if an economic incentive could also be found, surely all this could be implemented much more easily.
Another objective that the research could set is to install the sensors only at the busiest roads while maintaining observability on the entire network. This is certainly a challenge from a mathematical point of view as it would be necessary to carry out a study on the topology of the model itself and find algorithms that have an acceptable efficiency in terms of margins of error of the estimates.

Chapter 9

References

Bibliography

[1] United Nations Department of Economic and Social Affairs. Population Facts, "A World of Cities." August 2014.

[2] INRIX. INRIX 2019 Global Traffic Scorecard. March 9, 2020.

[3] Juniper Research. Smart Cities – What's in It for Citizens. November 2018.

[4] The World Health Organization. Ambient Air Pollution: Health Impacts.

[5] Rodrigue, Jean-Paul. "The Geography of Transport Systems." 2017.

[6] 20 Down to Earth. "Switching to Public Transport Could Cut 40 % of Vehicular Emissions by 2050." September 17, 2015.

[7] https://www.mef.gov.it/focus/Il-Piano-Nazionale-di-Ripresa-e-Resilienza-PNRR/

[8] SRSustainableUrbanMobilityIT

[9] Relazione - A9-0108/2023 Parlamento Europeo REPORT on the new EU framework for urban mobility

[10] Report RdS/2011/330

[11] Relazione Speciale 2022, IT, Corte dei Conti,Mobilità urbana sostenibile nell'UE: senza l'impegno degli Stati membri non potranno essere apportati miglioramenti sostanziali

[12] Hu, S.R., Peeta, S., Chu, C., 2009. Identification of vehicle sensor locations for link-based network traffic applications. Transportation Research Part B 43 (8–9), 873–894.

[13] A graphical approach to identify sensor locations for link flow, (2013) inference Sheng-xue He

[14] Synergistic sensor location for link flow inference without path enumeration: A node-based approach ManWo Ng

[15] Optimal Sensor Placement And Density Estimation In Large-Scale Traffic Networks Martin Rodriguez Vega

[16] Location of turning ratio and flow sensors for flow reconstruction in large traffic networks Martin Rodriguez-Vega, Carlos Canudas de Wit, Hassen Fourati

[17] La vita in coda degli italiani: "Nel traffico 23 giorni all'anno" di PAOLO G. BRERA

[18] Networkx Documentation https://networkx.org/

[19] OSM nx Documentation

[20] Libretto introduzione a OpenStreetMap 2017/09 pdf

[21] Porto: "in via di risoluzione i rallentamenti del traffico nell'area dell'ingresso est" https://www.comune.catania.it/informazioni/cstampa/default.aspx?cs=81901

[22] PTV Group, https://www.ptvgroup.com/it

[23] Tutorial ANM BestPracticeGuide: From OSM via PTV Visum to PTV Vissim

[24] Manual PTV Visum, PTV Group

[25] Manual PTV Vissim, PTV Group

[26] Piano Generale del Traffico Urbano, 6 ottobre 2014

[27] 2009 International Conference on Measuring Technology and Mechatronics Automation

[28] P. N. Shivakumar and Kim Ho Chew. "A Sufficient Condition for Nonvanishing of Determinants." In: Proceedings of the American Mathematical Society 43.1 (1974), p. 63 (cit. on p. 22).

[29] B D Greenshields et al. "A study of traffic capacity." In: Highway Research Board. Washington, USA, 1934, pp. 448–477 (cit. on p. 2).

[30] M J Lighthill and G B Whitham. "On kinematic waves II. A theory of traffic flow on long crowded roads." In: Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 229.1178 (1955), pp. 317–345 (cit. on p. 2).

[31] Paul I. Richards. "Shock Waves on the Highway." In: Operations Research 4.1 (1956), pp. 42–51 (cit. on p. 2).

[32] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms, Seconda Edizione. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Sezione 23.2: The algorithms of Kruskal and Prim, pp.567–574. Edizione Italiana da pag. 480 a pag. 490.

[33] Robert Tarjan: Depth-first search and linear graph algorithms. In: SIAM Journal on Computing. Vol. 1 (1972), No. 2, P. 146-160

[34] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 22.3: Depth-first search, pp. 540–549.